SERVERLESS PREDICTION OF PEPTIDE PROPERTIES WITH RECURRENT NEURAL NETWORKS

A PREPRINT

Mehrad Ansari
Department of Chemical Engineering
University of Rochester
mehrad.ansari@rochester.edu

Andrew D. White*
Department of Chemical Engineering
University of Rochester
andrew.white@rochester.edu

May 18, 2022

ABSTRACT

We present three deep learning sequence prediction models for hemolysis, solubility, and resistance to non-specific interactions of peptides that achieve comparable results to the state-of-the art models. These predictive models share a common architecture of bidirectional recurrent neural networks (LSTM). These models are implemented in JavaScript so that they can be run on a static website without use of a dedicated server. This removes the cost, and long-term management of a server, while still enabling open and free access to the models. This "serverless" prediction model is a demonstration of edge computing bioinformatics and removes the dependence on cloud providers or self-hosting of resource-rich academic institutions. This is feasible because of the continued track of Moore's law and ubiquitous hardware acceleration of deep learning computations on new phones and desktops.

1 Introduction

There is a growing increase in the number of web-based implementations of deep learning frameworks that provide convenient public access and ease of implementation [1–6]. Notably, many web servers have been developed for sequence design tasks, like analysis of RNA, DNA, or proteins. For example, survival analysis based on mRNA data (GENT2 [7], PROGgeneV2 [8], SurvExpress [9], MEXPRESS [10], etc.), studying prognostic implications of non-coding RNA (PROGmiR [11], SurvMicro [12], OncoLnc [13], TANRIC [14]), survival analysis based on protein (TCPAv3.0 [15], TRGAted [16]) and DNA (MethSurv [17], cBioPortal [18]) data, and multiple areas of assessing cancer therapeutics [19]. These scientific web servers and web-based services allow for the availability of complex inference algorithms to a much broader user community and promote open science. This is especially important because of the disparities between lower and higher income nations, where there are disparities in the types of research activities that can be performed [20]. Bioinformatics-related research, the topic of this work, mostly takes place at those nations privileged with resource-rich institutions, where there are adequate computational resources. Yet, web-based implementations can broaden access to these methods.

Beyond disparities among institutions, web-based implementations are also a mechanism for reproducibility in science. In peptides specifically, Melo et al. [21] argue that deep learning sequence design should be accomplished by free public access to the (1) source code, (2) training and testing data, (3) published findings. However, this is not often true; Littmann[22] found in an analysis of ML research articles in bio-medicine and life sciences published between 2011 and 2016 found that only 50% released software, while 64% released data. Web-based servers do not fit the exact definition of open science (due to lack of source code access), but they do accomplish the goal of enabling others with broader expertise to build on previous advances, and are often more accessible and convenient than access to model and source code alone.

*To whom correspondence should be addressed

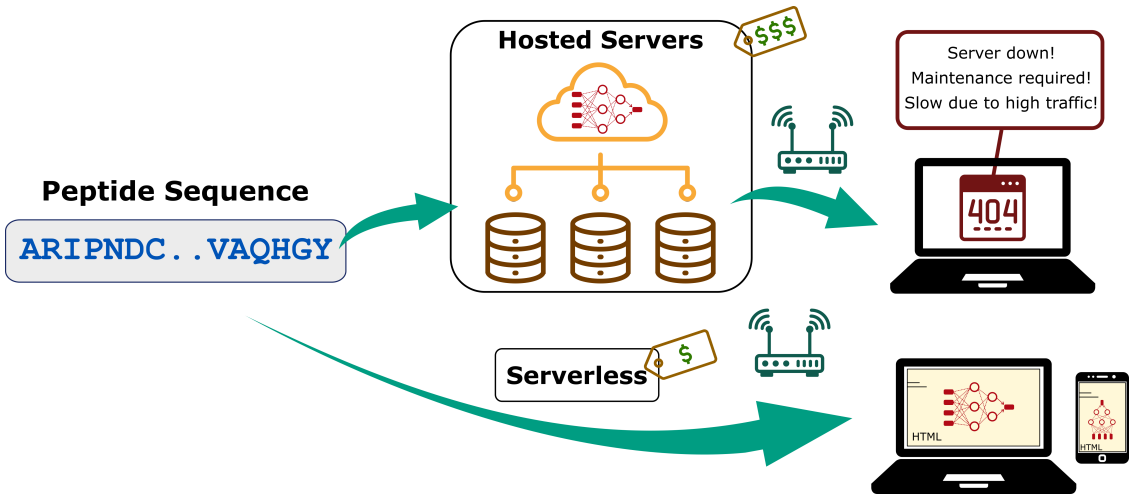

Figure 1: Conventional web-based bioinformatics frameworks vs the proposed serverless approach.

