Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2024, **3**, 977

Learning peptide properties with positive examples only

Mehrad Ansari and Andrew D. White *

Deep learning can create accurate predictive models by exploiting existing large-scale experimental data, and guide the design of molecules. However, a major barrier is the requirement of both positive and negative examples in the classical supervised learning frameworks. Notably, most peptide databases come with missing information and low number of observations on negative examples, as such sequences are hard to obtain using high-throughput screening methods. To address this challenge, we solely exploit the limited known positive examples in a semi-supervised setting, and discover peptide sequences that are likely to map to certain antimicrobial properties *via* positive-unlabeled learning (PU). In particular, we use the two learning strategies of adapting base classifier and reliable negative identification to build deep learning models for inferring solubility, hemolysis, binding against SHP-2, and non-fouling activity of peptides, given their sequence. We evaluate the predictive performance of our PU learning method and show that by only using the positive data, it can achieve competitive performance when compared with the classical positive–negative (PN) classification approach, where there is access to both positive and negative examples.

Received 5th November 2023
Accepted 30th March 2024

DOI: 10.1039/d3dd00218g

rsc.li/digitaldiscovery

1 Introduction

As short-chain amino acids, peptides have attracted growing attention in pharmaceutics,[1–3] therapeutics,[4–6] immunology,[7–9] and biomaterials design.[10–12] However, the development of novel peptides remains a challenge due to poor pharmacokinetic properties that restrict the design space and necessitate unnatural amino acids or cyclization, increasing the complexity of their design.[13] Computational design and data-driven discovery strategies have arisen as promising low-cost techniques in the pre-experiment phase to expedite the process of generating accurate predictions of peptide properties, and shortlist promising candidates for follow-up experimental validation. Some examples of these successful applications include single nucleotide polymorphisms (SNP) and small-indel calling,[14] estimating the impact of non-coding variants on DNA-methylation,[15] as well as for the prediction of protein function,[16] structure,[17,18] and protein–protein interactions.[19] Sequence-based learning strategies aim at mapping peptide's natural biological function to its sequence. In a supervised learning setting, this is done by training on sequence-function examples. This means that sequence–function relationships are learned by iteratively training on samples of different classes (*i.e.* positive and negative examples in binary classification). The performance of the classifier is highly dependent on the quality of the training samples and the ratio of the positive and negative samples.[20,21] In bioinformatics, a variety of supervised-learning algorithms, such as support vector machines,[22] random forest,[23] logistic regression,[24] and naive Bayesian classifier,[25] have been successfully applied to develop classification models.

However, lack of negative examples in numerous biological applications[26–29] limits the feasibility of constructing such reliable classifiers. As an example, medical information records typically contain the positively diagnosed diseases of a patient, and the absence of a diagnostic record does not necessarily rule out a disease for him/her. Most high-throughput screening methods solely focus on identifying the positive examples, thus, it is much more straightforward to confirm a property than to ascertain that it does not hold. As an example, a potential binding site is confirmed if a protein binds to a target, but failure to bind only means that the binding conditions were not satisfied under a given experimental setting. With the technological advances, identifying specific properties can be improved, and biological samples formerly not known to have a property can now be classified with confidence. As an example, ref. 30 demonstrated on the changes in protein glycosylation site labeling throughout four time points over 10 years. Another example is protein–protein interaction (PPI),[31,32] where experimentally validated interacting and non-interacting protein pairs are used as positive and negative examples, respectively. However, the selection of non-interacting protein pairs can be challenging for two reasons: (1) as more novel PPIs are constantly being discovered over time, some non-interacting protein pairs (*i.e.* negative examples) might be mislabeled. (2)

Department of Chemical Engineering, University of Rochester, Rochester, NY, 14627, USA. E-mail: andrew.white@rochester.edu

The positive examples are significantly outnumbered by a large number of protein pairs for which no interactions have been identified. Similar situations can be found in drug–drug interaction identification,[33] small non-coding RNA detection,[34] gene function[35,36] and phage–bacteria interaction[37] prediction, and biological sequence classification.[38,39]

To address the challenges above, we demonstrate on a positive-unlabeled (PU) learning framework to infer peptide sequence-function relationships, by solely exploiting the limited known positive examples in a semi-supervised setting. Semi-supervised learning techniques are a special instance of weak supervision,[40,41] where the training is based on partially labeled training data (*i.e.* labeled data can be either positive or both positive and negative samples). PU learning builds classification models by primarily leveraging a small number of labeled positive samples and a huge volume of unlabeled samples (*i.e.* a mixture of both *positive* (P) and *negative* (N) samples).[42] Depending on how the *unlabeled* (U) data is handled, existing PU learning strategies are divided into two categories. (1) *Reliable negative identification*: this category identifies *reliable negatives* (RN) within U, and then performs ordinary supervised (PN) learning;[43,44] (2) *adapting the base classifier*: this treats the U samples as N with smaller weights (biased learning) and adapts the conventional classifiers to directly learn from P and U samples.[45,46] The former *reliable negative identification* strategies rely on heuristics to identify the RN, and they have been widely used in none-coding RNA identification,[34] none-coding RNA-disease association,[47] gene function prediction,[35,48] disease gene identification,[26,49,50] and single-cell RNA sequencing quality control.[51] On the other hand, *adapting the base classifier* algorithms are Bayesian-based approaches that focus on estimating the ratio of positive and negative samples in U (class prior), which then can be applied for classification using the Bayes' rule. One major limitation is that their performance largely depends on good choices of weights of U samples, which are computationally expensive to tune.[52] Thus, compared to the first strategy, there has been a fewer use cases of them in the literature.[53–56] An excellent overview of PU leaning strategies can be found in ref. 42 and 20 also systematically reviewed the implementation of 29 PU learning methods in a wide range of biological topics.

In this work, we take advantage of the flexibility of reliable negative identification PU strategy, and discover peptide sequences that are likely to map to certain properties (Fig. 1). Specifically, we demonstrate on a two-step technique, where Step 1 handles the deficiency of negative training examples by extracting a subset of the U samples that can be confidently labeled as N (*i.e.* RN). Subsequently, step 2 involves training a deep neural network classifier using the P and the extracted RN, and applying it to the remaining pool of U. Reliable negative identification in step 1, is an adaption of the *Spy* technique formerly employed in handling unlabeled text data.[43] In this approach, some randomly selected positive samples are defined as spies, and are intentionally mislabeled as negatives. The reliable negative examples are found within the unlabeled samples for which the posterior probability is lower than the posterior probability of the spies. We use our approach to predict different peptide properties, such as hemolysis, resistance to non-specific interactions (non-fouling), and solubility.

