Journal Title Here, 2024, pp. 1–11

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

AutoPeptideML: Automated Machine Learning for Building Trustworthy Peptide Bioactivity Predictors

Raul Fernandez-Diaz,[1,2,3,*] Rodrigo Cossio-Pérez,[2,3,4] Clement Agoni,[2,3] Hoang Thanh Lam,[1] Vanessa Lopez[1] and Denis C. Shields[2,3,*]

[1]IBM Research, Dublin, IBM Technology Campus Damastown Industrial Park Mulhuddart, D15 HN66, Dublin, Ireland, [2]School of Medicine, University College Dublin, Belfield, D04 C1P1, Dublin, Ireland, [3]Conway Institute of Biomolecular and Biomedical Science, University College Dublin, Belfield, D04 C1P1, Dublin, Ireland and [4]Department of Science and Technology, National University of Quilmes, Roque Sáenz Peña 352, Bernal, B1876, Provincia de Buenos Aires, Argentina
[*]Corresponding authors. raul.fernandezdiaz@ucdconnect.ie, denis.shields@ucd.ie
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Automated machine learning (AutoML) solutions can bridge the gap between new computational advances and their real-world applications by enabling experimental scientists to build trustworthy models. We consider the design of such an AutoML tool for developing peptide bioactivity predictors. We analyse different design choices concerning data acquisition and negative class definition, homology partitioning for the construction of independent evaluation sets, the use of protein language models as a general sequence representation method, and model selection and hyperparameter optimisation. We have found that the definition of the negative class has a significant impact in the perceived performance of the models with differences up to 40%; the use of homology partitioning leads to more strict evaluation with up to 50% drops in perceived performance; the use of protein language models achieves state-of-the-art performance across different tasks; and the introduction of hyperparameter optimisation enables simpler machine learning models to perform similarly to more complex architectures. Finally, we integrate the conclusions drawn from this study into AutoPeptideML, an end-to-end, user-friendly application that enables experimental researchers to build trustworthy models, facilitating compliance with community guidelines. The source code, documentation, and data are available at `https://github.com/IBM/AutoPeptideML` and a dedicated web-server is available at `http://peptide.ucd.ie/AutoPeptideML`.

Introduction

Recent machine learning (ML) and deep learning (DL) algorithms have had tremendous impact on biochemistry research, particularly in the areas of drug discovery [34] and structural bioinformatics [23]. However, there are still technical challenges that need to be addressed before the full potential of these advances can be realised. They include working with small dataset sizes [3], sparsely labelled data [1], minimising sequence homology between training and evaluation sets to properly test the model generalisation capabilities [50], and a lack of general-purpose sequence representation methods [39].

One task that epitomises most of the key issues present in the prediction of biological function from sequence is peptide bioactivity prediction. Most available datasets are small and lacking in explicit negative labels, there is no standard representation method for peptide sequences, nor is there any standardised practice regarding control of sequence homology between training and evaluation sets [3]. Additionally, there is significant demand for custom ML predictors as there are many different bioactivities relevant for diverse applications ranging from drug discovery [51] to understanding metabolic or inflammatory signalling pathways [59, 16]. Currently, the development of models for new bioactivities is a tricky process that involves both ML expertise and domain knowledge [3].

Automated machine learning (AutoML) seeks to address these challenges by incorporating strategies that enable users with little or no ML knowledge to build high-quality custom predictors [21, 56]. Such a system would be able to assist researchers in designing more efficient experimental workflows informed by robust and reliable predictions [18] accelerating scientific discovery. A key feature of such automation is that it should integrate best practices and follow community guidelines in ML implementation [50] leading to more trustworthy models [24] that avoid the typical pitfalls of model over-prediction and lack of generalisability and produce reproducible and interpretable outputs.

Data acquisition and curation. The first step in the development of any ML model is obtaining and curating the data. The main challenge regarding data acquisition for peptide bioactivity prediction is the lack of explicit negative samples. There is a variety of strategies proposed for addressing this problem that include drawing random fragments from proteins

© The Author 2024. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

in databases like UniProt or SwissProt [1, 27, 4], drawing actual random peptides from the same databases [36, 32, 10, 1], and using peptides with a specific bioactivity as the negative class [1, 6, 31]. In all cases, there are important tradeoffs regarding the generalizability of the model. In the first and second approaches, the differential features between positive and negative classes could be explained by a myriad of confounding factors that do not have a direct bearing on their bioactivity. In the third approach, the opposite is true, positive and negative classes are so similar to each other that the model hyper-focuses on the specific differential features between both bioactivities, which hinders its generalizability. If the evaluation set has been constructed following any of these strategies, it will inherit the same biases and thus will not be able to properly diagnose them. Here, we explore an intermediate solution: to draw the negative peptides from a database with multiple bioactivities, instead of a single bioactivity. This approach generates a distribution of negative peptides that is broad enough (by covering several distinct bioactivities) as to generalise adequately, but that is similar enough to the positive peptide distribution (by also being bioactive peptides) as to minimise confounding factors.

Sequence representation. The use of domain knowledge for building effective peptide representations can simplify the process of training ML models and bypass some of the limitations inherent to small sample sizes [20]. There exists a plethora of different representations that can be used based on peptide composition [31, 4, 9, 7, 5], evolutionary profile [53], and physico-chemical profile [36, 4, 7, 9]. The emergence of sequence language models for either DNA [22] or protein sequences [25] has the potential of streamlining sequence representation beyond those methods. The sequence language models are DL algorithms trained to predict masked monomers (nucleotides or residues) in a sequence allowing the models to capture the conditional probability that any given monomer will occupy a certain position given the rest of the sequence. Consequently, they acquire the ability to provide high-dimensional vector representations of the sequences containing implicit structural, functional, and evolutionary properties. It is important to note that the data used during this process relies on sequence databases like UniRef [45] which do not contain explicit information about their properties, rather just nucleotide or residue distributions [25]. Although peptide sequences are less constrained than the bulk of their training data, recent studies [14, 13] have demonstrated that protein language model (PLM) representations obtained from models from the ESM (Evolutionary Scale Modelling) family [40, 25], can be used to train ML algorithms that predict peptide bioactivity with remarkable success, which led us to select them as a general representation method.

Independence of evaluation dataset. The independence between training and evaluation datasets is essential to properly evaluate any ML predictor [50]. Achieving this independence, however, is not straight-forward when working with biological sequences, because similar sequences tend to share structural and functional features [42]. The standard practice in protein ML development is to perform homology reduction which is a technique that clusters the sequences using algorithms like CD-HIT [17] to then preserve only the representative sequences from each cluster which is particularly useful when working with large data sets. However, when working with small datasets, the resulting loss of information hinders the models substantially [12]. Homology partitioning is an alternative approach that removes the minimal number of sequences to ensure that each cluster is independent from the others [47]. Here, we propose an alternative partitioning algorithm that generalises well to short sequences and maximises the difference between training and evaluation sets.