Thus, there is a compelling argument to continue web-based tools. There are, however, two major drawbacks: source-code can be inaccessible as discussed above and the reliance on third-party or self-hosted servers. Deep learning inference often requires GPUs and this requires a specialized hosting service or a complex self-hosted set-up. This creates difficult ongoing expenses, and many tools are thus only available for a limited time after publication. Additionally, there can be low incentives to increase capacity. Popular tools, like RoseTTAFold[23], can have days-long queues. The expense and deployment problems also can create disparities in impact of research between resource-rich and low-resource institutions, because not all researchers can afford to create web-based implementations.

To address the challenges above, we demonstrate a *serverless* deep learning web-based server, `https://peptide.bio`, that predicts peptide properties using recurrent neural networks (RNN) via users' local devices. These trained models are implemented in JavaScript and are exported to the user's web browser exactly as stored. The users simply make predictions by running these pre-trained models on a web browser on their local machines, or even cell phones. This removes hosting costs and the conventional dependence on cloud providers or self-hosting of resource-rich academic institutions. Although we make some compromises here on model size and complexity, we expect the continued improvement of hardware (i.e. Moore's law [24]) to increase the type of models possible in JavaScript each year. This serverless approach should accelerate reproducible ML science, while also lowering the gap of between resource-rich universities and the rest.

This manuscript is organized as follows: We start by providing a brief overview of some comparable predictive sequence-based models for the classification tasks in this work (hemolysis, solubility and non-fouling) in Section 1.1. In Section 2, we describe the datasets, architecture of our deep learning models, the choices for the hyperparameters, as well as a high level overview of the methods used in the previous comparable sequence-based models in the literature. This is followed by evaluating the model in a comparative setting with the state-of-art models in Section 3. Finally, we conclude the paper in Section 4, with a discussion of the implications of our findings.

1.1 Previous work

Quantitative structure–activity relationship (QSAR) modelling is a well-established field of research that aims at mapping sequence and structural properties of chemical compounds to their biological activities [25]. QSAR models have been successfully applied to ACE-inhibitory peptides [26–28], antimicrobial peptides [29–32], and antioxidant peptides [33–35]. For solubility predictions, DSResSol [36] improved prediction accuracy (ACC) to 79.6% by identifying long-range interaction information between amino acid k-mers with dilated convolutional neural networks and outperformed all existing models such as DeepSol [37], PaRSnIP [38], SoluProt [39] and PROSO II [40]. HAPPENN [41] forms the state-of-art model for hemolytic activity prediction with ACC of 85.7% and has better performance compared with HemoPI [42] and HemoPred [43].

2 Methods

2.1 Datasets

2.1.1 Hemolysis

Hemolysis is defined as the disruption of erythrocyte membranes that decrease the life span of red blood cells and causes the release of Hemoglobin. Identifying hemolytic antimicrobial is critical to their applications as non-toxic and safe measurements against bacterial infections. However, distinguishing between hemolytic and non-hemolytic peptides is complicated, as they primarily exert their activity at the charged surface of the bacterial plasma membrane. Timmons and Hewage [41] differentiate between the two whether they are active at the zwitterionic eukaryotic membrane, as well as the anionic prokaryotic membrane. In this work, the model for hemolytic prediction is trained using data from the Database of Antimicrobial Activity and Structure of Peptides (DBAASP v3 [44]). The activity is defined by extrapolating a measurement assuming dose response curves to the point at which 50% of red blood cells (RBC) are lysed. If the activity is below $100\frac{\mu g}{ml}$, it is considered hemolytic. Each measurement is treated independently, so sequences can appear multiple times. The training data contains 9,316 positive and negative sequences of only L- and canonical amino acids.

