BIOLOGY Methods & Protocols

Biology Methods and Protocols, 2021, 1–16

https://doi.org/10.1093/biomethods/bpab021
Advance Access Publication Date: 23 November 2021
Methods Article

METHODS ARTICLE

MultiPep: a hierarchical deep learning approach for multi-label classification of peptide bioactivities

Alexander G. B. Grønning [1,*], Tim Kacprowski[2,3] and Camilla Schéele [1]

[1]Novo Nordisk Foundation Center for Basic Metabolic Research, Faculty of Health and Medical Sciences, University of Copenhagen, 2200 Copenhagen, Denmark, [2]Division Data Science in Biomedicine, Peter L. Reichertz Institute for Medical Informatics, TU Braunschweig and Hannover Medical School, 38106 Braunschweig, Germany and [3]Braunschweig Integrated Centre for Systems Biology (BRICS), 38106 Braunschweig, Germany

*Correspondence address. Novo Nordisk Foundation Center for Basic Metabolic Research, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark. Tel: +4551888742; E-mail: alexander.groenning@sund.ku.dk

Abstract

Peptide-based therapeutics are here to stay and will prosper in the future. A key step in identifying novel peptide-drugs is the determination of their bioactivities. Recent advances in peptidomics screening approaches hold promise as a strategy for identifying novel drug targets. However, these screenings typically generate an immense number of peptides and tools for ranking these peptides prior to planning functional studies are warranted. Whereas a couple of tools in the literature predict multiple classes, these are constructed using multiple binary classifiers. We here aimed to use an innovative deep learning approach to generate an improved peptide bioactivity classifier with capacity of distinguishing between multiple classes. We present MultiPep: a deep learning multi-label classifier that assigns peptides to zero or more of 20 bioactivity classes. We train and test MultiPep on data from several publically available databases. The same data are used for a hierarchical clustering, whose dendrogram shapes the architecture of MultiPep. We test a new loss function that combines a customized version of Matthews correlation coefficient with binary cross entropy (BCE), and show that this is better than using class-weighted BCE as loss function. Further, we show that MultiPep surpasses state-of-the-art peptide bioactivity classifiers and that it predicts known and novel bioactivities of FDA-approved therapeutic peptides. In conclusion, we present innovative machine learning techniques used to produce a peptide prediction tool to aid peptide-based therapy development and hypothesis generation.

Keywords: peptide bioactivity prediction; peptide therapeutics; deep learning; machine learning

Introduction

Identifying bioactivities of peptides is becoming increasingly important. Illuminating the biological activity of peptides will help shedding light on their impact on various processes and help revealing peptides that can be used for therapies against diseases and disorders. A number of biologically available peptides have already now proven efficient for use as drugs for various diseases, including diabetes, osteoporosis, hypertension, cancer, and a variety of endocrine disorders [1, 2]. Though peptide-based drug development is still hampered by certain challenges, such as physiochemical instability and short half-life [3, 4], it can be expected that the future with its technical

Received: 8 September 2021; **Revised:** 28 October 2021; **Editorial Decision:** 11 November 2021; **Accepted:** 17 November 2021

advances will contain more therapies involving peptides [2, 3]. At the same time, mass spectrometry-based peptidomics technologies are rapidly developing, allowing for the discovery of previously unknown molecules [5, 6]. Although a subset of these peptides likely represent potent metabolic signaling molecules, the field currently lacks an efficient approach to scan for specific bioactivities, which further allow meaningful follow-up experiments for target validation. Thus, accurately predicting the bioactivity of peptides is important for identifying novel drug candidates as well as new and unknown peptides. Many tools have been developed for predicting the bioactivity of peptides. Table 1 shows a selection of important state-of-the-art binary classifiers that can group peptides into different bioactivity classes.

Though many of the peptide bioactivity predictors presented in Table 1 offer a wealth of additional information next to a prediction score [9, 13–15], a common denominator for all of them is that they are binary classifiers. For example, the classifier PeptideRanker can predict whether a peptide has a bioactivity or not, and Deep-AmPEP can distinguish short antimicrobial peptides (AMPs) from short non-AMPs. This automatically implies that peptides either have one or none bioactivity, which is not in line with reality [15–25].

Getting an overview of peptides' bioactivities is important when seeking to unravel therapeutics-relevant peptides. Further, an overview of peptides' bioactivity profiles is vital when analyzing mass spectrometry-based peptidomics data, where a key step is to rank peptide candidates for downstream functional studies. A few tools that can predict multiple bioactivity classes simultaneously exist in the literature. PEPred-suite [26] uses eight random forest (RF) models to predict how peptides belong to eight different classes. Peptipedia [27] is another tool that, among other things, can predict different bioactivities of peptides by using 44 RF models trained on data from a large number of peptide databases. A combination of several machine learning algorithm types has been utilized for the prediction of peptide bioactivities (Table 1). However, through the years, deep learning-based models have delivered state-of-the-art results in the field of biological sequence analysis [9, 8, 28, 29]. And though much have been achieved already regarding peptide bioactivity prediction (Table 1), the possibilities with deep learning are endless, and allows for continuous improvements. Here, we present MultiPep, a deep neural network multi-label classifier that can predict the bioactivity of peptides based on 20 bioactivity classes. The tool can take peptides of length 2–200 amino acids that can consist of natural amino acids (A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V). MultiPep utilizes convolutional neural networks [30] for predicting the peptide class belonging, and can classify peptides into zero or more bioactivity classes based on their intrinsic amino acid patterns. The architecture of MultiPep is inspired by a dendrogram from a hierarchical clustering of our defined bioactivity classes' mutual overlaps (Fig. 1). An overlap occurs when two or more classes contain the same peptides. The 20 output nodes of the network are divided into "network class-clades," which have outputs that are agglomeratively combined until two single binary predictions are made. The architecture ensures that extra penalties are added while training, if a sequence is predicted to be in, for example, a wrong network class-clade.

As a new approach to finding the optimal network parameters, we save the parameters of each network class-clade individually whenever the performance has improved on the validation set. This is opposed to saving all weights of the network when the overall performance has improved. We show that this approach is better than saving all weights of a network simultaneously whenever performance increases.

Multi-label classification of imbalanced datasets is approached in different ways [31–33]. Here, we test our customized version of Matthews Correlation Coefficient (MCC) function as parameter-free loss function for our new benchmark multi-label dataset with an imbalanced class-size distribution. We combine our MCC loss function with binary cross entropy (BCE), and we demonstrate that this approach is superior to the use of class-weighted BCE when training on our dataset and when using loss as model selection criteria.

Further, we show that MultiPep can compete with and surpass a set of state-of-the-art binary classifiers on seven different peptide bioactivity classes (Table 1), and that MultiPep can outperform Peptipedia on large set of compared bioactivity classes. Finally, we use our trained models to predict FDA-approved therapeutic peptides and we present how the predictions are in accordance with literature findings or can be used to generate novel hypotheses ready for further analysis and wet laboratory testing. For a quick overview of the workflow of our study, see Supplementary Fig. S1.

Materials and methods

Data and databases

The data used for training, validation, and testing of our models were downloaded from APD3 [15], BioDADPep [16], BIOPEP-UWM [18], CancerPPD [19], CAMPR3 [20], DBAASP [21], LAMP2 [22], NeuroPedia [23], NeuroPep [24], PeptideDB [25], and SATPdb [17].

APD3, CAMPR3, DBAASP, LAMP2, and SATPdb all contain many peptides with experimentally validated antimicrobial function, such as "antibacterial," "antifungal," and "antivirus." But, also peptides annotated with many other types of bioactivities ranging from "toxic" and "chemotactic" to "wound-healing" and "enzyme inhibitor." The databases are both curated (CAMPR3, DBAASP, and SATPdb) and based on peptides found in the literature (APD3) and scientific literature and authoritative databases (LAMP2).

Table 1: State-of-the-art binary classifiers for bioactivity prediction

Prediction class	Algorithm type(s)	Name/description of tool
General bioactivity	Neural network	PeptideRanker [7]
Antimicrobial peptides	Neural network	Convolutional long short-term memory neural network [8]
		Deep-AmPEP30 [9]
Anticancer peptides	Support vector machine	mACPpred [10]
Neuropeptide peptides	Various	PredNeuroP [11]
Toxic peptides	Support vector machine	ToxinPred [12]
Hemolytic peptides	Various	HLPpred-Fuse [13]
Antioxidant peptides	Neural network	AnOxPePred [14]

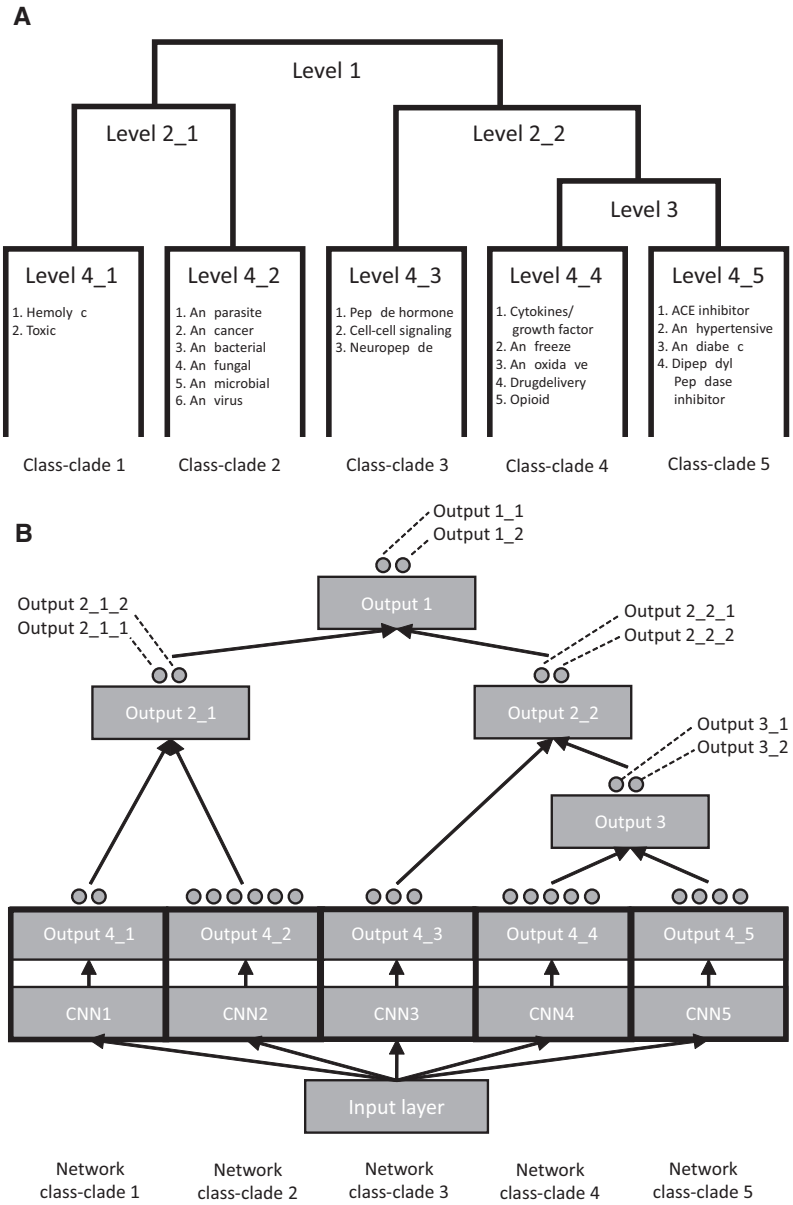

Figure 1: Dendrogram template and overall architecture of the convolutional neural network. (**A**) Dendrogram template. From the bottom, the five class clades can be seen with the bioactivity classes of the clades written within each of them. Above the class clades are connecting levels that amalgamate all the class clades and complete the dendrogram. For visualization purposes, not all leaves of the class clades are shown. (**B**) The overall architecture of MultiPep. From the bottom, the input layer passes input data to the "network class-clades," which consists of a CNN and an output layer. All layers with "Output" in their names are output layers. Above the network class clades are upper-level output layers that connect all the network class clades. The gray-filled circles on top of the output layers indicate the number of output nodes in the layers. The output nodes that are not explicitly named represent core bioactivity classes.

