Complex & Intelligent Systems (2022) 8:477–487
https://doi.org/10.1007/s40747-021-00531-6

ORIGINAL ARTICLE

Investigating deep feedforward neural networks for classification of transposon-derived piRNAs

Alisson Hayasi da Costa[1] · Renato Augusto Corrêa dos Santos[2,3] · Ricardo Cerri[1]

Received: 4 February 2021 / Accepted: 5 September 2021 / Published online: 21 September 2021

Abstract

PIWI-interacting RNAs (piRNAS) form an important class of non-coding RNAs that play a key role in gene expression regulation and genome integrity by silencing transposable elements. However, despite the importance of piRNAs and the large application of deep learning in computational biology, there are few studies of deep learning for piRNAs prediction. Still, current methods focus on using advanced architectures like CNN and variations. This paper presents an investigation on deep feedforward network models for classification of human transposon-derived piRNAs. We developed a lightweight predictor (when compared to other deep learning methods) and we show by practical evidence that simple neural networks can perform as well as better than complex neural networks when using the appropriate hyperparameters. For that, we train, analyze and compare the results of a multilayer perceptron with different hyperparameter choices, such as numbers of hidden layers, activation functions and optimizers, clarifying the advantages and disadvantages of each choice. Our proposed predictor reached a F-score of 0.872, outperforming other state-of-the-art methods for human transposon-derived piRNAs classification. In addition, to better access the generalization of our proposal, we also showed it achieved competitive results when classifying piRNAs of other species.

Keywords piRNAs classification · Neural networks · Deep learning · Hyperparameters optimization

Introduction

PIWI-interacting RNAs (piRNAs) comprise a class of small non-coding RNAs (ncRNAs) of approximately 24–31 nucleotides (although this range may change across different species) [1,2] that are present in a wide range of eukaryotes, from sponges to humans [1,3], where they are expressed predominantly in the gonads [1,4–6].

There are two main known classes of piRNAs: transposon-derived and mRNA-derived. Transposon-derived piRNAs are the most abundant and well-known [3,6]. In *Drosophila melanogaster*, studies showed that transposon-derived piRNAs are generated from genomic heterochromatic regions, in which the repertoire of all TEs is present. In these regions, arrays of defective transposable elements (TEs), termed "piRNA clusters", are transcribed into full transcripts that emerge from either one or both strands, followed by cleavage to produce the piRNAs [3].

The best-known role of piRNAs is silencing of TEs (from which they are generated) in the germline cells similar to other RNA-based mechanisms such as microRNAs (miRNAs) and small interfering RNAs (siRNAs) [1,3–7]. In brief, after maturation, piRNAs bound with PIWI proteins – a germline-specific subclass of the Argonaute family [1] – to form piRNA-induced silencing complexes (piRISC) that can recognize and silence complementary RNA targets at both the transcriptional and post-transcriptional levels [1,3–5].

Although TEs have a significant role in evolution, their mobility in the genome can generate deleterious mutations leading to biological problems, such as infertility

✉ Ricardo Cerri
cerri@ufscar.br

Alisson Hayasi da Costa
ahcosta@estudante.ufscar.br

Renato Augusto Corrêa dos Santos
renatoacsantos@gmail.com

[1] Department of Computer Science, Federal University of São Carlos, São Carlos, Brazil

[2] School of Pharmaceutical Sciences of Ribeirão Preto, University of São Paulo, Ribeirão Preto, Brazil

[3] Institute of Biology, State University of Campinas, Campinas, Brazil

[4,7]. Therefore, silencing of TEs by piRNAs is indispensable to protect the integrity of genomes in germline cells against harmful transposons [4,5,8], especially in animals that undergo obligate sexual reproduction, making this class of small ncRNAs guardians of the genome [4].

The importance of piRNAs brings out the need for efficient identification methods, capable of distinguishing the piRNA sequences from other ncRNAs. However, the development of computational tools for this task is complex [9]. Despite the genomic locations of piRNA clusters being often conserved between related species (such as mouse and humans), piRNAs are extremely diverse and known for lack in sequence conservation [2,10,11]. For example, as presented by Weick and Miska [12], in both *Drosophila melanogaster* and vertebrates, mature piRNAs are slightly longer than miRNAs and siRNAs, with sequences between 24 and 31 nucleotides in length, have a preference for a $5'$ uracil, and possess a $3'$-most sugar that is $2'$-*O*-methylated. On the other hand, *Caenorhabditis elegans* piRNAs are 21 nt long but share the $5'$ and $3'$ features of piRNAs in other organisms. Therefore, due to the large diversity of piRNA sequences, developing computational methods based on common structural-sequence features among species is challenging [13], making the use of deep learning very attractive [14].

Deep learning is now one of the most active fields in machine learning and has successfully performed many complex tasks such as image and speech recognition, machine translation, text and audio generation, etc. In computational biology, deep learning is attractive mainly due to the ability to learn a robust representation directly from raw input data, including bases of DNA sequences or pixel intensities of microscopy images [15]. On the other hand, traditional machine learning algorithms require hard laboratory work to extract relevant features to build reliable models [16].

Since 2016, many methods based on deep learning for solving computational biology problems have been published [15]. For example, U-Net [17] is a famous convolutional neural network (CNN) developed for biomedical image segmentation with great performance on tasks such as retinal vessel, skin cancer and lung nodule segmentation. In the context of genome regulation, Xiong et al. [18] developed a deep learning model that scores how strongly genetic variants affect RNA splicing, a critical step in gene expression whose disruption contributes to many diseases, including cancers and neurological disorders. DNN-PCA-GWO [19], is a method to predict diabetic retinopathy using a CNN-based architecture. Aledhari et al. [20] developed a model based on deep feedforward neural nets to detect people’s emotions in real-time using voice and biofeedback, achieving an accuracy of 85% in determining the emotional scale. More recently, DeepMind published AlphaFold [21], a method that predicts the 3D structure of a protein from its amino acid sequence.

Still, there are few studies on the application of deep learning to identify piRNAs [14,16]. Most methods are based on traditional machine learning techniques, like support vector machines and random forests. Some examples are piRPred [22], IpiRId [23], piRNAPredictor [13] and 2L-piRNA [24]. With the exception of piRNAPredictor, all methods have limitation, which include the need for genomic and epigenomic information, restrict application on specific organisms, and performance problems in different datasets [14].