This manuscript is organized as follows: in Section 2, we describe the datasets, architecture of the deep learning models, and our choices for the hyperparameters. This is followed by evaluating the model in a comparative setting with the classical PN classifier in Section 3. Finally, we conclude the paper in Section. 4, with a discussion of the implications of our findings.

2 Materials and methods

2.1 Datasets

2.1.1 Hemolysis. Hemolysis is referred to the disruption of erythrocyte membranes that decrease the life span of red blood cells and causes the release of Hemoglobin. It is critical to identify non-hemolytic antimicrobial peptides as a non-toxic and safe measure against bacterial infections. However, distinguishing between hemolytic and non-hemolytic peptides is a challenge, since they primarily exert their activity at the charged surface of the bacterial plasma membrane. In this work, the hemolysis classifier is trained using data from the Database of Antimicrobial Activity and Structure of Peptides (DBAASP v3 (ref. 57)). Hemolytic activity is defined by extrapolating a measurement assuming dose response curves to the point at which 50% of red blood cells are lysed. Activities below 100 $\mu g\ ml^{-1}$, are considered hemolytic. The data contains 9316 sequences (19.6% positives and 80.4% negatives) of only L- and canonical amino acids. Each measurement is treated independently, so sequences can appear multiple times. This experimental dataset contains noise, and in some observations ($\sim$40%), an identical sequence appears in both negative and positive class. As an example, sequence "RVKRVWPLVIRTVIAGYNLYRAIKKK" is found to be both hemolytic and non-hemolytic in two different lab experiments (*i.e.* two different training examples).

2.1.2 Solubility. This data contains 18 453 sequences (47.6% positives and 52.4% negatives) based on PROSO II,[58] where 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.[59] 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."[58]

2.1.3 Non-fouling. Non-fouling is defined as resistance to non-specific interactions, and this data is obtained from ref. 60. A non-fouling peptide (positive example) is defined using the mechanism proposed in ref. 61. Briefly,[61] 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

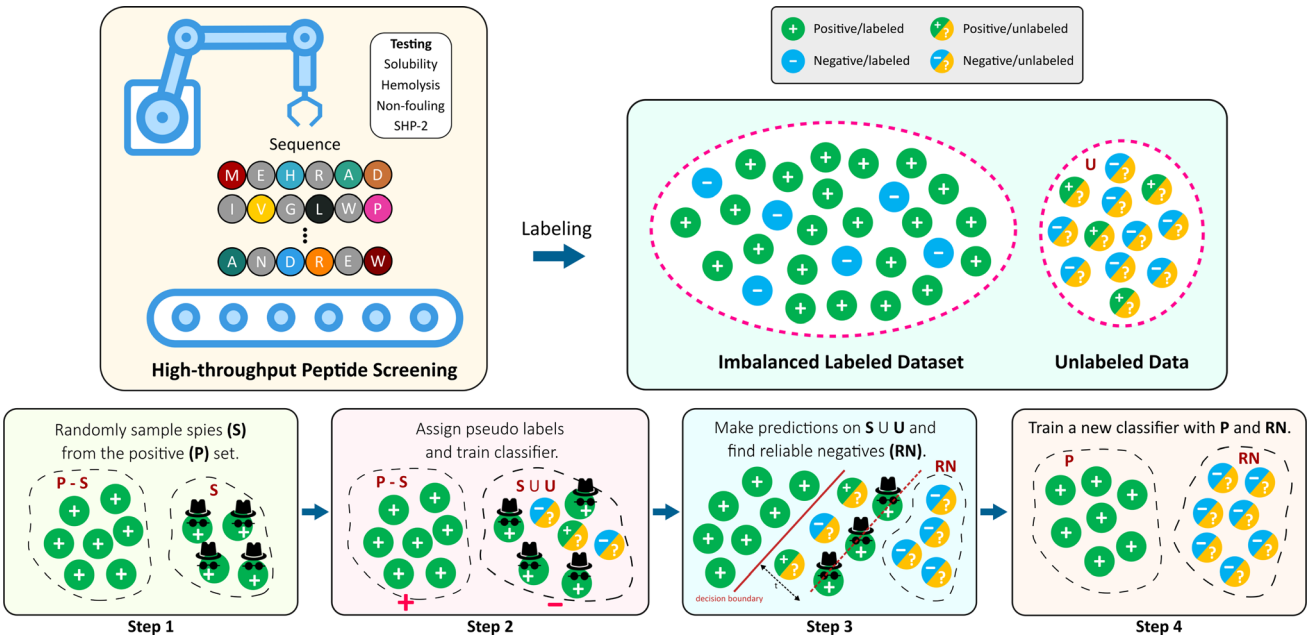

Fig. 1 Overview of this work. High-throughput screening methods are commonly good at identifying positive examples, leaving imbalanced datasets (skewed towards the positive class) that are not suitable for supervised learning algorithms. In this work, we use the positive examples only to distinguish between the positive and negative samples using Spy technique.

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.[62] Positive data contains 3600 sequences. Negative examples are based on 13 585 sequences (79.1% of dataset are negatives) 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 dataset. Samples are weighted to account for the class imbalance caused by the negative examples dataset size. This dataset is gathered based on the mechanism proposed in ref. 61.

2.1.4 SHP-2. SHP-2 is a ubiquitous protein tyrosine phosphatase, whose activity is regulated by phosphotyrosine (pY)-containing peptides generated in response to extracellular stimuli. SHP-2 is involved in processes such as cell growth, differentiation, migration, and immune response.[63] The SHP-2 dataset contains fixed-length peptides (5 AA residues) optimized for binding to N-SH2 domain, obtained from ref. 64. Total dataset size is 300, with 50% positive examples (Table 1).