CancerPDD contains experimentally validated anticancer peptides collected from published research articles, patents, and from other databases. PeptideDB and BIOPEP-UWM contain peptides belonging to many different bioactivity classes. PeptideDB contains bioactive peptides from 2820 metazoan species extracted through searches in Swiss-Prot and Trembl protein databases, using BLAST alignment tools and other *in silico* methods, whereas BIOPEP-UWM is a smaller but continuously curated database.

BioDADPep is a database that contains peptides with anti-Type I and/or Type II diabetes properties found based on literature search on PubMed and searches on other published databases. The NeuroPep database contains neuropeptides extracted from the literature, UniProt and other databases, such as Neuropedia. All the entries in NeuroPep have been manually checked [24]. Neuropedia is a database with neuropeptides based on in-house mass spectrometry data. For all databases, where it was possible, only peptides with experimentally validated bioactivity were downloaded. Duplicates and peptide sequences containing unnatural amino acids (B, J, O, U, X, and Z) were removed as well.

Defined bioactivity classes

Based on the downloaded peptides and their annotations, 20 bioactivity classes were defined (see Supplementary

Information and Supplementary Table S1). The complete list of classes and how the used databases have contributed to their generation can be seen in Table 2. Some database class contributions were based on overlaps. For instance, the class "ACE inhibitor" stems only from the database BIOPEP-UWM, however, the database LAMP2 contributed with 39 peptides, because it had peptides from another class that overlapped with the peptides from the "ACE inhibitor" class.

Bioactivity classes were only created if 100 or more peptides were available for that class, and if the class had more than two peptides that were unique for that specific class. We believed that if the classes did not live up to these criteria, the classes would be to information-poor for a classifier to use. Some of the classes had major overlaps, while other classes had none (Supplementary Fig. S2). The largest class was the Antibacterial class with 14 362 peptides and the smallest was the Opioid class with 117 peptides. Analyses to see how the defined bioactivity classes differed in terms of overall peptide lengths and amino acids distributions were made (Supplementary Figs S3 and S4). Some of the bioactivity classes are very general (antimicrobial, peptide hormone, and antihypertensive) while other classes are more specific (ACE inhibitor, opioid, and dipeptidyl peptidase inhibitor). No negative class with peptides without bioactivity was defined, as the many different classes generated negative classes for each other. If required, the applied databases can be used to acquire more information about the classes.

Overall network architecture

MultiPep is a deep neural network implemented using Keras (http://www.keras.io) and Tensorflow [34] in Python. MultiPep's architecture is inspired by a dendrogram from a hierarchical clustering of our defined bioactivity classes' mutual overlaps (Fig. 1A and Supplementary Fig. S2). The max-normalized class overlap values were clustered using the complete linkage type and Euclidean distance as metric. We used the hierarchical clustering algorithm provided by SciPy [35]. The cutoff for the clustering was based on SciPy's default cutoff value. Based on the classes' divisions, a dendrogram template that consisted of five lower level class clades with two, six, three, five, and four classes in each, respectively, was created. Above the class clades, upper levels that connected the lower levels until all class clades were amalgamated were made (Fig. 1A).

Using the dendrogram template, our deep neural network was constructed (Fig. 1B). The network has "network class clades," which consist of small convolutional neural networks (CNNs) and class-clade-specific output layers. See the section "Class-clade-specific convolutional neural networks" for a description of the CNNs. The network class-clade-associated output layers is connected to upper level output layers. Only the network class-clade-associated output layers (Output 4_1, Output 4_2, Output 4_3, Output 4_4, and Output 4_5 in Fig. 1B) contain weights. These layers are regulated by the sigmoid activation function. The upper level output layers (Output 1, Output 2_1, Output 2_2, and Output 3 in Fig. 1B) are concatenation layers that merge the two connected lower level output layers' maximum output values. Table 3 shows which bioactivity classes the upper level output layers represent. To provide an example, if Output level_2_2_2 outputs 0.8 given a peptide, this peptides is predicted to belong to one of the bioactivity classes "antifreeze," "cytokines/growth factors," "antioxidative," "drugdelivery," "opioid," "ACE inhibitor," "antihypertensive," "antidiabetes," or "dipeptidyl peptidase inhibitor" by a score of 0.8.

Class-clade-specific convolutional neural networks

The CNNs from the different network class clades have the same architecture, which is shown in Fig. 2. Each of the CNNs are connected to the input layer. The input layer passes n input examples as a 3D tensor of size $(n \times 4000 \times 1)$ to seven 1D convolutional layers within the different CNNs. The input data are passed as a 3D tensor, as this is required by Keras' 1D convolutional layers. See the section "Generating data for training and testing" for a more detailed explanation of the input data size.

The convolutional layers have 40 kernels with no bias parameters and they convolve over 4, 6, 10, 16, 22, 30, and 40 one-hot encoded amino acids, respectively. Their strides are 20, which is the length of 1 one-hot encoded amino acids. The convolutional layers perform valid convolutions. The 3D tensors that are outputted by the convolutional layers are max pooled along their second axes. The max pooled convolutional output values are concatenated, such that a $(n \times 280)$ tensor is created.

Table 2: Classes and databases.

	Classes	CAMP3	LAMP2	APD3	SATPdb	DBAASP	BIOPEP-UWM	PeptideDB	NeuroPedia	CancerPDD	BioDADPep	NeuroPep	Total class size
0	ACE inhibitor	1	39	1	687	29	973	4	1	1	65	3	973
1	Antibacterial	2104	11 516	2858	4533	10310	512	1059	16	265	31	37	13 538
2	Anticancer	245	1575	419	1177	1919	104	121	1	440	14	6	2426
3	Antidiabetes	11	46	17	125	41	189	9	3	2	1091	3	1112
4	Antifreeze	0	0	0	0	0	0	192	0	0	0	0	192
5	Antifungal	1457	4747	1955	2818	4172	219	559	11	197	22	27	5342
6	Antihypertensive	1	73	6	1664	51	783	10	3	5	107	8	1672
7	Antimicrobial	1882	13 446	2125	8887	3681	239	2544	15	225	9	57	14 362
8	Antioxidative	10	38	28	81	35	649	6	4	1	15	5	675
9	Antiparasite	133	479	190	342	388	46	94	5	20	6	10	503
10	Antivirus	735	4219	990	3918	1611	90	196	8	36	6	14	4500
11	Cellcellsignaling	11	38	15	679	30	22	378	392	0	3	391	682
12	Cytokines_ growthfactors	46	66	64	24	30	11	3729	0	1	0	1	3760
13	Dipeptidyl peptidase inhibitor	0	58	2	169	50	459	6	4	1	158	6	459
14	Drugdelivery	11	108	11	1484	96	17	44	29	5	1	45	1484
15	Hemolytic	183	1299	342	1279	1045	61	111	0	51	1	2	1339
16	Neuropeptide	18	70	26	473	33	108	2101	552	2	4	3822	3926
17	Opioid	0	5	0	27	1	117	12	9	2	0	16	117
18	Peptidehormone	22	149	28	499	33	34	6943	495	1	2	2077	6943
19	Toxic	411	1746	523	3840	1416	310	2404	2	76	1	10	5793

Table 3: Bioactivity classes represented by upper-level output layers

Layer outputs	Combined classes
Output_1_1	Hemolytic, toxic, antiparasite, anticancer, antibacterial, antifungal, insecticides, antimicrobial, and antivirus
Output_1_2	Cell–cell signaling, neuropeptide, peptide hormone, antifreeze, cytokines/growth factors, antioxidative, drugdelivery, opioid, ACE inhibitor, antihypertensive, antidiabetes, and dipeptidyl peptidase inhibitor
Output_2_1_1	Hemolytic and toxic
Output_2_1_2	Antiparasite, anticancer, antibacterial, antifungal, insecticides, antimicrobial, and antivirus
Output_2_2_1	Cell–cell signaling, neuropeptide, and peptide hormone
Output_2_2_2	Antifreeze, cytokines/growth factors, antioxidative, drugdelivery, opioid, ACE inhibitor, antihypertensive, antidiabetes, and dipeptidyl peptidase inhibitor
Output_3_1	Antifreeze, cytokines/growth factors, antioxidative, drugdelivery, and opioid
Output_3_2	ACE inhibitor, antihypertensive, antidiabetes, and dipeptidyl peptidase inhibitor

Each of the n input examples are then normalized by a division of their summed absolute values. This is done to ensure that the network focuses on amino acid patterns and does not predict based on sequence length. Dropout with a rate = 0.2 is applied on the normalization layer. The normalization layer is connected to a chain of three dense layers with 500 nodes where dropout is applied on each of them (rate = 0.5). See the section "Training settings and network initialization" to read more about the used regularization techniques. In the top of Fig. 2, the CNN is connected to an output layer. This could, for instance, be "CNN1" that is connected to "Output 4_1" in Fig. 1B. All layers with weights that are not output layers consist of Parametric Rectified Linear Units (PReLU) [36].

Generating data for training and testing

Models based on 10-fold cross-validation (CV) were trained. To do this, the peptide classes were divided into 10 bins of similar sizes. To distribute the peptides classes uniformly across the bins, all peptides belonging to all specific sets of classes were counted. If the count value was greater than 10, the peptides were split into the 10 bins, and if the count value was lower than 10, the peptides were uniformly distributed at random across the bins. Like in Reference [11], all peptide sequences were one-hot encoded before introduced to the MultiPep. All sequences were zero-padded until they reached a length of 200 one-hot encoded amino acids.

Training sets, validation sets, and testing sets with a size distribution of 0.8, 0.1, and 0.1, respectively, were generated. For all peptide sequences, labels or targets were created. One indicated membership of a class and zero indicated that they were not members of the specific class.

Training settings and network initialization

The start parameters of the convolutional layers were initialized using orthogonal weights. The initial weights of the dense non-output layers were sampled from a uniform distribution with min = −1 and max = 1. The initial weights of the output layers were sampled from a uniform distribution with min = 0.001 and max = 0.05. The bias parameters of the dense layers and output layers were initialized as zeros.