Only in the last few years deep learning started to be used for piRNAs prediction. The first method developed is called piRNN [14], which is a CNN with 2,774,722 parameters and was created to identify *Human* and *Drosophila melanogaster* piRNAs. More recent methods are 2L-piRNADNN [25] and piRNA(2L)-PseKNC [26], techniques whose objective is to identify piRNAs and their functions, in particular piRNAs of the *Mus musculus* organism.

However, improvements are needed. Both piRNN and piRNA(2L)-PseKNC use CNN, an architecture whose computational cost is high, training is slow and hyperparameter fine-tuning is difficult. Also, a large amount of data is needed to create a robust model and avoid overfitting. Moreover, none of these methods have been tested to predict transposon-derived piRNAs.

Considering the lack of works in the literature regarding applications of deep neural network models for piRNAs prediction, this paper presents an investigation on deep feedforward network (DFN) models for classification of human transposon-derived piRNAs. We developed a lightweight predictor (when compared to CNN-based architectures), and we show by practical evidence that simple neural networks can perform as well as better than complex neural networks when using the appropriate hyperparameters. We train, analyze and compare the results of a multilayer perceptron with different hyperparameters choices, such as number of hidden layers, activation functions and optimizers, clarifying the advantages and disadvantages of each choice.

Using 8 times less parameters than piRNN, our proposal, called piRNet, reached an average *F*-score of 0.872, outperforming piRNN (average *F*-score of 0.834) and traditional machine learning algorithms for human transposon-derived piRNAs. When applied to other datasets of *Human* piRNAs, *Mus musculus* piRNAs and other piRNAs from piRBASE, piRNet achieved competitive results compared with piRNN, 2L-piRNA and piRNA(2L)-PseKNC, showing that although piRNet has been tuned to identify human transposon-derived piRNAs, it is also possible to use it for other organisms with great performance. Additionally, we hope that this analysis encourages the use of multilayer perceptrons in other classes of small ncRNAs.

The remainder of this paper is organized as follows. “Methodology” explains the methodology used for the data acquisition, feature extraction from sequences, prepro-

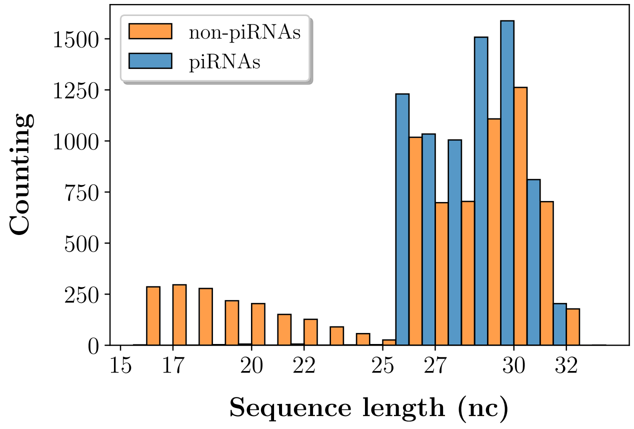


Fig. 1 Length distribution of piRNAs and non-piRNAs sequences (human transposon-derived piRNAs)

cessing algorithms applied to data, network architectures, hyperparameters chosen, the process of training and testing for hyperparameter optimization, and comparison procedure with other methods. "Results and discussion" presents the results obtained by the DFNs proposed, including analysis of the impact of each hyperparameter choice in the final result. Also in "Results and discussion" we show comparisons of our model with other traditional machine learning algorithms and piRNN, and show that our architecture can be used to identify piRNAs from different species with great performance. Finally, "Conclusions" presents the conclusions obtained in this work, and future research directions.

Methodology

Data acquisition and feature extraction process

To search for the best hyperparameter values for our proposal, we constructed variations of a transposon-derived piRNA benchmark dataset. The best neural network in this benchmark was then compared with other methods from the literature, using the datasets from these respective studies. Since piRNN provides the source code of their proposal, we also compared our proposal with piRNN in the transposon-derived piRNA benchmark dataset.

The benchmark dataset used for the experiments has a total of 14,810 samples, where 7405 are transposon-derived piRNA sequences (positive samples) and 7405 are pseudo-piRNA sequences (negative samples). All samples are *Human* ncRNAs and were obtained from the supplementary material provided in the work of Luo et. al. [13], where it is possible to find details about its construction. Figure 1 shows the sequence lengths of positive and negative samples.

To analyze the results of each hyperparameter choice in or proposal, we split the benchmark dataset into two equally disjoint subsets: training subset and test subset. The training subset was used for hyperparameter tuning and the test subset was used for comparison with piRNN. Table 1 presents the proportion of positive and negative samples in each subset.

Table 1 Proportion of positive and negative samples in each subset from the benchmark dataset (human transposon-derived piRNAs)

Dataset	Positive samples	Negative samples	Total
Training subset	3703	3703	7406
Test subset	3702	3702	7404

From the data collected, three different sequence-feature sets were extracted using the Pse-in-One-2.0 tool (local version) [27].

Spectrum profile Also named k-mer, counts the occurrences of k-mer motif frequencies (k-length contiguous strings) in sequences.

Mismatch profile Also counts the occurrences of k-mers, but allows max m ($m \leq k$) inexact matching, which is the penalization of spectrum profile.

Subsequence profile Considers not only the contiguous k-mers but also the non-contiguous k-mers, and a penalty factor w ($0 \leq w \leq 1$) is used to penalize the gap of non-contiguous k-mers [13].

Furthermore, since we have parameters that can be adjusted for obtaining of each feature set (k, m and w, where k is present in all three features), we adopted different values: for k-mers we adopted $k = 1, 2, 3, 4$. For mismatch profile we used (k, m) = (1,0), (2,1), (3,1), (4,1) and for subsequence profile (k,w) = (1,1), (2,1), (3,1), (4,1). Thus, a total of 340 attributes per sequence (sum of 4^1, 4^2, 4^3, 4^4, where k is the exponent) were obtained in each of the feature sets.

