Briefings in Bioinformatics, 22(6), 2021, 1–12

https://doi.org/10.1093/bib/bbab245
Problem Solving Protocol

Porpoise: a new approach for accurate prediction of RNA pseudouridine sites

Fuyi Li†, Xudong Guo†, Peipei Jin, Jinxiang Chen, Dongxu Xiang, Jiangning Song and Lachlan J.M. Coin

Corresponding authors. Fuyi Li, Department of Microbiology and Immunology, The Peter Doherty Institute for Infection and Immunity, The University of Melbourne, 792 Elizabeth Street, Melbourne, Victoria 3000, Australia. Tel: +61 3 8344 9927; E-mail: fuyi.li@unimelb.edu.au; Jiangning Song, Biomedicine Discovery Institute, Monash University, Melbourne, VIC 3800, Australia. Tel: +61 3 8344 3831; E-mail: Jiangning.song@monash.edu; Lachlan J.M. Coin, Department of Microbiology and Immunology, The Peter Doherty Institute for Infection and Immunity, The University of Melbourne, 792 Elizabeth Street, Melbourne, Victoria 3000, Australia. Tel: +61 3 9902 9304; E-mail: lachlan.coin@unimelb.edu.au
†The first two authors contributed equally to this work

Abstract

Pseudouridine is a ubiquitous RNA modification type present in eukaryotes and prokaryotes, which plays a vital role in various biological processes. Almost all kinds of RNAs are subject to this modification. However, it remains a great challenge to identify pseudouridine sites via experimental approaches, requiring expensive and time-consuming experimental research. Therefore, computational approaches that can be used to perform accurate *in silico* identification of pseudouridine sites from the large amount of RNA sequence data are highly desirable and can aid in the functional elucidation of this critical modification. Here, we propose a new computational approach, termed Porpoise, to accurately identify pseudouridine sites from RNA sequence data. Porpoise builds upon a comprehensive evaluation of 18 frequently used feature encoding schemes based on the selection of four types of features, including binary features, pseudo *k*-tuple composition, nucleotide chemical property and position-specific trinucleotide propensity based on single-strand (PSTNPss). The selected features are fed into the stacked ensemble learning framework to enable the construction of an effective stacked model. Both cross-validation tests on the benchmark dataset and independent tests show that Porpoise achieves superior predictive performance than several state-of-the-art approaches. The application of model interpretation tools demonstrates the importance of PSTNPs for the performance of the trained models. This new method is anticipated to facilitate community-wide efforts to identify putative pseudouridine sites and formulate novel testable biological hypothesis.

Key words: RNA pseudouridine sit; ebioinformatics; sequence analysis; machine learning; stacking ensemble learning

Fuyi Li received his PhD in Bioinformatics from Monash University, Australia. He is currently a Research Officer Bioinformatics in the Department of Microbiology and Immunology, Peter Doherty Institute for Infection and Immunity, the University of Melbourne, Australia. His research interests are bioinformatics, computational biology, machine learning, and data mining.
Xudong Guo received his MEng degree from Ningxia University, China. His research interests are bioinformatics and data mining.
Peipei Jin is a chief technician in the Department of Clinical Laboratory of Ruijin Hospital, affiliated with Shanghai Jiao Tong University School of Medicine, Shanghai, China. Her research interests are thrombosis and hemostasias, especially on coagulation factor in hereditary haemorrhagic disease and platelet membrane proteins with thrombosis.
Jinxiang Chen received his MEng degree from Northwest A&F University, China. His research interests are bioinformatics, big data and deep learning.
Dongxu Xiang is currently a master student in the Faculty of Engineering and Information Technology, The University of Melbourne. His research interests are data mining, machine learning and bioinformatics.
Jiangning Song is an associate professor and group leader in the Monash Biomedicine Discovery Institute, Monash University. He is also a member of the Monash Data Futures Institute, Monash University. His research interests include bioinformatics, computational biomedicine, computer vision, machine learning and pattern recognition.
Lachlan J.M. Coin is a professor and group leader in the Department of Microbiology and Immunology at the University of Melbourne. He is also a member of the Department of Clinical Pathology, University of Melbourne. His research interests are bioinformatics, machine learning, transcriptomics and genomics.
Submitted: 14 April 2021; **Received (in revised form):** 19 May 2021

Introduction

Pseudouridine (Ψ) is an essential and ubiquitous type of RNA modification, which is known as the "fifth RNA nucleotide." Ψ modifications have been extensively found in multiple types of RNAs from both eukaryotes and prokaryotes, including tRNA, mRNA and rRNA [1]. Numerous studies have shown that pseudouridine plays a critical role in the molecular mechanisms, such as the stabilization of RNA structure [2, 3], RNA-protein or RNA-RNA interactions [4], regulation of the entry site binding process [5] and the metabolism of RNAs [6, 7]. In addition, the deficiency of Ψ modification is associated with various diseases. For example, the mutations in Ψ modification are found to be associated with lung cancer and duykeratosis congenita [8]. The key to understanding the mechanisms and other functional roles of pseudouridine is identifying the corresponding pseudouridine sites. However, pseudouridine sites' experimental detection is still challenging, requiring extensive laboratory work and considerable expense [7, 9]. Therefore, computational methods that can accurately predict pseudouridine sites based on RNA sequence information would be beneficial. These computational methods may provide important insights into the functional roles of pseudouridine.

Table 1 provides a summary of the existing approaches for pseudouridine site prediction. From Table 1, we can see that that several computational predictors of pseudouridine sites to complement experimental studies have been developed in recent years. For example, Li *et al.* [10] proposed a support vector machine (SVM)-based model, PPUS, to predict the PUS-specific pseudouridine sites of *Homo sapiens* and *Saccharomyces cerevisiae*. Chen *et al.* [11] proposed iRNA-PseU by combing the chemical properties of nucleotides and pseudo nucleotide composition (PseKNC) encoding scheme with SVM to train a prediction model to identify the pseudouridine sites in *H. sapiens*, *S. cerevisiae* and *Mus musculus*. The dataset collected by Chen *et al.* [11] has been further used in the follow-up studies. He *et al.* [12] applied five different types of encoding schemes to extract sequence features from the RNA segments to develop the PseUI. PseUI is also based on the SVM algorithm combined with a sequential forward feature selection strategy to optimize the model performance. Tahir *et al.* [13] developed iPseU-CNN, a two-layer convolutional neural network model that combines the one-hot encoding scheme to predict pseudouridine sites. Liu *et al.* [14] created XG-PseU by applying the eXtreme Gradient Boosting (XGBoost) and forward feature selection method. Bi *et al.* [15] developed an ensemble learning algorithm, EnsemPseU, by integrating SVM, XGBoost, Naïve Bayes (NB), k-nearest neighbor (KNN) and random forest (RF). Lv *et al.* [16] developed a random forest-based method, RF-PseU, by applying the light gradient boosting machine (lightGBM) and incremental feature selection strategy. Saad *et al.* [17] proposed a convolutional neural network-based method, called MU-PseUDeep, which employs both the sequence and secondary structure features to improve prediction performance. In addition, Song *et al.* proposed PIANO [18] and PSI-MOUSE [19] by incorporating both sequence-based features and genome-derived features for pseudouridine-site prediction and annotation. The genome-derived features can provide additional information and have been demonstrated to be efficient in improving the model performance of RNA modification prediction [20]. However, it usually required well-annotated data such as genome coordinates to derive the relevant genomic features.

Despite developmental progress in computational approaches for predicting pseudouridine sites, several shortcomings remain in these approaches that need to be addressed to develop better methods. The first is that the prediction performance for the existing predictors is still unsatisfactory. For example, the most recent tool, RF-PseU, which performed best among these published tools developed using the same datasets, can only achieve 75% for *H. sapiens* and 77% for *S. cerevisiae* in terms of accuracy on the independent test datasets. In addition, the existing methods only use several commonly used feature encoding schemes to train the model, and they did not systematically analyze and assess the state-of-the-art feature encoding schemes and machine learning algorithms in pseudouridine site prediction. Moreover, it is hard to interpret the current state-of-the-art neural network models to understand which sequence-derived features play critical roles in the predictors and why positive and negative predictions are made.