The Adam [37] update function was used with a learning rate of 0.0005. The remaining hyper-parameters for the Adam update function were set to the Keras default values. The batch size was 128 and dropout with a rate of 20% and 50% on the normalization layer and the dense layers, respectively, was applied (Fig. 2). As a final regularization technique, early stopping after 20 epochs was used. All data were shuffled before an epoch was started. To acquire optimal models, weights for the network class clades were saved whenever their performances on the validation set had improved. Additionally, the weights of the entire network were saved whenever the overall performance had increased. The early-stopping count-down was set to zero whenever the parameters of a "network class clade" were saved. The performance was measured in terms of loss.

The customized MCC loss function

To increase the loss penalty during training, the BCE loss function was extended with a customized version of the MCC. The MCC function is given by

$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \tag{1}$$

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives. MCC values range from −1 to 1, where −1 indicates complete disagreement and 1 indicates a perfect agreement [38]. Here, the $\mathrm{MCC}_{\mathrm{loss}}$ function for a single sigmoid output node is calculated by

$$\mathrm{TP}^* = \sum_i^N \hat{y}_i y_i, \tag{2}$$

$$\mathrm{TN}^* = \sum_i^N (1-\hat{y}_i)(1-y_i), \tag{3}$$

$$\mathrm{FP}^* = \sum_i^N \hat{y}_i (1-y_i), \tag{4}$$

$$\mathrm{FN}^* = \sum_i^N (1-\hat{y}_i) y_i, \tag{5}$$

$$\mathrm{MCC}_{\mathrm{loss}} = 1 - \frac{\mathrm{TP}^*\cdot\mathrm{TN}^* - \mathrm{FP}^*\cdot\mathrm{FN}^*}{\sqrt{(\mathrm{TP}^*+\mathrm{FP}^*)(\mathrm{TP}^*+\mathrm{FN}^*)(\mathrm{TN}^*+\mathrm{FP}^*)(\mathrm{TN}^*+\mathrm{FN}^*)}}, \tag{6}$$

where TP^*, TN^*, FP^*, and FN^* are modified under certain circumstances. If $y_i = 0$ for all instances in a mini-batch, then TP^* is set to be minimum 1. If $y_i = 1$ for all instances in a mini-batch, then TN^* is set to be minimum 1. If $\hat{y}_i = 0$ for all instances in a mini-batch, then FP^* is set to be minimum 1. If $\hat{y}_i = 1$ for all instances in a mini-batch, then FN^* is set to be minimum one. These modifications are done to avoid that the denominator of the $\mathrm{MCC}_{\mathrm{loss}}$ function becomes zero.

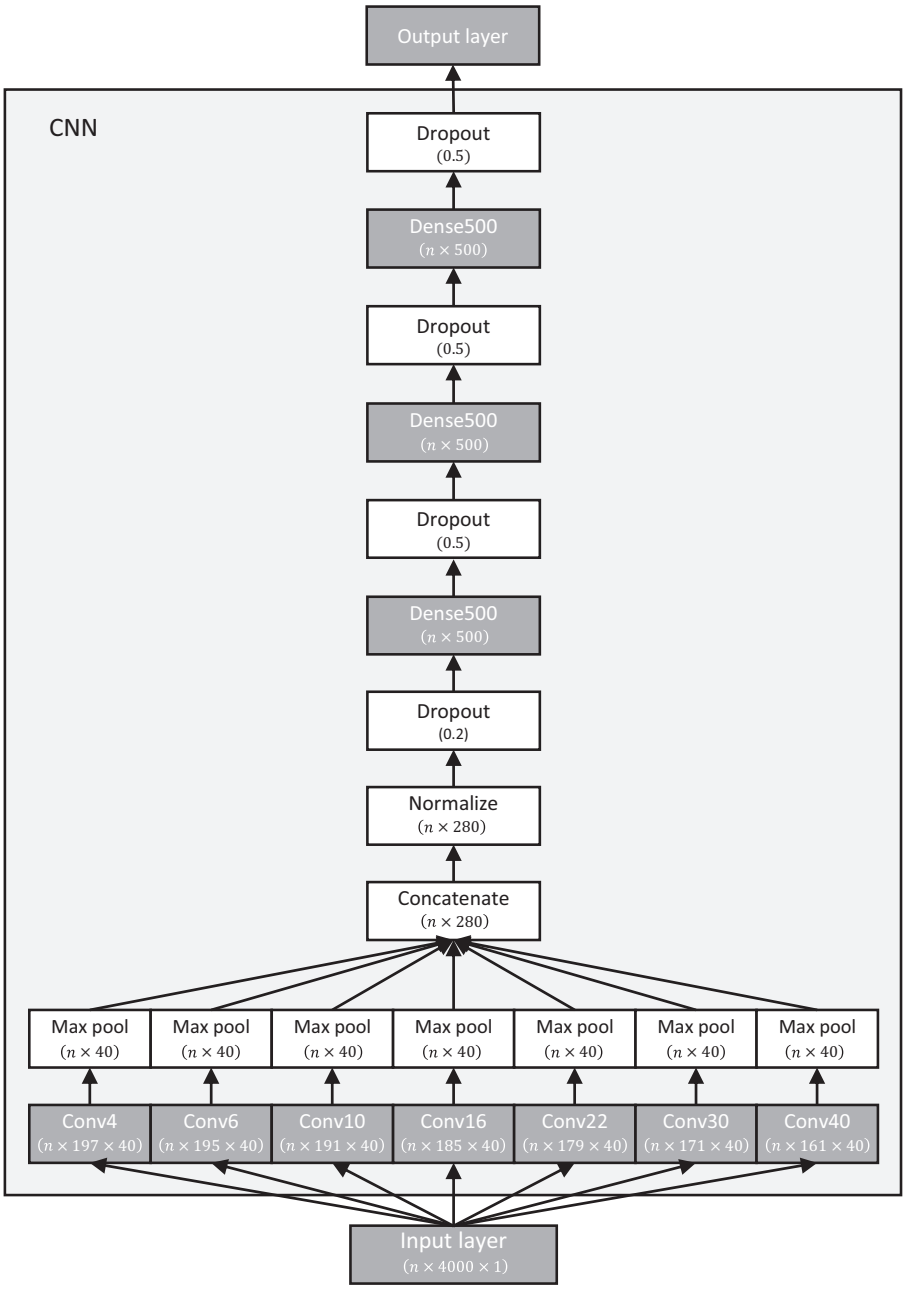

Figure 2: Architecture of network class-clade convolutional neural networks. All layers with dark-gray backgrounds (except for the input layer) have weights, whereas the layers with white backgrounds are mathematical operations or dropout layers. Below the name of each layer (except for the dropout layers), the sizes of the layers' output tensors are written. Arrows show the flow of information and how the layers are connected. What constitutes the convolutional neural network (CNN) is encapsulated by a light-gray box. "Conv4" means a 1D convolutional layer with a kernel size of four one-hot-encoded amino acids. The same logic applies to the remaining convolutional layers. "Dense500" means a dense layer with 500 nodes.

Again, with a single sigmoid output node, the loss for a single predicted mini-batch ranges between 0 and 2. Zero is given to a perfectly predicted mini-batch and two is given to perfectly miss-predicted mini-batch that contains both $y_i = 1$ and $y_i = 0$ instances. In the situation where $y_i = 1$ and $\hat{y}_i = 0$ or $y_i = 0$ and $\hat{y}_i = 1$ for all instances in a mini-batch, the loss can never reach exactly 2 due to the TP* and TN* definitions. So, to avoid "losing" loss while training, the loss is set to 2 if $|\sum_i^N y_i - \sum_i^N \hat{y}_i| = N$, where N is the number of examples in the mini-batch. In the situation where $y_i = 1$ and $\hat{y}_i = 1$ or $y_i = 0$ and $\hat{y}_i = 0$ for all instances in a mini-batch, the loss can never reach exactly 0 due to the FP* and FN* definitions. To circumvent this, the MCC_{loss} was implemented such that its outputs are multiplied by 0 if $|\sum_i^N y_i - \sum_i^N \hat{y}_i| = 0$.

The complete loss function with MCC_{loss}

The entire loss function for a single output node given all instances of mini-batch is defined as

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] + \text{MCC}_{\text{loss}}, \quad (7)$$

where N is the number of examples in the mini-batch. The loss function can be seen as two-step function where the MCC_{loss} values are calculated for the entire batch in the first step and then added to the BCE loss values, which then are averaged. During the updating steps while training, the loss is summed across all output nodes.

Loss function with class weights

Models using class weights were trained, which is a standard way of training models on data with imbalanced class-size distributions [39]. The class weights were based on the data from the training sets and were generated using the "compute_class_weight" function from Scikit-learn [40]. This means that a class weight, Cl_j, was generated for every j class. While training, the loss of a single output node was calculated using

$$\text{Loss}_{\text{Cl}} = -\frac{1}{N}\sum_{i=1}^{N}\left[(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \cdot \text{Cl}_j\right], \quad (8)$$

where N is the number of examples in the mini-batch and Cl_j is the class weight of the class in question. During the updating steps while training, the loss is summed across all output nodes.

Webtool and stand-alone-program

A Numpy [41] version of MultiPep based on the trained weights was implemented. It is available on https://agbg.shinyapps.io/MultiPep/. Users can choose between using the average of all CV models' prediction as the final output or to get the max of all CV models' predictions of the different classes as the final output. This is true for the webtool and the stand-alone-program. The webtool rounds the predictions to three decimals, while the stand-alone-program provides a higher number of decimals. The stand-alone-program is available at https://github.com/scheelelab/MultiPep. If predicting more than 500 in one batch of peptide sequences, we recommend using the stand-alone-program.

Performance metrics

The performance of MultiPep and peptide predictors compared with MultiPep was evaluated using the following metrics:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \quad (9)$$

$$\text{F1 score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \quad (10)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (11)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (12)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (13)$$

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

Results

Architecture rationale

We designed the architecture of MultiPep with inspiration from a dendrogram from a hierarchical clustering of our defined bioactivity classes' mutual overlaps. The performed clustering of the max-normalized class overlap values provided a proxy for the classes' peptide similarities. We hypothesized that the network would make more exact predictions when we grouped similar classes together and connected them to the same CNN. The CNNs would then be forced to learn how to distinguish between similar peptide patterns using the same network parameters. Further, the hierarchical structure of MultiPep ensures that an extra penalty is added while training, if a peptide is predicted to be, for example, in a wrong class clade or if the output layer of the network class clade produces only false negatives or positives. This happens because a, for example, false positive in an erroneous class clade will propagate through all connected upper output layers and thus produce more loss via these layers.

MCC loss function rationale

It has been suggested that the MCC function is a good performance metric for imbalanced data sets [42]. Thus, inspired by the work in Abhishek and Hamarneh [43], we found it intriguing to use a customized version of MCC function as parameter-free loss function for our data set with an imbalanced class–size distribution. However, the MCC function as is was in our opinion not suited for a classifier that, while training, iteratively looks at mini-batches of training examples. Especially not when training on data sets with small classes. For example, the MCC function will include division by 0, if for example, only true negatives or true positives are predicted in a mini-batch (Equation 6). Our loss function is a combination of the standard BCE and our MCC_{loss} function, as we found that this combination yielded the best results. In this context, the MCC_{loss} function produces extra penalties for the predictions with a low MCC score.