2.2 Model architecture

We build a recurrent neural network (RNN) to identify the position-invariant patterns in the peptide sequences, using a sequential model from Keras framework[66] and the TensorFlow deep learning library back-end.[67] In specific, the RNN employs bidirectional Long Short-Term Memory (LSTM) networks to capture long-range correlations between the amino acid residues. Compared to the conventional RNNs, LSTM networks with gate control units can learn dependency information between distant residues within peptide sequences more effectively.[68–70] An overview of the RNN architecture is shown in Fig. 2. This architecture is identical to the one used in our recent work in edge-computing cheminformatics.[65]

The input peptide sequences are integer encoded as 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]. For implementation purposes during the training step, the maximum length of the vector is fixed at 200, padding zeros to shorter length sequences. 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 sequence is first fed to an embedding layer with trainable weights, where the indices of discrete symbols (*i.e.* essential amino acids), into a representation of a fixed-length vector of defined size.

The embedding layer output either goes to a double stacked bi-LSTM layer (for solubility and hemolysis) or a single LSTM layer (for SHP-2 and non-fouling), to identify patterns along a sequence that can be separated by large gaps. 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,[64] 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 predict the peptide biological activity as the probability of the label being positive.

Table 1 Summary of used datasets. For more details, refer to ref. 65

	Hemolysis	Solubility	Non-fouling	SHP-2
Definition	Hemolysis is the process by which red blood cells (RBCs) rupture and release their contents, mainly Hemoglobin, into the surrounding plasma or extracellular fluid. Based on DBAASP v3.[57]	Solubility was defined in PROSO II[58] as a sequence that was transfectable, expressible, secretable, separable, and soluble in *E. coli* system	Resistance to non-specific interactions. Gathered using the mechanism proposed in ref. 61	SHP-2 is a protein encoded by the PTPN11 gene in humans. It is a non-receptor protein tyrosine phosphatase that plays a critical role in various cellular signaling pathways[63]
Total size	9316	18 453	17 185	300
Positive examples	19.6%	47.6%	20.9%	50.0%
Length range	1–190 AA residues	19–198 AA residues	5–198 AA residues	5 AA residues

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 curve (AUROC) and accuracy (ACC). Readers are encouraged to refer to ref. 65 for more details on the model architecture and its hyperparameters. We compile our Keras model using Adam optimizer[71] with a binary cross-entropy loss function, which is defined as

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

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.

2.3 Positive-unlabeled learning

Let $\vec{x}$ be an example, and $y \in \{0, 1\}$ the true binary label for the instance $\vec{x}$. If $\vec{x}$ is a positive example, $y = 1$, otherwise $y = 0$. Let $s = 1$, if example $\vec{x}$ is labeled, and $s = 0$, if $\vec{x}$ is unlabeled. Only positive examples are labeled (*i.e.* $p(s = 1|\vec{x}, y = 0) = 0$). In other words, the probability that a negative example appears in the labeled set is zero. On the other hand, the unlabeled set $p(s = 1|\vec{x}, y = 0) = 0$ can contain positive $(y = 1|\vec{x}, s = 0)$ or negative $(y = 0|\vec{x}, s = 0)$ examples. The goal is to learn a probabilistic binary classifier as a function $f(\vec{x})$, such that $f(\vec{x}) = p(y = 1|\vec{x})$, *i.e.* the conditioned probability of being positive given a feature vector $\vec{x}$.

In this work, we focus on two PU learning strategies; Adapting Base Classifier and Reliable Negative Identification.

2.3.1 Adapting base classifier. Adapting base classifier, also known as class prior estimation, are Bayesian-based methods that adapt the base classifier (*i.e.* SVM) to estimate the expected ratio of positive or negative examples in the unlabeled set. Note that in this work, we use an RNN as our base classifier. This approach simply tries to adjust the probability of being positive estimated by a traditional classifier trained with positive and unlabeled examples, where the unlabeled is treated as the negative class. The positive likelihood score $p(y = 1|\vec{x})$ is estimated by ref. 72 as

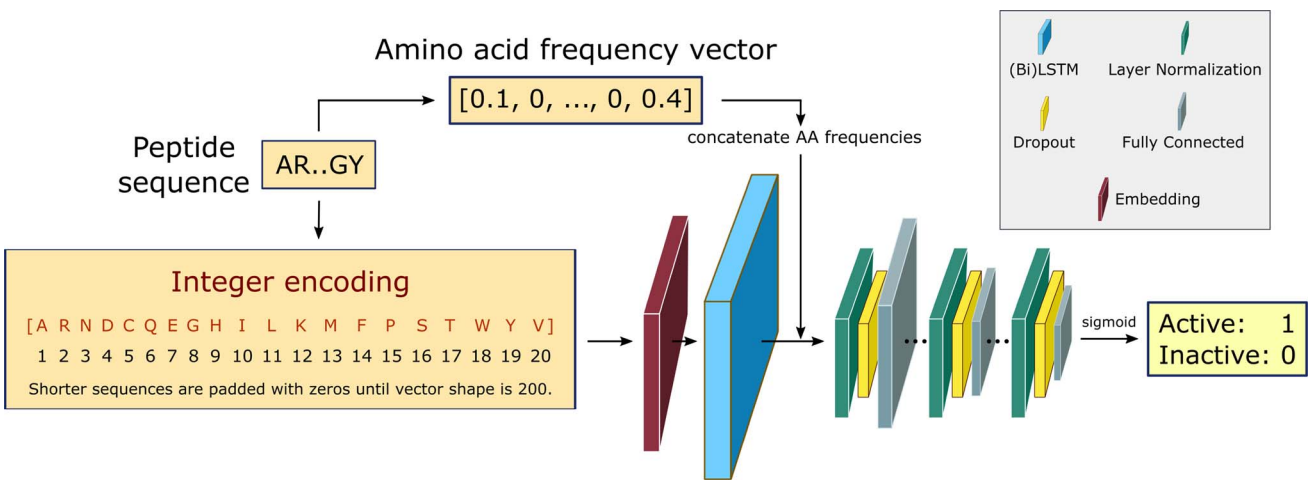

Fig. 2 RNN architecture.[65] Padded integer encoded sequences are first fed to a trainable embedding layer, yielding a semantically more compact representation of the input essential amino acids. The use of bidirectional LSTMS and direct inputs of amino acid frequencies prior to the fully connected layers, improves the learning of bidirectional dependency between distant residues within a sequence. The fully connected layers are down-sized in three consecutive steps *via* layer normalization and dropout regularization. The final layer outputs the probability of being active for the desired training task using a sigmoid activation function.