Therefore, in this study, we develop Porpoise (**P**redict**or** of RNA **p**seud**o**uri**d**ine **s**ite**s**) to identify pseudouridine sites of *H. sapiens*, *S. cerevisiae* and *M. musculus*. Firstly, we have comprehensively evaluated and compared 18 different types of sequence-based feature encoding schemes combined with nine commonly used machine learning algorithms. The performance assessed using cross-validation tests across three species. Then, each classifier's optimal features are selected through a two-step feature selection strategy and a best base classifiers combination for each species is selected through extensive performance comparison. Empirical performance benchmarking tests demonstrate that Porpoise can significantly improve the predictive performance of pseudouridine sites across the three species compared to state-of-the-art methods. Moreover, to interpret the superior performance of Porpoise, we further employ the Shapley Additive exPlanation algorithm to improve the model interpretation and highlight the most important features for Porpoise.

Materials and methods

Overall framework of Porpoise

The design and performance evaluation process of Porpoise is summarized in Figure 1. As can be seen, there exist four major steps, which include dataset cleaning/preparation, feature engineering, stacked model training and optimization and performance evaluation. At the first step, we collect the benchmark training and independent test datasets from the related literature and online databases [11]. At the second step, we comprehensively extract 18 types of RNA sequence-based features and evaluate each type of features coupled with nine commonly used machine learning algorithms. The scatter plot in the "feature engineering" stage in Figure 1 is the t-SNE plot of the pseudouridine sites of three different species based on the PSTNPss features. At the third step, we construct a series of stacked ensemble learning models through different combinations of base classifiers and optimize the model for each of the three species. In the fourth step, we comprehensively evaluate the optimized stacked models by performing cross-validation and independent tests against several existing state-of-the-art approaches. Finally, we implement and make publicly available an online webserver of Porpoise.

Data collection

Current predictors for RNA pseudouridine sites, including iRNA-PseU [11], PseUI [12], iPseU-CNN [13], XG-PseU [14] and

Table 1. Characteristics of the existing approaches for RNA pseudouridine site prediction

Tool[a]	Year	Webserver[b]	Features	Algorithm
PPUS [10]	2015	Yes	Binary	SVM
iRNA-PseU [11]	2016	Decommissioned	NCP, ND, PseKNC	SVM
PseUI [12]	2018	Yes	NAC, DNC, PseDNC, PSNP, PSDP	SVM
iPseU-CNN [13]	2019	No	One-hot	CNN
XG-PseU [14]	2020	Yes	NAC, DNC, TNC, ND, NCP, One-hot	SVM
EnsemPseU [15]	2020	No	Kmer, Binary, ENAC, NCP, ND	Ensemble
RF-PseU [16]	2020	Yes	Binary, ANF, NCP, EIIP, ENAC, CKSNAP	RF
MU-PseUDeep [17]	2020	No	One-hot, SS	CNN
PIANO [18]	2020	Yes	SCP, PSNP, Genome-derived features	SVM
PSI-MOUSE [19]	2020	Yes	NCP, ND, Genome-derived features	SVM

Abbreviations: Binary—binary features; SVM—support vector machine; NCP—nucleotide chemical property; ND—nucleotide density; PseKNC—pseudo nucleotide composition; NAC—nucleic acid composition; DNC—di-nucleotide composition; PseDNC—pseudo nucleic acid composition; PSNP—position-specific nucleotide propensity; PSDP—position-specific dinucleotide propensity; CNN—convolutional neural networks; TNC—tri-nucleotide composition; ENAC—enhanced nucleic acid composition; ANF—accumulated nucleotide frequency; EIIP—electron-ion interaction pseudopotentials of trinucleotide; CKSNAP—composition of *k*-spaced nucleic acid pairs; RF—random forest; SS—RNA secondary structures context; SCP—structural chemical properties

[a]The URL addresses for the listed tools are as follows: PPUS—http://lyh.pkmu.cn/ppus/; iRNA-PseU—http://lin.uestc.edu.cn/server/iRNA-PseU; PseUI—http://zhulab.ahu.edu.cn/PseUI/; XG-PseU—http://www.bioml.cn/; RF-PseU—http://rfpsu.aibiochem.net/; PIANO—http://piano.rnamd.com/; PSI-MOUSE—http://piano.rnamd.com/

[b]Yes—The publication is accompanied with a webserver/tool and it is still working; decommissioned—the webserver/tool is no longer available; no—the publication has no webserver or tool.

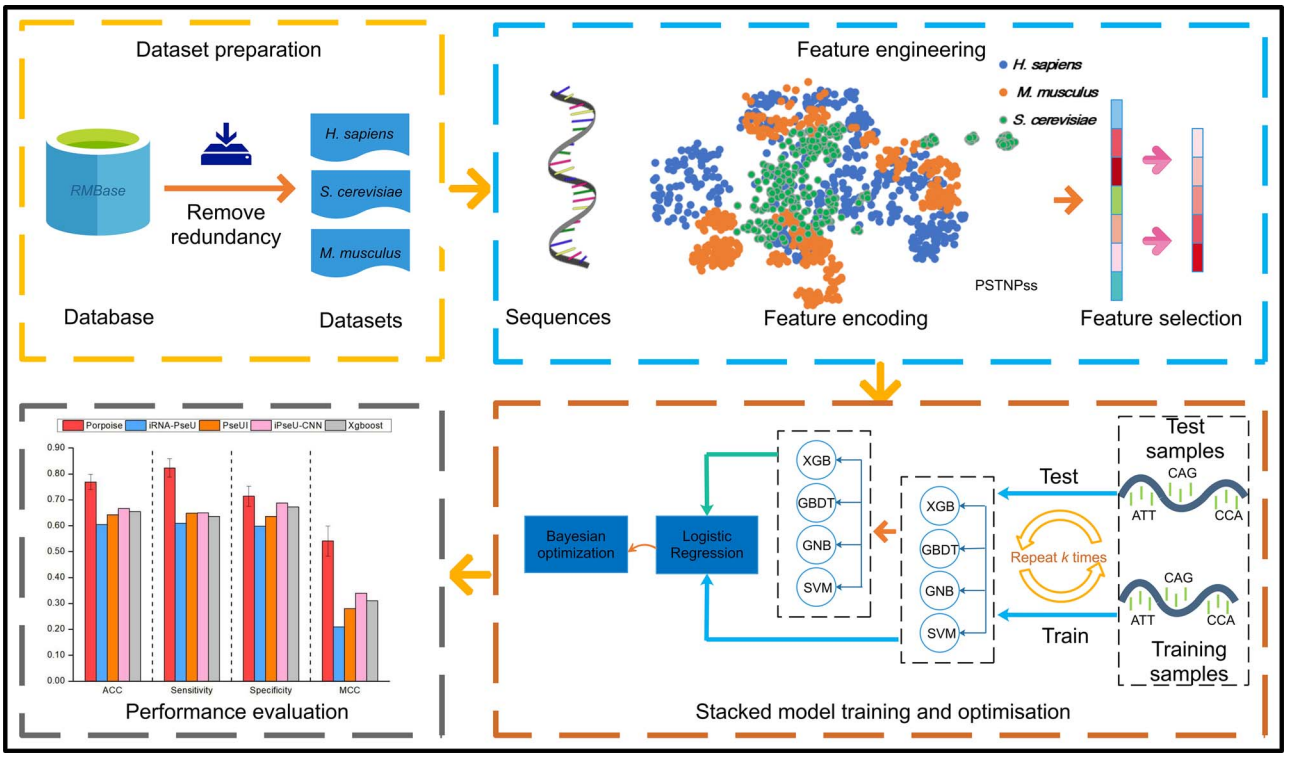

Figure 1. The overall framework of Porpoise. There are four major steps, including dataset preparation, feature engineering, model training and optimization and performance evaluation.