To study the behavior of each model in different data sparsity and distribution scenarios we applied two feature scaling algorithms on feature sets: Min–Max normalization (popularly known as normalization), which scales and translates each feature individually such that it is in a range of 0–1; and Z-score normalization, which transforms the original data distribution into a normal distribution with zero mean and unit variance [28].

To verify the generalization of our proposal to different organisms, we also tested the best model resulting from the fine-tuning in the training subset (transposon-derived piRNAs) in three additional datasets composed by: *Human* piRNAs, *Mus musculus* piRNAs and other piRNAs from piRBASE (Generic). These datasets were obtained, respectively, from the studies of Wang et al. (piRNN) [14], Liu et al. (2L-piRNA) [24] and Khan et al. (piRNA(2L)-PseKNC) [26]. So

Table 2 Proportion of positive and negative samples in additional datasets used to assess generalizability of our method

Dataset (Organism)	Positive samples	Negative samples	Total
Human [14]	32,826	32,826	65,652
Mus musculus [24]	1408	1408	2816
Generic [26]	500	500	1000

we also compared our proposal with the methods proposed in each study. Table 2 presents the proportion of positive and negative samples in each additional dataset. Details about the construction of each one can be found in their respective works.

Neural network architecture and hyperparameters

We implemented a DFN with eight different hyperparameter configurations, where each one is characterized by the number of hidden layers, activation function and optimizer used. As for the number of hidden layers, variations of three and five hidden layers were implemented with 340 units per layer (number equivalent to the input array dimension), and a dropout layer with 0.5 dropout ratio between all layers [29]. These numbers were chosen to verify how increasing depth can improve or impair the generalization capacity of the model, together with the activation function and the optimizer used [30–32]. In the output layer, a single neuron with sigmoid was used to predict if a sequence is a transposon-derived piRNA or not [31].

The activation functions selected were logistic function (sigmoid) [30] and rectified linear unit (ReLU) [33]. Sigmoid is very efficient to deal with sparse data, but when used in a neural net with large number of layers, the gradients may become vanishingly small, preventing learning from occurring [30,31]. On the other hand, ReLU is the most used activation function in very deep models, especially due to its sparse activation and better gradient propagation, enabling neural networks to have a large number of hidden layers decreasing vanishing gradient. Nevertheless, the sparse activation along with the large natural sparsity of data can produce an accumulation of large error gradient values, resulting in large updates to the network weights and consequently, a very unstable model [34,35]. Thus, given the pros and cons of each activation function, we analyzed the efficiency of both in the classification of piRNAs.

Moreover, we used the glorot weight initialization [30] in layers with sigmoid activations to minimize the occurrence of vanishing gradient. In layers with ReLU, we used the He et al. weight initialization [34] to promote a faster and efficient convergence, in addition to dealing with vanishing and exploding gradient problems.

Table 3 The eight DFN configurations implemented

Model	Hidden lyers	Activation	Optimizer
DFN1	3	ReLU	Adam
DFN2	3	ReLU	SGD
DFN3	3	Sigmoid	Adam
DFN4	3	Sigmoid	SGD
DFN5	5	ReLU	Adam
DFN6	5	ReLU	SGD
DFN7	5	Sigmoid	Adam
DFN8	5	Sigmoid	SGD

The adopted optimizers were the stochastic gradient descent (SGD) [36], with a learning rate of 0.01 and Nesterov momentum of 0.9, and the adaptive moment estimation (Adam), with default parameters provided in its original paper [37]. Adam is an adaptive learning rate optimization algorithm that utilises both momentum and scaling, combining the benefits of RMSProp and SGD with momentum. The optimizer is designed to be appropriate for non-stationary objectives and problems with very noisy or sparse gradients [38]. When used with multilayer perceptrons and CNNs, achieve great performance and fast convergence on multi-classification tasks in datasets such as MNIST and CIFAR-10 [37]. In contrast, SGD is the most traditional optimizer for neural networks and other optimization-based ML algorithms and was still used in many relevant works such as Xception [39], ResNet [40] and Faster R-CNN [41] with great results. Finally, the cost function used was the log loss [42].

All implementations and experiments were performed using Python 3.8.2 [43], TensorFlow 2.1 [44], and scikit-learn 0.23.1 [28]. Table 3 presents the eight hyperparameter configurations implemented, where the *X* in DFNX stands for the number (identification) of a DFN with a specific combination of hyperparameters.

Training and model comparison

All DFN models were trained with 256 epochs, batch size of 32 and evaluated using the tenfold cross-validation [45] in the training subset. The best performing model in the hyperparameter selection step was tested in our test subset (composed by *Human* transposon-derived piRNAs), and also in the other datasets from Table 2. Figure 2 presents the complete pipeline used for the execution of the experiments

Evaluation metrics

We used five evaluation metrics to assess the performance of the models. In the hyperparameter analysis step, we used: recall (REC) (Eq. 1), precision (PRE) (Eq. 2) and F-score (F)