$$f(\vec{x}) = p(y = 1|\ \vec{x}) = \frac{p(s = 1|\vec{x})}{p(s = 1|y = 1, \vec{x})}, \quad (2)$$

where $p(s = 1|\ \vec{x})$ is the likelihood of the example $\vec{x}$ being labeled (thus, being positive), learned from the labeled and unlabeled data. $p(s = 1|y = 1, \vec{x})$ denotes the posterior probability of the example $\vec{x}$, *i.e.* positive sample being labeled as positive in the training data. Assuming that the labeled positive samples are chosen completely randomly from all positive examples, $p(s = 1|y = 1,\ \vec{x})$ is treated as a constant factor (c) for all the samples, that can be obtained through a validation (held-out) set.[53] This *"selected completely at random"* assumption can be also written as $c = p(s = 1|y = 1, \vec{x}) = p(s = 1|y = 1)$, where c is a constant probability that a positive sample is labeled. This assumption is analogous to the *"missing completely at random"* assumption that is made when learning data with missing values.[73–75] Among the empirical estimators for c proposed in [72], we use the following average:

$$c = p(s = 1|y = 1) = \frac{\sum_{x \in R} p(s = 1|\vec{x})}{\sum_{x \in V} p(s = 1|\vec{x})}, \quad (3)$$

where V is the validation set, drawn in the same manner as the training set, and $R \subseteq V$ is a set of positive examples in V. A threshold is adjusted within range $(0 - 1/c)$ to discriminate if the sample belongs to the positive or negative class, by maximizing Cohen's kappa coefficient.[76] It is important to note that the [72] algorithm was not developed to handle noisy labeled data. In addition, the theory behind its estimator limits its use to classify conditional distributions with non-overlapping support.[77]

2.3.2 Reliable negative identification. Reliable negative identification adopts two independent algorithms: (1) identify the reliable negatives (RN) within the unlabeled set given the likelihood and (2) train a binary classifier to distinguish the labeled positive examples from the identified RN set. This approach is based on two assumptions of smoothness and separability, which simply means that all the positive examples are similar to the labeled examples, and that the negative examples are very different from them, respectively.[42] Several techniques have been proposed to extract the reliable negatives or positives from the unlabeled set, such as Spy,[43] Cosine-Rocchio,[78] Rocchio,[44] 1DNF,[79] PNLH,[80] and Augmented Negatives,[81] and DILCA.[82]

In this work, we use Spy to find the reliable negatives. First, a small randomly selected group of positive examples (S) are removed and put in the unlabeled data as spies. This allows us to define new datasets P_s and U_s, respectively. The percentage of positive instances used as spies is defined by *spy-rate* (in this work, we use 0.2). Then, a classifier f_1 is trained based on P_s and U_s. Next, the boundary of RN under the rule that most of the spies are classified as positives is found, based on spy-tolerance (ε). ε determines what percentage of spies can remain in the unlabeled set when the decision boundary threshold (t_s) is calculated (in this work, we use 0.05). In other words, t_s is the posterior likelihood such that all added spies during training f_1 are classified as positives. All samples in U_s, whose posterior likelihood is smaller than t_s are considered RN. Finally, we train a new classifier f_2 given original positive samples (P) and the found RN.

Algorithm 1 Reliable Negative Identification with Spy

```
Initialize RN = {}
Randomly sample spy-rate% from P
P_s = P − S, U_s = U ∪ S
Assign P_s to y = 1 class, and U_s to y = 0 class
Train classifier f_1 on P_s and U_s
Use f_1 to classify U_s and find p(y = 1|U_s)
Adjust t_s such that ratio% of the positively classed U_s is less than ϵ
for x ∈ U_s do
    if p(y = 1|x) < t_s then
        RN = RN ∪ x
        U_s = U_s − x
    end if
end for
return RN, U_s
Train classifier f_2 on P and RN
```

3 Results and discussion

In this section, we evaluate the estimated generalization error of our PU approach, and compare it with the classical PN classification, where both positive and negative examples are available for training. Note that the test data contains fixed unobserved real positive and negative examples with a consistent ratio across all PN and PU case studies. Thus, regardless of the size of the unlabeled data generated, the performance metrics can be fairly compared. We take two approaches to generate the unlabeled data: (1) unlabeled Data Generated from Positive and Negatives Samples. In this setting, the unlabeled data is generated from a mixture of known positive and negative examples for each task. (2) Unlabeled Data Generated from Mutated Positive Samples. Given a distribution of positive examples, we generate unlabeled examples by randomly breaking the positive examples into sub-sequences, and filling up a similar-length sequence, with these sub-sequences.

Table 2 Performance comparison between PU learning and classical PN learning for different prediction tasks, with the unlabeled data generated from positive and negatives samples. PN models are trained by having access to both positive and negative data, based on our earlier work in ref. 65. The test data contains fixed unobserved real positive and negative examples with a consistent ratio across all PN and PU case studies

		PU		PN	
Task	PU method	ACC (%)	AUROC	ACC (%)	AUROC
Hemolysis	Adapting base classifier	83.1	0.78	84.0	0.84
Hemolysis	Reliable negative identification	84.1	0.80		
Non-fouling	Adapting base classifier	93.8	0.93	82.0	0.93
Non-fouling	Reliable negative identification	95.0	0.93		
Solubility	Adapting base classifier	53.0	0.59	70.0	0.76
Solubility	Reliable negative identification	86.7	0.68		
SHP-2	Adapting base classifier	84.1	0.87	83.3	0.82
SHP-2	Reliable negative identification	90.2	0.93		

Duplicate sequence are removed after the generation step. This allows us to generate the unlabeled data, by creating mutations of the positive examples *without* any knowledge on what the true negative examples are, thus, making our approach agnostic with respect to the unknown ground-truth of distribution of the negative peptide examples in the sequence space.

3.1 Unlabeled data generated from positive and negatives samples

Performance comparison between our PU learning methods and classical PN learning for different prediction tasks are presented in Table 2. Results for all the PN models are based on our earlier work in ref. 65. For every task, we make comparisons of the model accuracy (ACC%), and the area under the receiver operating characteristic curve (AUROC), using the two the Adapting Base classifier, and the Reliable Negative Identification PU methods. Across all prediction tasks, with one exception of Hemolysis and Solubility with the Adapting Base Classifier method, the accuracy of our PU methods are considerably higher than the PN classification. Comparing the two PU methods, it is observed that Reliable Negative Identification outperforms Adapting Base Classifier method for all prediction tasks. Surprisingly, for the non-fouling and SHP-2 predictions, both PU methods outperform the PN classifier.