2.1.2 Solubility

The training data contains 18,453 positive and negative sequences based on data from PROSO II [40]. Solubility was estimated by retrospective analysis of electronic laboratory notebooks. The notebooks were part of a large effort called the Protein Structure Initiative and consider sequences linearly through the following stages: Selected, Cloned, Expressed, Soluble, Purified, Crystallized, HSQC (heteronuclear single quantum coherence), Structure, and deposited in PDB [45]. The peptides were identified as soluble or insoluble by "Comparing the experimental status at two time points, September 2009 and May 2010, we were able to derive a set of insoluble proteins defined as those which were not soluble in September 2009 and still remained in that state 8 months later." [40]

2.1.3 Non-fouling

Data for predicting resistance to non-specific interactions (non-fouling) is obtained from [30]. positive data contains 3,600 sequences. Negative examples are based on 13,585 sequences coming from insoluble and hemolytic peptides, as well as, the scrambled positives. The scrambled negatives are generated with lengths sampled from the same length range as their respective positive set, and residues sampled from the frequency distribution of the soluble data set. Samples are weighted to account for the class imbalance caused by the dataset size for negative examples. A non-fouling peptide (positive example) is defined using the mechanism proposed in White et al. [46]. Briefly, White et al. showed that the exterior surfaces of proteins have a significantly different frequency of amino acids and this increases in aggregation prone environments, like the cytoplasm. Synthesizing self-assembling peptides that follow this amino acid distribution and coating surfaces with the peptides creates non-fouling surfaces. This pattern was also found inside chaperone proteins, another area where resistance to non-specific interactions is important[47].

2.2 Model Architecture

To identify the position-invariant patterns in the peptide sequences, we build a deep neural network (DNN), using a sequential model from Keras framework [48] and the TensorFlow deep learning library back-end [49]. In specific, the DNN employs bidirectional Long Short Term Memory (LSTM) networks to capture long-range sequence correlations. Compared to the conventional RNNs, LSTM networks with gate control units (input gate, forget gate, and output gate) can learn dependency information between distant residues within peptide sequences more effectively [50–52]. They can also partly overcome the problem of vanishing or exploding gradients in the back-propagation phase of training conventional RNNs [53]. We use a bidirectional LSTM (bi-LSTM) to enhance the capability of our model in learning bidirectional dependence between N-terminal and C-terminal amino acid residues. An overview of the DNN architecture is shown in Figure 2.

Peptide sequences are represented as integer encoded vectors of shape 200, where the integer at each position in the vector corresponds to the index of the amino acid from the alphabet of the 20 essential amino acids: [A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V]. Maximum length of the peptide sequence is fixed at 200, and all sequences with higher lengths are excluded. For those sequences with shorter lengths, zeros are padded to the integer encoding representation to keep the shape fixed at 200 for all examples, to allow input sequences with flexible lengths. Every integer encoded peptide sequence is first fed to an embedding layer. The embedding layer enables us to convert the indices of discrete symbols (i.e. essential amino acids), into a representation of a fixed-length vector of defined size.

This is beneficial in the sense of creating a more compact representation of the input symbols, as well as yielding semantically similar symbols close to one another in the vector space. This embedding layer is trainable, and its weights can be updated during training along with the others layers of the RNN.

The output from the embedding layer either goes to a double stacked bi-LSTM layer or a single LSTM layer, to identify patterns along a sequence that can be separated by large gaps. The former is used in predicting solubility and hemolysis, whereas the latter is for predicting peptide's resistance to non-specific interactions (non-fouling). The rationale behind this choice for the non-fouling model is that the bi-LSTM layer did not contribute to a better performance, when compared with the LSTM layer (same ACC and AUROC of %82 and 0.93, respectively). The output from the LSTM layer is then concatenated with the relative frequency of each amino acid in the input sequences. This choice is partially based on our earlier work [54], and helps with improving model performance. The concatenated output is then normalized and fed to a dropout layer with a rate of 10%, followed by a dense neural network with ReLU activation function. This is repeated three times, and the final single-node dense layer uses a sigmoid activation function to force the final prediction as a value between 0 and 1. This scalar output shows the probability of the label being positive for the corresponding predicted peptide biological activity.

The hyperparameters are chosen based on a random search that resulted the best model performance in terms of the Area Under the Receiver Operating Characteristic (AUROC) curve [55] and accuracy. The AUROC shows the model's ability to discriminate between positive and negative examples as the discrimination threshold is varied, and the accuracy is defined as the ratio of correct predictions to the total number of predictions made by the model. The embedding layer has the same input dimension of 21 (alphabet length added by one to account for the padded zeros), and output dimension of 32. The LSTM layer has 64 units, and the first, second and third dense layers have 64, 16 and 1 units, respectively. We train with Adam optimizer [56] of binary cross-entropy loss function, which is defined as

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)], \tag{1}$$

where y_i is the true value of the ith example, $\hat{y}_i$ is the corresponding prediction, and N is the size of the dataset. The learning rate is adapted using a cosine decay schedule with an initial learning rate of 10^{-3}, decay steps of 50 and minimum of 10^{-6}. Data split for training, validation and test is 81%, 9% and 10%, respectively. To avoid overfitting, we add early stopping with patience of 5 that restores model weights from the epoch with the maximum AUROC on the validation set during training.

