Journal Title Here, 2022, pp. 1–11

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

PAPER

AutoPeptideML: A study on how to build more 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 considered the effect of different design choices in the development of peptide bioactivity binary predictors and found that the choice of negative peptides and the use of homology-based partitioning strategies when constructing the evaluation set have a significant impact on perceived model performance providing more realistic estimation of the performance of the model when exposed to new data. We also show that the use of protein language models to generate peptide representations can both simplify the computational pipelines and improve model performance, and that state-of-the-art protein language models perform similarly regardless of size or architecture. Finally, we integrate these results into an easy-to-use AutoML tool to support the development of new robust predictive models for peptide bioactivity by biologist without a strong machine learning expertise. Source code, documentation, and data are available at `https://github.com/IBM/AutoPeptideML` and a dedicated web-server at `http://peptide.ucd.ie/AutoPeptideML`.

Introduction

Peptides are short amino acid chains comprised of 3 to 50 residues that can have multiple roles in different organisms from mediating intercellular signalling pathways to protecting from microbial infection (1). Due to their versatility, they have recently gained a lot of attention from the pharmaceutical and food industries as these biological properties, or bioactivities, make them excellent candidates for drug or nutraceutical discovery (1). In this context, there is a growing demand for predictive models that can accelerate the discovery or design of peptides targeting new bioactivities (2).

Novel developments in machine and deep learning algorithms have revolutionised the field of computational biochemistry offering robust predictive models for protein structure (3) or molecular properties (4). Despite the advancements in the field, developing and evaluating predictive models is still an arduous process that requires both domain expertise and technical skills (2). Thus, most predictive models target broad and general applications while solutions for more narrow use cases, like specific peptide bioactivities, remain underdeveloped (2).

Here, we analysed key steps in the development life-cycle of peptide bioactivity predictors from data gathering to evaluation, and integrated the conclusions we reached into an automated machine learning system. The aim of this system is to allow both experimental and computational researchers to build custom predictors and provide a robust and interpretable evaluation. Peptide bioactivity, in the context of this study, refers to the binary quality of either possessing a certain biological function/property or not; thus, the predictive modelling task we are automating is binary classification.

There are a fair number of studies exploring the construction of peptide bioactivity predictors targeting individual bioactivities like antimicrobial, enzyme inhibition, brain-blood barrier penetration, and so on (2). A review of the existing literature revealed five key steps where automation would not only greatly simplify the task, but could also lead to more reliable models. They are: 1) data gathering for the negative class, 2) dataset partitioning, 3) computational representation of the peptides, 4) model training and hyperparameter optimisation, and 5) reporting of model evaluation.

Data gathering for the negative class. Binary classifiers need to be exposed to positive and negative examples. Finding positive examples for peptide bioactivity is relatively simple as there are databases, repositories, and prior literature describing the function and role of a multitude of peptides (5). However, there are few repositories enumerating peptides that do not

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

present a certain function or property (2). Further, there is no consensus in the literature as to how negative peptides should be chosen: some works opt for randomly choosing fragments of proteins from UniProt or SwissProt to constitute their negative subset (6; 7; 8), others look for actual peptides (defined as proteins shorter than 50 residues) in the same databases (9; 10; 11; 6), and yet others use peptides with a known bioactivity that is different from the one they are concerned about (6; 12; 13). If we consider the first and second approaches, where the negative peptides are drawn from a different distribution to that of the positives (random protein fragments and random peptides), the differences between positive and negative peptides could be explained by a myriad of confounding factors that do not have a direct bearing on their specific bioactivity, but that are related to the differences between generally bioactive peptides and random sequences. In the third approach, the opposite is true, positive and negative peptides may be so similar to each other that the model hyper-focuses on the specific differential features between both bioactivities, hindering its ability to generalise. In this paper, we explored the effect of introducing 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.