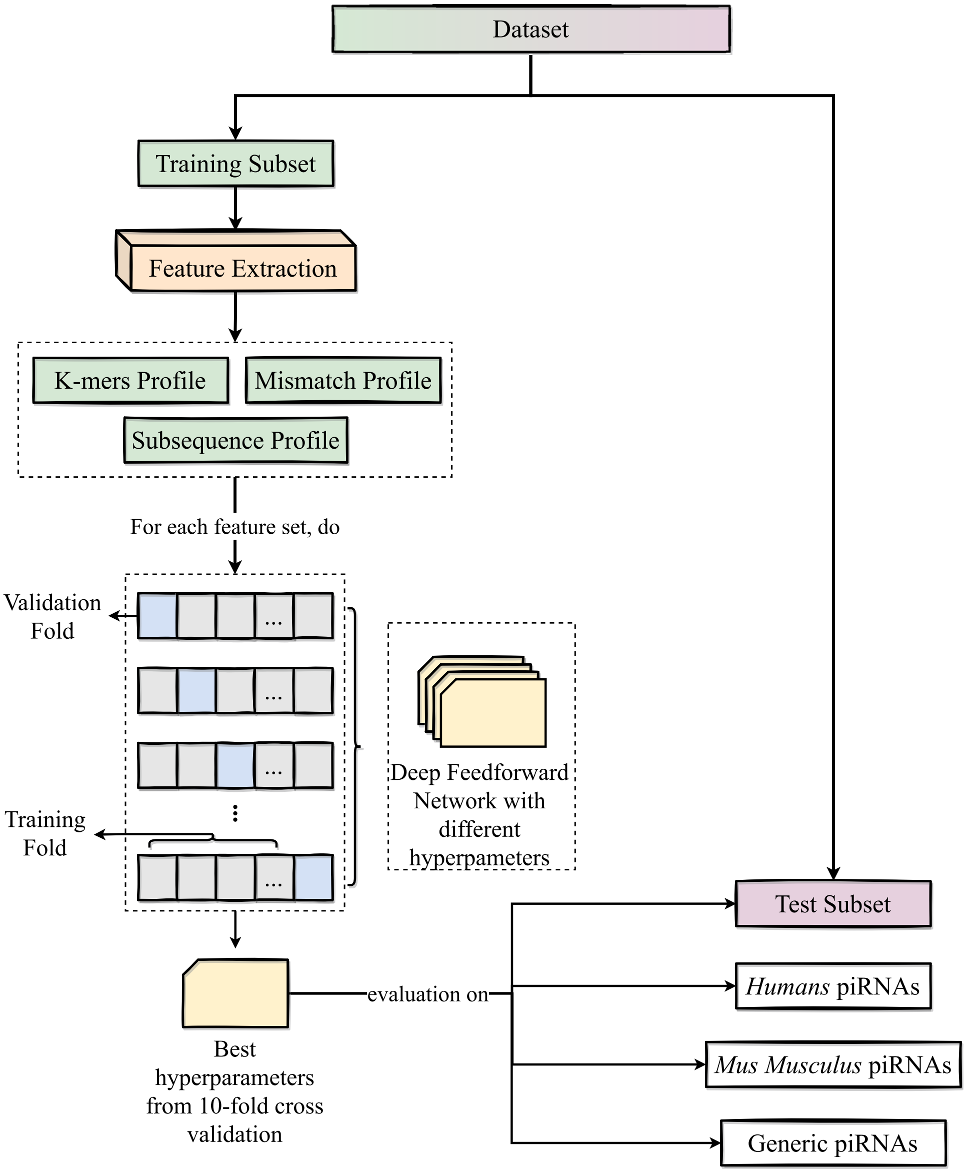


Fig. 2 Pipeline of the execution of the experiments. First, we split the human transposon-derived piRNAs dataset into two subsets: training and test. Then, we extract three feature sets and for each feature set we find the best hyperparameters configuration. The best configuration, including feature, is tested on datasets of different studies

(Eq. 3). In the test and for comparing different methods, we also included accuracy (ACC) (Eq. 4) and specificity (SP) (Eq. 5). In the equations, tp, tn, fp and fn stand, respectively, for the number of true positive and number of true negative samples, and for number of false positive and number of false negative samples.

$$\mathrm{REC} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}, \tag{1}$$

$$\mathrm{PRE} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}, \tag{2}$$

$$F = \frac{2\mathrm{tp}}{2\mathrm{tp}+\mathrm{fp}+\mathrm{fn}}, \tag{3}$$

$$\mathrm{ACC} = \frac{\mathrm{tp}+\mathrm{tn}}{\mathrm{tp}+\mathrm{tn}+\mathrm{fp}+\mathrm{fn}}, \tag{4}$$

$$\mathrm{SP} = \frac{\mathrm{tn}}{\mathrm{tn}+\mathrm{fp}}. \tag{5}$$

Results and discussion

To understand the results, it is necessary to clarify the distribution of the training data (samples of transposon-derived piRNAs), since raw data and preprocessed data (i.e after applying feature scaling) affect the results in different ways.

Since k-mers count the occurrences of k-mer motif frequencies in sequences, the values are limited to a range of 0–1, making the application of Min–Max scaling not as efficient as it could be.

Mismatch and subsequence in their raw state are composed of positive integers, with mismatch ranging from 0 to

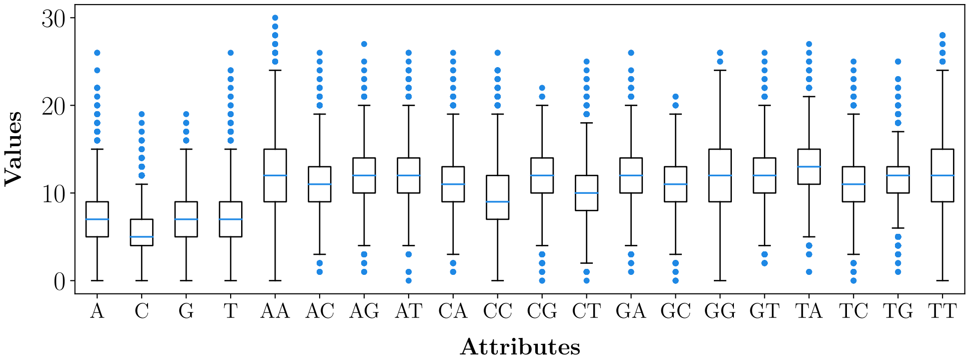


Fig. 3 Sparsity of the first 20 attributes of original Mismatch profile. The x-axis represents the variation of data in each attribute produced by 4^1 (1–4) mismatchs count concatenated with 4^2 (5–20) and y-axis the variation interval of count