We have combined these three approaches above (bioactive sampling of negatives, PLM representation, and homology partitioning) into AutoPeptideML, a user-friendly AutoML application that allows researchers without a strong computational background to build their own peptide bioactivity predictors easily, while supporting their reproducibility and robust evaluation. The tool is made available as an open source project under MIT licence and a dedicated web-server. We demonstrate that the application of the method has good performance against a series of benchmark peptide bioactivity prediction datasets, evaluating not only performance, but also the effect of different design choices on performance which can be particularly marked in smaller datasets. Thus, the software not only provides a facility for biologists to make predictions from peptide datasets, but also encourages compliance with, and appreciation of, best practices in machine learning.

Methods

Algorithm

The primary objective behind the design of AutoPeptideML is to provide a user-friendly tool that does not require extensive technical knowledge to use, while still remaining highly versatile. This is achieved through a pipeline that guarantees compliance with community guidelines such as DOME (Data, Optimisation, Model, and Evaluation) [50], ensuring a robust scientific validation (see Supplementary). A visual summary of the workflow can be found in Figure 1.

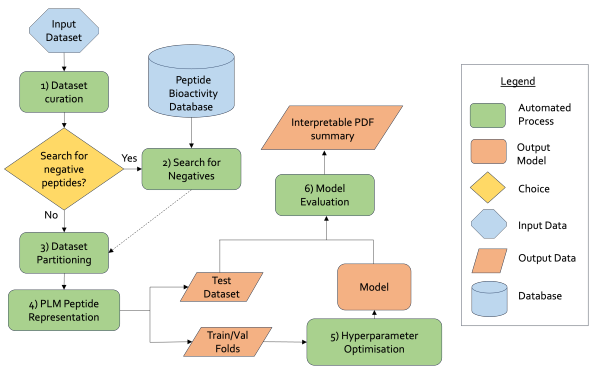

Fig. 1. Visual summary of the AutoPeptideML workflow.

Inputs. AutoPeptideML only requires a dataset of peptides known to be positive for the bioactivity of interest.

Outputs. AutoPeptideML generates a directory containing the generated model, along with all the necessary information to understand its capabilities and reproduce its training. The output contains: 1) the experimental configuration used to run AutoPeptideML, 2) all the intermediary datasets generated by the program, 3) the optimal configurations found during hyperparameter optimisation, 4) the evaluation metrics for both the n-fold cross-validation and against the holdout independent evaluation set, including a set of visualisations to facilitate interpreting the results, 5) the predictive model itself, and 6) an interpretable summary including all information regarding figures and main metrics

with accompanying explanations of their meaning to guide researchers not specialised in ML on how reliable the model generated is.

Step 1 - Dataset curation. Dataset curation preprocesses the data to remove all sequences that are either repeated or contain non-standard amino acids. If both positive and negative peptides are provided, the program balances the classes by oversampling the underrepresented class and continues to Step 3; if negative peptides are not provided, it executes Step 2 to build the negative set.

Step 2 - Automatic search for negative peptides. The automatic search for negative peptides queries a database of bioactive peptides. To avoid introducing false negative peptides into the negative subset, the algorithm accepts an optional input containing a list of bioactivity tags that the user considers may overlap with the bioactivity of interest and should, therefore, be excluded. To ensure that the negative peptides are drawn from a similar distribution to the positive peptides and minimise the number of confounding factors, the system calculates the lengths of all peptides in the positive peptides and organises the lengths into a histogram with bin size of 5. Then, for each bin in the histogram, it queries the reference database for as many peptides as present in the bin, with lengths between its lower and upper bounds. If there are not enough peptides in the database, the remaining peptides are drawn from the next bin.

Regarding the reference database, the system uses a pre-prepared subset of the Peptipedia database [37], named AutoPeptideML-Peptipedia. The original Peptipedia database integrates information from 30 peptide bioactivity databases collecting almost 97,331 bioactive peptides labelled with 128 bioactivities (version `29_03_2023`). AutoPeptideML-Peptipedia is the result of removing all sequences with non-standard residues or without any known bioactivity and contains 92,092 peptides (see Supplementary).

Step 3 - Dataset Partitioning. Dataset partitioning divides a dataset into an overall training dataset, validation datasets drawn from within the training dataset using tenfold cross-validation, and an evaluation (test) set. The evaluation set is only evaluated once in the process, to assess the performance of the final choice of model [50]. By default, the proportion of training:evaluation data is 80:20. The division is performed in two subsequent steps.

The first division relies on a new homology partitioning algorithm, which concers three steps. First, it computes pairwise alignments between all sequences in the dataset and uses the sequence identity measurements obtained to construct a pairwise similarity matrix. Local alignments (Smith-Waterman [43]) are computed using `MMSeqs2` [44] with and without prior k-mer prefiltering and global alignments (Needleman-Wunch [30]), with `EMBOSS` [41]. Second, the pairwise matrix is used to cluster the sequences using the connected components algorithm as implemented in `NetworkX` [19]. Third, it iteratively assigns the smallest clusters to the evaluation partition. During this step, the balance between positive and negative samples is checked before assigning clusters to the evaluation partition so that only clusters that would not lead to an imbalanced partition are considered. The underlying idea is that the clusters with fewer neighbours are more dissimilar to the main bulk of sequences in the dataset. Therefore, the evaluation set built with the most dissimilar sequences is not only independent from the train set, but it is a more close approximation to an out-of-distribution evaluation. Furthermore, by exploiting the topology of the graph rather than manipulating the number of nodes, the method does not require the removal of any sequences, minimising the loss of information.

The second division concerns the training set which will be subdivided into n cross-validation folds. Two alternatives are implemented: random stratified n-fold division (as implemented in `scikit-learn` [35]) or the implementation of stratified n-fold homology partitioning in `GraphPart` [47]. Independence between training and validation folds is not as essential as between training and evaluation sets as the main objective of the validation set is to ensure that the algorithm is learning from the training examples. Therefore, it is reasonable for the validation set to belong to the same distribution as the training set and, consequently, the default option is the stratified tenfold random division.