Dataset partitioning. To evaluate predictive models it is necessary to divide the data into at least three distinct subsets: training, validation, and evaluation. The independence between training and evaluation datasets is essential to properly evaluate any ML predictor (14). Achieving this independence, however, is not straight-forward when working with biological sequences, because similar sequences tend to share structural and functional features (15). Community guidelines recommend building evaluation sets that do not share homologous sequences with the training set either by homology reduction (14) or homology partitioning (16). Despite this, most of the peptide bioactivity predictors reviewed (13; 8; 17; 9; 18; 19) do not introduce any correction for homology when partitioning their datasets and those that do, use high thresholds (80-90% of sequence identity, see Supplementary D), still allowing for similar sequences to be in different sets, which could lead to the overestimation of the performance of the models due to data leakage. Here, we explore the effect of introducing homology-based dataset partitioning for building evaluation subsets more suited for testing model generalisation.

Computational representation of the peptides. For a predictive model to be able to interpret the peptide sequences, they need to be translated into mathematical objects (vectors or matrices). The reviewed literature offers different options for performing this transformation that include statistics of: residue composition (13; 8; 17), evolutionary profile (20), or physico-chemical properties (9; 8; 18; 19). The consensus that can be drawn from the variety of combinations is that a different set of descriptors may be optimal for each predictive task, i.e., for every bioactivity. Thus, finding the best combination of descriptors is a crucial and intricate part of the modelling problem (2).

The advent of Protein Language Models (PLMs) like those from the ESM (evolutionary scale modelling) (21; 22) or RostLab (ProtBERT, Prot-T5-XL, or ProstT5) (23; 24) families has allowed for a much simpler and much richer way to perform this representation step. These models learn by masked residue reconstruction to predict the conditional probability that any given residue r would appear in the masked position i of a sequence s, $p(r_i|s)$. This probability has a relationship with the concept of conserved and unconserved positions that is often used when analysing multiple sequence alignments (3). The models are trained on a vast set of sequences from the UniRef database which includes not only protein sequences, but also peptides. Moreover, at least, two prior studies have demonstrated that models from the ESM family can be used for representing peptides overperforming traditional description strategies (25; 26). In this paper, we continue this line of research by considering two main questions: does model size have an impact on how suitable the representations are for describing peptides? and is there any significant difference between different classes of models?

Model training and hyperparameter optimisation. There are many different algorithms for fitting predictive models to a binary classification task and choosing between them is an extended trial-and-error task. Here, we explore an alternative approach: to use standard tools in the AutoML domain for performing hyperparameter bayesian optimisation of simpler machine learning models like k-nearest neighbours (K-NN), random forest (RF), or light gradient boosting (LightGBM), and ensembling them.

The main contributions of this paper are:

1. A new method for drawing the negative subsets in peptide bioactivity classifiers, and the analysis of its effect on model performance.
2. An analysis of the effect of performing homology-based dataset partitioning on the perceived performance of the models.
3. A systematic analysis of the performance of different PLMs as peptide representation methods.
4. An analysis of whether an ensemble of optimised traditional machine learning algorithms can compete with more complex neural network models.

These contributions have been integrated into a computational tool and webserver, named AutoPeptideML, that allows any researcher to build their own custom models for any arbitrary peptide bioactivity they are interested in, regardless of whether they have a computational background. The tool provides an output summary that facilitates the interpretation of the reliability of the predictor generated. Finally, it also provides support for using the generated models for predicting the bioactivity of a given set of peptides.

Materials and methods

Data acquisition. 18 different peptide bioactivity datasets containing positive and negative samples were used to evaluate the effect of the different methods. These datasets were selected from a previous study, evaluating the use of PLMs for general peptide bioactivity prediction (25). The datasets ranged in size from 200 to 20,000 peptides. Here, they are referred to as the "original" datasets.

Dataset with new negative peptides. For each of the original datasets, a new version was constructed using the new definition of negative peptides, termed "NegSearch". The negative peptides were drawn from a curated version of the Peptipedia database "APML-Peptipedia" comprised of 92,092

peptides representing 128 different activities (see Figure SA). To avoid introducing false negative peptides into the negative subset, all bioactivities that may overlap with the bioactivity of interest were excluded (see Table SB). To ensure that the negative peptides were drawn from a similar distribution to the positive peptides and thus minimise the number of confounding factors, for each dataset we calculated a histogram of the lengths of its peptides with bin size of 5. Then, for each bin in the histogram, we queried APML-Peptipedia for as many peptides as present in the bin, with lengths between its lower and upper bounds. If there were not enough peptides, the remaining peptides were drawn from the next bin.