3.2 Unlabeled data generated from mutated positive samples

Table 3 shows performance comparison between our PU learning method and classical PN learning for different prediction tasks. Considering the much better performance of Reliable Negative Identification compared to the Adapting Base Classifier observed in Table 2, we only consider the Reliable Negative Identification PU method for this unlabeled data generation scenario. Note that the solubility model in this setting showed a poor performance, and was excluded in our comparison. Considering the ACC and AUROC reported in Table 3, our PU method is able to reasonably discriminate between the positive and the reliable negatives identified from the generated unlabeled examples.

It is important to note that with the unlabeled data generation, we can control how large the size of the generated unlabeled examples are. The generated unlabeled : labeled ratio reported in Table 3 is fixed at 8.0. Next, we investigate the effect of unlabeled : labeled ratio on the performance of Reliable Negative Identification strategy across all prediction tasks in Fig. 3. Each point represents the average value of AUROC and ACC% (left and right panel, respectively) over 6 models trained with a different choice of randomly selected spy positives, and error bars show the magnitude of the standard deviation. Horizontal dashed lines show the performance of the PN classifier for each task represented as a baseline for performance comparison. With very small generated unlabeled samples (*i.e.* unlabeled : labeled ratio ≈ 2.0), the exploration of new examples that can qualify as reliable negatives will be largely limited. Thus, the trained f_2 classifier has a significantly lower performance compared to the baseline PN classifier and to the other PU models trained with higher generated unlabeled : labeled ratios. With larger unlabeled : labeled ratios (*i.e.* >10.0), we see a better prediction performance across all the tasks. There are

Table 3 Performance comparison between PU learning and classical PN learning for different prediction tasks, with the unlabeled data generated from mutated positive samples. Generated unlabeled is 8 times larger than the positive size. PN models are trained by having access to both positive and negative data, based on our earlier work in ref. 65. The test data contains fixed unobserved real positive and negative examples with a consistent ratio across all PN and PU case studies

		PU		PN	
Task	PU method	ACC (%)	AUROC	ACC (%)	AUROC
Hemolysis	Reliable negative identification	76.8	0.75	84.0	0.84
Non-fouling	Reliable negative identification	94.1	0.87	82.0	0.93
SHP-2	Reliable negative identification	84.8	0.91	83.3	0.82

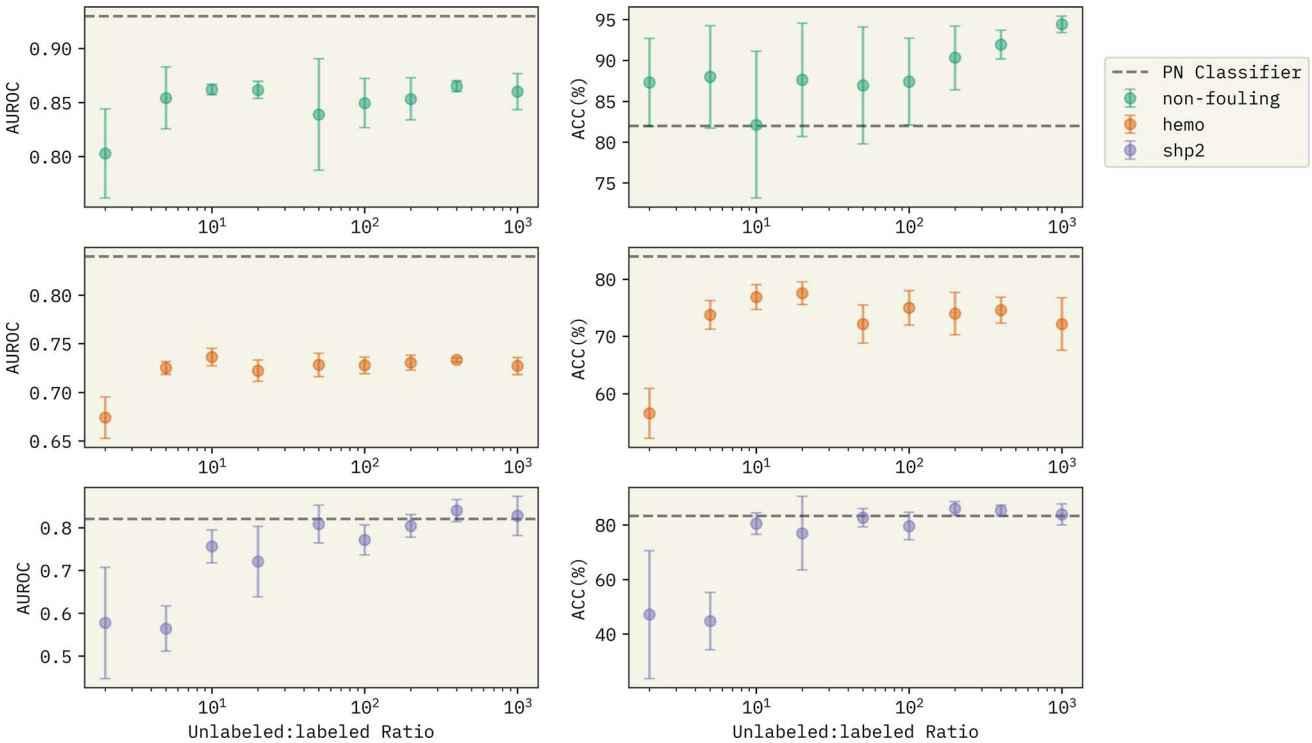

Fig. 3 Effect of generated unlabeled : labeled ratio on the performance of the Reliable Negative Identification strategy for the three prediction tasks. Horizontal dashed lines show the performance of the PN classifier from Table 3 used as a baseline for comparison. At the low ratio regime, the pool of unlabeled data is not big enough to obtain promising candidates as reliable negatives. With larger unlabeled : labeled ratios, the PU model gets to identify a better choice of sequences as reliable negatives, despite the major existing class imbalance in the traning data.

two significant observations; (1) with more unlabeled sequences generated, the trained PU models have a competitive performance with the PN models. In specific, for binding against SHP-2, we observe that the PU model beats the PN classifier in both AUROC and ACC%. 2. Surprisingly, the PU models become more confident in their predictions with the increase of the unlabeled : labeled ratio (compare magnitude of error bars in Fig. 3). This can bring a major advantage in implementing our approach in a generative setting, where we can predict the properties of new peptide sequences without having to worry much about the class imbalance between the positive and the negative examples, which can majorly reduce model performance, if the learning is supervised.