Step 4 - Peptide Representation. The ESM-2 model with 8 million parameters (ESM2-8M) ([25] is used to compute a representation for each residue in the peptide, using `Transformers` [54] and `PyTorch` [33]. The resulting peptide representation is a matrix with dimension $(n, 320)$ where n is the length of the peptide and 320 the representation size which is determined by the PLM architecture. A global representation for the whole sequence is then calculated by averaging across the individual residue representations for a final representation with dimension $(1, 320)$ [26].

The selection of this model is the result of an evaluation of the effect of model size on downstream performance (see Subsection 3.3). Nevertheless, users have the option to input their own peptide representations as a JSON file that may contain handcrafted features, learnt representations obtained from a different PLM or other type of protein representation learning algorithm [58], or a combination thereof.

Step 5 - Hyperparameter Optimisation and Model Training. Model selection and hyperparameter optimisation (HPO) are tedious and time-consuming tasks that require extensive trial-and-error [14, 3], and if not performed appropriately, can lead to model overfitting and lack of generalisability. AutoPeptideML automates and streamlines model development, by performing a greedy Bayesian Optimisation Search, as implemented by `Optuna` [2]. Users of either the CLI or the API are free to define the number of models that should be included in this optimization, as well as their hyperparameter search space. AutoPeptideML supports the following algorithms: K-nearest neighbours (KNN), light gradient boosting (LightGBM), support vector machine (SVM), random forest classification (RFC), extreme gradient boosting (XGBoost), simple neural networks like the multi-layer perceptron (MLP), and 1D-convolutional neural networks (1D-CNN). Model selection and HPO are conducted simultaneously in a cross-validation regime so that the metric to optimise is the average across n folds. Thus, the system is never exposed to the evaluation set, which is kept unseen until the final model evaluation [50].

Step 6 - Model Evaluation. The ensemble obtained in the previous step is evaluated against the hold-out evaluation set using a wide range of metrics that include accuracy, balanced accuracy, weighted precision, precision, F1, weighted F1, recall, weighted recall, area under the receiver-operating characteristic curve, matthew's correlation coefficient (MCC), jaccard similarity, and weighted jaccard similarity as implemented in the `scikit-learn` [35]. The plots generated include calibration curve, confusion matrix, precision-recall curve, and receiver-operating characteristic curve, as implemented in `sckit-plot` [29].

Step 7 - Prediction. AutoPeptideML can predict the bioactivity of new samples given a pre-trained model generated in Step 5. Predictions are a score within the range $[0, 1]$. This result can be interpreted as the probability of the peptide sequence having the target bioactivity, given the predictive model ($P(x \cap +|\text{MODEL})$). This step outputs a CSV file with the peptides sorted according to their predicted bioactivity probability.

Benchmark datasets and model implementation

Benchmarks. 16 different peptide bioactivity datasets containing positive and negative samples were used to evaluate the various modules of the algorithm. These datasets were selected from a previous study, evaluating the use of PLMs for general peptide bioactivity prediction [14]. The datasets ranged in size from 200 to 20,000. In order to evaluate homology partitioning (HP) and automated selection of negatives, two new benchmark sets were constructed from these original datasets named "NegSearch" and "NegSearch+HP". In both datasets positive samples were taken from the original bioactivity sets (see Table 1), and the negative peptides were automatically selected from other bioactivity classes, as described in Step 2. In the "NegSearch" sets, training and evaluation subsets were divided by random splitting, whereas in "NegSearch+HP", they were divided using the new homology partitioning algorithm described in Step 3. Therefore, "NegSearch" and "NegSearch+HP" only differ on the partitioning method employed and were composed of the same sets of peptides. In the original benchmarking datasets (see Table 1), the two anticancer and the two antimalarial sets respectively shared the same positive peptides and only differed on how the negative classes were defined.

AutoPeptideML configuration. Dataset partitioning used MMSeqs2 [44] with prior k-mer prefilter for calculating all pairwise sequence alignments in the train/evaluation division and random partitioning for creating 10 cross-validation folds. Sequence identity was calculated using the longest sequence as denominator and the maximum identity allowed between sequences in training and evaluation was 30% using. Automatic hyperparameter optimisation was performed separately for KNN, LightGBM and RFC and all models were ensembled. Thus, the final ensemble contained 10 instances (one per cross-validation fold) of each of these models for a total of 30 models. Final model predictions were the average of all 30 individual predictions.

Statistical Analysis. All statistical analyses were performed using the corresponding functions in `Pandas` [28], as detailed in the accompanying code.

UniDL4BioPep-A. We wished to compare our findings with a state-of-the-art framework, UniDL4BioPep [14], which trained a convolutional neural network on ESM2-8M representations for peptide bioactivity prediction. Analysis of the source code in the UniDL4BioPep project GitHub Repository [`https://github.com/dzjxzyd/UniDL4BioPep`] indicated that the hold-out test set is used during model training to determine the best checkpoint for evaluation. However, DOME guidelines recommend avoiding this practice [50], as it compromises the independence of the hold-out evaluation dataset, which can lead to the overestimation of model performance. Accordingly, an amended version of the UniDL4BioPep model, which we named UniDL4BioPep-A, was used. This version instead used the 10-fold cross-validation to determine the best possible checkpoint, maintaining the hold-out set unseen until final model evaluation. This implementation is written in PyTorch and was adapted from code provided by the UniDL4BioPep authors in a secondary project Github Repository `https://github.com/David-Dingle/UniDL4BioPep_ASL_PyTorch`.

Analysis of sequence homology between train and evaluation sets

The dependence between training and evaluation sets in both the original datasets and the new ones generated throughout the study was evaluated by first calculating pairwise alignments between the sequences in either set using MMSeqs2 (Version: `14-7e284`) [44] with prior k-mer prefiltering. The proportion of the proteins within the training dataset with at least one homologous protein in the evaluation set is used as independence measurement. Here, we considered that two proteins are homologous if they have a sequence identity above 30% using the longest sequence as denominator.

Results and discussion

AutoPeptideML is the first end-to-end AutoML solution proposed for building peptide bioactivity ML predictors. It combines three features: automatic selection of negatives from a background dataset of bioactive peptides to increase specificity, homology partitioning of datasets to evaluate generalisability, and implementation of a protein language model representation to streamline representation. One property of datasets of bioactive peptides is that they are often quite small. Thus, computational methods that embed knowledge from outside the training dataset, that are relevant to training, can serve to strongly streamline the efficiency of model estimation, reducing stochastic elements of parameter choice. We leveraged the ESM-8M PLM [25] in AutoPeptideML to generate a representation of each peptide as a vector of length 320, which is used as the set of features for a downstream model.