Dataset partitioning. Two different partitioning strategies were used to generate the training/evaluation subsets: A) random partitioning or B) a novel homology-based partitioning algorithm, which creates an independent evaluation set ensuring that there are no homologous sequences between training and evaluation. Briefly, the algorithm calculates pairwise alignments among all dataset sequences to form a pairwise similarity matrix. It then clusters these sequences based on the similarity matrix using the connected components algorithm. Lastly, it iteratively transfers the smallest clusters to the evaluation set. This process ensures that there are no sequences in the evaluation set with homologs in the training set. The datasets generated through this alternative strategy are referred to as "NegSearch+HP".

In both cases, the training set is further subdivided into 10 folds for cross-validation. This second division relies on random stratified partitioning, as implemented by `scikit-learn`, to create 10 cross-validation folds.

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 with prior k-mer prefiltering (27). The proportion of the peptides within the training dataset with at least one homologous peptide in the evaluation set is used as independence measurement. We considered that two peptides are homologous if they have a sequence identity above 30% using the longest sequence as denominator.

Peptide representations. In order to evaluate the PLM peptide representations, the following methods (21; 22; 23; 24) were evaluated: ESM2-8M, ESM2-35M, ESM2-150M, ESM2-650M, ESM1b, ProtBERT, Prot-T5-XL-UniRef50, ProstT5 (sequence mode), and one-hot encoding as a non-PLM-based baseline.

Model training. In order to evaluate the model training and hyperparameter optimisation step, hyperparameter optimisation was performed separately for KNN, LightGBM and RFC and all models were ensembled (see Table SC for more details about the hyperparameter optimisation). Thus, the final ensembles 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. This strategy is referred to as "Optimised ML ensemble" throughout the text. Our system was compared against an amended version of the UniDL4BioPep (25) framework, which we named "UniDL4BioPep-A". This amendment differs from the original in that, following community guidelines (14), it uses 10-fold cross-validation to determine the best possible checkpoint, instead of the hold-out evaluation set.

Results and Discussion

We have focused our study of peptide bioactivity prediction in the binary classification task of discriminating between peptides that show a biologically-relevant property or function and those which do not.

New Dataset construction

Analysis of original benchmarks. The results from the analysis of the independence between training and evaluation sets compiled in Table 1 indicate that for 13 of the 18 original datasets, at least 10% of the peptides in the training are similar to sequences in the evaluation dataset, which compromises their independence. If we consider the datasets (see Table SD), for which homology-based independence correction was used (ACE inhibitor, Antioxidant, Antiparasitic, Anti-MRSA, and Neuropeptides) we see that only two have less than 10% of their training sequences being similar to at least one sequence in the evaluation set. Interestingly, Anticancer 1 and 2, and Antimalarial 1 and 2 have the same sets of positive peptides, but Anticancer 1 defines its negatives as peptides with antimicrobial activity, which is a bioactivity that overlaps with the positive class (28), thus explaining the comparatively low performance achieved when compared with Anticancer 2, which draws its negative set from a collection of random peptides (Figure 1). 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 (11; 29). These results indicate that random negatives lead to better perceived model performance, even though this may not reflect their real world application.

Table 1. Proportion of sequences in training set with at least one similar peptide (sequence identity > 30%) in the evaluation set. NS: NegSearch datasets; NS+HP: NegSearch+HP datasets; *: Equivalent to Anticancer 1 for NS and NS+HP; **: Equivalent to Antimalarial 1 for NS and SN+HP

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

Effect of introducing new negative peptides. Table 1 shows that the introduction of new negative peptides (column NS) has a limited effect on the interdependence between training and evaluation subsets going so far as increasing it for certain datasets. This was to be expected since, for example, typically random peptides are less similar than other classes

of bioactive peptides. In contrast, the NS+HP dataset (fourth column) completely removes any interdependence between training and evaluation subsets.