EnsemPseU [15], are all trained and evaluated using the same datasets, previously constructed in the iRNA-PseU work [11]. These datasets were collected from the RMBase database [21], which included three benchmark datasets, named H_990 (*H. sapiens*), S_628 (*S. cerevisiae*) and M_944 (*M. musculus*), for model training; and two other independent test datasets, named H_200 and S_200, for performance validation and comparison between different methods. In this study, we use the same datasets to construct the models of Porpoise and benchmarked the performance of Porpoise against the state-of-the-art approaches. All the samples in both *H. sapiens* datasets (H_990 and H_200) and the *M. musculus* dataset (M_944)

have the RNA sequences with 21 nucleotides with uridine at the center. In comparison, the samples in *S. cerevisiae* (S_628 and S_200) have the RNA sequences with 31 nucleotides with uridine at the center. Both training and independent test datasets are balanced with the ratio of positive and negative samples set as 1:1. For the training datasets, H_990, S_628 and M_944 contained 495, 314 and 472 positive samples and the same number of negative samples. For independent test datasets, both H_200 and S_200 encompassed 100 positive and 100 negative samples, respectively. In addition, to further evaluate the performance of Porpoise and benchmark it against other state-of-the-art methods, we collect another independent test dataset from the m6A-Atlas database [22], a comprehensive public database for multiple RNA modifications. This independent test dataset includes 3,137 *H. sapiens*, 2,702 *M. musculus* and 733 *S. cerevisiae* pseudouridine sites and is named as *Psi_test*.

Feature engineering

In this study, we comprehensively test 18 types of RNA sequence encoding schemes to identify the best feature encoding combinations for training the stacked models, including binary feature, autocorrelation (AC), cross-covariance (CC), trinucleotide-based auto-cross covariance (TACC), accumulated nucleotide frequency (ANF), nucleic acid composition (NAC), pseudo nucleic acid composition (PseDNC), pseudo *k*-tuple composition (PseKNC), the composition of *k*-spaced nucleic acid pairs (CKSNAP), di-nucleotide composition (DNC), tri-nucleotide composition (TNC), electron-ion interaction pseudopotentials of trinucleotide (EIIP), enhanced nucleic acid composition (ENAC), nucleotide chemical property (NCP), electron-ion interaction pseudopotentials of trinucleotide (PseEIIP), position-specific trinucleotide propensity based on double-strand (PSTNPds), position-specific trinucleotide propensity based on single-strand (PSTNPss) and reverse complement Kmer (RCKmer). All these sequence encoding schemes can be calculated using our developed open-source *iLearn* and *iLearnPlus* software packages [23, 24]. Finally, four specific encoding schemes, including binary features, PseKNC, NCP and PSTNPss, are selected according to their predictive performance and used for training the stacked models. These four encoding schemes are described in the following sections.

Binary feature

The binary feature encoding encodes each nucleic acid as a four-dimensional binary vector, e.g. A is encoded as (1,0,0,0), C encoded as (0,1,0,0), G encoded as (0,0,1,0) and U encoded as (0,0,0,1), respectively. The binary feature encoding is applied for the final prediction model for the Porpoise model of *H. sapiens*.

Pseudo k-tuple composition (PseKNC)

The Pseudo *k*-tuple composition (PseKNC) feature is a type of pseudo nucleic acid composition feature that considers both the local and long-range sequence information [23]. PseKNC combines the *k*-tuple nucleotide composition, which is defined as:

$$D = [d_1, d_2, \ldots, d_{4^k}, d_{4^k+1}, \ldots, d_{4^k+\lambda},]^T \quad (1)$$

where

$$d_k = \begin{cases} \frac{f_u}{\sum_{i=1}^{4^k} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (1 \le u \le 4) \\ \frac{w\theta_{u-4^k}}{\sum_{i=1}^{4^k} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (4^k \le u \le 4^k + \lambda) \end{cases} \quad (2)$$

Table 2. Chemical structures and properties of the four nucleotide types

Chemical structure and property	Class	Nucleotides
Ring structure	Purine	A, G
	Pyrimidine	C, U
Functional group	Amino	A, C
	Keto	G, U
Hydrogen bond	Strong	C, G
	Weak	A, U

where f_u $(u = 1, 2, \ldots, 4^k)$ is the frequency of oligonucleotide, which is normalized to $\sum_{i=1}^{4^k} f_i = 1$, w is the factor and θ_j is defined as:

$$\theta_j = \frac{1}{L-j-1}\sum_{i=1}^{L-j-1} \Theta\left(R_iR_{i+1}, R_{i+j}R_{i+j+1}\right), (j = 1, 2, \ldots, \lambda; \lambda < L) \quad (3)$$

$\Theta(R_iR_{i+1}, R_{i+j}R_{i+j+1})$ in Equation (3) is the correlation function, which is defined as follows:

$$\Theta\left(R_iR_{i+1}, R_{j+1}R_{j+2}\right) = \frac{1}{\mu}\sum_{v=1}^{\mu}\left[P_v\left(R_iR_{i+1}\right) - P_v\left(R_{i+j}R_{i+j+1}\right)\right]^2 \quad (4)$$

where μ is the number of physicochemical indices. The PseKNC encoding used six indices, including 'Rise (RNA)', 'Roll (RNA)', 'Shift (RNA)', 'Slide (RNA)', 'Tilt (RNA)' and 'Twist (RNA)' for RNA sequences. $P_v(R_iR_{i+1})$ is the numerical value of the v-th $(v = 1, 2, \ldots, \mu)$ physicochemical index of the dinucleotide R_iR_{i+1} at the position i.

Nucleotide chemical property

It is common knowledge that there are four different types of nucleotides in RNA sequences, i.e. 'A' (adenine), 'G' (guanine), 'C' (cytosine) and 'U' (uracil). Different nucleotides have different chemical structures and chemical properties. According to their chemical properties, four nucleotides can be clustered into three different groups, as shown in Table 2.

For encoding, 'A' is encoded as (1,1,1), 'C' as (0,1,0), 'G' as (1,0,0), and 'U' as (0,0,1), respectively.

Position-specific trinucleotide propensity based on single strand

The Position-specific trinucleotide propensity based on single strand (PSTNPss) encoding scheme employs a statistical rule to encode the RNA sequences. In general, there are 64 (i.e. 4^3) types of trinucleotides (e.g., 'AAA', 'AAC', 'AAG', ..., 'UUU'). Therefore, for a given *L*-bp-long RNA sequence, the trinucleotide position specificity can be formulated as a 64×(*L*-2) matrix:

$$Z = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,L-2} \\ Z_{2,1} & Z_{2,2} & \cdots & Z_{2,L-2} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{64,1} & Z_{64,2} & \cdots & Z_{64,L-2} \end{bmatrix} \quad (5)$$

where $Z_{i,j} = F^+(3mer_i|j) - F^-(3mer_i|j), i = 1, 2, \ldots, 64, j = 1, 2, \ldots, L-2$. $F^+(3mer_i|j)$ and $F^-(3mer_i|j)$ represent the frequencies of the i-th trinucleotide ($3mer_i$) at the j-th position appearing in the positive (S^+) and negative (S^-) datasets, respectively. $3mer_1$ equals 'AAA',

$3mer_2$ equals 'AAC', ..., $3mer_{64}$ equals 'UUU'. Accordingly, an L-bp-long sequence can be represented as the following feature vector S:

$$S = [\varnothing_1, \varnothing_2, \ldots, \varnothing_{L-2}]^T \tag{6}$$

where T denotes the transpose operator and $\varnothing_u$ is defined as follows:

$$\varnothing_u = \begin{cases} Z_{1,u}, \text{when } N_u N_{u+1} N_{u+2} = \text{AAA} \\ Z_{2,u}, \text{when } N_u N_{u+1} N_{u+2} = \text{AAG} \\ \vdots \\ Z_{64,u}, \text{when } N_u N_{u+1} N_{u+2} = \text{UUU} \end{cases} \tag{7}$$

Therefore, in this study, the samples in H_990 and M_944 are encoded as $21-2=19$ PSTNPss features, while the samples in S_628 are represented as $31-2=29$ PSTNPss features.

Stacking ensemble learning framework of Porpoise

Porpoise is a meta-learning framework designed for improving the prediction of RNA pseudouridine sites. Stacking is an effective ensemble learning strategy that integrates the information of various classifiers to enable the construction of a robust prediction model. Such strategy has been successfully applied in a number of recent bioinformatics and computational biology studies [25–29]. The stacking strategy contains two major steps, and the corresponding classifiers at each step are termed as base-classifier and meta-classifier. A set of base-classifiers are applied and built at the first step, and the outputs of base classifiers are then used as the input to train the meta-classifier at the second step.