Model performances

Using data derived from multiple peptide databases, we conducted 10-fold CVs and tested the models on their associated test sets. We trained several models using two different loss functions. The neural network architecture and all hyperparameters and regularization settings were always the same. First, we trained models using our loss function with the MCC_{loss} penalty, where we applied our training scheme that saved the parameters of the best "network class clades" whenever performance on the validation sets increased. Secondly, while training these models, we saved the parameters of the entire network whenever the overall performance had increased. Thirdly, we ran a 10-fold CV using the weighted BCE loss function. We used the training scheme, where the parameters of the "network class clades" were saved whenever their performances increased and the training scheme where we saved all network parameters whenever the overall performance had increased. This produced 4×10 different models, which we used for a comparative study to find the best way of training the network.

Tables 4 and 5 show the mean and standard deviations of the performances of the trained models on the CV test sets using MCC and the F1 score as performance metrics. In the tables, the mean values have been compared and the standard

Table 4: Mean and standard deviation of MCC of CV models on test sets

Bioactivity class/output name	BCE + MCC_{loss}	BCE + MCC_{loss} − overall lowest loss	Weighted BCE	Weighted BCE—overall lowest loss
Output 1_1	0.839 ± 0.01	0.843 ± 0.01	0.803 ± 0.01	0.799 ± 0.01
Output 1_2	0.856 ± 0.01	0.859 ± 0.01	0.81 ± 0.02	0.847 ± 0.01
Output 2_1_1	0.782 ± 0.01	0.788 ± 0.01	0.764 ± 0.02	0.778 ± 0.01
Output 2_1_2	0.849 ± 0.01	0.853 ± 0.01	0.813 ± 0.01	0.802 ± 0.01
Output 2_2_1	0.895 ± 0.01	0.894 ± 0.01	0.875 ± 0.01	0.878 ± 0.01
Output 2_2_2	0.826 ± 0.01	0.83 ± 0.01	0.751 ± 0.02	0.811 ± 0.01
Output 3_1	0.798 ± 0.01	0.798 ± 0.01	0.711 ± 0.03	0.787 ± 0.02
Output 3_2	0.766 ± 0.03	0.765 ± 0.02	0.749 ± 0.02	0.751 ± 0.02
Hemolytic	0.531 ± 0.04	**0.539 ± 0.03**	0.514 ± 0.03	0.52 ± 0.03
Toxic	0.785 ± 0.01	**0.789 ± 0.01**	0.768 ± 0.02	0.782 ± 0.01
Antimicrobial	**0.675 ± 0.01**	0.674 ± 0.01	0.617 ± 0.01	0.587 ± 0.01
Antivirus	0.611 ± 0.02	**0.614 ± 0.02**	0.593 ± 0.01	0.569 ± 0.03
Antiparasite	0.281 ± 0.05	0.173 ± 0.1	**0.299 ± 0.06**	0.289 ± 0.07
Anticancer	0.51 ± 0.03	0.488 ± 0.02	**0.513 ± 0.03**	0.47 ± 0.03
Antibacterial	0.702 ± 0.01	**0.703 ± 0.01**	0.659 ± 0.01	0.644 ± 0.01
Antifungal	**0.487 ± 0.02**	0.471 ± 0.02	0.411 ± 0.03	0.335 ± 0.06
Cell–cell signaling	**0.553 ± 0.02**	0.552 ± 0.02	0.544 ± 0.03	0.548 ± 0.04
Neuropeptide	**0.78 ± 0.02**	0.771 ± 0.02	0.74 ± 0.02	0.757 ± 0.03
Peptide hormone	**0.845 ± 0.01**	0.844 ± 0.01	0.82 ± 0.01	0.828 ± 0.01
Antifreeze	0.966 ± 0.04	0.976 ± 0.03	0.976 ± 0.03	**0.977 ± 0.02**
Cytokines/growth factors	0.932 ± 0.01	**0.936 ± 0.01**	0.851 ± 0.02	0.923 ± 0.01
Antioxidative	0.467 ± 0.06	0.461 ± 0.06	0.378 ± 0.09	**0.483 ± 0.07**
Drugdelivery	**0.598 ± 0.05**	0.588 ± 0.04	0.248 ± 0.13	0.556 ± 0.05
Opioid	0.701 ± 0.11	0.676 ± 0.11	0.67 ± 0.11	**0.733 ± 0.1**
ACE inhibitor	0.511 ± 0.02	**0.511 ± 0.02***	0.502 ± 0.03	0.498 ± 0.03
Antihypertensive	**0.668 ± 0.03**	0.664 ± 0.02	0.652 ± 0.02	0.655 ± 0.02
Antidiabetes	**0.596 ± 0.03**	0.58 ± 0.03	0.587 ± 0.03	0.587 ± 0.03
Dipeptidyl peptidase inhibitor	0.564 ± 0.05	**0.564 ± 0.04**	0.557 ± 0.03	0.559 ± 0.08
Average of bioactivity classes:	**0.638**	0.629	0.595	0.615
Best total	**8**	7	2	3

Notes: Bold values indicate the model with the best performance. The asterisk symbols indicate that more than three decimals are needed to reveal the highest values. At the two bottom rows, "Best total" shows the number of times the average of the models are the best and "Average of bioactivity classes" shows the average of the columns above, but only for the bioactivity classes.

deviations are stated to provide readers with an overview of the robustness of the models. Similar tables showing the performance on the test sets, but using precision, recall, and accuracy as performance metrics are available in the Supplementary Information (Supplementary Tables S2–S4). Additionally, plots of receiver operating characteristic (ROC) curves of MultiPep models' predictions of the test sets can be found in the Supplementary Information (Supplementary Figs S5–S14). Tables showing the performance on the validation sets can as well be found in the Supplementary Information (Supplementary Tables S5–S9). The classification threshold was 0.5 and all mean values in the tables have been rounded to three decimals, whereas all standard deviation values have been rounded to two decimals. We did not use the upper layers' predictions for the comparisons, as we were only interested in finding how the models performed on the bioactivity classes. The performances on the test sets and validation sets of the individual CV models trained using the "save the individual network class-clade" training scheme can be found in Supplementary Tables S10–S29.

The performance comparisons suggest that the approach where we save the parameters of the network class clades whenever their performances increase is slightly better than saving all parameters when whenever the overall performance has increased. However, if taking the performances on the validation sets into account, it seems to be favorable to use the "save the individual network class clade" approach. Further, the study indicates that using our combined BCE and MCC_{loss} loss function in general produces better classifiers than when using weighted BCE as loss function. Altogether, the performance of our models trained using the "save the individual network class clade" training scheme seems to produce reasonable and robust classifiers with an average MCC score above 0.5, except for the "antiparasite," "antifungal," and "antioxidative" class.

Comparisons against state-of-the-art binary classifiers

We next aimed to compare MultiPep against existing state-of-the-art binary bioactivity classifiers. We could not find a range of tools necessary for comparing all bioactivity classes predicted by MultiPep. Thus, we chose a few different classifiers based on their performances and bioactivity focus (Tables 1 and 6). We only used the tools' pre-trained models available on the tools' webpages or on GitHub (Table 6). As many of the chosen classifiers did not take peptides of length 2–200, it was necessary to

Table 5: Mean and standard deviation of F1 score of CV models on test sets

Bioactivity class/output name	**BCE + MCC_{loss}**	**BCE + MCC_{loss} − overall lowest loss**	Weighted BCE	Weighted BCE—overall lowest loss
Output 1_1	0.936 ± 0.0	0.937 ± 0.0	0.922 ± 0.0	0.921 ± 0.0
Output 1_2	0.909 ± 0.01	0.911 ± 0.01	0.874 ± 0.01	0.903 ± 0.01
Output 2_1_1	0.801 ± 0.01	0.807 ± 0.01	0.782 ± 0.02	0.799 ± 0.01
Output 2_1_2	0.928 ± 0.0	0.929 ± 0.0	0.911 ± 0.0	0.907 ± 0.0
Output 2_2_1	0.915 ± 0.01	0.914 ± 0.01	0.898 ± 0.01	0.902 ± 0.01
Output 2_2_2	0.857 ± 0.01	0.86 ± 0.01	0.787 ± 0.02	0.844 ± 0.01
Output 3_1	0.82 ± 0.01	0.821 ± 0.01	0.723 ± 0.04	0.81 ± 0.02
Output 3_2	0.781 ± 0.02	0.781 ± 0.02	0.766 ± 0.02	0.768 ± 0.02
Hemolytic	0.538 ± 0.04	**0.543 ± 0.03**	0.52 ± 0.03	0.528 ± 0.03
Toxic	0.8 ± 0.01	**0.806 ± 0.01**	0.78 ± 0.02	0.799 ± 0.01
Antimicrobial	0.776 ± 0.01	**0.777 ± 0.01**	0.741 ± 0.01	0.724 ± 0.01
Antivirus	0.62 ± 0.02	**0.621 ± 0.02**	0.596 ± 0.02	0.558 ± 0.04
Antiparasite	0.267 ± 0.06	0.151 ± 0.1	**0.293 ± 0.06**	0.28 ± 0.07
Anticancer	0.5 ± 0.04	0.478 ± 0.02	**0.515 ± 0.03**	0.447 ± 0.06
Antibacterial	0.793 ± 0.01	**0.793 ± 0.01**	0.764 ± 0.01	0.753 ± 0.01
Antifungal	**0.547 ± 0.02**	0.535 ± 0.02	0.478 ± 0.03	0.394 ± 0.07
Cell-cell signaling	**0.552 ± 0.02**	0.551 ± 0.02	0.531 ± 0.03	0.536 ± 0.04
Neuropeptide	**0.798 ± 0.02**	0.79 ± 0.02	0.761 ± 0.02	0.777 ± 0.03
Peptide hormone	**0.868 ± 0.01***	0.868 ± 0.01	0.847 ± 0.01	0.854 ± 0.01
Antifreeze	0.965 ± 0.04	0.976 ± 0.03	0.976 ± 0.03	**0.976 ± 0.02***
Cytokines/growth factors	0.937 ± 0.01	**0.94 ± 0.01**	0.863 ± 0.02	0.929 ± 0.01
Antioxidative	0.468 ± 0.06	0.466 ± 0.07	0.318 ± 0.11	**0.481 ± 0.07**
Drugdelivery	**0.593 ± 0.05**	0.583 ± 0.04	0.166 ± 0.12	0.552 ± 0.05
Opioid	0.694 ± 0.11	0.667 ± 0.11	0.654 ± 0.11	**0.725 ± 0.1**
ACE inhibitor	**0.506 ± 0.02**	0.493 ± 0.02	0.505 ± 0.03	0.501 ± 0.03
Antihypertensive	**0.667 ± 0.03**	0.66 ± 0.02	0.652 ± 0.02	0.656 ± 0.02
Antidiabetes	**0.598 ± 0.03**	0.584 ± 0.03	0.586 ± 0.03	0.589 ± 0.03
Dipeptidyl peptidase inhibitor	0.562 ± 0.05	**0.563 ± 0.04**	0.543 ± 0.04	0.55 ± 0.08
Average of bioactivity classes:	**0.652**	0.642	0.604	0.63
Best total:	**8**	7	2	3

Notes: Bold values indicate the model with the best performance. The asterisk symbol indicates that more than three decimals are needed to reveal the highest values. At the two bottom rows, "Best total" shows the number of times the average of the models are the best, and "Average of bioactivity classes" shows the average of the columns above, but only for the bioactivity classes.