Overall, choice of negatives from among other classes of bioactivities typically reduces the perceived performance of the models (see Figure 1). Let us consider, for example, the antimalarial prediction datasets. The ability to contrast the positives with other bioactive classes (Antimalarial NegSearch) is a tougher challenge than distinguishing the positives from random protein fragments (Antimalarial 2) or random peptides from Uniprot (Antimalarial 1). This observation can be extended to most datasets where the more restricted selection criteria of the new negative peptides coincides with a drop in apparent performance. Notably, the Antiviral dataset is the only case where performance increased after narrowing the negative class definition, and the reason behind this discrepancy remains unclear. 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.

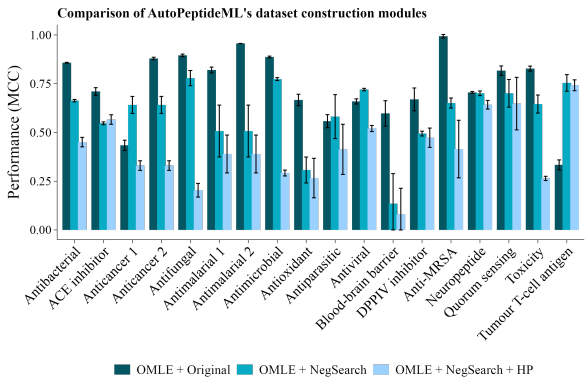

Fig. 1. Evaluation of AutoPeptideML's dataset construction modules. Error bars reflect the standard deviation across three replicates. OMLE: Optimised ML ensemble; Original: Original benchmark; NegSearch: Dataset with new negative peptides; HP: Homology-based dataset partitioning module.

Effect of homology-based partitioning. The difference between Original and NegSearch datasets highlights the importance of considering the similarity between the positive and negative peptides. We also considered the importance of controlling the similarity between training and evaluation subsets by comparing the performance between the NegSearch datasets (where no similarity correction was introduced) and NegSearch+HP (where we introduced homology-based partitioning).

The effect observed across most datasets is a further drop in perceived model performance (Figure 1). This indicates that alternative methods tend to overestimate model performance by not properly diagnosing model overfitting to the training subset, supporting the use of homology partitioning techniques (14; 16).

Protein Language Models as peptide representation methods

Recent studies have reported the use of PLMs for predicting peptide bioactivity (25; 26), however, they have not been compared to a naive baseline representation like one-hot encoding; nor has there been an evaluation on which PLM may be more suited for peptide representation.

Baseline. First, we compare the PLMs to a naive baseline representation (one-hot encoding). Figure 2 shows that generally PLMs are significantly better representation methods across datasets, though in specific datasets one-hot encoding appears to achieve similar performance.

Model size. We evaluated four different PLM models from the ESM family with increasing size: ESM2-8M (8 million parameters), ESM2-35M (35 million parameters), ESM2-150M (150 million parameters), ESM2-650M (650 million parameters). We also evaluated ESM1b-650M (650 million parameters), from a previous version of ESM. We performed the comparisons with the NegSearch+HP datasets to ensure that we were properly evaluating model generalisation. Figure 2 shows that there is no significant difference between models across all datasets and no correlation between model size and performance.

Type of model. We further compared the ESM models to the main models from the RostLab family: ProtBERT, Prot-T5-XL-UniRef50, and Prost-T5. Figure 2 shows that even though for certain datasets there might be significantly better models, when the effect is analysed across all datasets there is no significant difference between the different models or families.

All things considered, the ESM2-8M model achieves a commendable balance between enhanced performance relative to one-hot encoding and minimal computational size. Notably, the utilization of PLM encoding for peptide representation yielded improvements in performance, despite PLMs being primarily designed for protein applications. This underscores the potential of PLM encoding to be effectively extended to peptides, in line with previous studies (26; 25).

Hyperparameter optimisation and ensembling of simple ML models as an alternative to neural networks

We compared the use of hyperparameter optimisation and ensembling of simple machine learning models against the UniDL4BioPep (25) approach of using a fixed 1D-CNN. To contextualise the current state of general methods, we compare against the baseline of dataset-specific models that combine handcrafted dataset-specific features with the best performing machine (or deep) learning algorithms. The self-reported values for the handcrafted models referenced in Supplementary D are included with the evaluation in the original set of benchmarks to assess the contributions of both general purpose frameworks.