Benchmark construction analysis

Analysis of original benchmarks. An analysis of the partition strategies used to construct the original benchmarks (see Table 1) shows that among the 18 datasets, only 8 consider the homology relationships between training and evaluation sets, of which four of them [36] perform an approximation to homology maximisation between the training and evaluation sets, by clustering the peptides according to their feature similarities and assigning 80% of the samples in each cluster to training and 20% to testing, and the remaining randomly assign peptides to either train or evaluation. The problem with homology maximisation is that it deliberately increases the non-independence of training and evaluation sets, thus impeding the proper evaluation of generalisability or the diagnosis of over-fitting. Another four studies perform homology reduction at a very high similarity threshold, which has the advantage of not reducing the size of the dataset for training, but the disadvantage of not significantly increasing the independence of the evaluation set. To consider the effects of the homology partitioning we constructed a second set of datasets (termed NegSearch+HP), where homology partitioning (see Methods - Algorithm Step 3) strongly minimised similarity between training and evaluation datasets.

Table 4 indicates that for 13 of the 18 original datasets, at least 10% of the peptides in the training set share homology

Table 1. Original benchmark datasets. AB: Antibacterial; ACE: ACE inhibitor; AC: Anticancer; AF: Antifungal; AMA: Antimalarial; AMI: Antimicrobial; AO: Antioxidant; AP: Antiparasitic; AV: Antiviral; BBB: Brain-blood barrier crossing; DPPIV: DPPIV inhibitor; MRSA: Anti-MRSA; NP: Neuropeptide; QS: Quorum sensing; TOX: Toxic; TTCA: Tumor T-cell antigens; Ref: Reference to best reported model in the literature.

Dataset	Negative Class	Partitioning	Ref
AB [36]	Random peptides	Homology maximisation	[36]
ACE [27]	Random protein fragments	Homology reduction (90%)	[27]
AC1 [1]	Antimicrobial peptides	Random	[5]
AC2 [1]	Random protein fragments	Random	[1]
AF [36]	Random peptides	Homology maximisation	[36]
AMA1 [10, 55]	Random peptides	Random	[10]
AMA2 [10, 1]	Random protein fragments	Random	[10]
AMI [36]	Random peptides	Homology maximisation	[36]
AO [31]	Experimental + random peptides	Homology reduction (90%)	[31]
AP [57]	Random peptides	Homology reduction (90% for positives and 60% for negatives)	[57]
AV [36]	Random peptides	Homology maximisation	[36]
BBB [12]	Random peptides	Homology reduction (90%)	[12]
DPPIV [6]	Random + Bioactive	Random	[8]
MRSA [7]	Random peptides	Homology reduction (80%)	[7]
NP [4]	Random protein fragments	Homology reduction (90%)	[11]
QS [38]	Random peptides	Random	[52]
TOX [53]	Random peptides	Random	[53]
TTCA [9]	T-cell antigens not associated to disease	Random	[9]

with at least one sequence in the evaluation dataset, which compromises their independence. Looking at the partitioning strategies used to generate the datasets, ACE inhibitor, Antioxidant, Antiparasitic, Anti-MRSA, and Neuropeptides all have been constructed with homology reduction strategies with high threshold 80-90%. Of the five datasets, there are two with only 1% of their training sequences being homologous to at least one sequence in the evaluation set, thus showing how homology reduction strategies, although insufficient for fully ensuring the independence, can, nonetheless, reduce the magnitude of the problem. The datasets built with the approximation to homology maximisation (Antibacterial, Antifungal, Antimicrobial, and Antiviral) all show relatively high proportion of homologous sequences in training (26-47%) as expected. Finally, the middle column (Table 4) shows that this effect is similar when the negatives are automatically chosen from other bioactivity classes. However, when homology partitioning is applied, the dependence between training and

Table 2. Proportion of sequences in training set with at least one homolog in the evaluation set. We consider a sequence to be homolog to another when their sequence identity is superior to 30%. NS: AutoPeptideML's negative search module (Algorithm - Step 2); HP: AutoPeptideML's homology partitioning algorithm (Algorithm - Step 3).

Dataset	Original	NS	NS+HP
Antibacterial	0.36	0.64	0.0
ACE inhibitor	0.01	0.03	0.0
Anticancer 1	0.59	0.50	0.0
Anticancer 2	0.30		
Antifungal	0.34	0.58	0.0
Antimalarial 1	0.41	0.15	0.0
Antimalarial 2	0.10		
Antimicrobial	0.47	0.63	0.0
Antioxidant	0.01	0.0	0.0
Antiparasitic	0.11	0.32	0.0
Antiviral	0.26	0.44	0.0
Blood-brain barrier	0.04	0.04	0.0
DPPIV inhibitor	0.05	0.04	0.0
Anti-MRSA	0.15	0.34	0.0
Neuropeptide	0.24	0.24	0.0
Quorum sensing	0.18	0.09	0.0
Toxicity	0.56	0.56	0.0
Tumor T-cell antigens	0.00	0.03	0.0

evaluation sets is largely eliminated, with no peptide pairs displaying a similarity above 30%.

Benchmarks construction. We applied the AutoPeptideML automated analysis to the benchmark datasets, using the default settings for negative selection, homology partitioning, hyperparameter optimisation and protein language model representation. As the automatic search for negatives is a random process, and this stochasticity can influence model performance, every experiment was performed in triplicate. Figure 2, shows that the original datasets in general appear to outperform the automatically assigned peptides in terms of predictive performance. Thus, the definition used for the negative class has a significant impact on the perceived performance of the models. A closer inspection of the results highlights this observation. Particularly interesting are the pairs of datasets Anticancer 1 and 2, and Antimalarial 1 and 2 which have been created with the same set of positive peptides and differ on the definition used in the original benchmark [1] to create the negative set. Anticancer 1 defines its negatives as peptides with antimicrobial activity, which is a bioactivity that overlaps heavily with the positive class [48], thus explaining the comparatively low performance achieved when compared with Anticancer 2, which draws its negative set from a collection of random peptides. Similarly, Antimalarial 1 draws its negative class from a distribution of peptides from UniProt which is a narrower distribution than the one used in Antimalarial 2, random protein fragments [10, 55]. These results indicate that broad negative class definitions lead to better perceived model performance, even though this may not reflect their real world-application.