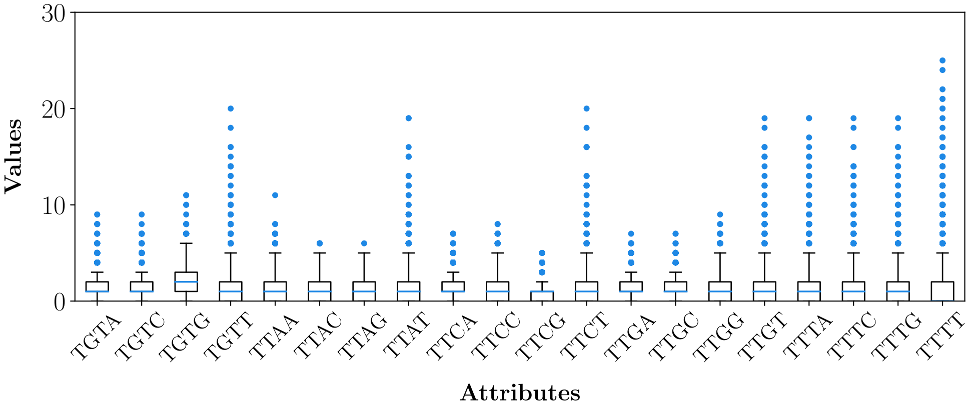


Fig. 4 Sparsity of the last 20 attributes of the original Mismatch profile. The x-axis represents the variation data in the last 20 attributes produced by 4^4 (85–340) mismatches count and y-axis the variation interval

30 and subsequence ranging from 0 to 14950. These very sparse interval not only express the diversity of input samples but also the presence of several outliers. In addition, since the piRNA sequences are small in length, the higher the value of k, the lower the frequency. Consequently, the first 84 attributes (sum of 4^1, 4^2, 4^3) have a slightly better behavior than the other 256 attributes (equivalent to 4^4), as shown in Figs. 3 and 4. Therefore, the use of Min–Max normalization and Z-score normalization (although sensitive to outliers) is extremely useful and necessary to reduce the interval to which the features belong, facilitating the convergence of the neural net.

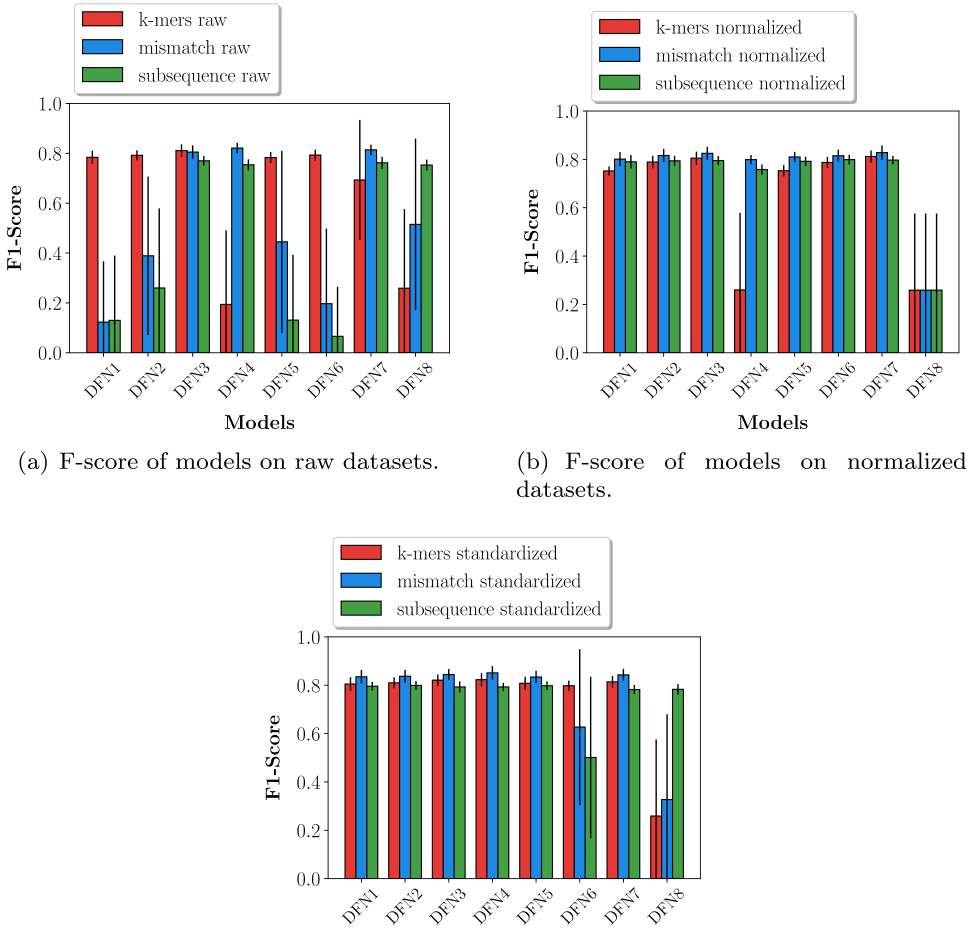
Analysis of hyperparameter optimization

Considering the combination of hyperparameters, features and feature scaling algorithms, a total of 72 results were obtained. To present the results in a illustrative way, Fig. 5 shows the F-scores achieved by the models in the hyperparameter optimization step. Note that the names of the models used for each configuration are defined in Table 3, where the X in DFNX stands for the number (identification) of the hyperparameters combination (deep feedforward network number X).

One important characteristic of deep learning models is the high computational power generated by the large number of hidden layers. However, from the results presented in Fig. 5, we can see that increasing the depth of the neural nets did not improve their performance. Although DFN7 achieved the best result in k-mers and mismatch (both normalized) (with the same being observed for DFN6 in normalized subsequence), the differences between the results obtained compared with the neural nets with three hidden layers are irrelevant. Even models with ReLU activation, which allows neural nets with a large number of hidden layers, did not obtain better results. Thus, it is observed that in the case of piRNAs prediction (and possibly other short ncRNAs), very deep models may not produce good results, even with an activation function adequate for deep models. It is also worth mentioning that the excess of complexity of a model can lead to the occurrence of overfitting [36], besides including an unnecessary high computational cost.

Regarding the activation functions, ReLU has not been effective in producing better results in deeper neural nets, but obtained good results in the preprocessed datasets (even on standardized datasets, whose presence of negative values is significant). ReLU also had no vanishing gradient problem, considering that the increase of layers did not harm the performance. However, its performances on raw mismatch and subsequence were very poor, regardless of the number of layers or optimizer used, as can be seen in Fig. 5a. Note also from Fig. 5c that DFN6 had poor performances in standard mismatch and subsequence features.

Fig. 5 F-score of models on each dataset variation (i.e., raw, normalized and standardized). Each bar indicates the average achieved by the model in the tenfold cross validation on that dataset (indicated by color in legend). Black lines on each bar indicates the standard deviation



(a) F-score of models on raw datasets.