Comparison to handcrafted models. 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 best handcrafted models for each specific dataset. However, this comparison is performed with the original datasets and it is likely that the results are inflated due to the effect of negative class choice and lack of training-evaluation independence and might not properly reflect the behaviour of the models in the real-world. It is impossible to compare the handcrafted models in the new datasets as 1) in a majority of cases the code for reproducing the training is not available and 2) the process for generating the models is highly specific to the datasets and partitions used.

The true value of automatic models emerges from the observation that, even though, in a disadvantaged

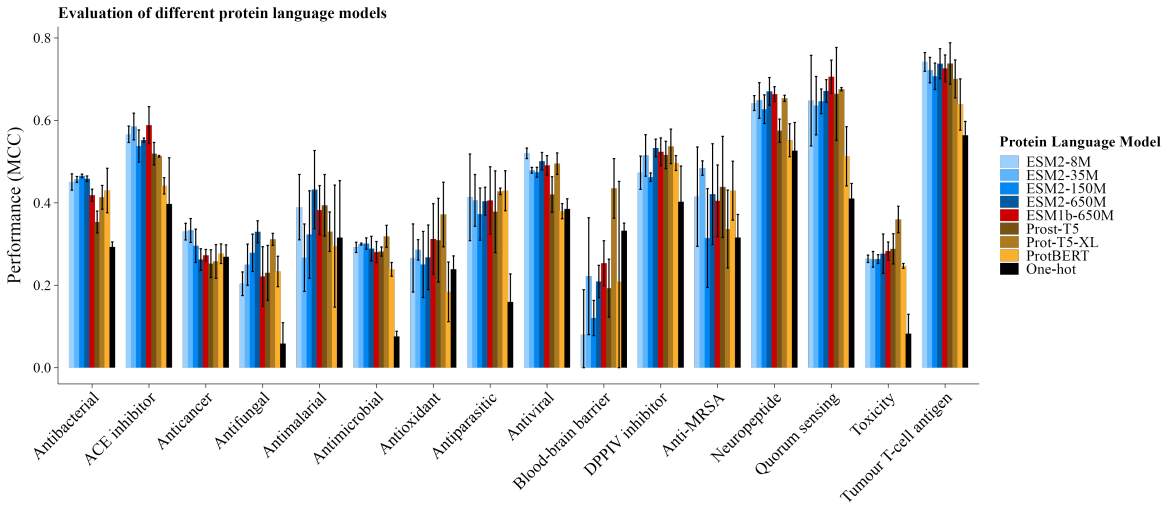

Fig. 2. Evaluation of different protein language models. Error bars reflect the standard deviation across three replicates.

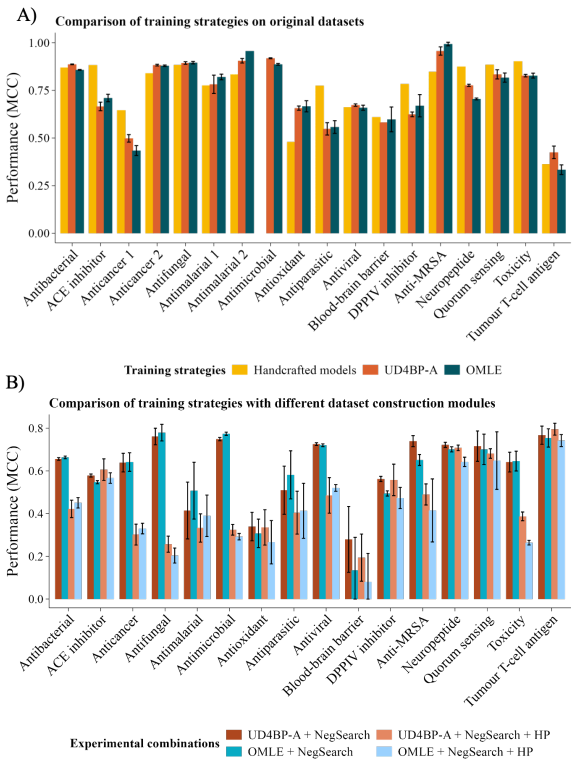

Fig. 3. A: **Comparison of training strategies on original datasets**. B: **Comparison of training strategies with different dataset construction modules**. Error bars reflect the standard deviation across three replicates. OMLE: Optimised ML ensemble; UD4BP-A: UniDL4BioPep-A