Previous models for peptide prediction tasks use a variety of deep learning and classical machine learning methods. The prediction server PROSO II employs a two-layered structure, where the output of a primary Parzen [57] window model for sequence similarity and a logistic regression classifier of amino acid k-mer composition, are fed to a second-level logistic regression classifier. HAPPENN uses normalized features selected by SVM and ensemble of Random Forests,

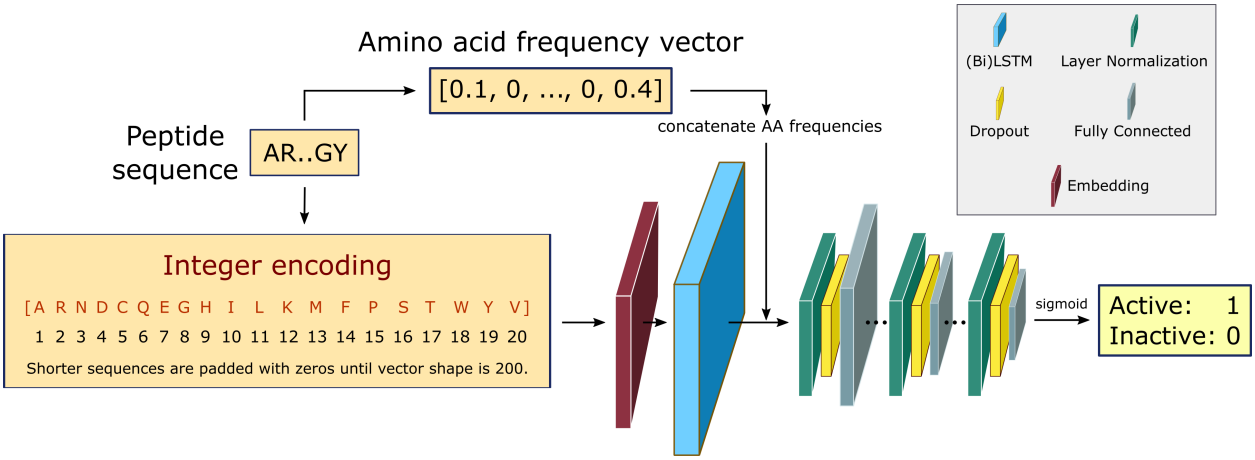

Figure 2: DNN architecture. Fixed-length integer encoded sequences are first fed to a trainable embedding layer, yielding a semantically more compact representation of the input essential amino acids. The bidirectional LSTMS and direct inputs of amino acid frequencies prior to the fully connected layers, improves the learning of bidirectional dependency between distant residues within a sequence. The fully connected layers are down-sized in three consecutive steps with layer normalization and dropout regularization. The final layer uses a sigmoid activation to output a scalar that shows the probability of being active for the desired training task.

which are fed to a deep neural network with batch normalization and dropout regularization to prevent overfitting. DSResSol, that takes advantage of the integration of Squeeze-and-Excitation (SE) [58] residual networks [59] with dilated convolutional neural networks [60]. In specific, the model includes five architectural units, including a single embedding layer, nine parallel initial CNNs with different filter sizes, nine parallel SE-ResNet blocks, three parallel CNNs, and fully connected layers

3 Results

Table 1 shows the classification performance for all the three tasks, along with a comparison between our RNN model and the state-of-the-art methods. All models achieve the same result range as the state-of-the-art methods. We compare the feature extraction capability of our DNN with other unconditional protein language models that provide a pre-trained sequence representations, that transfer well to supervised tasks. In specific, we train two machine learning model on the hemolytic dataset, using UniRep [61, 62] representation of the peptide sequences, followed by a logistic regression, and a Random Forests [63] classifier. Our DNN architecture slightly outperforms both models in terms of AUROC. Our predictive model for the solubility task has the lowest accuracy of 69.0% amongst all, and this is mostly attributed to the difficulty associated with solubility prediction in bioinformatics. The one-hot representation of peptides followed by an RNN results the best hemolysis model in terms of AUROC in [64]. The choice of one-hots requires training features specific to each position though, so we do not expect the model to generalize. In contrast, our model is length-agnostic and will have a relatively smaller generalization error for sequences with lengths it has not observed before. Moreover, this removes the need for having training data at each position for each amino acid.