(b) F-score of models on normalized datasets.

(c) F-score of models on standardized datasets

In contrast, some models with sigmoid had the performance impaired by increasing the number of hidden layers, mainly on raw and normalized k-mers. For example, DFN4 and DFN8 on raw k-mers (Fig. 5a) performed very poorly and unstable, while DFN7 presented some instability. The only model with sigmoid and good results was DFN3. Moreover, DFN8 is the worst model, with a poor performance in seven of nine experiments, as shown in Fig. 5. However, although some models with sigmoid have obtained poor results in some datasets, in raw datasets the performance was as good as the best results obtained in preprocessed data.

We can infer that the poor performances obtained by the neural nets with sigmoid must have occurred due to the vanishing gradient, since models with 3 layers (and sigmoid) had good results in general, while models with 5 layers did not perform well. In addition, it is known in the literature that neural networks with a large number of hidden layers together with sigmoid activation tend to have such a problem. At the same time, the properties that make sigmoid incapable of being used with many layers make it very powerful for dealing with the immense sparsity of the data, outliers and any other problem in the raw data, since models with sigmoid achieved good results in raw datasets and the DFN3 achieved good results in all datasets. Table 4 shows cases with the occurrence of vanishing gradient in neural nets with sigmoid.

Table 4 Possible vanishing gradient in neural nets with sigmoid in (raw) k-mers feature set. Note that the increase in the number of hidden layers in the neural net with sigmoid produced an unstable model

Model	Hidden Layers	Activation	F-score
DFN1	3	ReLU	0.784 ± 0.026
DFN3	3	Sigmoid	0.811 ± 0.025
DFN5	5	ReLU	0.783 ± 0.022
DFN7	5	Sigmoid	0.693 ± 0.241

Considering ReLU, it had no vanishing gradient problems, but the sparse activation of ReLU may have caused the exploding gradient problem, since models with ReLU

Table 5 Possible exploding gradient in neural nets with ReLU on (raw) subsequence feature set. Note that regardless of the number of layers, neural nets with ReLU were unable to learn (underfitting)

Model	Hidden layers	Activation	F-score
DFN1	3	ReLU	0.130 ± 0.260
DFN3	3	Sigmoid	0.770 ± 0.019
DFN5	5	ReLU	0.131 ± 0.262
DFN7	5	Sigmoid	0.762 ± 0.024

(mainly with SGD) were unable to learn from training data with raw mismatch and subsequence features. In these data the sparsity is large, which can lead to very large gradient and, consequently, large updates to the neural net weights, producing an unstable model. Table 5 shows cases with the occurrence of exploding gradient in neural nets with ReLU.

The optimization algorithms chosen also have significant impacts on the analyzed models. From Fig. 5, we can see that several models with SGD failed to successfully execute the classification task due to the feature used or poor choice of hyperparameters. For example, DFN4 was unable to learn from both raw k-mers and normalized k-mers, and DFN8 was unable to learn from practically all datasets.

Instead, DFN3 and DFN7, whose number of layers and activation function used correspond to the same ones used in DFN4 and DFN8, reached a great performance in all datasets. Note that Adam was a better choice not only for models with sigmoid, but also models with ReLU. After all, since standardized datasets contain negative values, many neurons tend to be inactive (i.e. only 0 outputs), preventing learning from occurring. Thus, comparing the results obtained by DFN5 and DFN6 in standardized datasets, it is clear that the use of an optimization algorithm such as Adam is much more indicated in this case than the SGD. Adam is much more efficient to deal with noisy data or outliers, sparse gradients and bad hyperparameter choices, as can be seen in our experimental results.

Comparison with other methods considering transposon-derived piRNAs

The best performing model in the hyperparameter optimization step was DFN3, which has three hidden layers, sigmoid activation and Adam optimizer. Figure 6 presents the described neural net architecture. Considering that the best performance of DFN3 was in standardized mismatch (i.e., the mismatch profile rescaled by Z-score normalization), only these features were used in the comparison with other literature methods.

To verify the performance of DFN3 in predicting human transposon-derived piRNAs in comparison with other methods, we used piRNN, since it is the only predictor developed to classify human piRNAs (although it was not specifically proposed for transposon-derived piRNAs). Besides piRNN, we also built a support vector machine (SVM) and random forest (RF) following exactly the same procedure used for our neural network. The best configuration for SVM was $C = 7.0, \gamma = 0.0005$ and radial basis function kernel (where C is the penalty parameter and γ is the kernel coefficient gamma). For RF, we used 500 trees and entropy criterion.

The results obtained by all methods are shown in Table 6. We show the average results after a tenfold cross-validation in the test subset. To compare the computational cost between DFN3 and piRNN, Table 6 also shows the total number of trainable parameters ("Total params" column) for both neural networks.

From the results obtained by all predictors, we can see that our proposed model outperformed piRNN in all evaluation measures, specially Recall. The computational cost of our method is also much lower than piRNN with a total number of parameters approximately 8 times smaller. Regarding the SVM, it achieved excellent performance with better Precision than our method, but Recall and F-score were lower. Considering RF, although it obtained a Recall value close to piRNN, it was the worst performing predictor. Thus, it is clear that despite the success and good performance of CNNs in classification tasks and their wide use in computational biology, DFNs can perform such prediction tasks as well as better than CNNs, achieving good results with less computational resources.

Generalization to different scenarios (non transposon-derived piRNAs)

To access the generalization and learning capabilities of our proposal in predicting other types of piRNAs, we executed our best neural network (DFN3) in the datasets provided by three other studies:

- *Human* piRNAs provided by Wang et al. (piRNN method) [14];
- *Mus musculus* piRNAs, provided by Liu et al. (2L-piRNA method) [24];
- other piRNAs from piRBASE (Generic), provided by Khan et al. (piRNA(2L)-PseKNC method) [26].

For a fair comparison with these methods, we downloaded the provided datasets and performed a cross-validation experiments just like described in the respective works: a tenfold cross-validation was executed for the *Human* piRNAs (piRNN method), while a fivefold cross-validation was executed for the other datasets (2L-piRNA and piRNA(2L)-PseKNC methods).