comparison they still attain performances comparable to those of handcrafted models and thus enable their application to previously unstudied bioactivities, suggesting that their predictions will roughly perform similarly but requiring significantly less effort and technical expertise to develop. Therefore, instances where the model exhibits inferior performance, such as with the Anticancer 1 and Antiparasitic datasets, should not deter its use. The primary intent of these models is to pioneer research in previously unexamined bioactivities.

Comparison of an optimised ML ensemble with a neural network. When compared with Fig 3.A, Figure 3.B shows that when automatic negative selection from a bioactive dataset are introduced, both ML and neural network 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-based 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 show similar performance across the tasks. 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 neural network model while being more computationally efficient.

AutoPeptideML

All the findings described thus far, were used to guide the development of AutoPeptideML, a computational tool and webserver that allows researchers to easily build strong peptide bioactivity predictors and provide a robust evaluation that complies with community guidelines (see Supplementary E for a more through description of the AutoML system).

AutoPeptideML can be used in two regimes: *Model builder* and *Prediction*. In the first mode new predictive models are created automatically from a single file with known positive peptides for the bioactivity of interest. In the second mode, any predictive model generated through the *model builder* can then be used to predict how likely are a set of peptides to have the desired bioactivity.

The outputs that the program generates are:

- *Model builder:* When used to develop new predictors, AutoPeptideML outputs a model fitted to predict the bioactivity of interest, a folder with all information necessary for reproducing the model, and an interpretable summary of the model capabilities.
- *Prediction:* AutoPeptideML can also be used to leverage existing predictors. In this case, it outputs a list of the problem peptides sorted in descending order of predicted bioactivity (higher bioactivity first) and a measure of the uncertainty of each prediction.

Conclusions

The definition of the negative class used for building peptide bioactivity predictos 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 here to define training and evaluation sets impacts model generalisability and thus has an impact on apparent performance of models of up to 50%. The magnitude of these effects suggests that the model performance reported for existing tools might be inflated due to data leakage.

The combination of PLM peptide representations and an optimised ensemble of simple ML models 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 and more complex DL algorithms (UniDL4BioPep-A), even though the former is more computationally efficient.

Using PLM for computing peptide representations is a significantly better strategy than using a one-hot encoding (a naive representation) for most of the datasets considered. However, the choice of PLM family or model size does not affect significantly the performance of the predictors.

Finally, we present AutoPeptideML as a computational tool and webserver that researchers without technical expertise to develop predictive models for any custom peptides bioactivity and facilitates compliance with community guidelines for predictive modelling in the life-sciences. It is able to handle several key steps in the peptide bioactivity predictor development life-cycle including: 1) data gathering, 2) homology-based dataset partitioning, 3) model selection and hyperparameter optimisation, 4) robust evaluation, 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 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 and development of new models by researchers not specialised in ML.

Acknowledgements

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.

Competing interests

No competing interest is declared.