Comparing AUROC and ACC in Tables 2 and 3, we observe that Reliable Negative Identification with mutated positive samples has a relative lower performance compared to the other scenario, where the unlabeled data is generated from a distribution of positive and negative examples. Despite this minor lower performance, using the new unlabeled sequence generation, one can explore the newly unlabeled samples, and make predictions on peptide properties by only having access to the examples from one class (*i.e.* positive). The sequence-based peptide property prediction in this work is limited to four different tasks. However, with the positive data available, this work can be further extended to developing predictive models for inferring other peptide properties.

4 Conclusions

We have showed a semi-supervised learning framework to infer the mapping from peptides' sequence to function for properties such as hemolysis, solubility, non-fouling, and binding against SHP-2. Our positive unlabeled learning method aims at identifying likely negative candidates (reliable negatives) from the generated unlabeled sequences, given random permutations of subsequences within the available positive samples. The reliable negative identification strategy is agnostic with respect to the model architecture used, giving generality. Our method will be most beneficial in biology screening experiments, where most high-throughput screening methods solely focus on identifying the positive example. All PU models showed a comparative predictive ability and robustness across the different prediction tasks, when compared to training with both positive and negative examples.

Moreover, our approach is fundamentally agnostic to negative data, a practical stance considering the rarity of such truly meaningful data in biological datasets, where typically only successful experiments are reported, not the failures. PU learning offers a significant advantage in such imbalanced datasets, where the typical approach of oversampling to address class imbalance can lead to model bias in a supervised learning setting. This learning strategy can provide a robust feasible path towards estimating how amino acids positional substitutions can affect peptide's functional response for unknown

sequences, enhancing the model's ability to generalize to new data, and accelerate the design and discovery of novel therapeutics.

Data and code availability

All data and code used to produce results in this study are publicly available in the following GitHub repository: **https://github.com/ur-whitelab/pu-peptides**.

Conflicts of interest

The authors declare no conflicts of interest.

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.

References

1 J. B. Sperry, C. J. Minteer, J. Tao, R. Johnson, R. Duzguner, M. Hawksworth, *et al.*, Thermal stability assessment of peptide coupling reagents commonly used in pharmaceutical manufacturing, *Org. Process Res. Dev.*, 2018, **22**(9), 1262–1275.

2 L. Ferrazzano, D. Corbisiero, G. Martelli, A. Tolomelli, A. Viola, A. Ricci, *et al.*, Green solvent mixtures for solid-phase peptide synthesis: A dimethylformamide-free highly efficient synthesis of pharmaceutical-grade peptides, *ACS Sustain. Chem. Eng.*, 2019, **7**(15), 12867–12877.

3 M. Muttenthaler, G. F. King, D. J. Adams and P. F. Alewood, Trends in peptide drug discovery, *Nat. Rev. Drug Discovery*, 2021, **20**(4), 309–325.

4 D. J. Drucker, Advances in oral peptide therapeutics, *Nat. Rev. Drug Discovery*, 2020, **19**(4), 277–289.

5 K. Sato, M. P. Hendricks, L. C. Palmer and S. I. Stupp, Peptide supramolecular materials for therapeutics, *Chem. Soc. Rev.*, 2018, **47**(20), 7539–7551.

6 F. Araste, K. Abnous, M. Hashemi, S. M. Taghdisi, M. Ramezani and M. Alibolandi, Peptide-based targeted therapeutics: Focus on cancer treatment, *J. Controlled Release*, 2018, **292**, 141–162.

7 B. P. Lazzaro, M. Zasloff and J. Rolff, Antimicrobial peptides: Application informed by evolution, *Science*, 2020, **368**(6490), eaau5480.

8 A. Nelde, T. Bilich, J. S. Heitmann, Y. Maringer, H. R. Salih, M. Roerden, *et al.*, SARS-CoV-2-derived peptides define heterologous and COVID-19-induced T cell recognition, *Nat. Immunol.*, 2021, **22**(1), 74–85.

9 N. Furukawa and A. S. Popel, Peptides that immunoactivate the tumor microenvironment, *Biochim. Biophys. Acta, Rev. Cancer*, 2021, **1875**(1), 188486.

10 L. Zhang, J. Lu and T. Waigh, Electronics of peptide-and protein-based biomaterials, *Adv. Colloid Interface Sci.*, 2021, **287**, 102319.

11 J. N. Sloand, M. A. Miller and S. H. Medina, Fluorinated peptide biomaterials, *Pept. Sci.*, 2021, **113**(2), e24184.

12 C. Karavasili and D. G. Fatouros, Self-assembling peptides as vectors for local drug delivery and tissue engineering applications, *Adv. Drug Delivery Rev.*, 2021, **174**, 387–405.

13 A. C. L. Lee, J. L. Harris, K. K. Khanna and J. H. Hong, A comprehensive review on current advances in peptide drug development and design, *Int. J. Mol. Sci.*, 2019, **20**(10), 2383.

14 R. Poplin, P. C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, *et al.*, A universal SNP and small-indel variant caller using deep neural networks, *Nat. Biotechnol.*, 2018, **36**(10), 983–987.

15 H. Zeng and D. K. Gifford, Predicting the impact of non-coding variants on DNA methylation, *Nucleic Acids Res.*, 2017, **45**(11), e99, DOI: **10.1093/nar/gkx177**.

16 M. Kulmanov, M. A. Khan and R. Hoehndorf, DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier, *Bioinformatics*, 2018, **34**(4), 660–668.

17 B. Zhang, J. Li and Q. Lü, Prediction of 8-state protein secondary structures by a novel deep learning architecture, *BMC Bioinf.*, 2018, **19**(1), 1–13.

18 J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, *et al.*, De novo design of protein structure and function with RFdiffusion, *Nature*, 2023, **620**(7976), 1089–1100.