Table 6: Comparisons against state-of-the-art peptide bioactivity predictors

	MCC	F1 score	Precision	Recall	Accuracy
MultiPep	**0.804**	**0.91**	**0.983**	0.848	**0.897**
Neural network by Veltri *et al.* [8], link	0.526	0.836	0.773	**0.909**	0.78
MultiPep	**0.813**	**0.94**	**0.984**	**0.9**	**0.917**
Deep-AmPEP30, link	0.657	0.9	0.92	0.88	0.858
RF-AmPEP30, link	0.712	0.914	0.94	0.889	0.879
MultiPep	**0.604**	0.631	**1.0**	0.461	**0.824**
mACPpred, link	0.459	**0.653**	0.512	**0.9**	0.688
MultiPep	**0.879**	**0.892**	**0.959**	0.833	**0.973**
PredNeuroP, link	0.698	0.722	0.579	**0.959**	0.901
MultiPep	**0.677**	**0.77**	**0.997**	**0.627**	**0.817**
ToxinPred, link	0.567	0.687	0.943	0.54	0.76
MultiPep	**0.724**	**0.725**	**1.0**	0.569	**0.928**
HLPred-Fuse, link	0.435	0.511	0.355	**0.908**	0.708
MultiPep	**0.588**	**0.552**	**1.0**	0.381	**0.913**
AnOxPePred, link	0.29	0.394	0.377	**0.413**	0.822

Notes: Bold values indicate that the performance is better than the compared tool(s). All values in the table have been rounded to three decimals. Links to the webtools/github of the tools are inserted next to the names of the tools in the first column from the left.

create different sets of peptide sequences that were compatible with the tools' input formats (see Supplementary Table S30 for sizes of peptide sets).

For the tool by Veltri *et al.* [8], Deep-AmPEP30 and RF-AmPEP30, the positive peptides were from the "antibacterial" class whereas the negative peptides were drawn from the

classes "cell–cell signaling," "neuropeptide," and "peptide hormone." For mACPred, ToxinPred, HLPpred-Fuse, and AnOxPePred, the positive peptides were from the classes "anticancer," "toxic," "hemolytic," and "antioxidative," respectively, and the negative peptides were as well drawn from the classes "cell–cell signaling," "neuropeptide," and "peptide hormone." For PredNeuroP, we sampled positive peptides from the "neuropeptide" class and negative peptides from the classes "antiparasite," "anticancer," "antibacterial," "antifungal," "antimicrobial," and "antivirus." For the tool by Veltri *et al.* [8], we used their newest model for the predictions. For ToxinPred, we used the tool's Swiss-Prot-based SVM model. For AnOxPePred, we used the tools "Peptide Mode" and only used its "Free Radical Scavenger"-prediction mode.

Additionally, we compared MultiPep with THPep, a tool specialized in finding tumor homing peptides [44]. Here, the positive peptides were from the class "anticancer" and the negative peptides were from the classes "cell–cell signaling," "neuropeptide," and "peptide hormone."

For all peptide sets created, we ensured that there were no overlaps between the positive and negative peptides. We did not make any effort to create balanced data sets, as this should not matter for the comparisons. For the comparisons, we used the MultiPep model trained using our "save the individual network class-clade" approach that produced the lowest overall loss on the test set (Supplementary Tables S15 and S25). All peptides from the above-described peptide sequence sets were drawn from the model's test set, as the peptides in this set contain unseen peptides that have not been part of model's training. We did not consider if the sequences in the smaller sequence sets were part of the different tools' training sets. All peptide sequence sets are available on https://github.com/scheelelab/MultiPep. We observed that MultiPep in general outperforms existing peptide prediction tools when subject to our model's test set (Table 6 and Supplementary Table S31). While the recall ability of MultiPep is bested by the other tools except for Deep-AmPEP30, RF-AmPEP30, and ToxinPred, MultiPep consistently outperforms the other tools on the MCC score, precision, and accuracy.

Comparing prediction error of MultiPep against PeptideRanker

As an additional analysis, we compared MultiPep with PeptideRanker. PeptideRanker is a binary classifier that can predict general bioactivity of peptides. Thus, we found it interesting to compare MultiPep against this tool. Using PeptideRanker, we predicted all peptides of the test set of the model with the lowest loss on the test trained using our "save the individual network class-clade" training scheme (Supplementary Tables S15 and S25). The set contained 4585 peptides and we did not consider whether any of the used peptides from out test set were part of the training data of PeptideRanker. For both MultiPep and PeptideRanker, instead of using peptides with no bioactivity, we used all peptides of the test set as true positives and found the error of the tool's predictions. We calculated the error of PeptideRanker's predictions by $\frac{\sum_{i=1}^{N} 1-\text{round}(p)}{N}$, where N is the number of examples in the test set, p is the predictions, and round() is a function that rounds values to their nearest integer. The rounding function implies that the threshold for the predictions was 0.5. For MultiPep, we took the predictions on the Output_1 output layer and found how they diverged from the labels of that layer. We calculated $\frac{\sum_{i=1}^{N}\sum_{j=1}^{k}|y_j-\text{round}(p_j)|}{N}$, where N is the number of examples in the test set, $\boldsymbol{y}$ is an element of the row vector $\boldsymbol{y}$ with labels, $\boldsymbol{p}$ is an element of the row vector $\boldsymbol{p}$ with predictions, and round() is a function that rounds values to their nearest integer. Also, to make a more direct estimate of whether MultiPep can assign a prediction above 0.5 to bioactive peptides, we calculated an additional error percentage by $\frac{\sum_{i=1}^{N} 1-\text{round}(\max(\boldsymbol{p}))}{N}$, where N and $\boldsymbol{p}$ are the same as above and max() is a function that finds the maximum value of the row vector $\boldsymbol{p}$. With this approach, we calculated the prediction errors (Table 7). This analysis shows that MultiPep, with a low prediction error, successfully identifies a higher number of bioactive peptide from a list of peptides with known bioactivity.

Table 7: Prediction error of MultiPep compared with PeptideRanker

Program names	Rounded prediction error
MultiPep (Output_1, max)	0.074
MultiPep (Output_1)	0.140
PeptideRanker	0.299

Note: Final prediction errors rounded to three decimals.

MultiPep versus peptipedia

Peptipedia is a tool that can predict many different bioactivity classes simultaneously. The authors describe how they have trained 44 different RF models that each can predict belonging to a specific bioactivity class [27]. As described for the comparisons above, we used the MultiPep model trained using our "save the individual network class-clade" approach that produced the lowest overall loss on the test set (Supplementary Tables S15 and S25). We used the test set of this model to generate a test-subset that was used as input to the Peptipedia models (Supplementary Table S30). Noteworthy, as the Peptipedia webtool was unavailable during the writing of this article, we retrieved the Peptipedia predictions via personal communication with the authors of the article presenting Peptipedia [27]. In the prediction data received by the Peptipedia authors, 41 bioactivity classes were included. The Peptipedia prediction is available on https://github.com/scheelelab/MultiPep. Both MultiPep and Peptipedia can predict many different bioactivity classes. Supplementary Table S32 presents an overview of classes that were deemed comparable and incomparable, which was based on the name of the classes. For this grouping of the classes, it was taken into account how the MultiPep classes were generated (Supplementary Information). If a MultiPep bioactivity class was deemed comparable with more than one Peptipedia bioactivity class, the predictions of these Peptipedia classes were merged such that, for every predicted peptide, the max value of the predictions was used. Comparable bioactivity classes are expected to contain similar and identical peptides. Table 8 shows how MultiPep and Peptipedia performed on the generated test-subset. With the exception of the recall score for the antiparasite and the antifungal classes, we observed that MultiPep outperforms Peptipedia on all performances for predicting the compared bioactivity classes.

MultiPep models versus logistic regression models

We next aimed to compare our MultiPep deep learning models with linear logistic regression models. Therefore, we trained 10 logistic regression models using our peptide data and our 10-fold CV scheme. We used the logistic regression functionalities

Table 8: Performance of MultiPep and Peptipedia

	MCC		F1 score		Precision		Recall		Accuracy	
Bioactivity classes	*M*	*P*	*M*	*P*	*M*	*P*	*M*	*P*	*M*	*P*
Hemolytic	**0.599**	0.023	**0.609**	0.064	**0.655**	0.044	**0.569**	0.115	**0.977**	0.895
Toxic	**0.785**	−0.052	**0.798**	0.172	**0.952**	0.115	**0.686**	0.341	**0.952**	0.551
Antimicrobial	**0.689**	0.065	**0.791**	0.48	**0.806**	0.355	**0.777**	0.74	**0.863**	0.463
Antivirus	**0.634**	0.021	**0.647**	0.171	**0.834**	0.116	**0.528**	0.324	**0.939**	0.666
Antiparasite	**0.281**	0.039	**0.269**	0.031	**0.409**	0.016	0.2	**0.533**	**0.988**	0.647
Anticancer	**0.563**	0.08	**0.564**	0.142	**0.746**	0.086	**0.453**	0.419	**0.961**	0.719
Antibacterial	**0.721**	0.254	**0.811**	0.497	**0.775**	0.477	**0.851**	0.519	**0.876**	0.671
Antifungal	**0.493**	0.211	**0.556**	0.32	**0.544**	0.214	0.569	**0.637**	**0.889**	0.668
Neuropeptide	**0.802**	0.249	**0.82**	0.296	**0.815**	0.377	**0.825**	0.243	**0.967**	0.896
Drugdelivery	**0.653**	0.111	**0.65**	0.143	**0.804**	0.133	**0.545**	0.154	**0.98**	0.937
Antihypertensive	**0.679**	0.0	**0.672**	0.0	**0.557**	0.0	**0.848**	0.0	**0.982**	0.978
Antidiabetes	**0.641**	0.046	**0.646**	0.05	**0.697**	0.097	**0.602**	0.034	**0.986**	0.973

Notes: Bold values indicate that the performance is better than the compared tool. All values in the table have been rounded to three decimals. M, MultiPep; P, Peptipedia.

provided by the Python package Scikit-learn [40]. We initialized the following hyper-parameters as follows: random_state = 1234, max_iter = 1000, and class_weight = "balanced." The remaining hyper-parameters were set to their default values. The peptides were one-hot encoded before introduced to the logistic regression models. When comparing the performances of the two model types, it can be seen that MultiPep outperforms the logistic regression models (Tables 4 and 5 and Supplementary Tables S2–S9, S33, and S34). Nevertheless, the logistic regression models seem to have a generally better recall performance.