To allow for transparency between users and developers, details of the models' performance, training procedures, intended use and ethical considerations have been incorporated as model cards [65] on https://peptide.bio/. Model cards present information about how the model was trained, its intended use, caveats about its use, and any ethical or practical concerns when using model predictions.

Table 1: Performance comparison on the testing dataset. Best performing method for each task is in bold. Our approach is highlighted with an asterisk.

Method	**Task**	**ACC(%)**	**AUROC**
Embedding + Bi-LSTM*	Hemolysis	84.0	**0.84**
UniRep + Logistic Regression	Hemolysis	82.0	0.81
UniRep + Random Forests	Hemolysis	84.0	0.78
HAPPENN [41]	Hemolysis	**85.7**	-
one-hots + RNN [64]	Hemolysis	76.0	0.87
Embedding + LSTM*	Non-fouling	**82.0**	**0.93**
Embedding + Bi-LSTM*	Solubility	70.0	0.76
PROSO II [40]	Solubility	75.4	-
DSResSol [36]	Solubility	**79.6**	-

To evaluate the contribution of different architectural components to the model's performance, we conducted a set of ablative experiments on the solubility model only. In each ablation trial, an architectural component is removed and the corresponding test AUC and accuracy is reported via a 5-fold cross-validation on the solubility dataset. We remove the effect of regularization techniques (see methods in Section 2) in our ablation trials by disregarding the early-stopping callback, and fixing the number of training epochs to 50. The learning rate is also set to a fixed value of 10^{-3}. This is the reason for the lower performance of the "full model."

The results from our ablation study are shown in Table 2, sorted by the highest AUROC. The table is sorted by the highest AUROC. We point out that the AUROC of the solubility model has a significant drop from 0.76 to 0.68 after removing the regularization callbacks and fixing the learning rate in our cross-validation analysis. Removing amino acid count frequencies, dropout and layer normalization layer both reduced AUROC by about 2%. The removal of the 1st and 2nd dense layers decreased performance by about 5%. Finally, our ablation analysis shows that the Bi-LSTM is the most contributing component of the architecture, as its removal decreased AUROC by about 10%. Indeed, the bidirectionality feature of Bi-LSTM layers boosts the performance by enabling additional learning of the dependence between N-terminal and C-terminal amino acid residues.

Table 2: Ablation trials to evaluate the contribution of model's architectural components in the classification performance on the solubility dataset via 5-fold cross-validation. For comparison, the performance of the model with full architecture (as shown in Figure 2) is highlighted with an asterisk.

Change in Architecture	**ACC(%)**	**AUROC**
None*(full model)	63.1 ± 4.1	0.683 ± 0.046
Removing AA count frequencies	62.2 ± 2.9	0.667 ± 0.027
Removing dropout	61.7 ± 2.4	0.667 ± 0.030
Removing layer normalization	62.2 ± 3.4	0.661 ± 0.030
Removing 1st and 2nd dense layers	59.8 ± 2.0	0.637 ± 0.021
Removing bidirectionality of LSTM layer	56.1 ± 1.1	0.580 ± 0.015

4 Discussion

We present three sequence-based classifiers to predict hemolysis, solubility, and resistance to non-specific interactions of peptides and achieve competitive results compared with state-of-the art models. The hemolytic model predicts the ability for a peptide to lyse red blood cells, and is intended to be applied to peptides between 1 and 190 residues, L- and canonical amino acids. Hemolysis training dataset is from sequences thought to be antimicrobial or clinically relevant, so it may not generalize to all possible peptides. The solubility model is trained with data mostly containing long sequences, thus, it may not be as applicable to solid-phase synthesized peptides. The model accuracy is low. Its intended use is for peptides or proteins expressed in E. coli that are less than 200 residues long, and may provide solubility predictions more broadly applicable. The non-fouling model predicts the ability for a peptide to resist non-specific interactions, and is intended to be applied to short peptides between 2 and 20 residues. The non-fouling training data mostly contains short sequences, where negative examples have insoluble peptides overrepresented, so the accuracy may be inflated if only comparing soluble peptides.

5 Conclusions

Our proposed DNN models allow for automatic extraction of features from peptide sequences, and removes the reliance on domain experts for feature construction. Moreover, these models are implemented in JavaScript, so that they can run on a static website through a browser on users' phone or desktop. This *serverless* approach removes the conventional dependence of DNN models in bioinformatics on third-party hosted servers, thus, reduces cost, increases flexibility, accessibility and promotes open science.