In this study, we comprehensively assess the performance of nine commonly used machine learning algorithms for RNA pseudouridine site prediction, including Adaptive Boosting (AdaBoost) [30], extremely randomized trees (ERT), extreme gradient boosting (XGBoost) [31], Gradient Boosting Decision Tree (GBDT) [32], logistic regression (LR), k-nearest neighbors (KNN), support vector machine (SVM), random forest (RF) and Gaussian Naive Bayes (GaussianNB). Firstly, we evaluate the predictive performance of all the 18 types of features. For each machine learning algorithm, we train 18 individual classifiers based on each type of features and select the best performing one according to the Matthew's Correlation Coefficient (MCC) as the candidate base classifier. All the classifiers are built and optimized using the scikit-learn package in Python with ten times 10-fold cross-validation tests. Using this strategy, we obtained nine candidate base classifiers in accordance with nine different machine learning algorithms. In order to determine the optimal combination of base-classifiers for the stacked models, we evaluate eight different combinations of base classifiers. Firstly, we rank these nine base classifiers in terms of their classification performance. We set C as the ranked pool of candidate base classifiers, $C = \{c_1, c_2, \ldots c_8, c_9\}$, where c_1 achieved the best MCC. Eight base-classifier combinations are generated by taking base classifiers from C, where Ensemble 1 includes $\{c_1, c_2\}$, Ensemble 2 includes $\{c_1, c_2, c_3\}$, Ensemble 3 includes $\{c_1, c_2, c_3, c_4\}$, Ensemble 4 includes $\{c_1, c_2, c_3, c_4, c_5\}$, Ensemble 5 includes $\{c_1, c_2, c_3, c_4, c_5, c_6\}$, Ensemble 6 includes $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$, Ensemble 7 includes $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ and Ensemble 8 includes $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$, respectively. Logistic regression is used as the meta-classifier to train the stacked model. We comprehensively evaluate these eight base-classifier combinations for each of the three species and select the combination that achieved the best performance as the final model, then optimize the hyperparameters of the final model using the Bayesian optimization algorithm [33]. We implement our stacking strategy using the Stacking Cross-Validation algorithm provided in the '*mlxtend*' package [34] in Python.

Performance evaluation

The predictive performance of Porpoise and other existing methods is evaluated and compared in terms of several commonly used performance measures [35–37], including sensitivity (Sn), specificity (Sp), precision, accuracy (ACC), F1 score, Matthew's Correlation Coefficient (MCC) and area under the receiver-operating curves (AUC). Sn, Sp, Precision, ACC, F1 and MCC are, respectively, defined as:

$$\begin{aligned} Sn &= \frac{TP}{TP+FN} \\ Sp &= \frac{TN}{TN+FP} \\ Precision &= \frac{TP}{TP+FP} \\ Acc &= \frac{TP+TN}{TP+TN+FP+FN} \\ F1 &= \frac{2\times Precision\times Sn}{Precision+Sn} \\ MCC &= \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \end{aligned} \tag{8}$$

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives, respectively.

Results and discussion

Performance evaluation of different feature encoding schemes

In this section, we comprehensively investigated and evaluated the performance of variant models trained using 18 different feature encodings in combination with nine commonly used machine learning algorithms (as detailed in Stacking ensemble learning framework of Porpoise section) by performing ten times 10-fold cross-validation tests. We built the corresponding 18 classifiers with the default parameters based on different feature encoding schemes for each machine learning algorithm and evaluate the model performance based on the average performance metrics of ten times 10-fold cross-validation tests. The performance evaluation results are provided in Supplementary Tables S1–S9. As mentioned in Stacking ensemble learning framework of Porpoise section, we selected the best-performing one in terms of MCC as the candidate base classifier for each algorithm. As a result, nine candidate base classifiers for each species are summarized in Supplementary Table S10.

Overall, we do not observe any single type of features that was consistently outperforming other features for any species, and the predictive performance of the models trained using single types of features was generally limited. Despite this, we make several important observations: First, two gradient boosting algorithms, XGBoost and GBDT, achieved the best performance among the nine compared machine learning algorithms. In particular, XGBoost achieved the best performance on H_990 and M_944. In contrast, GBDT achieved the best performance on S_628. Here, it should be noted that we ranked the classifiers according to the MCC values, as we want to construct the best classifier to achieve a better balance between sensitivity and specificity. Second, we find that the PSTNPss features contributed more to the model performance as opposed to other types of features, particularly for these two best-performing algorithms XGBoost and GBDT. For example, the XGBoost model trained using the PSTNPss features outperformed across all

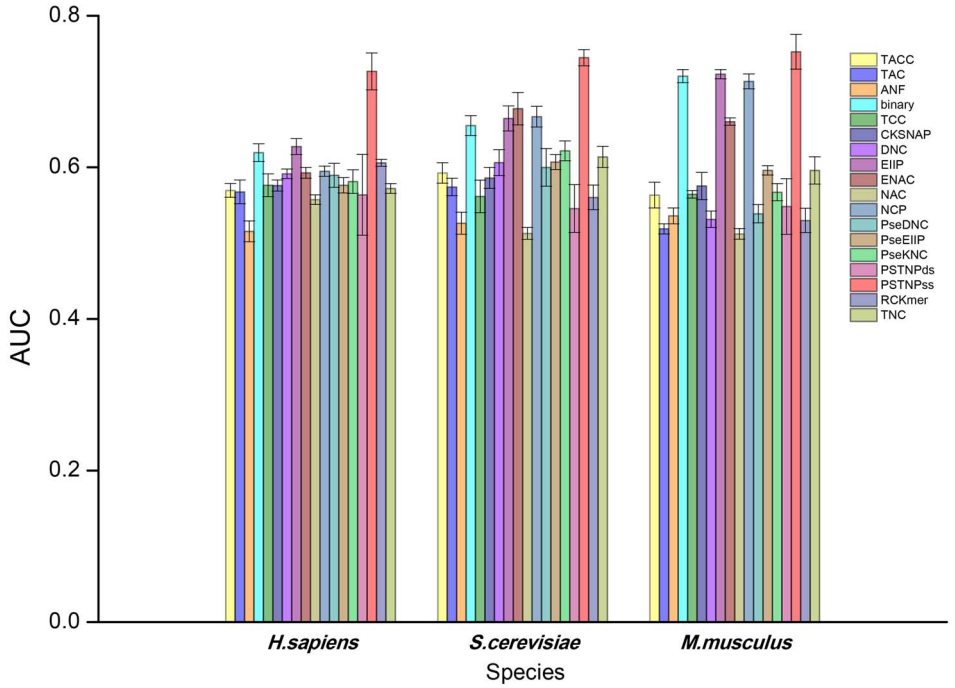

Figure 2. Performance comparison of variant XGBoost models trained using 18 different types of features in terms of AUC value.

three species compared with the XGBoost models trained using 18 different types of features in accuracy, sensitivity, MCC and AUC (Supplementary Table S3). Figure 2 provides a comparison of the AUC values of 18 XGBoost models trained using these different types of features. XGBoost models trained using the PSTNPss features perform the best amongst 18 compared models across all three species. In addition, the GBDT model trained with PSTNPss features also achieved the best performance on H_990 and M_944 datasets in terms of all performance metrics, with the only exception of the specificity on H_990, where the RCKmer-based model performed the best (Supplementary Table S4). Third, the prediction performance of all base classifiers was not satisfying. This indicates that the single model trained with a single type of features could not effectively address this prediction task. As such, we were highly motivated to develop an effective ensemble learning strategy to enhance the model performance.

The optimal base-classifier combinations

Porpoise employed the stacking strategy to build the ensemble learning model. In order to identify the optimal base classifiers for the stacked model from the nine candidate base classifiers, we evaluated eight different combinations of the base classifiers as detailed in Stacking ensemble learning framework of Porpoise section. The performance evaluation results of three species are summarized in Supplementary Tables S11–S13 and Figure 3. As can be seen, it is clear that, for H_990, Ensemble3 achieved the best performance in terms of almost all performance metrics, with the only exception of Specificity. For S_628 and M_944, Ensemble1 secured the best-prediction performance with respect to almost all performance evaluation metrics. This means that the stacking model could achieve a competitive performance based on a few base classifiers on all these three datasets. Specifically, Ensemble3 of H_990 stacked an XGBoost and a GBDT classifier trained with PSTNPss features, a GaNB model trained with PseKNC and an SVM model trained with binary features. The optimal combination for S_628 ensembled a GBDT model and an XGBoost model trained using the PSTNPss features. As a comparison, Ensemble1 of M_944 stacked an XGBoost classifier trained using the PSTNPss features and an AdaBoost classifier trained with the NCP features.