19 S. Hashemifar, B. Neyshabur, A. A. Khan and J. Xu, Predicting protein–protein interactions through sequence-based deep learning, *Bioinformatics*, 2018, **34**(17), i802–i810.

20 F. Li, S. Dong, A. Leier, M. Han, X. Guo, J. Xu, *et al.*, Positive-unlabeled learning in bioinformatics and computational biology: a brief review, *Briefings Bioinf.*, 2022, **23**(1), bbab461.

21 K. Sidorczuk, P. Gagat, F. Pietluch, J. Kała, D. Rafacz, L. Bąkała, *et al.*, Benchmarks in antimicrobial peptide prediction are biased due to the selection of negative data, *Briefings Bioinf.*, 2022, **23**(5), bbac343.

22 E. Byvatov and G. Schneider, Support vector machine applications in bioinformatics, *Appl. Bioinf.*, 2003, **2**(2), 67–77.

23 A. L. Boulesteix, S. Janitza, J. Kruppa and I. R. König, Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, 2012, **2**(6), 493–507.

24 T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel and K. Lange, Genome-wide association analysis by lasso penalized logistic regression, *Bioinformatics*, 2009, **25**(6), 714–721.

25 Q. Wang, G. M. Garrity, J. M. Tiedje and J. R. Cole, Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy, *Appl. Environ. Microbiol.*, 2007, **73**(16), 5261–5267.

26 P. Yang, X. L. Li, J. P. Mei, C. K. Kwoh and S. K. Ng, Positive-unlabeled learning for disease gene identification, *Bioinformatics*, 2012, **28**(20), 2640–2647.

27 A. Vasighizaker, A. Sharma and A. Dehzangi, A novel one-class classification approach to accurately predict disease-gene association in acute myeloid leukemia cancer, *PLoS One*, 2019, **14**(12), e0226115.
28 Z. Cheng, S. Zhou and J. Guan, Computationally predicting protein-RNA interactions using only positive and unlabeled examples, *J. Bioinf. Comput. Biol.*, 2015, **13**(03), 1541005.
29 H. Song, B. J. Bremer, E. C. Hinds, G. Raskutti and P. A. Romero, Inferring protein sequence-function relationships with large-scale positive-unlabeled learning, *Cell Syst.*, 2021, **12**(1), 92–101.
30 F. Li, Y. Zhang, A. W. Purcell, G. I. Webb, K. C. Chou, T. Lithgow, *et al.*, Positive-unlabelled learning of glycosylation sites in the human proteome, *BMC Bioinf.*, 2019, **20**(1), 1–17.
31 H. Liu, M. Torii, G. Xu, Z. Hu and J. Goll, Learning from positive and unlabeled documents for retrieval of bacterial protein-protein interaction literature, In *Linking Literature, Information, and Knowledge for Biology*, Springer, 2010, pp. 62–70.
32 C. Kılıç and M. Tan, Positive unlabeled learning for deriving protein interaction networks, *Netw. Model. Anal. Health Inform. Bioinform.*, 2012, **1**(3), 87–102.
33 P. N. Hameed, K. Verspoor, S. Kusljic and S. Halgamuge, Positive-unlabeled learning for inferring drug interactions based on heterogeneous attributes, *BMC Bioinf.*, 2017, **18**(1), 1–15.
34 C. Wang, C. Ding, R. F. Meraz and S. R. Holbrook, PSoL: a positive sample only learning algorithm for finding non-coding RNA genes, *Bioinformatics*, 2006, **22**(21), 2590–2596.
35 X. M. Zhao, Y. Wang, L. Chen and K. Aihara, Gene function prediction using labeled and unlabeled data, *BMC Bioinf.*, 2008, **9**(1), 1–14.
36 N. Bhardwaj, M. Gerstein and H. Lu, Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique, *BMC Bioinf.*, 2010, **11**(1), 1–8.
37 J. F. López, J. A. L. Sotelo, D. Leite and C. Peña-Reyes, Applying one-class learning algorithms to predict phage-bacteria interactions, In *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, IEEE, 2019, pp. 1–6.
38 Y. Xiao and M. R. Segal, Biological sequence classification utilizing positive and unlabeled data, *Bioinformatics*, 2008, **24**(9), 1198–1205.
39 P. Bhadra, J. Yan, J. Li, S. Fong and S. W. Siu, AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest, *Sci. Rep.*, 2018, **8**(1), 1–10.
40 Z. H. Zhou, A brief introduction to weakly supervised learning, *Natl. Sci. Rev.*, 2018, **5**(1), 44–53.
41 J. Zhang, C. Y. Hsieh, Y. Yu, C. Zhang and A. Ratner, A survey on programmatic weak supervision, arXiv, 2022, preprint, arXiv:220205433, DOI: **10.48550/arXiv.2202.05433**.
42 J. Bekker and J. Davis, Learning from positive and unlabeled data: A survey, *Mach. Learn.*, 2020, **109**(4), 719–760.
43 B. Liu, W. S. Lee, P. S. Yu and X. Li, Partially supervised classification of text documents, In *ICML*, Sydney, NSW, 2002, vol. 2, pp. 387–394.
44 X. Li and B. Liu, Learning to classify texts using positive and unlabeled data, In *IJCAI*, 2003, vol. 3. pp. 587–592.
45 W. S. Lee and B. Liu, Learning with positive and unlabeled examples using weighted logistic regression, In *ICML*, 2003, vol. 3, pp. 448–455.
46 B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu, Building text classifiers using positive and unlabeled examples, In *Third IEEE international conference on data mining*, IEEE, 2003, pp. 179–186.
47 H. Wei, Y. Xu and B. Liu, iPiDi-PUL: identifying Piwi-interacting RNA-disease associations based on positive unlabeled learning, *Briefings Bioinf.*, 2021, **22**(3), bbaa058.
48 Y. Chen, Z. Li, X. Wang, J. Feng and X. Hu, Predicting gene function using few positive examples and unlabeled ones, *BMC Genomics*, 2010, **11**(2), 1–9.
49 P. Yang, X. Li, H. N. Chua, C. K. Kwoh and S. K. Ng, Ensemble positive unlabeled learning for disease gene identification, *PLoS One*, 2014, **9**(5), e97079.
50 G. H. Jowkar and E. G. Mansoori, Perceptron ensemble of graph-based positive-unlabeled learning for disease gene identification, *Comput. Biol. Chem.*, 2016, **64**, 263–270.
51 F. Yan, Z. Zhao and L. M. Simon, EmptyNN: A neural network based on positive and unlabeled learning to remove cell-free droplets and recover lost cells in scRNA-seq data, *Patterns*, 2021, **2**(8), 100311.
52 R. Kiryo, G. Niu, M. C. Du Plessis and M. Sugiyama, Positive-unlabeled learning with non-negative risk estimator, *Adv. Neural Inf. Process Syst.*, 2017, **30**, 1674–1684.
53 L. Cerulo, C. Elkan and M. Ceccarelli, Learning gene regulatory networks from only positive and unlabeled data, *BMC Bioinf.*, 2010, **11**(1), 1–16.
54 V. Pejaver, J. Urresti, J. Lugo-Martinez, K. A. Pagel, G. N. Lin, H. J. Nam, *et al.*, Inferring the molecular and phenotypic impact of amino acid variants with MutPred2, *Nat. Commun.*, 2020, **11**(1), 1–13.
55 Z. Li, L. Hu, Z. Tang and C. Zhao, Predicting HIV-1 protease cleavage sites with positive-unlabeled learning, *Front. Genet.*, 2021, **12**, 658078.
56 Y. Chu, H. Zhang and L. Zhang, Function Prediction of Peptide Toxins with Sequence-Based Multi-Tasking PU Learning Method, *Toxins*, 2022, **14**(11), 811.
57 M. Pirtskhalava, A. A. Amstrong, M. Grigolava, M. Chubinidze, E. Alimbarashvili, B. Vishnepolsky, *et al.*, DBAASP v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics, *Nucleic Acids Res.*, 2021, **49**(D1), D288–D297.
58 P. Smialowski, G. Doose, P. Torkler, S. Kaufmann and D. Frishman, PROSO II–a new method for protein solubility prediction, *FEBS J.*, 2012, **279**(12), 2192–2200.
59 H. M. Berman, J. D. Westbrook, M. J. Gabanyi, W. Tao, R. Shah, A. Kouranov, *et al.*, The protein structure initiative structural genomics knowledgebase, *Nucleic Acids Res.*, 2009, **37**(suppl_1), D365–D368.