Acknowledgements

Research reported in this work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R35GM137966. We thank the Center for Integrated Research Computing (CIRC) at University of Rochester for providing computational resources and technical support.

Data and Code Availability

All data and code used to produce results in this study are publically available in the following GitHub repository: https://github.com/ur-whitelab/peptide-dashboard. The JavaScript implementation of the models is available at https://peptide.bio/.

References

[1] Seungwoo Hwang, Zhenkun Gou, and Igor B Kuznetsov. Dp-bind: a web server for sequence-based prediction of dna-binding residues in dna-binding proteins. *Bioinformatics*, 23(5):634–636, 2007.

[2] Shuai Zeng, Ziting Mao, Yijie Ren, Duolin Wang, Dong Xu, and Trupti Joshi. G2pdeep: a web-based deep-learning framework for quantitative phenotype prediction and discovery of genomic markers. *Nucleic Acids Research*, 49(W1):W228–W236, 2021.

[3] Castrense Savojardo, Pier Luigi Martelli, Piero Fariselli, and Rita Casadio. Deepsig: deep learning improves signal peptide detection in proteins. *Bioinformatics*, 34(10):1690–1696, 2018.

[4] Jielu Yan, Pratiti Bhadra, Ang Li, Pooja Sethiya, Longguang Qin, Hio Kuan Tai, Koon Ho Wong, and Shirley WI Siu. Deep-ampep30: improve short antimicrobial peptides prediction with deep learning. *Molecular Therapy-Nucleic Acids*, 20:882–894, 2020.

[5] Kotaro Tsutsumi, Khodayar Goshtasbi, Adwight Risbud, Pooya Khosravi, Jonathan C Pang, Harrison W Lin, Hamid R Djalilian, and Mehdi Abouzari. A web-based deep learning model for automated diagnosis of otoscopic images. *Otology & Neurotology*, 42(9):e1382–e1388, 2021.

[6] Guangyuan Li, Balaji Iyer, VB Surya Prasath, Yizhao Ni, and Nathan Salomonis. Deepimmuno: deep learning-empowered prediction and generation of immunogenic peptides for t-cell immunity. *Briefings in bioinformatics*, 22(6):bbab160, 2021.

[7] Seung-Jin Park, Byoung-Ha Yoon, Seon-Kyu Kim, and Seon-Young Kim. Gent2: an updated gene expression database for normal and tumor tissues. *BMC medical genomics*, 12(5):1–8, 2019.

[8] Chirayu Pankaj Goswami and Harikrishna Nakshatri. Proggenev2: enhancements on the existing database. *BMC cancer*, 14(1):1–6, 2014.

[9] Raul Aguirre-Gamboa, Hugo Gomez-Rueda, Emmanuel Martínez-Ledesma, Antonio Martínez-Torteya, Rafael Chacolla-Huaringa, Alberto Rodriguez-Barrientos, Jose G Tamez-Pena, and Victor Trevino. Survexpress: an online biomarker validation tool and database for cancer gene expression data using survival analysis. *PloS one*, 8 (9):e74250, 2013.

[10] Alexander Koch, Tim De Meyer, Jana Jeschke, and Wim Van Criekinge. Mexpress: visualizing expression, dna methylation and clinical tcga data. *BMC genomics*, 16(1):1–6, 2015.

[11] Chirayu Pankaj Goswami and Harikrishna Nakshatri. Progmir: a tool for identifying prognostic mirna biomarkers in multiple cancers using publicly available data. *Journal of clinical bioinformatics*, 2(1):1–8, 2012.

[12] Raul Aguirre-Gamboa and Victor Trevino. Survmicro: assessment of mirna-based prognostic signatures for cancer clinical outcomes by multivariate survival analysis. *Bioinformatics*, 30(11):1630–1632, 2014.

[13] Jordan Anaya. Oncolnc: linking tcga survival data to mrnas, mirnas, and lncrnas. *PeerJ Computer Science*, 2:e67, 2016.

[14] Jun Li, Leng Han, Paul Roebuck, Lixia Diao, Lingxiang Liu, Yuan Yuan, John N Weinstein, and Han Liang. Tanric: an interactive open platform to explore the function of lncrnas in cancer. *Cancer research*, 75(18):3728–3737, 2015.