Based on the optimal combinations of base-classifiers, we next conducted feature selection and hyperparameter optimization for each base classifier. Although each base classifier is only trained using one single type of features, some of the features are of high dimension and particularly sparse, which might have a negative impact on model training. Therefore, it might be helpful to further apply feature selection to identify more relevant and contributing features for model training. In this study, we employed a two-step feature selection strategy that combines the mRMR (minimum redundancy maximum relevance) [38] and incremental feature selection (IFS) algorithms [39, 40] to identify the optimal feature subsets. At the first step, we ranked all the initial features according to the information gain. We set F as the whole set of all ranked features, $F = \{f_1, f_2, \ldots f_{n-1}, f_n\}$, where n is the number of features. Then, the IFS was applied to the training dataset by performing ten times 10-fold cross-validation tests to evaluate the relative importance of all the features included in F. At each iteration, IFS constructed n features subsets by adding one feature from F to the candidate feature subset F_c. The i-th feature subset can be defined as $F_{ci} = \{f_1, f_2, \ldots f_i\}$. As a result, the feature subset that achieved the highest MCC value is regarded as the optimal feature subset.

Following feature selection, we employed the Bayesian optimization algorithm [33] to optimize the hyper-parameters of the stacked models. The best-performing parameter values and the corresponding numbers of features after feature selection on the training dataset are provided in Supplementary Table S14. In addition, IFS feature selection results are shown in Supplementary Figures S1–S3. These results demonstrate that, for H_990, all 19 PSTNPss features were selected as optimal features for

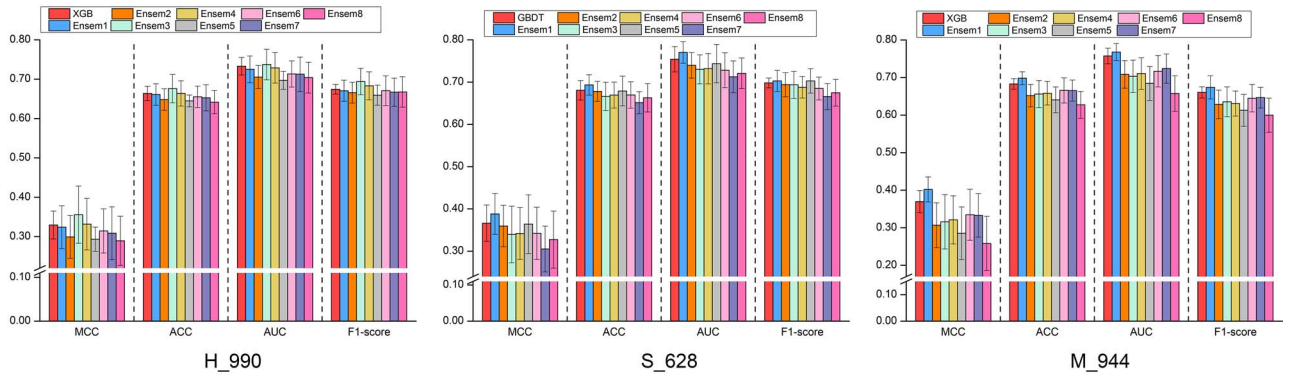

Figure 3. Performance comparison of different combinations of base classifiers.

XGBoost and GBDT classifiers; for S_628, 22 out of 29 PSTNPss features were selected as the optimal features for GBDT and XGBoost classifiers, respectively; while for M_944, 15 out of 19 PSTNPss features were chosen as the optimal features for the XGBoost classifier. Furthermore, after feature selection, the total number of features fed into the stacked model was relatively small. There were only 109 features for H_990, 44 features for S_628 and 50 features for M_944, respectively. To some extent, feature selection helps reduce the stacked models' complexity, alleviate the resources required for model training and improve the model interpretability.

The base classifiers of Porpoise were all trained using a single type of features. For example, on H_990, XGBoost only applied the PSTNPss features for model training, feature selection and parameter optimization. In comparison, a variety of diverse features could provide more useful information for improving the performance of machine learning algorithms. Therefore, a question will naturally arise—Will the models perform better if an optimal subset of features could be selected from a more comprehensive and larger feature set? To address this question, we subsequently compared and evaluated two contrasting feature selection strategies. We named the feature selection strategy used in Porpoise as Strategy 1, while Strategy 2 is also used for making a comparison. In Strategy 2, we first ranked all the 18 types of features using the mRMR algorithm. As the dimensionality of all 18 types of features is very high, which requires intensive computing resources to execute the IFS strategy. To deal with this, we used a percentage-based IFS strategy to select the optimal features and reduce the computing time. This percentage-based IFS strategy was conducted ten times in the following manner: in the first time, the top 10% of the ranked features were used to train and evaluate the model performance; in the second time, the top 20% of the ranked features were employed to train and evaluate the models, so on and as so forth; each time 10% more features were added, and the model performance was accordingly evaluated; in the 10-th time, all the features are fed into the classifier. Finally, the feature subset that attained the highest MCC value was regarded as the optimal feature subset. Both Strategy 1 and Strategy 2 were repeated ten times to calculate the average performance.

Moreover, we conducted a student's *t*-test to assess the statistical significance of the MCC values between Strategy 1 and Strategy 2. Figure 4A plots the distributions of the MCC scores for Strategy 1 and Strategy 2 evaluated on the training datasets, while Figure 4B shows the distributions of the MCC scores on the independent test datasets. The results indicate that Strategy 1 indeed significantly outperformed Strategy 2 in terms of MCC on both benchmark training and independent test datasets, with an only exception on the independent test dataset of *S. cerevisiae* (S_200), for which the *P*-value is not significant. Nevertheless, it is apparent that the average MCC values of Strategy 1 are higher than those of Strategy 2; on the other hand, the standard deviation of Strategy 1 is lower than that of Strategy 2. This suggests that Strategy 1 provided better performance than Strategy 2 and achieved a more stable performance than Strategy 2. Therefore, we adopt Strategy 1 to optimize Porpoise models in the following studies.

Performance comparison between Porpoise and state-of-the-art methods

In this section, we employed the same benchmark training and independent test datasets previously used by several state-of-the-art methods to rigorously evaluate and compare the predictive performance of Porpoise against the other methods. Tables 3 and 4, respectively, summarize the performance comparison results of Porpoise and state-of-the-art predictors including iRNA-PseU [11], PseUI [12], iPseU-CNN [13], XG-PseU [14], EnsemPseU [15], RF-PseU [16] and MU-PseUDeep [17] on the same benchmark training and independent test datasets. As can be seen from Table 3, Porpoise achieved the overall best performance compared with the other seven tools across all three species *H. sapiens*, *S. cerevisiae* and *M. musculus* on the same training datasets in terms of accuracy and MCC. On H_990, Porpoise achieved the best performance in accuracy, sensitivity and MCC, while MU-PseUDeep achieved the best specificity. Porpoise's accuracy, sensitivity and MCC were 5.93, 18.21 and 6.05% higher than the second-best method MU-PseUDeep, respectively. On S_628, Porpoise secures the best performance in terms of all the four measures i.e. accuracy, sensitivity, specificity and MCC, which were 4.89, 3.01, 2.37, and 8.78%, respectively, higher than the second-best performance value. The performance improvement of Porpoise on M_944 was relatively small, on which it achieved the best accuracy, specificity and MCC with a marginal increase of 1.75, 4.67 and 1.85%, respectively.