The definition used in this study to generate the new datasets is a middle point between Anticancer 1 and 2. This is reflected in the drop in apparent performance observed in most datasets with the exception of Anticancer 1 (which has a narrower definition) and Tumour T-cell antigens (which also use a narrower negative class definition, namely, T-cell antigens not known to be associated with any disease). The Antiviral

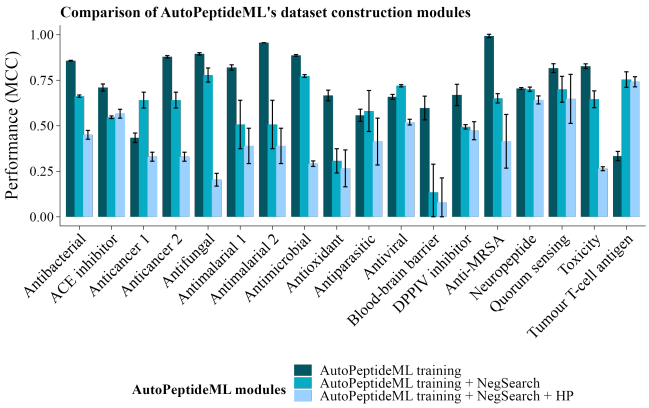

Fig. 2. Evaluation of AutoPeptideML's dataset construction modules. Error bars reflect the standard deviation across three replicates. NegSearch: AutoPeptideML's negative search module (Algorithm - Step 2); HP: AutoPeptideML's homology partitioning algorithm (Algorithm - Step 3).

dataset is the only case where performance increased after narrowing the negative class definition and the reason behind this discrepancy remains unclear.

This hypothesis is further supported by the results of Table 2 showcasing the independence between training and evaluation sequences. The dataset with the new bioactive negative peptides for dataset Anticancer 1 has 50% of its training sequences with at least one homolog in the evaluation set, this value is an intermediate point between the original Anticancer 1 and Anticancer 2 (59% and 30%) which is a difference that reflects the further difference in apparent performance in Figure 2. Overall, these results show that using bioactive peptides as a negative class generally leads to more challenging classification problems than distinguishing between a certain bioactivity and random peptides. Generally speaking, the definition of negative class depends on the problem the researcher wants to tackle specifically, however, for the purposes of an automated tool, the proposed definition balances specificity with usability and interpretability.

Regarding homology partitioning, it manifests as a drop in perceived model performance. This indicates that alternative methods tend to overestimate model performance by not properly diagnosing model overfitting. However, a thorough comparison among alternative homology reduction and partitioning methods remains an open research question.

Effect of HPO and model ensemble on performance

The current approaches for peptide bioactivity predictors can be divided into 2 categories: dataset-specific models [3] and general model development frameworks [14]. The former, as the name suggests, refers to those models that have been designed for a specific dataset while the latter, comprises general approaches that can be applied to any given dataset. AutoPeptideML would belong to the second category. UniDL4BioPep [14] is another algorithm that can be regarded as a general-purpose framework in this domain. Therefore, it was used as a baseline against which AutoPeptideML can be compared.

An inherent problem when comparing algorithms is that the data used to train and evaluate the models have to be the same. To do this a custom implementation of UniDL4BioPep was prepared, named UniDL4BioPep-A (see Methods) which selects the best checkpoint out of a 10-fold cross-validation. The training module of AutoPeptideML was evaluated against UniDL4BioPep-A in both the old and the new sets of benchmarks. The new benchmarks were again partitioned using both the random and homology partitioning strategies, to determine whether using an ensemble of simple ML models (AutoPeptideML) was a better strategy for avoiding overfitting in small datasets than the more complex CNN model (UniDL4BioPep-A). The self-reported values for the handcrafted models referenced in Table 1 are included with the evaluation in the original set of benchmarks to contextualise the contributions of both general purpose frameworks.

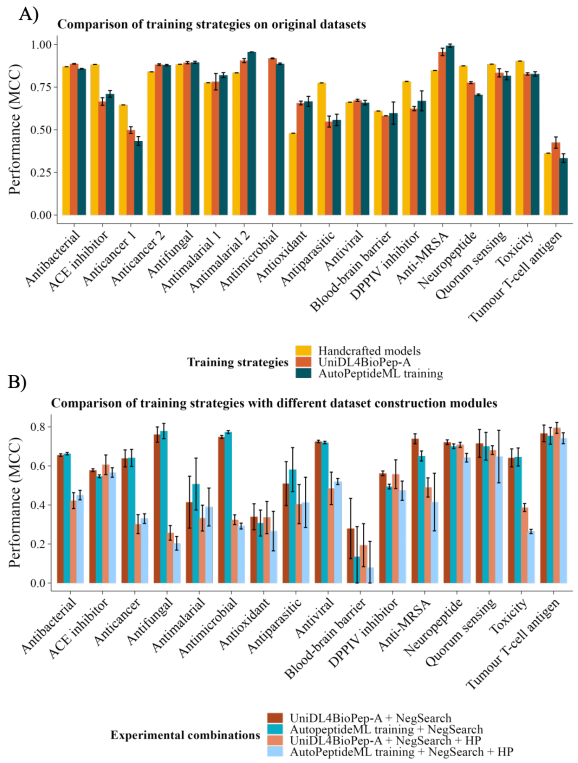

Fig. 3. A: Comparison of training strategies on original datasets. B: Comparison of training strategies with different dataset construction modules. Error bars reflect the standard deviation across three replicates. NegSearch: AutoPeptideML's negative search module (Algorithm - Step 2); HP: AutoPeptideML's homology partitioning algorithm (Algorithm - Step 3).

Figure 3.A shows that when applied to a literature derived benchmark set of positive and negative datasets, the two general purpose PLM-enabled bioactivity predictors have a performance comparable with the literature self-reported performance of models handcrafted for each specific dataset. Secondly, the two general purpose predictors, one of which is based on a convolutional neural network (UniDL4BioPep-A), and the other on an ensemble of three simpler ML models (AutoPeptideML), both have comparable performance.

When compared with Fig 3.A, Figure 3.B shows that when automatic negative selection from a bioactive dataset are introduced, both ML and DL general purpose models show an equivalent drop in apparent performance, reflecting the more challenging task of predicting specificity versus other

bioactives. This performance drops further in both models when homology partitioning is introduced, illustrating the effect of the lack of independence on inflating the results on the literature. Remarkably, there is no significant evidence of greater overfitting on the part of the DL model, despite the small dataset sizes, this might be due to the relatively small number of parameters within the 1D-CNN. Both approaches are better than the other in 8 out 16 datasets, showing how similar their performance is. Overall, these results allow us to conclude that the hyperparameter optimisation enables the ensemble of ML models to achieve comparable performance to a more complex DL model while being more computationally efficient.