[15] Mei-Ju May Chen, Jun Li, Yumeng Wang, Rehan Akbani, Yiling Lu, Gordon B Mills, and Han Liang. Tcpa v3. 0: an integrative platform to explore the pan-cancer analysis of functional proteomic data. *Molecular & Cellular Proteomics*, 18(8):S15–S25, 2019.

[16] Nicholas Borcherding, Nicholas L Bormann, Andrew P Voigt, and Weizhou Zhang. Trgated: A web tool for survival analysis using protein data in the cancer genome atlas. *F1000Research*, 7, 2018.

[17] Vijayachitra Modhukur, Tatjana Iljasenko, Tauno Metsalu, Kaie Lokk, Triin Laisk-Podar, and Jaak Vilo. Methsurv: a web tool to perform multivariable survival analysis using dna methylation data. *Epigenomics*, 10(3):277–288, 2018.

[18] Jianjiong Gao, Bülent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling*, 6(269):pl1–pl1, 2013.

[19] Hong Zheng, Guosen Zhang, Lu Zhang, Qiang Wang, Huimin Li, Yali Han, Longxiang Xie, Zhongyi Yan, Yongqiang Li, Yang An, et al. Comprehensive review of web servers and bioinformatics tools for cancer prognosis analysis. *Frontiers in oncology*, 10:68, 2020.

[20] Mike May and Herb Brody. Nature index 2015 global. *Nature*, 522(7556):S1–S1, 2015.

[21] Marcelo CR Melo, Jacqueline RMA Maasch, and Cesar de la Fuente-Nunez. Accelerating antibiotic discovery through artificial intelligence. *Communications biology*, 4(1):1–13, 2021.

[22] Maria Littmann, Katharina Selig, Liel Cohen-Lavi, Yotam Frank, Peter Hönigschmid, Evans Kataka, Anja Mösch, Kun Qian, Avihai Ron, Sebastian Schmid, et al. Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nature Machine Intelligence*, 2(1):18–24, 2020.

[23] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[24] Chris A Mack. Fifty years of moore's law. *IEEE Transactions on semiconductor manufacturing*, 24(2):202–207, 2011.

[25] Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12):4977–5010, 2014.

[26] Baichuan Deng, Xiaojun Ni, Zhenya Zhai, Tianyue Tang, Chengquan Tan, Yijing Yan, Jinping Deng, and Yulong Yin. New quantitative structure–activity relationship model for angiotensin-converting enzyme inhibitory dipeptides based on integrated descriptors. *Journal of agricultural and food chemistry*, 65(44):9774–9781, 2017.

[27] Yu-Tang Wang, Daniel P Russo, Chang Liu, Qian Zhou, Hao Zhu, and Ying-Hua Zhang. Predictive modeling of angiotensin i-converting enzyme inhibitory peptides using various machine learning approaches. *Journal of agricultural and food chemistry*, 68(43):12132–12140, 2020.

[28] Xiao Guan and Jing Liu. Qsar study of angiotensin i-converting enzyme inhibitory peptides using svhehs descriptor and osc-svm. *International Journal of Peptide Research and Therapeutics*, 25(1):247–256, 2019.

[29] Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, Michael Tartakovsky, Grigol Managadze, Maya Grigolava, George I Makhatadze, and Malak Pirtskhalava. Predictive model of linear antimicrobial peptides active against gram-negative bacteria. *Journal of chemical information and modeling*, 58(5):1141–1151, 2018.

[30] Rainier Barrett, Shaoyi Jiang, and Andrew D White. Classifying antimicrobial and multifunctional peptides with bayesian network models. *Peptide Science*, 110(4):e24079, 2018.

[31] YONGZHONG LU, QIAN QIU, DAOLE KANG, and JIE LIU. Qsar modeling of antimicrobial peptides based on their structural and physicochemical properties. *Journal of Biology and Nature*, pages 120–126, 2018.

[32] Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero Dos Santos, Pin-Yu Chen, et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.

[33] Nan Chen, Ji Chen, Bo Yao, and Zhengguo Li. Qsar study on antioxidant tripeptides and the antioxidant activity of the designed tripeptides in free radical systems. *Molecules*, 23(6):1407, 2018.

[34] Baichuan Deng, Hongrong Long, Tianyue Tang, Xiaojun Ni, Jialuo Chen, Guangming Yang, Fan Zhang, Ruihua Cao, Dongsheng Cao, Maomao Zeng, et al. Quantitative structure-activity relationship study of antioxidant tripeptides based on model population analysis. *International journal of molecular sciences*, 20(4):995, 2019.