The superior performance of Porpoise on the benchmark training datasets suggests that the stacking ensemble learning strategy employed can enhance the effectiveness of model training. However, to examine if models trained using this strategy is subjected to overfitting, we further performed an independent test to evaluate and validate the trained models using the independent test datasets. Using the same independent

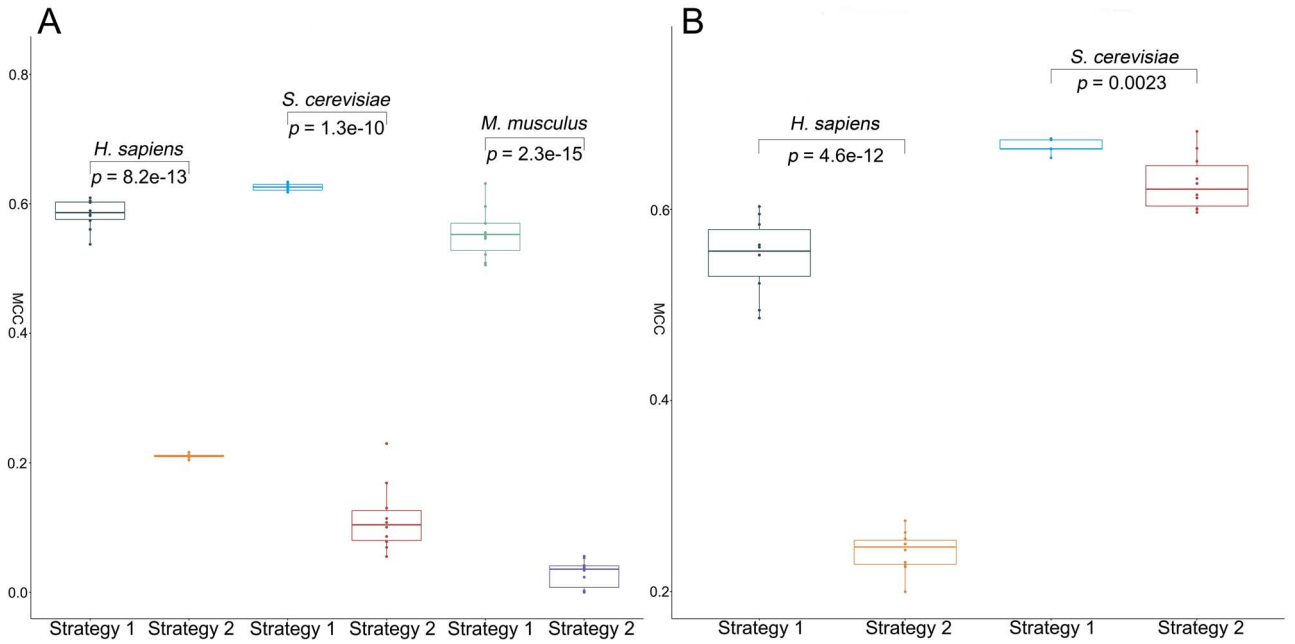

Figure 4. Boxplots showing that Strategy 1 significantly outperformed Strategy 2 in terms of MCC score on both (**A**) benchmark training and (**B**) independent test datasets.

Table 3. Performance comparison of Porpoise and seven state-of-the-art methods on the same benchmark training datasets

Species (dataset)	Method	ACC (%)	Sn (%)	Sp (%)	MCC (%)
***H. sapiens* (H_990)**	Porpoise (10-fold)	**78.53**	**89.11**	67.94	**58.45**
	iRNA-PseU	60.40	61.01	59.80	21.00
	PseUI	64.24	64.85	63.64	28.00
	iPseU-CNN	66.68	65.00	68.78	34.00
	XG-PseU	65.44	63.64	67.24	31.00
	EnsemPseU	66.28	63.46	69.09	33.00
	RF-PseU (10-fold)	64.30	66.10	62.60	29.00
	RF-PseU (LOO)	64.00	65.90	62.60	29.00
	MU-PseUDeep	72.60	70.90	**81.00**	52.40
***S. cerevisiae* (S_628)**	Porpoise (10-fold)	**81.69**	**81.21**	**82.17**	**63.38**
	iRNA-PseU	64.49	64.65	64.33	29.00
	PseUI	65.13	62.74	67.52	30.00
	iPseU-CNN	68.15	66.36	70.45	37.00
	XG-PseU	68.15	66.84	69.45	37.00
	EnsemPseU	74.16	73.88	74.45	49.00
	RF-PseU (10-fold)	74.80	77.20	72.40	49.00
	RF-PseU (LOO)	75.80	78.20	73.40	52.00
	MU-PseUDeep	76.80	74.20	79.80	54.60
***M. musculus* (M_944)**	Porpoise (10-fold)	**77.75**	77.83	**77.67**	**55.55**
	iRNA-PseU	69.07	73.31	64.83	38.00
	PseUI	70.44	74.58	66.31	41.00
	iPseU-CNN	71.81	74.79	69.11	44.00
	XG-PseU	72.03	76.48	67.57	45.00
	EnsemPseU	73.85	75.43	72.25	48.00
	RF-PseU (10-fold)	74.80	73.10	76.50	50.00
	RF-PseU (LOO)	74.50	72.70	75.20	48.00
	MU-PseUDeep	76.00	**80.00**	73.00	53.70

Notes: 10-fold—10-fold cross-validation; LOO—leave-one-out cross-validation. Bold values indicate the best performance in terms of the corresponding measure.

test datasets, we evaluated and compared the predictive performance of Porpoise with that of other existing methods. The results are summarized in Table 4. As can be seen, the Porpoise models also achieved a very competitive performance on both H_200 and S_200. In particular, on H_200, Porpoise achieved the best accuracy, sensitivity and MCC, which were 2.35, 4.3 and 7.13% higher than the second-best performance values, respectively. On S_200, Porpoise achieved the best performance in terms of three performance measures, with an increase of 6.5, 3.0 and 13.27% in accuracy, sensitivity and MCC, respectively.

Table 4. Performance comparison of Porpoise and six state-of-the-art methods evaluate on the same independent test datasets

Species (dataset)	Method	ACC (%)	Sn (%)	Sp (%)	MCC (%)
***H. sapiens* (H_200)**	Porpoise	**77.35**	**82.30**	72.40	**55.13**
	XG-PseU	67.50	68.00	67.00	35.00
	iPseU-CNN	69.00	77.72	60.81	40.00
	PseUI	65.50	63.00	68.00	31.00
	RNA-PseU	61.50	58.00	65.00	23.00
	EnsemPseU	69.50	73.00	66.00	39.00
	RF-PseU (10-fold)	75.00	78.00	72.00	50.00
	RF-PseU (LOO)	74.00	74.00	**74.00**	48.00
***S. cerevisiae* (S_200)**	Porpoise	**83.50**	**88.00**	**79.00**	**67.27**
	XG-PseU	71.00	75.00	67.00	42.14
	iPseU-CNN	73.50	68.76	77.82	47.00
	PseUI	68.50	65.00	72.00	37.00
	iRNA-PseU	60.00	63.00	57.00	20.00
	EnsemPseU	75.00	85.00	65.00	51.00
	RF-PseU (10-fold)	77.00	75.00	79.00	54.00
	RF-PseU (LOO)	74.50	70.00	79.00	49.00

Notes: 10-fold—10-fold cross-validation; LOO—leave-one-out cross-validation. Bold values indicate the best performance in terms of the corresponding measure.

In addition, the performance on the independent test was also very close to that on the cross-validation test on the benchmark training datasets: for example, Porpoise achieved an accuracy of 77.35% and MCC of 55.13% on H_200, which was very close to the performance on the training dataset, on which Porpoise attained an accuracy of 78.53% and MCC of 58.45%, respectively. Besides, Porpoise achieved an accuracy of 83.5% and MCC of 67.27% on S_200, which was slightly higher than that on the training dataset (81.69 and 63.38%, respectively). Taken together, these results demonstrated that Porpoise achieved a competitive performance and compared favorably with the existing state-of-the-art methods, and there was no overfitting for the Porpoise models.

Model interpretation based on the kernel SHAP

In this section, we employed the Shapley Additive explanation (SHAP) algorithm [41] to help interpret the Porpoise models. Although we applied the mRMR feature selection algorithm to rank the features and selected the optimal features using an incremental feature selection. However, the directionality of these optimal features was still unknown. The directionality here means when a feature takes a higher value, we do not know whether this feature indeed corresponds to larger or smaller feature importance for the performance of the classifier. To address this, we used the SHAP algorithm, which can assign each feature an importance value for the model prediction, thereby providing an interpretation of the stacked model of Porpoise [28, 42]. Specifically, in the current work, we used the kernel SHAP instead of the SHAP algorithm to interpret the stacked model of Porpoise, as the stacked model is too complicated to interpret using the latter, which is designed for explaining the single-tree ensemble models. The kernel SHAP applies a specially weighted local linear regression to estimate the SHAP values for any model. Figure 5 shows the top 20 features ranked based on the SHAP value for the three different species, where each row represents the SHAP value distributions of a feature. The color of data points in the figure indicates the corresponding feature value, and redder points mean higher feature values, while bluer points indicate lower feature values. The color (high or low) and the SHAP (positive or negative) value indicate the features' directionality.