Effect of ESM-2 model size on downstream performance

Recent studies focused on language model development have reported that there exists a direct correlation between model size and downstream performance on natural language processing tasks [46, 49]. The same phenomenon has been observed for PLMs for different protein representation related tasks [25]. While PLMs have been utilised in predicting peptide bioactivity [14, 13], the relationship between model size and downstream performance has not been previously explored.

We evaluated four different PLM models from the ESM family with increasing size: ESM2-8M (8M parameters), ESM2-35M (35M parameters), ESM2-150M (150M parameters), ESM2-650M (650M parameters). We also evaluated ESM1b-650M (650M parameters), from a previous version of ESM. We performed the comparisons with auto-negative and homology partitioning strategies in place.

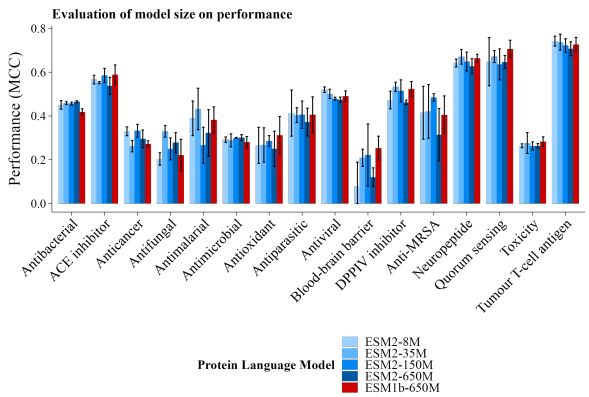

Fig. 4. Model size effect on performance. Error bars reflect the standard deviation across three replicates.

Figure 4 shows that there is no significant difference between models across all datasets and no correlation between model size and performance. Consequently, the smallest model is a valid default representation method. This result does not align with prior literature that describes improved downstream performance of protein predictors as model size increases [25, 15]. There are two factors, however, that may explain this apparent incongruity. First, peptide sequences are less constrained than that of proteins and, therefore, the evolutionary rules available to the larger models may not be as relevant for informing downstream models. Secondly, the shorter length of peptide sequences limits the context available for the attention mechanism within the PLM model, thus, reducing the importance of model size.

Conclusion

AutoPeptideML is a computational tool that allows non-specialised users to develop ML models for the prediction of peptide bioactivity and facilitates compliance with ML community guidelines for the life-sciences. It is able to handle several key steps in the peptide bioactivity predictor development life-cycle including: 1) automatic search for negative samples, 2) dataset partitioning using a novel algorithm for homology partitioning, 3) hyperparameter search optimisation, 4) evaluation of soft-voting ensemble with best simple ML models generated, and 5) prediction of new samples. Further, the output is generated in the form of a PDF summary easily interpretable by researchers not specialised in ML; alongside a directory that ensures reproducibility by containing all necessary information for re-using and re-training the models.

The definition of the negative class used for building peptide bioactivity datasets has a significant impact on the model performance of up to 40% and has to be controlled in order to properly understand the predictions of the models being built. The partitioning strategy employed to define training and evaluation sets impacts model generalisability and thus has an impact on apparent performance of models of up to 50%.

The training module in AutoPeptideML reaches state-of-the-art performance when compared both to an alternative general-purpose-framework and dataset-specific, handcrafted models across a set of 18 different datasets. Furthermore, there is no significant difference between using an ensemble of simple ML algorithms (AutoPeptideML) and more complex DL algorithms (UniDL4BioPep-A), even though the former is more computationally efficient. Finally, PLM model size does not affect significantly downstream performance on the peptide bioactivity datasets evaluated in this study.

The foundational principles underlying the issues described and solutions implemented throughout this study are relevant for the application of trustworthy ML predictors for any other biosequence (e.g., DNA, RNA, proteins, peptides, DNA methylation, etc.) and their automation facilitates the rigorous evaluation of new models by researchers not specialised in ML.

Acknowledgments

The authors would like to thank Silvia González López for designing the AutoPeptideML logo and Marcos Martínez Galindo for his technical assistance during the deployment of the AutoPeptideML webserver.

A. Peptipedia length distribution

Figure 5 describes the distribution of the lengths of the peptides comprising the subset of Peptipedia generated in this study and named "AutoPeptideML-Peptipedia".

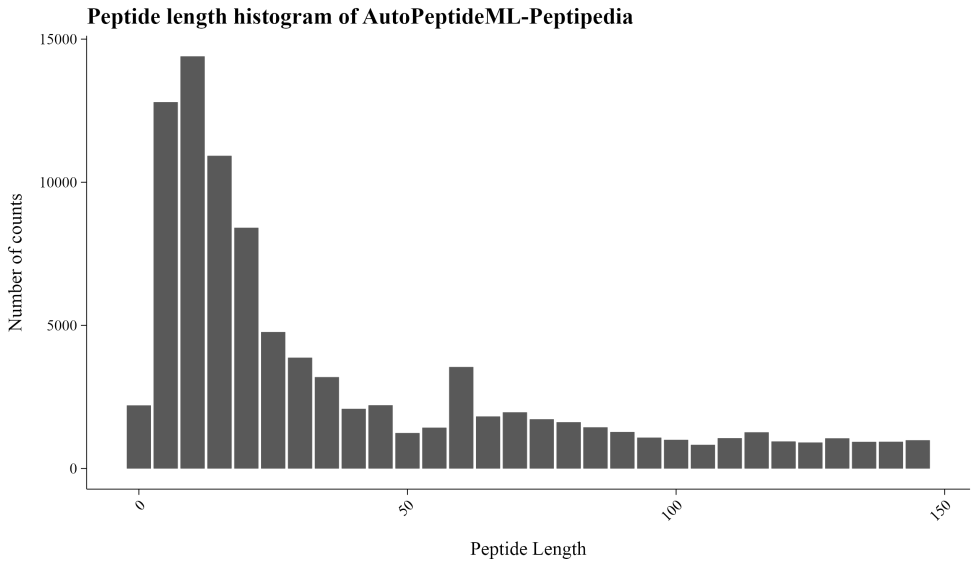

Fig. 5. Histogram of peptide lengths with bin size 5.

B. Recommendations for using AutoPeptideML and reporting its results

This section explores how the structure of the outputs from AutoPeptideML facilitates compliance with DOME guidelines [50], nevertheless, it is important to note that no system can fully avoid its misuse or abuse and the ultimate responsibility of following proper guidelines and accurately reporting the results lies in the final users.

- **Data:** The algorithm ensures independence between the optimisation (training) and evaluation (test) sets. The hyperparameter optimisation and model selection, which can be considered as meta-optimisation strategies, relies on n-fold cross-validation and maintains the independence of the evaluation set. Further, the constraints upon the algorithm in the web-server application impedes malpractices like the manual curation of parameters to meta-optimise the results in the independent test sets.