[35] Tobias Hegelund Olsen, Betül Yesiltas, Frederikke Isa Marin, Margarita Pertseva, Pedro J García-Moreno, Simon Gregersen, Michael Toft Overgaard, Charlotte Jacobsen, Ole Lund, Egon Bech Hansen, et al. Anoxpepred: using deep learning for the prediction of antioxidative properties of peptides. *Scientific reports*, 10(1):1–10, 2020.

[36] Mohammad Madani, Kaixiang Lin, and Anna Tarakanova. Dsressol: A sequence-based solubility predictor created with dilated squeeze excitation residual networks. *International Journal of Molecular Sciences*, 22(24): 13555, 2021.

[37] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.

[38] Reda Rawi, Raghvendra Mall, Khalid Kunji, Chen-Hsiang Shen, Peter D Kwong, and Gwo-Yu Chuang. Parsnip: sequence-based protein solubility prediction using gradient boosting machine. *Bioinformatics*, 34(7):1092–1098, 2018.

[39] Jiri Hon, Martin Marusiak, Tomas Martinek, Antonin Kunka, Jaroslav Zendulka, David Bednar, and Jiri Damborsky. Soluprot: prediction of soluble protein expression in escherichia coli. *Bioinformatics*, 37(1):23–28, 2021.

[40] Pawel Smialowski, Gero Doose, Phillipp Torkler, Stefanie Kaufmann, and Dmitrij Frishman. Proso ii–a new method for protein solubility prediction. *The FEBS journal*, 279(12):2192–2200, 2012.

[41] P. Brendan Timmons and Chandralal M. Hewage. Happenn is a novel tool for hemolytic activity prediction for therapeutic peptides which employs neural networks. *Scientific reports*, 10(1):1–18, 2020.

[42] Kumardeep Chaudhary, Ritesh Kumar, Sandeep Singh, Abhishek Tuknait, Ankur Gautam, Deepika Mathur, Priya Anand, Grish C Varshney, and Gajendra PS Raghava. A web server and mobile app for computing hemolytic potency of peptides. *Scientific reports*, 6(1):1–13, 2016.

[43] Thet Su Win, Aijaz Ahmad Malik, Virapong Prachayasittikul, Jarl E S Wikberg, Chanin Nantasenamat, and Watshara Shoombuatong. Hemopred: a web server for predicting the hemolytic activity of peptides. *Future medicinal chemistry*, 9(3):275–291, 2017.

[44] Malak Pirtskhalava, Anthony A Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, and Michael Tartakovsky. Dbaasp v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic acids research*, 49(D1):D288–D297, 2021.

[45] Helen M Berman, John D Westbrook, Margaret J Gabanyi, Wendy Tao, Raship Shah, Andrei Kouranov, Torsten Schwede, Konstantin Arnold, Florian Kiefer, Lorenza Bordoli, et al. The protein structure initiative structural genomics knowledgebase. *Nucleic acids research*, 37(suppl_1):D365–D368, 2009.

[46] Andrew D White, Ann K Nowinski, Wenjun Huang, Andrew J Keefe, Fang Sun, and Shaoyi Jiang. Decoding nonspecific interactions from nature. *Chemical Science*, 3(12):3488–3494, 2012.

[47] Andrew D White, Wenjun Huang, and Shaoyi Jiang. Role of nonspecific interactions in molecular chaperones through model-based bioinformatics. *Biophysical journal*, 103(12):2484–2491, 2012.

[48] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[49] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[50] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

[51] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.

[52] Yilin Ye, Jian Wang, Yunwan Xu, Yi Wang, Youdong Pan, Qi Song, Xing Liu, and Ji Wan. Mathla: a robust framework for hla-peptide binding prediction integrating bidirectional lstm and multiple head attention mechanism. *BMC bioinformatics*, 22(1):1–12, 2021.

[53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[54] Rainier Barrett and Andrew D White. Investigating active learning and meta-learning for iterative peptide design. *Journal of chemical information and modeling*, 61(1):95–105, 2020.

[55] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[57] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[58] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[60] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arxiv 2016. *arXiv preprint arXiv:1511.07122*, 2016.

[61] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[62] Eric J Ma and Arkadij Kummer. Reimplementing unirep in jax. *bioRxiv*, 2020.

[63] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[64] Alice Capecchi, Xingguang Cai, Hippolyte Personne, Thilo Köhler, Christian van Delden, and Jean-Louis Reymond. Machine learning designs non-hemolytic antimicrobial peptides. *Chemical Science*, 12(26):9221–9232, 2021.

[65] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.