References

1. 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*, vol. 7, no. 1, p. 48, 2022.
2. 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*, vol. 8, pp. 148570–148594, 2020.
3. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," *BioRxiv*, 2022.
4. S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, and M. J. Ahsan, "Machine learning in drug discovery: a review," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 1947–1999, 2022.
5. 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*, vol. 2021, 2021.
6. 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*, vol. 22, no. 3, p. bbaa153, 2021.
7. 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*, vol. 35, no. 16, pp. 2757–2765, 2019.
8. 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*, vol. 19, no. 9, pp. 3732–3740, 2020.
9. 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*, vol. 61, no. 6, pp. 3141–3157, 2021.
10. 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*, vol. 38, no. 24, pp. 5368–5374, 2022.
11. 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*, vol. 7, no. 45, pp. 41082–41095, 2022.
12. 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*, vol. 19, no. 10, pp. 4125–4136, 2020.
13. 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*, vol. 10, no. 1, p. 21471, 2020.
14. 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*, vol. 18, no. 10, pp. 1122–1127, 2021.
15. B. Rost, "Twilight zone of protein sequence alignments," *Protein engineering*, vol. 12, no. 2, pp. 85–94, 1999.
16. 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*, vol. 5, no. 4, p. lqad088, 2023.
17. 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*, vol. 11, no. 1, p. 3017, 2021.
18. 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*, vol. 7, no. 36, pp. 32653–32664, 2022.
19. 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*, vol. 599, p. 113747, 2020.
20. 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*, vol. 22, no. 5, p. bbab041, 2021.
21. R. M. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives, "Transformer protein language models are unsupervised structure learners," *bioRxiv*, 2020.
22. 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*, vol. 379, no. 6637, pp. 1123–1130, 2023.
23. 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*, vol. 44, no. 10, pp. 7112–7127, 2021.
24. M. Heinzinger, K. Weissenow, J. G. Sanchez, A. Henkel, M. Steinegger, and B. Rost, "Prostt5: Bilingual language model for protein sequence and structure," *bioRxiv*, pp. 2023–07, 2023.
25. Z. Du, X. Ding, Y. Xu, and Y. Li, "Unidl4biopep: a universal deep learning architecture for binary classification in peptide bioactivity," *Briefings in Bioinformatics*, p. bbad135, 2023.
26. W. Dee, "Lmpred: Predicting antimicrobial peptides using pre-trained language models and deep learning," *Bioinformatics Advances*, vol. 2, no. 1, p. vbac021, 2022.
27. M. Steinegger and J. Söding, "Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature biotechnology*, vol. 35, no. 11, pp. 1026–1028, 2017.
28. 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*, vol. 25, no. 12, p. 2850, 2020.
29. 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*, vol. 22, no. 6, p. bbab209, 2021.
30. 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*, pp. 1–11, 2022.
31. 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*, vol. 61, no. 1, pp. 525–534, 2021.
32. 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*, vol. 204, pp. 189–198, 2022.
33. 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*, vol. 23, no. 5, 2022.
34. A. Rajput, A. K. Gupta, and M. Kumar, "Prediction and analysis of quorum sensing peptides based on sequence features," *PLoS One*, vol. 10, no. 3, p. e0120066, 2015.
35. 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*, vol. 21, no. 1, pp. 106–119, 2020.

A. APML-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`). APML-Peptipedia is the result of removing all sequences with non-standard residues or without any known bioactivity and contains 92,092 peptides (see Supplementary). Figure 1 describes the distribution of the lengths of the peptides comprising APML-Peptipedia.

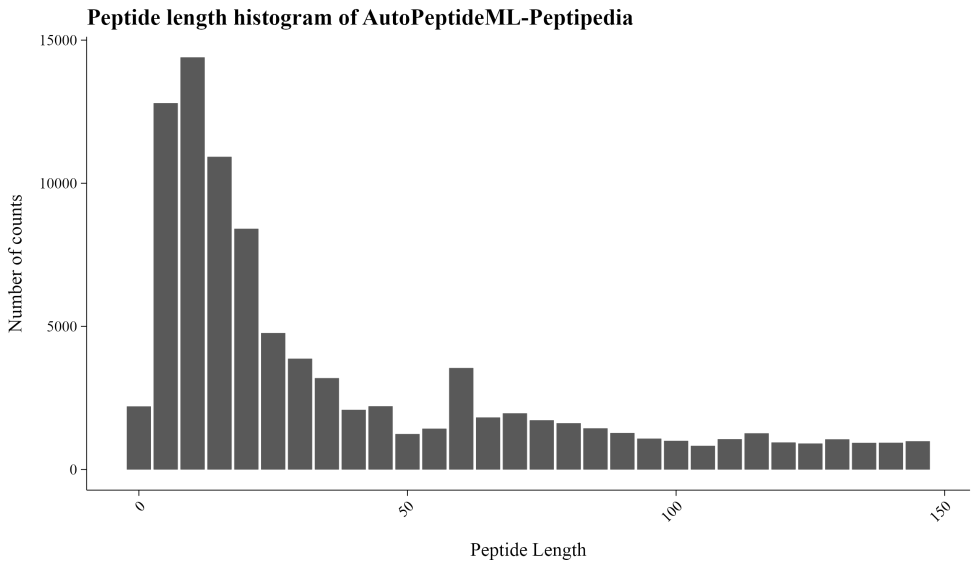

Fig. 1. Histogram of peptide lengths with bin size 5.