For each panel in Figure 5, the top 20 ranked features sorted according to the sum of SHAP values incorporating all training samples. The points with different colors illustrate the distribution of each feature's impact on the output of the Porpoise model, where a SHAP value of >0 indicates that the prediction favors the positive class (i.e. pseudouridine site). In contrast, a SHAP value of <0 means that the prediction tends to be of the negative class (i.e. non-pseudouridine site). The results revealed that the PSTNPss features play a predominant role for Porpoise, which was reflected by the presence of all the top 20 important features for *H. sapiens* and *S. cerevisiae* models, and the top 15 critical features for the *M. musculus* model were PSTNPss features. Altogether, these results once again confirm and highlight the importance and contribution of the PSTNPss features to the performance of Porpoise.

Performance evaluation on the *Psi_test* dataset

In this section, we further evaluated the predictive performance of Porpoise and compared it with other state-of-the-art methods on the *Psi_test* dataset collected from the m6A-Atlas database. We submitted the FASTA sequences in the *Psi_test* dataset to the webservers of the existing approaches and accordingly calculated the percentage of corrected predicted samples. For Porpoise, we used the model trained on the training datasets to perform the prediction. The performance comparison results are provided in Table 5. As can be seen, Porpoise achieved the best predictive performance and secured 80.5, 83 and 88.1% in *H. sapiens*, *M. musculus* and *S. cerevisiae*, which was 2.8, 5.2, and 9.9% higher than the second-best performed values. In addition, we applied the webserver of Porpoise to make a time-usage comparison with other existing webservers and provided the comparison results in Supplementary Table S15. The results shown that Porpoise was the second-ranked time-saving webserver to generate the prediction results among the four compared webservers. PseUI was the most time-saving webserver for *H. sapiens* and *M. musculus*, while PseUI required considerable time for *S. cerevisiae*, 30 times longer than that of XG-PseU, which was ranked first for *S. cerevisiae*. It should be noted that this time-usage analysis was performed only for a rough time comparison purpose between different webservers, which may also be influenced by the status of the internet.

Table 5. Performance comparison of Porpoise and six state-of-the-art methods evaluate on the *Psi_test* dataset

Species	Methods			
	Porpoise	RF-PseU	PseUI	XG-PseU
H. sapiens	80.49% (2525/3137)	77.69% (2437/3137)	68.70% (2155/3137)	67.99% (2133/3137)
M. musculus	82.98% (2242/2702)	72.46% (1958/2702)	77.17% (2085/2702)	77.79% (2102/2702)
S. cerevisiae	88.27% (647/733)	78.17% (573/733)	74.76% (548/733)	76.67% (562/733)

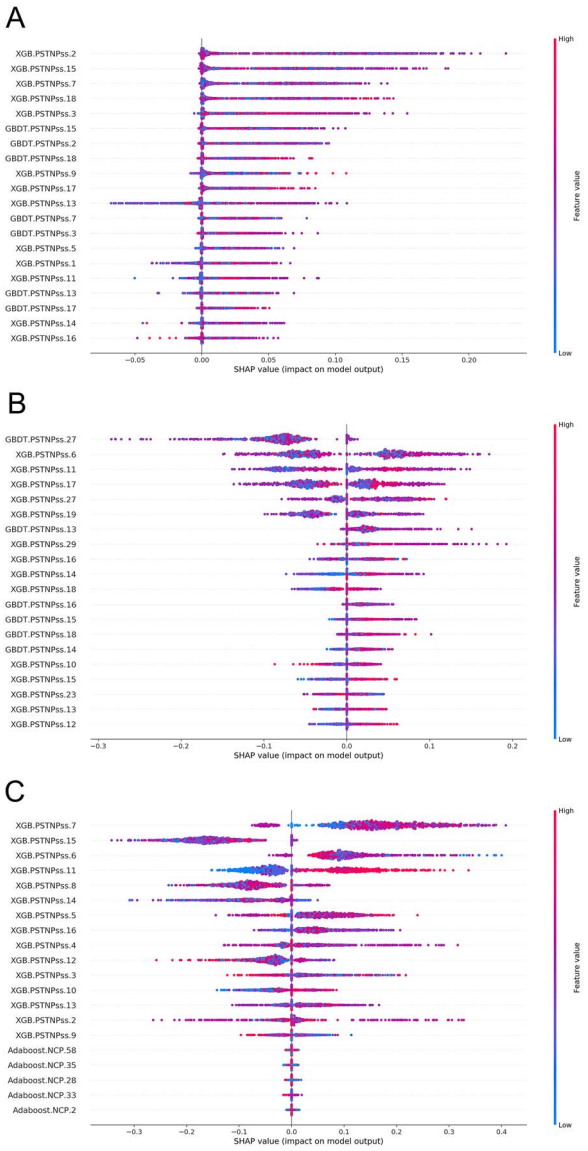

Figure 5. Top 20 features of Porpoise ranked by the SHAP algorithm for predicting species-specific RNA pseudouridine sites of (**A**) *H. sapiens*, (**B**) *S. cerevisiae* and (**C**) *M. musculus*.

Webserver and software development

In this section, to facilitate community-wide efforts in performing high-throughput analysis and prediction of novel potential RNA pseudouridine sites from RNA sequence data, we implemented an online webserver of Porpoise, which is publicly accessible at http://web.unimelb-bioinfortools.cloud.edu.au/Porpoise/. The web page of Porpoise was developed based on PHP, while the webserver is managed by Apache HTTP Server and configured in a 16-core Linux server machine with 64GB RAM and 500GB hard disk. To use the webserver, users only need to input their sequences of interest or upload an input sequence file in the FASTA format and specify the species type(s). Users' submitted tasks will be processed at the server-side to complete the prediction. Upon completion, the prediction results will be returned to the webpage or sent via the user's optionally provided email addresses. A detailed user manual for using the Porpoise web server can be found on the help page of the webserver. The server has been tested using several popular web browsers and works well, including Internet Explorer, Mozilla Firefox, Google Chrome and Safari. In addition, we also make available a stand-alone version of Porpoise, which can be downloaded from the download webpage. Users can download and run the trained models of Porpoise in cases where they need to process large-scale data sets locally.

Conclusion

In this study, we develop a new stacked ensemble learning approach termed Porpoise, to achieve more accurate and improved prediction of RNA pseudouridine sites across three species *H. sapiens*, *S. cerevisiae* and *M. musculus*. In order to select the optimal features and the best combination of base classifiers, we comprehensively evaluate 18 types of RNA sequence-based feature encoding schemes and test nine commonly used machine learning algorithms by performing extensive benchmarking tests. Consequently, an optimized stacked model for each species is constructed; performance comparison through both cross-validation and independent tests demonstrates the superior performance of Porpoise compared with several state-of-the-art predictors. The performance improvement of Porpoise can be attributed to the use of optimal feature selection algorithms and the stacking strategy adopted by Porpoise. In addition, we also confirm the importance of the selected features and aid in the model interpretation using the SHAP algorithm. Furthermore, an online webserver of Porpoise is developed and make freely available at http://web.unimelb-bioinfortools.cloud.edu.au/Porpoise/.

The performance improvement of Porpoise can be attributed to two major reasons: (i) comprehensive evaluation and selection of the most informative features for model training and (ii) extensive investigation of the optimal combination of the base classifiers for the stacked model. The feature selection strategy and stacking framework proposed in this study are generally applicable and can be extended to solve other bioinformatics tasks, such as DNA/RNA modification site prediction. We plan to extend this framework to develop improved prediction models of other species, such as *Arabidopsis thaliana*, in the future work. We anticipate Porpoise will serve as a valuable tool for facilitating the community-wide efforts for RNA pseudouridine site

identification and providing high-quality predicted pseudouridine sites for hypothesis generation and biological validation.