Prediction of FDA-approved therapeutic peptides

We wanted to test utility of MultiPep to classify FDA-approved therapeutic peptides. Thus, we generated a list of peptide sequences from THPdb [45], containing FDA-approved peptides. We ensured that none of the peptides were part of our training, validation, or test sets before we predicted. We predicted the example peptides from THPdb as is and did not use any additional information (Table 8). For the predictions, we used the average of all 10 CV models trained using our "save the individual network class-clade" training scheme. However, we also predicted the therapeutic peptides using the max score of the CV models (Supplementary Table S35). The threshold for shoving prediction scores was set to 0.15 (Table 9 and Supplementary Table S35).

Aprotinin is an inhibitor of plasmin, trypsin, chymotrypsin, kallikrein, thrombin, and activated protein C that is used to control bleeding during surgery [46]. However, MultiPep predicts that it has broad-spectrum antimicrobial properties, which is consistent with additional research on the effects of Aprotinin [47, 48].

Glatiramer acetate is a drug that is used to treat relapsing-remitting multiple sclerosis and has an average of 40–100 residues [49]. Interestingly, it has been found that Glatiramer acetate has efficient antibacterial properties [50], which is in line with the predictions of MultiPep. Moreover, MultiPep detected minor antiviral properties of Glatiramer acetate. However, this needs to be verified.

Lucinactant is used to treat respiratory distress syndrome in infants. It contains two phospholipids and a high concentration of sinapultide (also known as KL-4), a synthetic peptide designed to have similar activity to surfactant protein B [51, 52]. MultiPep classifies the peptide sinapultide as an antibacterial and hemolytic peptide with minor antimicrobial and toxic properties. Though Lucinactant is to be considered safe compared with alternative surfactants, it is associated with certain side-effects, such as transient pallor, dose interruption, and endotracheal obstruction [51], which may explain the toxic profile of sinapultide predicted by MultiPep. Further studies are needed to verify, if sinapultide has antibacterial properties.

Pramlintide is used for treatment of insulin-using patients with type 2 or type 1 diabetes mellitus [53]. Pramlintide is an analog of the neuroendocrine hormone amylin and it works via similar mechanisms [53]. To this end, MultiPep detected high association with the classes "neuropeptide" and "peptide hormone" which indicates that MultiPep was able to capture the general mechanism of action the peptide-drug.

Liraglutide, Teduglutide, and Metreleptin are analogs of GLP-1, GLP-2, and leptin, respectively [54–56]. They have all been predicted to belong to the classes "neuropeptide" and "peptide hormone," which is consistent with the general mechanism of action of GLP-1, GLP-2, and Leptin [57–60]. GLP-1 and GLP-2 have highly different bioactivity profiles [57], thus, it is exciting to see that GLP-2 has been predicted to have strong cell–cell signaling properties as well.

Bivalirudin is a direct thrombin inhibitor that is used to treat heparin-induced thrombocytopenia [61]. Interestingly, the MultiPep models have predicted that the peptide has both neuropeptide and peptide hormone properties. It will be interesting to see if these predictions can be verified by experimental procedures.

Sermorelin is a synthetic analog of growth hormone-releasing hormone (GHRH) and is used to treat children with growth hormone deficiency [62]. MultiPep classifies the peptide sequence of Sermorelin as a peptide hormone and a neuropeptide. This is fascinating when considering the general mechanism of action of the drug and when considering the findings that indicate that intravenous and subcutaneous sermorelin has been found to stimulate growth hormone secretion from the anterior pituitary [62].

The corticotropin from Questcor Pharmaceuticals contains a 39-amino-acid peptide natural form of adrenocorticotropic hormone (ACTH) [63]. It works by stimulating the adrenal cortex to secrete cortisol, corticosterone, aldosterone, and a few other weakly androgenic substances. In the body, corticotropin-releasing hormone (CRH) from the hypothalamus stimulates the release of ACTH from the anterior pituitary gland [63].

Table 9: Prediction of FDA-approved therapeutic peptides

Therapeutic peptides	Predictions	
Aprotinin (Trasylol, Bayer Pharmaceuticals, **Th1158**)	Antibacterial:	0.486
	Antimicrobial:	**0.627**
Glatiramer acetate (Copaxone, Teva Pharmaceutical Industries, **Th1113**)	**Antimicrobial:**	**0.815**
	Antivirus:	0.232
Lucinactant (*Surfaxin*, *Discovery Laboratories*, **Th1146**)	**Hemolytic:**	**0.631**
	Toxic:	0.321
	Antibacterial:	**0.977**
	Antimicrobial:	0.276
Pramlintide (Symlin, AstraZeneca, **Th1100**)	Cell–cell signaling:	0.430
	Neuropeptide:	**0.995**
	Peptide hormone:	**0.997**
Liraglutide (Saxenda, Novo Nordisk, **Th1124**)	Cell–cell signaling:	0.225
	Neuropeptide:	**0.893**
	Peptide hormone:	**0.982**
Teduglutide (Gattex, NPS Pharmaceuticals, **Th1137**)	**Cell–cell signaling:**	**1.0**
	Neuropeptide:	**0.999**
	Peptide hormone:	**1.0**
Bivalirudin (Angiomax, *The Medicines Company*, **Th1006**)	**Neuropeptide:**	**0.690**
	Peptide hormone:	**0.806**
Sermorelin (Sermorelin acetate, *Emd serono inc.*, **Th1157**)	**Neuropeptide:**	**0.812**
	Peptide hormone:	**0.997**
Metreleptin (Myalept, Amylin Pharmaceuticals, **Th1208**)	**Neuropeptide:**	**0.873**
	Peptide hormone:	**1.0**
Corticotropin (H.P. Acthar, Questcor Pharmaceuticals, **Th1104**)	**Neuropeptide:**	**0.839**
	Peptide hormone:	**0.999**
Thymalfasin (Zadaxin, SciClone Pharmaceuticals, **Th1110**)	Neuropeptide:	0.173
	Peptide hormone:	**0.567**
Aldesleukin (Proleukin, Chiron Corp., **Th1036**), **Anakinra** (Kineret, Amgen Inc., **Th1023**), **Filgrastim** (Neulasta, Amgen Inc., **Th1082**), **Interferon beta-1b** (Betaseron, Bayer, **Th1057**), **Interferon alfacon-1** (INFERGEN, Kadmon Pharmaceuticals, **Th1058**), **Interferon gamma-1b** (Actimmune, InterMune Inc., **Th1030**), **Peginterferon alfa-2a** (Pegasys, Hoffman-La Roche Inc., **Th1008**), **Oprelvekin** (Neumega, Genetics Institute Inc., **Th1033**), **Palifermin** (Kepivance, Amgen Inc., **Th1034**), and **Sargramostim** (Leucomax, Novartis, **Th1017**)	**Cytokines/growth factors:**	**1.0**

Notes: The names of the therapeutic peptides are written in Column 1 together with the peptides' brand names, designer company, and a link to the peptides in THPdb. The second column contains MultiPep's predictions. Predictions in bold are above threshold at 0.5. The predictions are based on the average of all 10 CV models. The threshold for showing prediction scores is 0.15.

Altogether, the predictions of MultiPep seem to be consistent with the general physiological properties of the hormone.

Thymalfasin is a synthetic analog of thymosin alpha 1 and is used for the treatment of chronic hepatitis B and C and as an immune enhancer for treating several other diseases [64]. Thymosin alpha 1 has a wide range of biological activities, which may explain its classification as a peptide hormone [64]. Further, MultiPep predictions indicate that the Thymalfasin may have minor neuropeptide properties. Remarkably, studies show that thymosin alpha 1 is found in the CNS of rats and can regulate the levels of nerve growth factor hormone and contribute to neurogenesis and cognition [65, 66].

Aldesleukin, anakinra, Interferon beta-1b, interferon alfacon-1, interferon-gamma-1b, Peginterferon alfa 2a, and Oprelvekin are recombinant versions of interleukins and interferons used to treat various illnesses [67–73]. Filgrastim, Palifermin, and Sargramostim are recombinant forms of human granulocyte colony stimulating factor, keratinocyte growth factor, and granulocyte–macrophage-colony stimulating factor, respectively [74–76]. All of these peptide drugs have been classified as "cytokines/growth factors," which indeed describe their overall peptide class.

Discussion

In this work, we construct and demonstrate the utility of a new multi-label classifier, MultiPep. Moreover, we test our novel loss function where BCE synergistically is merged with a customized version of the MCC function. Our results suggest that using our loss of function, instead of weighted BCE, is beneficial when training using large multi-label datasets containing an imbalanced class–size distribution. Whether this is a general tendency needs to be verified. Additionally, we use an innovative neural network architecture and a new training scheme to find optimal network parameters. We demonstrate that our novel training scheme on average produces models that are better than when saving all parameters of a network simultaneously whenever performance has increased. Further, we show that MultiPep on our data surpasses state-of-the-art peptide bioactivity classifiers, and that it can predict the general bioactivities

of FDA-approved therapeutic peptides. It should be noted that the comparisons with the other tools were made using our data only; thus, the results do not suggest that MultiPep is a better tool as such. If the classifiers were trained on the same data the outcome might have been different. The results demonstrate that MultiPep is better on the currently used data. This was consistent with our goal for the comparisons, namely to show how the different pre-trained models performed on a previously unseen dataset.

MultiPep is designed to be a tool that can grant scientists an overview of peptides' bioactivities. It does not offer a wealth of additional predictive information like other peptide bioactivity classifiers [9, 12–14]. Thus, we suggest that MultiPep can be used in a pipeline, where the predictions of MultiPep can be further evaluated using tools that have more narrow and specialized classification foci. All in the interest of the progress of research.

MultiPep is not the only tool that simultaneously can predict more bioactivity classes of peptides. Previous tools in the literature, PEPred-Suite and Peptipedia, also predicts multiple classes for peptides. The Peptipedia models performed rather poorly on the applied test-subset. As implied above, the Peptipedia models were trained on other data, and their bioactivity classes were probably defined differently. For the comparison, we deemed that a range of MultiPep bioactivity classes and Peptipedia bioactivity classes were comparable (Supplementary Table S32). We assumed that when the class names were similar or identical, then predictions of models trained on these classes would be somewhat overlapping. This did not seem to be the case. So, are the MultiPep and Peptipedia bioactivity classes just incomparable and of different origin? Though MultiPep's performance was better, the binary classifiers that were tested against MultiPep (Table 6) performed reasonably on our test subsets with our defined bioactivity classes. This indicates that MultiPep and these models agree on the definition of the used classes and that these bioactivity classes and included peptides contain essential class-specific features. Altogether, this suggests that the problem does not lie with the bioactivity classes defined in the data presented in this work. In addition, the peptide data used by MultiPep and Peptipedia stem from a number of identical databases [27]. Therefore, it is very unlikely that MultiPep's and Peptipedia's defined bioactivity classes are of completely different origin.

The difference in performance might rather be explained by the algorithms used to create the tools. PEPred-Suite and Peptipedia utilize individual RF models to predict bioactivity classes of peptides. MultiPep on the other hand uses a single deep learning-based model with sub-models to predict bioactivities of peptides. In other words, PEPred-Suite and Peptipedia solve many binary classification problems, whereas MultiPep turns it into a multi-label classification problem.