B. Search for Negative Peptides

Table 1 compiles the bioactivity tags excluded from the negative set when building the "NegSearch" datasets. The meaning behind these tags can be further expanded in the original publication (5).

Table 1. 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`

C. Default hyperparameter search space

Table 2 describes the hyperparameter space defined for all experiments using the "Optimised ML ensemble".

Table 2. 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. Review of original benchmarks

Table 3 contains a review of the datasets used as benchmarks throughout the study: their origin, how the negatives peptides were drawn, training-evaluation partitioning strategy, and the reference of the best handcrafted model reported.

Table 3. 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 (9)	Random peptides	Homology maximisation	(9)
ACE (7)	Random protein fragments	Homology reduction (90%)	(7)
AC1 (6)	Antimicrobial peptides	Random	(17)
AC2 (6)	Random protein fragments	Random	(6)
AF (9)	Random peptides	Homology maximisation	(9)
AMA1 (11; 29)	Random peptides	Random	(11)
AMA2 (11; 6)	Random protein fragments	Random	(11)
AMI (9)	Random peptides	Homology maximisation	(9)
AO (13)	Experimental + random peptides	Homology reduction (90%)	(13)
AP (30)	Random peptides	Homology reduction (90% for positives and 60% for negatives)	(30)
AV (9)	Random peptides	Homology maximisation	(9)
BBB (31)	Random peptides	Homology reduction (90%)	(31)
DPPIV (12)	Random + Bioactive	Random	(32)
MRSA (18)	Random peptides	Homology reduction (80%)	(18)
NP (8)	Random protein fragments	Homology reduction (90%)	(33)
QS (34)	Random peptides	Random	(35)
TOX (20)	Random peptides	Random	(20)
TTCA (19)	T-cell antigens not associated to disease	Random	(19)

E. AutoPeptideML

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) (14), ensuring a robust scientific validation (see below). A visual summary of the workflow can be found in Figure 2.

Users are free to define the number of models that should be included in the hyperparameter 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 (14).

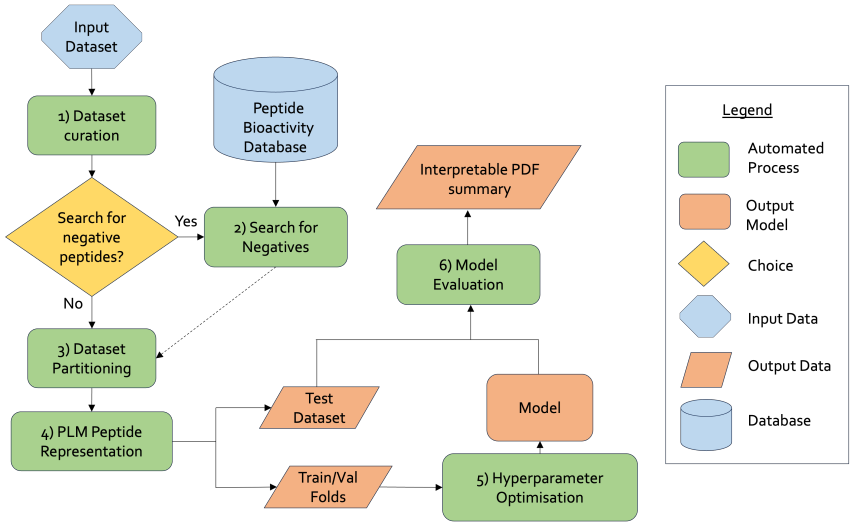

Fig. 2. Visual summary of the AutoPeptideML workflow.

Recommendations for using AutoPeptideML and reporting its results

This section explores how the structure of the outputs from AutoPeptideML facilitates compliance with DOME guidelines (14), 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.