Key Points

- We develop a novel stacking approach, termed Porpoise, for improved and robust prediction of RNA pseudouridine sites from RNA sequence information.
- We undertake a comprehensive benchmarking of the performance of 18 feature encoding schemes and nine commonly used machine learning algorithms for predicting RNA pseudouridine sites and identify the optimal base classifiers for each species-specific model.
- Extensive benchmarking and independent tests show Porpoise achieves a superior predictive performance compared with several state-of-the-art predictors.
- We implement an online webserver of Porpoise to facilitate community-wide efforts in high-throughput analysis and prediction of potential RNA pseudouridine sites from RNA sequence data, which is publicly available at http://web.unimelb-bioinfortools.cloud.edu.au/Porpoise/.

Funding

Doherty Institute at the University of Melbourne (to F.L.); NHMRC career development fellowship (APP1103384 to L.C.); NHMRC-EU project (grant GNT1195743 to L.C.); National Health and Medical Research Council of Australia (NHMRC) (APP1127948, APP1144652 to J.S.); the Australian Research Council (ARC) (LP110200333, DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965); a Major Inter-Disciplinary Research (IDR) project awarded by Monash University.

References

1. Ge J, Yu Y-T. RNA pseudouridylation: new insights into an old modification. *Trends Biochem Sci* 2013;**38**:210–8.
2. Charette M, Gray MW. Pseudouridine in RNA: what, where, how, and why. *IUBMB Life* 2000;**49**:341–52.
3. Davis DR, Veltri CA, Nielsen L. An RNA model system for investigation of pseudouridine stabilization of the codon-anticodon interaction in tRNALys, tRNAHis and tRNATyr. *J Biomol Struct Dyn* 1998;**15**:1121–32.
4. Basak A, Query CC. A pseudouridine residue in the spliceosome core is part of the filamentous growth program in yeast. *Cell Rep* 2014;**8**:966–73.
5. Jack K, Bellodi C, Landry DM, *et al*. rRNA pseudouridylation defects affect ribosomal ligand binding and translational fidelity from yeast to human cells. *Mol Cell* 2011;**44**:660–6.
6. Ma X, Zhao X, Yu YT. Pseudouridylation (Ψ) of U2 snRNA in *S. cerevisiae* is catalyzed by an RNA-independent mechanism. *EMBO J* 2003;**22**:1889–97.
7. Carlile TM, Rojas-Duran MF, Zinshteyn B, *et al*. Pseudouridine profiling reveals regulated mRNA pseudouridylation in yeast and human cells. *Nature* 2014;**515**:143–6.
8. Mei YP, Liao JP, Shen J, *et al*. Small nucleolar RNA 42 acts as an oncogene in lung tumorigenesis. *Oncogene* 2012;**31**:2794–804.
9. Li X, Zhu P, Ma S, *et al*. Chemical pulldown reveals dynamic pseudouridylation of the mammalian transcriptome. *Nat Chem Biol* 2015;**11**:592–7.
10. Li Y-H, Zhang G, Cui QPPUS. a web server to predict PUS-specific pseudouridine sites. *Bioinformatics* 2015;**31**:3362–4.
11. Chen W, Tang H, Ye J, *et al*. iRNA-PseU: identifying RNA pseudouridine sites. *Mol Ther Nucleic Acids* 2016;**5**:e332.
12. He J, Fang T, Zhang Z, *et al*. PseUI: pseudouridine sites identification based on RNA sequence information. *BMC Bioinformatics* 2018;**19**:306.
13. Tahir M, Tayara H, Chong KT. iPseU-CNN: identifying RNA pseudouridine sites using convolutional neural networks. *Mol Ther Nucleic Acids* 2019;**16**:463–70.
14. Liu K, Chen W, Lin H. XG-PseU: an eXtreme Gradient Boosting based method for identifying pseudouridine sites. *Mol Gen Genomics* 2020;**295**:13–21.
15. Bi Y, Jin D, Jia C. EnsemPseU: identifying pseudouridine sites with an ensemble approach. *IEEE Access* 2020;**8**:79376–82.
16. Lv Z, Zhang J, Ding H, *et al*. RF-PseU: a random forest predictor for RNA pseudouridine sites. *Front Bioeng Biotechnol* 2020;**8**:134.
17. Khan SM, He F, Wang D, *et al*. MU-PseUDeep: a deep learning method for prediction of pseudouridine sites. *Comput Struct Biotechnol J* 2020;**18**:1877–83.
18. Song B, Tang Y, Wei Z, *et al*. PIANO: a web server for pseudouridine-site (Psi) identification and functional annotation. *Front Genet* 2020;**11**:88.
19. Song B, Chen K, Tang Y, *et al*. PSI-MOUSE: predicting mouse pseudouridine sites from sequence and genome-derived features. *Evol Bioinformatics Online* 2020;**16**:1176934320925752.
20. Chen K, Wei Z, Zhang Q, *et al*. WHISTLE: a high-accuracy map of the human N6-methyladenosine (m6A) epitranscriptome predicted using a machine learning approach. *Nucleic Acids Res* 2019;**47**:e41.
21. Sun WJ, Li JH, Liu S, *et al*. RMBase: a resource for decoding the landscape of RNA modifications from high-throughput sequencing data. *Nucleic Acids Res* 2016;**44**:D259–65.
22. Tang Y, Chen K, Song B, *et al*. m6A-Atlas: a comprehensive knowledgebase for unraveling the N6-methyladenosine (m6A) epitranscriptome. *Nucleic Acids Res* 2021;**49**:D134–43.
23. Chen Z, Zhao P, Li F, *et al*. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Brief Bioinform* 2020;**21**:1047–57.
24. Chen Z, Zhao P, Li C, *et al*. iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. *Nucleic Acids Res* 2021. 10.1093/nar/gkab122.
25. Mishra A, Pokhrel P, Hoque MT. StackDPPred: a stacking based prediction of DNA-binding protein from sequence. *Bioinformatics* 2019;**35**:433–41.
26. Su R, Liu X, Xiao G, *et al*. Meta-GDBP: a high-level stacked regression model to improve anticancer drug response prediction. *Brief Bioinform* 2021;**21**:996–1005.
27. Verma A, Mehta S. A comparative study of ensemble learning methods for classification in bioinformatics. In: *Proceedings of the 7th International Conference on Cloud Computing Data Science and Engineering (Confluence 2017)*, IEEE, Noida, India, 2017:155–158.
28. Wei L, He W, Malik A, *et al*. Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework. *Brief Bioinform* 2020. 10.1093/bib/bbaa275.

29. Li F, Chen J, Ge Z, *et al*. Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework. *Brief Bioinform* 2021;**22**:2126–40.
30. Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: *ICML*. USA: Citeseer, 1996, 148–56.
31. Chen T, He T, Benesty M, *et al*. *Xgboost: Extreme Gradient Boosting*, R package version v0.4-2, 2015. https://cran.r-project.org/web/packages/xgboost/.
32. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat* 2001;1189–232.
33. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. arXiv preprint arXiv:1206.2944 2012.
34. Raschka S. MLxtend: providing machine learning and data science utilities and extensions to Python's scientific computing stack. *J Open Source Software* 2018;**3**:638.
35. Li F, Chen J, Leier A, *et al*. DeepCleave: a deep learning predictor for caspase and matrix metalloprotease substrates and cleavage sites. *Bioinformatics* 2020;**36**:1057–65.
36. Li F, Leier A, Liu Q, *et al*. Procleave: predicting protease-specific substrate cleavage sites by combining sequence and structural information. *Genomics Proteomics Bioinformatics* 2020;**18**:52–64.
37. Liu Q, Chen J, Wang Y, *et al*. DeepTorrent: a deep learning-based approach for predicting DNA N4-methylcytosine sites. *Brief Bioinform* 2021;**22**. 10.1093/bib/bbaa124.
38. Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 2005;**27**:1226–38.
39. Li F, Li C, Wang M, *et al*. GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. *Bioinformatics* 2015;**31**:1411–9.
40. Li F, Li C, Revote J, *et al*. GlycoMine(struct): a new bioinformatics tool for highly accurate mapping of the human N-linked and O-linked glycoproteomes by incorporating structural features. *Sci Rep* 2016;**6**:34595.
41. Lundberg SM. Lee S-I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017, 4765–74.
42. Bi Y, Xiang D, Ge Z, *et al*. An interpretable prediction model for identifying N(7)-methylguanosine sites based on XGBoost and SHAP. *Mol Ther Nucleic Acids* 2020;**22**:362–72.