Another difference is that PEPred-Suite and Peptipedia use elaborate techniques to encode the peptide sequences, and MultiPep uses a simple one-hot encoding. Many peptide encoding techniques exist and peptide encoding in general is a field that is gaining a lot of attention [77, 78]. PEPred-Suite uses adaptive feature representation strategy, where they, among other things, use 10 feature encoding algorithms, which together efficiently capture local and global compositional information and well as position-specific residue information and physiochemical information [26]. Peptipedia encodes peptides using representations of physicochemical properties and transforms them using Fourier transforms [27, 79]. Although some general selection rules have been suggested, it is difficult to find a single universally optimal peptide encoding technique [77, 78]. As it has been found that deep learning models require little encoding for the classification process [77], we chose to use a simple and, in our opinion, reliable encoding technique where all amino acids are equally similar or dissimilar (one-hot encoding). We then leave it to our models to find patterns and amino acid relationships.

Though our loss function where we combine our customized MCC function with BCE worked well in our study, it still has some shortcomings. For example, the loss calculations are more robust when the target list for a mini-batch contains both $y_i = 1$ and $y_i = 0$, and not as robust when all targets are either $y_i = 1$ or $y_i = 0$. Still, what we presented here is a great start for directly including the MCC function as a loss function for data sets with imbalanced class distributions.

MultiPep's training data consist of peptides derived from various databases. The peptides have only been filtered based on size and uniqueness and not filtered based on general sequence homology. In theory, this may cause a minor inflation of the performances of some bioactivity classes, since similar peptides for a given class may be more easy to classify correctly. Also, in theory, this can hurt the generalizability of machine learning tools like MultiPep; however, we show via our 10-fold CV and prediction of associated test sets that our tool can generalize to unseen data and that it is robust when training on different data sets. Further, we show based on the comparisons with other state-of-the-art tools on sets from our test set and the prediction of FDA-approved therapeutic peptides that our tool produce sound and meaningful predictions.

MultiPep can take peptides of length 2–200 amino acid residues, which is a feature that is only matched by PeptideRanker and THPep. The minor peptide-length restrictions allow for ready classification of a long range of peptides without the need to filter by size; a step that otherwise may hamper the evaluation of bioactivity properties of peptides of interest.

For the predictions of the FDA-approved therapeutic peptides, we ensured that none of these peptides were part of our training, validation of test sets before we predicted any of them. However, for some of the therapeutic peptides, there were peptides of high similarity in the benchmark data set. For example, Liraglutide and Metreleptin are analogs of GLP-1 and leptin, respectively, and GLP-1 and leptin are both present in our training data set. Thus, it can be expected that the predicted bioactivity profiles of such therapeutic peptides would be similar to those of the peptides on which they are based.

Again, for the FDA-approved peptides, the average of all CV models' outputs was used for generating the predictions. We believe that the ensemble consisting of all CV models together will provide predictions that are more accurate. MultiPep is first of all a classifier, which, in this case, means that the prediction scores are supposed to be interpreted in a binary fashion with a threshold at 0.5. However, as shown with the predictions of the FDA-approved therapeutic peptides, predictions below threshold might still indicate that given peptides have properties associated with a certain class. For example, Thymalfasin was predicted by MultiPep as a neuropeptide but with a score below threshold. However, this peptide has indeed been found to be localized in the brain and to regulate different cerebral-associated mechanisms. This phenomenon is due to the fact that at least one of the CV models had found that Thymalfasin should be classified as a neuropeptide (Supplementary Table S35). Therefore, we have integrated functionalities that allow users of MultiPep to find the max predictions or the average of predictions for a given class based on all MultiPep CV models. This is

both true for the webtool and the stand-alone program. This will enable users to choose between whether they want all models to have a vote or if one is enough. MultiPep cannot directly detect degradation products in mass spectrometry-based peptidomics data. However, the tool can classify peptides into zero or up to 20 classes, outperform state-of-the-art binary classifiers on the used data, and meaningfully predict bioactivities of FDA-approved therapeutic peptides. Altogether, this enables MultiPep, like no other peptide prediction tool, to estimate whether a peptide may have a bioactivity or not. In conclusion, we present a valuable tool for the emerging field of peptide research and, at the same time, provide conceptual advance in applications of deep learning neural networks.

Supplementary data

Supplementary data are available at *Biology Methods and Protocols* online.

Data availability

All data in this study are available either through https://github.com/scheelelab/MultiPep or via direct communication with the authors. The webtool is available at https://agbg.shinyapps.io/MultiPep/.

Code availability

Data and scripts for generating input data for MultiPep, the MultiPep stand-alone program, all MultiPep versions and parameters used in this article, as well as sequences predicted by the compared bioactive predictors are available at https://github.com/scheelelab/MultiPep.

Author contributions

C.S. and A.G.B.G. conceived and designed the project. MultiPep was implemented, trained, and tested by A.G.B.G. The MultiPep stand-alone program and webtool were implemented by A.G.B.G. All analyses and comparisons were carried out by A.G.B.G. All authors read and contributed to the manuscript.

Funding

Novo Nordisk Foundation Center for Basic Metabolic Research is an independent Research Center, based at the University of Copenhagen, Denmark and supported by an unconditional donation from the Novo Nordisk Foundation (www.cbmr.ku.dk) (Grant number NNF18CC0034900).

Conflict of interest

The authors declare no conflict of interests.

References

1. de la Torre BG, Albericio F. Peptide therapeutics 2.0. *Molecules* 2020;**25**:2293.
2. Lau JL, Dunn MK. Therapeutic peptides: historical perspectives, current development trends, and future directions. *Bioorg Med Chem* 2018;**26**:2700–7.
3. Fosgerau K, Hoffmann T. Peptide therapeutics: current status and future directions. *Drug Discov Today* 2015;**20**. doi:10.1016/j.drudis.2014.10.003.
4. Lee ACL, Harris JL, Khanna KK *et al.* A comprehensive review on current advances in peptide drug development and design. *IJMS* 2019;**20**:2383–21. doi:10.3390/ijms20102383.
5. Cunningham R, Ma D, Li L. Mass spectrometry-based proteomics and peptidomics for systems biology and biomarker discovery. *Front Biol (Beijing)* 2012;**7**:313–35. doi:10.100/s11515-012-1218-y.
6. Fricker LD. Limitations of mass spectrometry-based peptidomic approaches. *J Am Soc Mass Spectrom* 2015;**26**:1981–91. doi:10.1007/s13361-015-1231-x.
7. Mooney C, Haslam NJ, Pollastri G *et al.* Towards the improved discovery and design of functional peptides: common features of diverse classes permit generalized prediction of bioactivity. *PLoS ONE* 2012;**7**:e45012.doi:10.1371/journal.pone.0045012.
8. Veltri D, Kamath U, Shehu A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* 2018;**34**:2740–7. doi:10.1093/bioinformatics/bty179.
9. Yan J, Bhadra P, Li A *et al.* Deep-AmPEP30: improve short antimicrobial peptides prediction with deep learning. *Mol Ther Nucleic Acids* 2020;**20**:882–94. doi:10.1016/j.omtn.2020.05.006.
10. Boopathi V, Subramaniyam S, Malik A *et al.* MACppred: a support vector machine-based meta-predictor for identification of anticancer peptides. *IJMS* 2019;**20**:1964.doi:10.3390/ijms20081964.
11. Bin Y, Zhang W, Tang W *et al.* Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features. *J Proteome Res* 2020;**19**:3732–40. doi:10.1021/acs.jproteome.0c00276.
12. Gupta S, Kapoor P, Chaudhary K *et al.*; Open Source Drug Discovery Consortium. In silico approach for predicting toxicity of peptides and proteins. *PLoS ONE* 2013;**8**:e73957.doi:10.1371/journal.pone.0073957.
13. Hasan MM, Schaduangrat N, Basith S *et al.* HLPpred-Fuse: improved and robust prediction of hemolytic peptide and its activity by fusing multiple feature representation. *Bioinformatics* 2020;**36**:3350–6. doi:10.1093/bioinformatics/btaa160.
14. Olsen TH, Yesiltas B, Marin FI *et al.* AnOxPePred: using deep learning for the prediction of antioxidative properties of peptides. *Sci Rep* 2020;**10**:21471.doi:10.1038/s41598-020-78319-w.
15. Wang G, Li X, Wang Z. APD3: the antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res* 2016;**44**:D1087–93. doi:10.1093/nar/gkv1278.
16. Roy S, Teron R. BioDADPep: a bioinformatics database for anti-diabetic peptides. *Bioinformation* 2019;**15**:780–3. doi:10.6026/97320630015780.
17. Singh S, Chaudhary K, Dhanda SK *et al.* SATPdb: a database of structurally annotated therapeutic peptides. *Nucleic Acids Res* 2016;**44**:D1119–26. doi:10.1093/nar/gkv1114.
18. Minkiewicz P, Iwaniak A, Darewicz M. BIOPEP-UWM database of bioactive peptides: current opportunities. *Int J Mol Sci* 2019;**20**. doi:10.3390/ijms20235978.
19. Tyagi A, Tuknait A, Anand P *et al.* CancerPPD: a database of anticancer peptides and proteins. *Nucleic Acids Res* 2015;**43**:D837–43. doi:10.1093/nar/gku892.
20. Waghu FH, Barai RS, Gurung P *et al.* CAMPR3: a database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Res* 2016;**44**:D1094–7. doi:10.1093/nar/gkv1051.
21. Gogoladze G, Grigolava M, Vishnepolsky B *et al.* DBAASP: database of antimicrobial activity and structure of peptides. *FEMS Microbiol Lett* 2014;**357**:63–8. doi:10.1111/1574-6968.12489.