As shown in Table 7, our predictor achieved competitive results compared with piRNN considering the dataset of

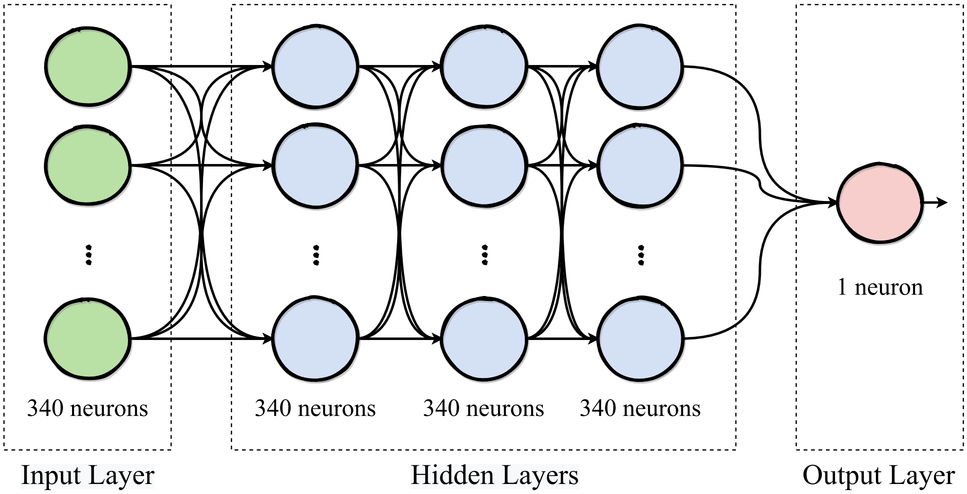


Fig. 6 Architecture with the best performance in the training subset (transposon-derived piRNAs)

Table 6 Comparison between DFN3, piRNN, SVM and RF on test subset and tenfold cross validation

Model	Recall	Precision	*F*-score	Total params
DFN3	0.893 ± 0.019	0.853 ± 0.018	0.872 ± 0.007	348,161
SVM	0.880 ± 0.011	0.860 ± 0.021	0.870 ± 0.009	–
piRNN	0.830 ± 0.025	0.838 ± 0.021	0.834 ± 0.011	2,774,722
RF	0.825 ± 0.015	0.744 ± 0.007	0.782 ± 0.005	–

Table 7 Comparison between DFN3 and piRNN in *Human* piRNAs

Model	Accuracy	Recall	Specificity
DFN3	0.95 ± 0.003	0.96 ± 0.005	0.94 ± 0.005
piRNN	0.95 ± 0.004	0.97 ± 0.012	0.97 ± 0.012

Table 8 Comparison between DFN3 and 2L-piRNA in *Mus musculus* piRNAs

Model	Accuracy	Recall	Specificity
DFN3	0.861	0.865	0.857
2L-piRNA	0.861	0.883	0.839

Table 9 Comparison between DFN3 and piRNA(2L)-PseKNC in Generic piRNAs

Model	Accuracy	Recall	Specificity
DFN3	0.906	0.909	0.903
piRNA(2L)-PseKNC	0.926	0.871	0.852

Human piRNAs. Since our method has 8 times less parameters, we can consider that the DFN3 is an option as good as better than piRNN.

As for *Mus musculus* piRNAs, our method also achieved competitive results with better specificity, lower recall and equal accuracy, as shown Table 8. Considering that DFN3 was adjusted for human transposon-derived piRNAs, competitive results in piRNAs classification of *Mus musculus* are quite satisfactory.

When compared to piRNA(2L)-PseKNC (other piRNAs from piRBASE (Generic)), our method also achieved satisfactory results, with better recall and specificity Table 9.

Conclusions

A deep feedforward network is a basic architecture, but powerful and capable of successfully perform classification tasks in computational biology, including piRNAs prediction. Although very deep architectures have high computational power, they did not necessarily achieve excellent results. Thus, it is very important to correctly fit the number of hidden layers, since a much more complex model than the problem can overfit with an unnecessary high computational cost.

ReLU activation function, although being the state-of-the-art in avoiding the vanishing gradient problem, is not a good choice when data has a large sparsity and many outliers, which is common for piRNA sequences and other ncRNAs (negative samples). Thus, the application of a feature scaling algorithm is essential when using ReLU.

On the other hand, sigmoid activation is very susceptible to the occurrence of vanishing gradient problems. However, it was very efficient to deal with the sparsity and outliers in the datasets used in our study, reaching great results before and after the feature scaling. Therefore, for both piRNAs and other ncRNAs, the use of sigmoid in DFNs may be a good solution.

The correct choice of the optimization algorithm also has a significant impact on the neural network performances, with Adam being a better choice than the SGD for the data in question.

Finally, our proposed model (piRNet) has achieved a great performance in human transposon-derived piRNAs classification, outperforming piRNN using 8 times less parameters, which suggests that simpler multilayer perceptrons can be classifiers as good as better than complex architectures. Furthermore, despite the choice of hyperparameters being driven to human transposon-derived piRNAs, piRNet achieved competitive results compared with several methods in various datasets.

As future works, activation functions like LeakyReLU, ELU and Swish, and optimizers like Nadam and AMSGrad should be tested. Finally, we plan to extend our study and proposed model to other small ncRNAs, such as miRNAs and siRNAs. This certainly can help computational biologists to build models with high classification performances.

Funding This work was supported by the National Council for Scientific and Technological Development (CNPq, from portuguese: Conselho Nacional de Desenvolvimento Científico e Tecnológico).

