Briefings in Bioinformatics, 21(3), 2020, 1047–1057

doi: 10.1093/bib/bbz041
Advance Access Publication Date: 24 April 2019
Problem solving protocol

iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data

Zhen Chen†, Pei Zhao†, Fuyi Li, Tatiana T. Marquez-Lago, André Leier, Jerico Revote, Yan Zhu, David R. Powell, Tatsuya Akutsu, Geoffrey I. Webb, Kuo-Chen Chou, A. Ian Smith, Roger J. Daly, Jian Li and Jiangning Song

Corresponding author: Jiangning Song, Biomedicine Discovery Institute, Department of Biochemistry and Molecular Biology and Monash Centre for Data Science, Monash University, Melbourne, Victoria 3800, Australia. Tel.: +61-3-9902-9304; Email: Jiangning.Song@monash.edu
†These two authors contributed equally to this work.

Zhen Chen is an assistant professor at the School of Basic Medical Science, Qingdao University, China. His research interests include protein bioinformatics, machine learning and analysis of next-generation sequencing.
Pei Zhao is an assistant professor in the State Key Laboratory of Cotton Biology, Institute of Cotton Research of Chinese Academy of Agricultural Sciences (CAAS), China. Her research interests are bioinformatics and analysis of next-generation sequencing in plants.
Fuyi Li received his BEng and MEng degrees in Software Engineering from Northwest A&F University, China. He is currently a PhD candidate in the Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Australia. His research interests are bioinformatics, computational biology, machine learning and data mining.
Tatiana Marquez-Lago is an associate professor in the Department of Genetics and the Department of Cell, Developmental and Integrative Biology, University of Alabama at Birmingham School of Medicine, USA. Her research interests include multiscale modeling and simulations, artificial intelligence, bioengineering and systems biomedicine. Her interdisciplinary laboratory studies stochastic gene expression, chromatin organization, antibiotic resistance reversal in bacteria and host-microbiota interactions in complex diseases.
André Leier is an assistant professor in the Department of Genetics, University of Alabama at Birmingham (UAB) School of Medicine, USA. He is also an associate scientist at the UAB Comprehensive Cancer Center. He received his PhD in computer science (Dr rer. nat.), University of Dortmund, Germany. He conducted postdoctoral research at the Memorial University of Newfoundland, Canada; the University of Queensland, Australia; and Eidgenössische Technische Hochschule (ETH) Zürich, Switzerland. His research interests are in biomedical informatics and computational and systems biomedicine.
Jerico Revote received his bachelor degree in computer science from Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Australia. He is working as a research engineer in the Monash eResearch Centre at Monash University, Australia. He is also currently a part-time PhD student in the Department of Biochemistry and Molecular Biology and Biomedicine Discovery Institute, Monash University. His research interests are machine learning, data mining, artificial intelligence, software development and scalable applications.
Yan Zhu received his PhD in microbiology from the Chinese Academy of Sciences. He is a research fellow at the Monash Biomedicine Discovery Institute and Department of Microbiology, Monash University, Australia. His research currently focuses on systems antimicrobial pharmacology and virtual cell modeling and targets an urgent global medical challenge due to antibiotic resistance and the lack of new antibiotics against Gram-negative 'superbugs'.
David R. Powell received his PhD in computer science from the Monash University, Melbourne, Australia. He is an associate professor and scientific director of the Monash Bioinformatics Platform at the Monash University. He is most interested in topics involving bioinformatics, in particular, interactive visualization of biological data.
Tatsuya Akutsu received his DEng degree in information engineering in 1989 from the University of Tokyo, Japan. He has been appointed as a professor since 2001 at the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan. His research interests include bioinformatics and discrete algorithms.
Geoffrey I. Webb received his PhD degree in computer science in 1987 from the La Trobe University. He is the director of the Monash Centre for Data Science and professor in the faculty of information technology at the Monash University. His research interests include machine learning, data mining, computational biology and user modeling.
Kuo-Chen Chou received his DSc degree in 1984 from Kyoto University, Japan. He is the founder and chief scientist of Gordon Life Science Institute. He is also a distinguished high impact professor and advisory professor of several universities. Professor Chou's research interests are in computational biology and biomedicine, protein structure prediction, low-frequency internal motion of protein/DNA molecules and its biological functions, diffusion-controlled reactions of enzymes, as well as graphic rules in enzyme kinetics and other biological systems.
A. Ian Smith completed his PhD at the Prince Henry's Institute Melbourne and Monash University, Australia. He is vice-provost (Research & Research Infrastructure) of Monash University. He is also a professorial fellow in the Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology at the Monash University where he runs his research group. His research applies proteomics technologies to study the proteases involved in the generation and metabolism of peptide regulators involved in both brain and cardiovascular function.
Roger J. Daly obtained his PhD from the University of Liverpool, UK. He is the head of the Department of Biochemistry and Molecular Biology and the Cancer Program within the Biomedicine Discovery Institute at the Monash University. His research applies cutting-edge technology platforms in mass spectrometry-based proteomics and kinomics to inform and refine the sub classification of particular cancers, identify novel therapeutic targets and new applications for existing therapies, and also identify companion biomarkers that help stratify patients for appropriate therapy.
Jian Li is a professor and group leader in the Monash Biomedicine Discovery Institute and Department of Microbiology, Monash University, Australia. He is a Web of Science 2015–2017 highly cited researcher in pharmacology & toxicology. He is currently a National Health and Medical Research Council principal research fellow. His research interests include the pharmacology of polymyxins and the discovery of novel, safer polymyxins.
Jiangning Song is an associate professor and group leader at the Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Australia. He is a member of the Monash Centre for Data Science, Faculty of Information Technology and an associate investigator of the Australian Research Council Centre of Excellence in advanced molecular imaging, Monash University. His research interests include bioinformatics, computational biomedicine, machine learning, data analytics and pattern recognition.