  The datasets generated during the automatic search for negative samples, the train/test partitions, and the n train/validation folds are included in the ZIP-compressed output file, thus making their release and sharing easy. The automatic search for negatives is also compliant with the recommendation that the distribution of the data is representative of the domain in which the model is going to be applied. The use of random seeds for any stochastic process improves the reproducibility when the same exact datasets are used, thus guaranteeing that different runs will produce similar results.
- **Optimisation:** Metrics for each fold in cross-validation are provided alongside the final evaluation metrics of the model so that train versus test error can be calculated as a measure of possible under- or over-fitting. The hyper-parameter configurations of the final models are included in the output file and are therefore easy to share.
- **Model:** PLMs are not directly explainable and it follows that models built on top of their representations are thus not explainable.
- **Evaluation:** Models are evaluated with a wide array of metrics and a PDF summary of the main model performance plots and evaluation metrics is provided with a guide on how to interpret them depending on different application contexts meant for researchers that are not familiar with ML concepts. Most common problems when analysing evaluation metrics arise when working with imbalanced evaluation datasets, the automatic dataset construction module bypasses this problem by generating balanced datasets.

C. Default hyperparameter search space

Table 3 describes the hyperparameter space defined for all experiments using the AutoPeptideML-training module.

Table 3. Default hyperparameter search space for the ensemble used throughout the paper.

Model	Trials	Hyperparameter search space			
		Name	**Type**	**Range**	**Log-scale**
KNN	10	K	integer	1-30	No
		Weights	categorical	uniform or distance	No
RFC	10	Max depth	integer	2-20	No
		Number of estimators	integet	10-100	No
LightGBM	10	Max depth	integer	1-30	Yes
		Number of leaves	integer	5-50	Yes
		Learning rate	float	10e-3 - 0.3	Yes

D. Search for Negative Peptides

Table 4 compiles the bioactivity tags excluded from the negative set during all executions of the AutoPeptideML "NegSearch" module for each specific dataset. The meaning behind this tags can be further expanded in the original publication [37].

Table 4. Overlapping classes excluded from the negative set for each of the benchmark datasets.

Dataset	Overlapping bioactivities
Antibacterial	`Antibacterial/antibiotic`
ACE inhibitor	`Blood_pressure,Blood_processes,Vasodilator,Vascular`
Anticancer	`Anicancer,Cytotoxic,Antitumour`
Antifungal	`Antifungal`
Antimalarial	`Antimalarial/antiplasmodial`
Antimicrobial	`Antimicrobial`
Antioxidant	`Antioxidant`
Antiparasitic	`Antiparasitic`
Antiviral	`Antiviral`
Blood-brain barrier	`Neuropeptide,Blood-brain_barrier_crossing`
DPPIV inhibitor	`Diabetic`
Anti-MRSA	`Antibacterial/antibiotic`
Neuropeptide	`Neuropeptide`
Quorum sensing	`Quorum_sensing`
Toxicity	`Cytotoxic,Neurotoxin,Toxic,Toxins`
Tumour T-cell antigens	`Immunological_activity`

References

1. P. Agrawal, D. Bhagat, M. Mahalwal, N. Sharma, and G. P. Raghava. Anticp 2.0: an updated model for predicting anticancer peptides. *Briefings in bioinformatics*, 22(3):bbaa153, 2021.
2. T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.
3. M. Attique, M. S. Farooq, A. Khelifi, and A. Abid. Prediction of therapeutic peptides using machine learning: computational models, datasets, and feature encodings. *IEEE Access*, 8:148570–148594, 2020.
4. Y. Bin, W. Zhang, W. Tang, R. Dai, M. Li, Q. Zhu, and J. Xia. Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features. *Journal of proteome research*, 19(9):3732–3740, 2020.
5. P. Charoenkwan, W. Chiangjong, V. S. Lee, C. Nantasenamat, M. M. Hasan, and W. Shoombuatong. Improved prediction and characterization of anticancer activities of peptides using a novel flexible scoring card method. *Scientific reports*, 11(1):3017, 2021.
6. P. Charoenkwan, S. Kanthawong, C. Nantasenamat, M. M. Hasan, and W. Shoombuatong. idppiv-scm: a sequence-based predictor for identifying and analyzing dipeptidyl peptidase iv (dpp-iv) inhibitory peptides using a scoring card method. *Journal of proteome research*, 19(10):4125–4136, 2020.
7. P. Charoenkwan, S. Kanthawong, N. Schaduangrat, P. Li', M. A. Moni, and W. Shoombuatong. Scmrsa: a new approach for identifying and analyzing anti-mrsa peptides using estimated propensity scores of dipeptides. *ACS omega*, 7(36):32653–32664, 2022.
8. P. Charoenkwan, C. Nantasenamat, M. M. Hasan, M. A. Moni, B. Manavalan, W. Shoombuatong, et al. Stackdppiv: A novel computational approach for accurate prediction of dipeptidyl peptidase iv (dpp-iv) inhibitory peptides. *Methods*, 204:189–198, 2022.
9. P. Charoenkwan, C. Nantasenamat, M. M. Hasan, and W. Shoombuatong. ittca-hybrid: Improved and robust identification of tumor t cell antigens by utilizing hybrid feature representation. *Analytical biochemistry*, 599:113747, 2020.
10. P. Charoenkwan, N. Schaduangrat, P. Lio, M. A. Moni, P. Chumnanpuen, and W. Shoombuatong. iamap-scm: A novel computational tool for large-scale identification of antimalarial peptides using estimated propensity scores of dipeptides. *ACS omega*, 7(45):41082–41095, 2022.
11. S. Chen, Q. Li, J. Zhao, Y. Bin, and C. Zheng. Neuropred-clq: incorporating deep temporal convolutional networks and multi-head attention mechanism to predict neuropeptides. *Briefings in Bioinformatics*, 23(5), 2022.
12. R. Dai, W. Zhang, W. Tang, E. Wynendaele, Q. Zhu, Y. Bin, B. De Spiegeleer, and J. Xia. Bbppred: sequence-based prediction of blood-brain barrier peptides with feature representation learning and logistic regression. *Journal of Chemical Information and Modeling*, 61(1):525–534, 2021.
13. W. Dee. Lmpred: Predicting antimicrobial peptides using pre-trained language models and deep learning. *Bioinformatics Advances*, 2(1):vbac021, 2022.
14. Z. Du, X. Ding, Y. Xu, and Y. Li. Unidl4biopep: a universal deep learning architecture for binary classification in peptide bioactivity. *Briefings in Bioinformatics*, page bbad135, 2023.
15. A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.
16. S. Fernandez-Tome, B. Hernandez-Ledesma, M. Chaparro, P. Indiano-Romacho, D. Bernardo, and J. P. Gisbert. Role of food proteins and bioactive peptides in inflammatory bowel disease. *Trends in Food Science & Technology*, 88:194–206, 2019.
17. L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.
18. S. Giguere, F. Laviolette, M. Marchand, D. Tremblay, S. Moineau, X. Liang, É. Biron, and J. Corbeil. Machine learning assisted design of highly active peptides for drug discovery. *PLoS computational biology*, 11(4):e1004074, 2015.
19. A. Hagberg and D. Conway. Networkx: Network analysis with python. *URL: https://networkx. github. io*, 2020.
20. D. R. Harper, A. Nandy, N. Arunachalam, C. Duan, J. P. Janet, and H. J. Kulik. Representations and strategies for transferable machine learning improve model performance in chemical discovery. *The Journal of Chemical Physics*, 156(7), 2022.
21. X. He, K. Zhao, and X. Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.
22. Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.
23. D. T. Jones and J. M. Thornton. The impact of alphafold2 one year on. *Nature methods*, 19(1):15–20, 2022.
24. D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi. Trustworthy artificial intelligence: a review. *ACM Computing Surveys (CSUR)*, 55(2):1–38, 2022.
25. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
26. M. Littmann, M. Heinzinger, C. Dallago, T. Olenyi, and B. Rost. Embeddings from deep learning transfer go annotations beyond homology. *Scientific reports*, 11(1):1–14, 2021.
27. B. Manavalan, S. Basith, T. H. Shin, L. Wei, and G. Lee. mahtpred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics*, 35(16):2757–2765, 2019.
28. W. McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.
29. R. Nakano, C. Wells, wikke, lugq, R. J. Liwag, M. Emery, L. Miranda, J. Engelman, F. Herfert, E. Can, D. Friedman, and C. Ren. scikit-plot: v0.3.5, May 2018.
30. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid

sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

31. T. H. Olsen, B. Yesiltas, F. I. Marin, M. Pertseva, P. J. García-Moreno, S. Gregersen, M. T. Overgaard, C. Jacobsen, O. Lund, E. B. Hansen, et al. Anoxpepred: using deep learning for the prediction of antioxidative properties of peptides. *Scientific Reports*, 10(1):21471, 2020.
32. Y. Pang, L. Yao, J. Xu, Z. Wang, and T.-Y. Lee. Integrating transformer and imbalanced multi-label learning to identify antimicrobial peptides and their functional activities. *Bioinformatics*, 38(24):5368–5374, 2022.
33. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
34. V. Patel and M. Shah. Artificial intelligence and machine learning in drug discovery and development. *Intelligent Medicine*, 2(3):134–140, 2022.
35. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
36. S. A. Pinacho-Castellanos, C. R. García-Jacas, M. K. Gilson, and C. A. Brizuela. Alignment-free antimicrobial peptide predictors: improving performance by a thorough analysis of the largest available data set. *Journal of Chemical Information and Modeling*, 61(6):3141–3157, 2021.
37. C. Quiroz, Y. B. Saavedra, B. Armijo-Galdames, J. Amado-Hinojosa, Á. Olivera-Nappa, A. Sanchez-Daza, and D. Medina-Ortiz. Peptipedia: a user-friendly web application and a comprehensive database for peptide research supported by machine learning approach. *Database*, 2021, 2021.
38. A. Rajput, A. K. Gupta, and M. Kumar. Prediction and analysis of quorum sensing peptides based on sequence features. *PLoS One*, 10(3):e0120066, 2015.
39. R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, and Y. Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.
40. R. M. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020.
41. P. Rice, I. Longden, and A. Bleasby. Emboss: the european molecular biology open software suite. *Trends in genetics*, 16(6):276–277, 2000.
42. B. Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.
43. T. F. Smith, M. S. Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
44. M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.
45. B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.
46. R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.
47. F. Teufel, M. H. Gíslason, J. J. Almagro Armenteros, A. R. Johansen, O. Winther, and H. Nielsen. Graphpart: homology partitioning for biological sequence analysis. *NAR genomics and bioinformatics*, 5(4):lqad088, 2023.
48. A. L. Tornesello, A. Borrelli, L. Buonaguro, F. M. Buonaguro, and M. L. Tornesello. Antimicrobial peptides as anticancer agents: functional properties and biological activities. *Molecules*, 25(12):2850, 2020.
49. H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
50. I. Walsh, D. Fishman, D. Garcia-Gasulla, T. Titma, G. Pollastri, J. Harrow, F. E. Psomopoulos, and S. C. Tosatto. Dome: recommendations for supervised machine learning validation in biology. *Nature methods*, 18(10):1122–1127, 2021.
51. L. Wang, N. Wang, W. Zhang, X. Cheng, Z. Yan, G. Shao, X. Wang, R. Wang, and C. Fu. Therapeutic peptides: Current applications and future directions. *Signal Transduction and Targeted Therapy*, 7(1):48, 2022.
52. L. Wei, J. Hu, F. Li, J. Song, R. Su, and Q. Zou. Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. *Briefings in Bioinformatics*, 21(1):106–119, 2020.
53. L. Wei, X. Ye, Y. Xue, T. Sakurai, and L. Wei. Atse: a peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. *Briefings in Bioinformatics*, 22(5):bbab041, 2021.
54. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.
55. X. Xiao, Y.-T. Shao, X. Cheng, and B. Stamatovic. iamp-ca2l: a new cnn-bilstm-svm classifier based on cellular automata image for identifying antimicrobial peptides and their functional types. *Briefings in bioinformatics*, 22(6):bbab209, 2021.
56. Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*, 2018.
57. W. Zhang, E. Xia, R. Dai, W. Tang, Y. Bin, and J. Xia. Predapp: predicting anti-parasitic peptides with undersampling and ensemble approaches. *Interdisciplinary Sciences: Computational Life Sciences*, pages 1–11, 2022.
58. Z. Zhang, C. Wang, M. Xu, V. Chenthamarakshan, A. Lozano, P. Das, and J. Tang. A systematic study of joint representation learning on protein sequences and structures, 2023.
59. L. Zhao, D. Li, X. Qi, K. Guan, H. Chen, R. Wang, and Y. Ma. Potential of food-derived bioactive peptides in alleviation and prevention of alzheimer's disease. *Food & Function*, 13(21):10851–10869, 2022.