Data availibility All datasets, materials and codes associated with the current submission are available at https://gitlab.com/biomal/pirnet. The current version of piRNet (available in the provided repository) was trained on all transposon-derived piRNAs used in the study (i.e. complete benchmark dataset). Any updates will be published on the provided link. In the future, we hope to make a web server available for use by scientists.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Iwasaki YW, Siomi MC, Siomi H (2015) Piwi-interacting RNA: its biogenesis and functions. Annu Rev Biochem 84(1):405–433
2. Ha H, Song J, Wang S, Kapusta A, Feschotte C, Chen KC, Xing J (2014) A comprehensive analysis of pirnas from adult human testis and their relationship with genes and mobile elements. BMC Genom 15(1):545
3. Rojas-Ríos P, Simonelig M (2018) pirnas and piwi proteins: regulators of gene expression in development and stem cells. Development 145(17):dev161786
4. Hirakata S, Siomi MC (2016) piRNA biogenesis in the germline: From transcription of piRNA genomic sources to piRNA maturation. Biochim Biophys Acta 1859(1):82–92
5. Siomi MC, Sato K, Pezic D, Aravin AA (2011) Piwi-interacting small rnas: the vanguard of genome defence. Nat Rev Mol Cell Biol 12(4):246–258
6. Han BW, Zamore PD (2014) pirnas. Curr Biol 24(16):R730–R733
7. Slotkin RK, Martienssen R (2007) Transposable elements and the epigenetic regulation of the genome. Nat Rev Genet 8:272–285
8. Lindsay MA, Griffiths-Jones S, Sato K, Siomi MC (2013) Piwi-interacting rnas: biological functions and biogenesis. Essays Biochem 54:39–52
9. Zhang Y, Wang X, Kang L (2011) A k-mer scheme to predict pirnas and characterize locust pirnas. Bioinformatics 27(6):771–776
10. Girard A, Sachidanandam R, Hannon GJ, Carmell MA (2006) A germline-specific class of small rnas binds mammalian piwi proteins. Nature 442(7099):199–202
11. Ozata DM, Gainetdinov I, Zoch A, O'Carroll D, Zamore PD (2018) Piwi-interacting rnas: small rnas with big functions. Nat Rev Genet 20:1
12. Weick EM, Miska EA (2014) pirnas: from biogenesis to function. Development 141(18):3458–3471
13. Luo L, Li D, Zhang W, Tu S, Zhu X, Tian G (2016) Accurate prediction of transposon-derived pirnas by integrating various sequential and physicochemical features. PLoS One 11(4):1–13
14. Wang K, Hoeksema J, Liang C (2018) pirnn: deep learning algorithm for pirna prediction. PeerJ 6:e5429 (**5429[PII]**)
15. Angermueller C, Pärnamaa T, Parts L, Stegle O (2016) Deep learning for computational biology. Mol Syst Biol 12(7):878
16. Jones W, Alasoo K, Fishman D, Parts L (2017) Computational biology: deep learning. Emerg Top Life Sci 1(3):257–274
17. Ronneberger O, Fischer P, and Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, New York, pp 234–241
18. Xiong HY, Alipanahi B, Lee LJ, Bretschneider H, Merico D, Yuen RK, Hua Y, Gueroussov S, Najafabadi HS, Hughes TR et al (2015) The human splicing code reveals new insights into the genetic determinants of disease. Science 347(6218):1254806. https://doi.org/10.1126/science.1254806
19. Alfian G, Syafrudin M, Fitriyani NL, Anshari M, Stasa P, Svub J, Rhee J (2020) Deep neural network for predicting diabetic retinopathy from risk factors. Mathematics 8(9):1620
20. Aledhari M, Razzak R, Parizi RM, Srivastava G (2021) Deep neural networks for detecting real emotions using biofeedback and voice. In: Del Bimbo A, Cucchiara R, Sclaroff S, Farinella GM, Mei T, Bertini M, Escalante HJ, Vezzani R (eds) Pattern recognition. ICPR International Workshops and Challenges. Springer International Publishing, Cham, pp 302–309
21. Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Žídek A, Nelson AW, Bridgland A et al (2020) Improved protein structure prediction using potentials from deep learning. Nature 577(7792):706–710
22. Brayet J, Zehraoui F, Jeanson-Leh L, Israeli D, Tahi F (2014) Towards a pirna prediction using multiple kernel fusion and support vector machine. Bioinformatics 30(17):i364–i370
23. Boucheham A, Sommard V, Zehraoui F, Boualem A, Batouche M, Bendahmane A, Israeli D, Tahi F (2017) Ipirid: integrative approach

for pirna prediction using genomic and epigenomic data. PLoS One 12(6):1–16

24. Liu B, Yang F, Chou KC (2017) 2l-pirna: a two-layer ensemble classifier for identifying piwi-interacting rnas and their function. Mol Therapy Nucl Acids 7:267–277
25. Khan S, Khan M, Iqbal N, Hussain T, Khan SA, Chou KC (2019) A two-level computation model based on deep learning algorithm for identification of pirna and their functions via chou's 5-steps rule. Int J Peptide Res Ther 26:1–15
26. Khan S, Khan M, Iqbal N, Khan SA, Chou KC (2020) Prediction of pirnas and their function based on discriminative intelligent model using hybrid features into chou's pseknc. Chemom Intell Lab Syst 203:104056
27. Liu B, Wu H, Chou KC (2017) Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of dna, rna, and protein sequences. Nat Sci 9(04):67
28. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
29. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
30. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Yee Whye T, Mike T (eds) Proceedings of the thirteenth international conference on artificial intelligence and statistics. PMLR, pp 249–256
31. LeCun YA, Bottou L, Orr GB, Müller KR (2012) Efficient backprop. Neural networks: tricks of the trade. Springer, New York, pp 9–48
32. Haykin SS, Haykin SS, Haykin SS, Elektroingenieur K, Haykin SS (2009) Neural networks and learning machines, vol 3. Pearson, Upper Saddle River
33. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. European conference on computer vision. Springer, New York, pp 818–833
34. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: 2015 IEEE international conference on computer vision (ICCV). pp 1026–1034
35. Ramachandran P, Zoph B, Le Q (2018) Searching for activation functions. arXiv:1710.05941
36. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, New York. http://www.deeplearningbook.org
37. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Bengio Y, LeCun Y (eds) 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings. arXiv:1412.6980
38. Ruder S (2015) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
39. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
40. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
41. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf
42. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press, New York
43. Van Rossum G, Drake FL (2009) Python 3 reference manual. CreateSpace, Scotts Valley
44. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M et al (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/. Software available from tensorflow.org
45. Bengio Y, Grandvalet Y (2004) No unbiased estimator of the variance of k-fold cross-validation. J Mach Learn Res 5(Sep):1089–1105