60 R. Barrett, S. Jiang and A. D. White, Classifying antimicrobial and multifunctional peptides with Bayesian network models, *Pept. Sci.*, 2018, **110**(4), e24079.
61 A. D. White, A. K. Nowinski, W. Huang, A. J. Keefe, F. Sun and S. Jiang, Decoding nonspecific interactions from nature, *Chem. Sci.*, 2012, **3**(12), 3488–3494.
62 A. D. White, W. Huang and S. Jiang, Role of nonspecific interactions in molecular chaperones through model-based bioinformatics, *Biophys. J.*, 2012, **103**(12), 2484–2491.
63 M. Marasco, J. Kirkpatrick, V. Nanna, J. Sikorska and T. Carlomagno, Phosphotyrosine couples peptide binding and SHP2 activation via a dynamic allosteric network, *Comput. Struct. Biotechnol. J.*, 2021, **19**, 2398–2415.
64 R. Barrett and A. D. White, Investigating Active Learning and Meta-Learning for Iterative Peptide Design, *J. Chem. Inf. Model.*, 2020, **61**(1), 95–105.
65 M. Ansari and A. D. White, Serverless prediction of peptide properties with recurrent neural networks, *J. Chem. Inf. Model.*, 2023, **63**(8), 2546–2553.
66 F. Chollet andKeras, *GitHub*, 2015, **https://github.com/fchollet/keras**.
67 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, Software available from tensorflow.org. Available from: **https://www.tensorflow.org/**.
68 I. Sutskever, J. Martens and G. E. Hinton, Generating text with recurrent neural networks, In *ICML*, 2011.
69 M. H. Segler, T. Kogej, C. Tyrchan and M. P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, *ACS Cent. Sci.*, 2018, **4**(1), 120–131.
70 Y. Ye, J. Wang, Y. Xu, Y. Wang, Y. Pan, Q. Song, *et al.*, MATHLA: a robust framework for HLA-peptide binding prediction integrating bidirectional LSTM and multiple head attention mechanism, *BMC Bioinf.*, 2021, **22**(1), 1–12.
71 D. P. Kingma and J. Ba: A method for stochastic optimization, *arXiv*, 2014, preprint, arXiv:14126980, DOI: **10.48550/arXiv.1412.6980**.
72 C. Elkan and K. Noto, Learning classifiers from only positive and unlabeled data, In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 213–220.
73 R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, 2019, vol. 793.
74 A. Smith and C. Elkan, A Bayesian network framework for reject inference, In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 286–295.
75 A. T. Smith and C. Elkan, Making generative classifiers robust to selection bias, In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 657–666.
76 C. Esposito, G. A. Landrum, N. Schneider, N. Stiefl and S. Riniker, GHOST: adjusting the decision threshold to handle imbalanced data in machine learning, *J. Chem. Inf. Model.*, 2021, **61**(6), 2623–2640.
77 S. Jain, M. White and P. Radivojac, Estimating the class prior and posterior from noisy positives and unlabeled data, *Adv. Neural Inf. Process Syst*, 2016, **29**, 2693–2701.
78 H. Yu, J. Han and K. C. Chang, PEBL: Web page classification without negative examples, *IEEE Trans. Knowl. Data Eng.*, 2004, **16**(1), 70–81.
79 T. Peng, W. Zuo and F. He, SVM based adaptive learning method for text classification from positive and unlabeled documents, *Knowl. Inf. Syst.*, 2008, **16**(3), 281–301.
80 G. P. C. Fung, J. X. Yu, H. Lu and P. S. Yu, Text classification without negative examples revisit, *IEEE Trans. Knowl. Data Eng.*, 2005, **18**(1), 6–20.
81 X. L. Li and B. Liu, Learning from positive and unlabeled examples with different data distributions, In *European Conference on Machine Learning*, Springer, 2005, pp. 218–229.
82 D. Ienco, R. G. Pensa and R. Meo, From context to distance: Learning dissimilarity for categorical data clustering, *ACM Trans. Knowl. Discov. Data*, 2012, **6**(1), 1–25.