22. Ye G, Wu H, Huang J *et al*. LAMP2: a major update of the database linking antimicrobial peptides. *Database* 2020;**2020**: 61.doi:10.1093/database/baaa061.
23. Kim Y, Bark S, Hook V *et al*. NeuroPedia: neuropeptide database and spectral library. *Bioinformatics* 2011;**27**:2772–3. doi: 10.1093/bioinformatics/btr445.
24. Wang Y, Wang M, Yin S *et al*. NeuroPep: a comprehensive resource of neuropeptides. *Database (Oxford)* 2015;**2015**: bav038.doi:10.1093/database/bav038.
25. Liu F, Baggerman G, Schoofs L *et al*. The construction of a bioactive peptide database in metazoa. *J Proteome Res* 2008;**7**: 4119–31. doi:10.1021/pr800037n.
26. Wei L, Zhou C, Su R *et al*. PEPred-Suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics* 2019;**35**:4272–80. doi: 10.1093/BIOINFORMATICS/BTZ246.
27. Quiroz C, Saavedra YB, Armijo-Galdames B *et al*. Peptipedia: a user-friendly web application and a comprehensive database for peptide research supported by machine learning approach. *Database* 2021;**2021**:1–9. doi: 10.1093/DATABASE/BAAB055
28. Alipanahi B, Delong A, Weirauch MT *et al*. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015;**33**:831–8. doi: 10.1038/nbt.3300.
29. Bjørnholt Grønning AG, Doktor TK, Larsen SJ *et al*. DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning. *Nucleic Acids Res* 2020;**48**:7099–118. doi: 10.1093/nar/gkaa530.
30. LeCun Y, Boser B, Denker JS *et al*. *Advances in Neural Information Processing Systems 2*. San Francisco (CA): Morgan Kaufmann Publ Inc., 1990, 396–404.
31. Maxwell A, Li R, Yang B *et al*. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC Bioinformatics* 2017;**18**. doi:10.1186/s12859-017-1898-z.
32. Zhang X, Zhao H, Zhang S *et al*. A novel deep neural network model for multi-label chronic disease prediction. *Front Genet* 2019;**10**:351.doi:10.3389/fgene.2019.00351.
33. Zou Z, Tian S, Gao X *et al*. mlDEEPre: multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Front Genet* 2019;**10**. doi:10.3389/fgene.2018.00714.
34. Abadi M, Barham P, Chen J *et al*. TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems, Design and Implementation, OSDI 2016, 2016, 265–83. Available at: http://arxiv.org/abs/1605.08695
35. Virtanen P, Gommers R, Oliphant TE *et al*.; SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* 2020;**17**:261–72. doi: 10.1038/s41592-019-0686-2.
36. He K, Zhang X, Ren S *et al*. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. arXiv: 1502.01852v1.
37. Kingma DP, Lei Ba J. Adam: a method for stochastic optimization. arXiv: 1412.6980v9.
38. Baldi P, Brunak S, Chauvin Y *et al*. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 2000;**16**:412–24. doi:10.1093/bioinformatics/16.5.412.
39. Wang S, Liu W, Wu J *et al*. Training deep neural networks on imbalanced data sets. Proceedings of the International Joint Conference on Neural Networks. Institute of Electrical and Electronics Engineers Inc., 2016, 4368–74. doi:10.1109/IJCNN.2016.7727770.
40. Pedregosa F, Varoquaux G, Gramfort A *et al*. Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.
41. Harris CR, Millman KJ, van der Walt SJ *et al*. Array programming with NumPy. *Nat Res* 2020; **585**:357–62. doi: 10.1038/s41586-020-2649-2.
42. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 2020;**21**:6.doi: 10.1186/s12864-019-6413-7.
43. Abhishek K, Hamarneh G. Matthews correlation coefficient loss for deep convolutional networks: application to skin lesion segmentation. Proceedings of the International Symposium on Biomedical Imaging, 2021 April, 2020, 225–9. doi:10.1109/ISBI48211.2021.9433782.
44. Shoombuatong W, Schaduangrat N, Pratiwi R *et al*. THPep: a machine learning-based approach for predicting tumor homing peptides. *Comput Biol Chem* 2019;**80**:441–51. doi: 10.1016/J.COMPBIOLCHEM.2019.05.008.
45. Usmani SS, Bedi G, Samuel JS *et al*. THPdb: database of FDA-approved peptide and protein therapeutics. *PLoS ONE* 2017; **12**:e0181748.doi:10.1371/journal.pone.0181748.
46. Mahdy AM, Webster NR. Perioperative systemic haemostatic agents. *Br J Anaesth* 2004;**93**:842–58. doi:10.1093/bja/aeh227.
47. Ibrahim H, Aoki T, Pellegrini A. Strategies for new antimicrobial proteins and peptides: lysozyme and aprotinin as model molecules. *Curr Pharm Des* 2002;**8**:671–93. doi: 10.2174/1381612023395349.
48. Pellegrini A, Thomas U, Bramaz N *et al*. Identification and isolation of the bactericidal domains in the proteinase inhibitor aprotinin. *Biochem Biophys Res Commun* 1996;**222**: 559–65. doi:10.1006/bbrc.1996.0783.
49. Weber MS, Hohlfeld R, Zamvil SS. Mechanism of action of glatiramer acetate in treatment of multiple sclerosis. *Neurotherapeutics* 2007;**4**:647–53. doi:10.1016/J.NURT.2007.08.002.
50. Christiansen SH, Murphy RA, Juul-Madsen K *et al*. The immunomodulatory drug glatiramer acetate is also an effective antimicrobial agent that kills Gram-negative bacteria. *Sci Rep* 2017;**7**:16.doi:10.1038/s41598-017-15969-3.
51. Garner SS, Cox TH. Lucinactant: new and approved, but is it an improvement? *J Pediatr Pharmacol Ther* 2012;**17**:206–10. doi:10.5863/1551-6776-17.3.206.
52. Lal MK, Sinha SK. Surfactant respiratory therapy using surfaxin/sinapultide. *Ther Adv Respir Dis* 2008;**2**:339–44. doi: 10.1177/1753465808097113.
53. Edelman S, Maier H, Wilhelm K. Pramlintide in the treatment of diabetes mellitus. *BioDrugs* 2008;**22**:375–86. doi: 10.2165/0063030-200822060-00004.
54. Meehan CA, Cochran E, Kassai A *et al*. Metreleptin for injection to treat the complications of leptin deficiency in patients with congenital or acquired generalized lipodystrophy. *Exp Rev Clin Pharmacol* 2016;**9**:59–68. doi:10.1586/17512433.2016.1096772.
55. Russell-Jones D. Molecular, pharmacological and clinical aspects of liraglutide, a once-daily human GLP-1 analogue. *Mol Cell Endocrinol* 2009;**297**:137–40. doi:10.1016/j.mce.2008.11.018.
56. Jeppesen PB. Teduglutide, a novel glucagon-like peptide 2 analog, in the treatment of patients with short bowel syndrome. *Therap Adv Gastroenterol* 2012;**5**:159–71. doi: 10.1177/1756283X11436318.
57. Janssen P, Rotondo A, Mulé F et al. Review article: a comparison of glucagon-like peptides 1 and 2. *Aliment Pharmacol Ther* 2013;**37**:18–36. doi:10.1111/apt.12092.
58. Klok MD, Jakobsdottir S, Drent ML. The role of leptin and ghrelin in the regulation of food intake and body weight in

humans: a review. *Obes Rev* 2007;**8**:21–34. doi:10.1111/j.1467-789X.2006.00270.x.
59. Drucker DJ. Glucagon-like peptide 2. *J Clin Endocrinol Metab* 2001;**86**:1759–64. doi:10.1210/jcem.86.4.7386.
60. MacDonald PE, El-Kholy W, Riedel MJ *et al*. The multiple actions of GLP-1 on the process of glucose-stimulated insulin secretion. *Diabetes* 2002;**51**:S434–42. doi:10.2337/diabetes2007.s434.
61. Sun Z, Lan X, Li S *et al*. Comparisons of argatroban to lepirudin and bivalirudin in the treatment of heparin-induced thrombocytopenia: a systematic review and meta-analysis. *Int J Hematol* 2017;**106**:476–83. doi:10.1007/s12185-017-2271-8.
62. Prakash A, Goa KL. Sermorelin: a review of its use in the diagnosis and treatment of children with idiopathic growth hormone deficiency. *BioDrugs* 1999;**12**:139–57.
63. Gettig J, Cummings JP, Matuszewski KHP. Acthar gel and cosyntropin review. *Pharm Ther* 2009;**34**:250–7. Available at: /pmc/articles/PMC2697107/.
64. Dominari A, Iii DH, Pandav K *et al*. Thymosin alpha 1: a comprehensive review of the literature. *World J Virol* 2020;**9**:67–78. doi:10.5501/wjv.v9.i5.67.
65. Turrini P, Aloe L. Evidence that endogenous thymosin alpha-1 is present in the rat central nervous system. *Neurochem Int* 1999;**35**:463–70. doi:10.1016/S0197-0186(99)00084-4.
66. Wang G, He F, Xu Y *et al*. Immunopotentiator thymosin alpha-1 promotes neurogenesis and cognition in the developing mouse via a systemic Th1 bias. *Neurosci Bull* 2017;**33**:675–84. doi:10.1007/s12264-017-0162-x.
67. Wilde MI, Faulds D. Oprelvekin: a review of its pharmacology and therapeutic potential in chemotherapy-induced thrombocytopenia. *BioDrugs* 1998;**10**:159–71. doi:10.2165/00063030-199810020-00006.
68. Rotte A, Bhandaru M, Zhou Y *et al*. Immunotherapy of melanoma: present options and future promises. *Cancer Metastasis Rev* 2015;**34**:115–28. doi:10.1007/s10555-014-9542-0.
69. Fleischmann RM, Schechtman J, Bennett R *et al*. Anakinra, a recombinant human interleukin-1 receptor antagonist (r-metHuIL-1ra), in patients with rheumatoid arthritis: a large, international, multicenter, placebo-controlled trial. *Arthritis Rheum* 2003;**48**:927–34. doi:10.1002/art.10870.
70. Peng F, Wang Y, Sun L *et al*. PEGylation of proteins in organic solution: a case study for interferon beta-1b. *Bioconjug Chem* 2012;**23**:1812–20. doi:10.1021/bc300081f.
71. Balan V, Rosati MJ, Anderson MH *et al*. Successful treatment with novel triple drug combination consisting of interferon-γ, interferon alfacon-1, and ribavirin in a non-responder HCV patient to pegylated interferon therapy. *Dig Dis Sci* 2006;**51**:956–9. doi:10.1007/s10620-006-9349-0.
72. Devane JG, Martin ML, Matson MA. A short 2 week dose titration regimen reduces the severity of flu-like symptoms with initial interferon gamma-1b treatment. *Curr Med Res Opin* 2014;**30**:1179–87. doi:10.1185/03007995.2014.899209.
73. Matthews SJ, McCoy C. Peginterferon alfa-2a: a review of approved and investigational uses. *Clin Ther* 2004;**26**:991–1025. doi:10.1016/S0149-2918(04)90173-7.
74. Damiani G, McCormick TS, Leal LO *et al*. Recombinant human granulocyte macrophage-colony stimulating factor expressed in yeast (sargramostim): a potential ally to combat serious infections. *Clin Immunol* 2020;**210**:108292.doi:10.1016/j.clim.2019.108292.
75. McDonnell AM, Lenz KL. Palifermin: role in the prevention of chemotherapy- and radiation-induced mucositis. *Ann Pharmacother* 2007;**41**:86–94. doi:10.1345/aph.1G473.
76. Dale DC, Crawford J, Klippel Z *et al*. A systematic literature review of the efficacy, effectiveness, and safety of filgrastim. *Support Care Cancer* 2018;**26**:7–20. doi:10.1007/s00520-017-3854-x.
77. Spänig S, Heider D. Encodings and models for antimicrobial peptide classification for multi-resistant pathogens. *BioData Min* 2019;**12**:1–29. doi:10.1186/S13040-019-0196-X.
78. Spänig S, Mohsen S, Hattab G *et al*. A large-scale comparative study on peptide encodings for biomedical classification. *NAR Genomics Bioinform* 2021;**3**. doi:10.1093/NARGAB/LQAB039.
79. Medina-Ortiz D, Contreras S, Amado-Hinojosa J *et al*. Combination of digital signal processing and assembled predictive models facilitates the rational design of proteins, 2020. Available at: https://arxiv.org/abs/2010.03516v1.