Submitted: 20 January 2019; **Received (in revised form):** 28 February 2019

Abstract

With the explosive growth of biological sequences generated in the post-genomic era, one of the most challenging problems in bioinformatics and computational biology is to computationally characterize sequences, structures and functions in an efficient, accurate and high-throughput manner. A number of online web servers and stand-alone tools have been developed to address this to date; however, all these tools have their limitations and drawbacks in terms of their effectiveness, user-friendliness and capacity. Here, we present *iLearn*, a comprehensive and versatile Python-based toolkit, integrating the functionality of feature extraction, clustering, normalization, selection, dimensionality reduction, predictor construction, best descriptor/model selection, ensemble learning and results visualization for DNA, RNA and protein sequences. *iLearn* was designed for users that only want to upload their data set and select the functions they need calculated from it, while all necessary procedures and optimal settings are completed automatically by the software. *iLearn* includes a variety of descriptors for DNA, RNA and proteins, and four feature output formats are supported so as to facilitate direct output usage or communication with other computational tools. In total, *iLearn* encompasses 16 different types of feature clustering, selection, normalization and dimensionality reduction algorithms, and five commonly used machine-learning algorithms, thereby greatly facilitating feature analysis and predictor construction. *iLearn* is made freely available via an online web server and a stand-alone toolkit.

Key words: bioinformatics; integrated platform; sequence analysis; machine learning; automated modeling; data clustering; feature selection; biomedical data mining

Introduction

With the rapid accumulation of DNA, RNA and protein sequences in the post-genomic age, there is an ever-widening gap between sequencing data and their analysis and annotations [1]. Studying various attributes in these sequences is of crucial importance for both basic research (structure and function studies of DNA, RNA or proteins) and drug target development [2]. However, many of these attributes are impossible to discern through human inspection alone; in other cases, related wet laboratory experiments are prohibitively costly, both in time and financially, thereby limiting the analysis process and validation of sequences. For example, the number of protein sequences in the latest version of the NCBI NR database has reached 171 418 145 (as of 2018/10/09), while the number of well-annotated sequences in Swiss-Prot is only 557 491 (version 2018_6). Consequently, computational methods for biological sequence function prediction/analysis are urgently needed.

The majority of prediction tasks in the field of biological sequence analysis can be formulated as a classification problem, i.e. binary classification tasks or multi-class classification tasks [2], such as prediction of protein structural and function classes [3], fold recognition [4, 5], protein–protein interactions, protein–ligand interactions [6, 7], subcellular locations [8], enzyme substrates [9–11], protein function sites [12–14], identification of noncoding RNAs [15], enhancer identification [16], RNA target prediction [17] and RNA-binding protein identification [18, 19], among many others. With the assistance of machine-learning algorithms, many efficient computational methods have been developed to expeditiously predict/analyze various attributes of biological sequences based on their sequence information alone [16, 20–25]. As done in many recent publications (see for example [26, 27]), there exist several major steps for the development of a useful sequence-based statistical predictor for a biological system. The latter comes down to making the following steps very clear: (i) how to construct or select a valid benchmark data set to train and test the predictor, (ii) how to formulate the biological sequence samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target to be predicted, (iii) how to introduce or develop a powerful algorithm (or engine) to operate the prediction, (iv) how to properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor and (v) how to establish a user-friendly web-server for the predictor that is accessible to the public. A general flowchart of computational methods based on machine-learning algorithms for biological sequences analysis is shown in Figure 1. In addition, we have also added the feature analysis step (i.e. feature clustering, selection, dimensionality reduction and feature normalization), descriptor performance evaluation and ensemble learning step into the framework of *iLearn*.

From these steps, it is worth noting that feature extraction/calculation has special relevance for sequence analysis; transforming sequences into an effective mathematical expression truly reflecting the intrinsic relation with the attribute to be predicted can significantly affect performance of any resulting model [28]. For this purpose, several web servers and stand-alone software packages have been developed, calculating various structural and physicochemical features. Examples include PROFEAT [29, 30], PseAAC [31], propy [32], repDNA [25], PseAAC-General [33], protr/ProtrWeb [34], Rcpi [6], PseKRAAC [35], Pse-in-One [24], POSSUM [36], BioSeq-Analysis [2] and our previously developed iFeature [37].

Predictor construction is another key step for meaningful sequence analysis. To speed up analysis of ever-growing available biological sequences, many machine-learning algorithms have been used to predict structural and functional properties and to assist in the annotation of genomic and proteomic data [38, 39]. Some examples of these algorithms are support vector machine (SVM) [40], random forest (RF) [41], artificial neutral network (ANN) [42], *k*-nearest neighbors (KNN) [43], logistic regression (LR) [44], etc. Lastly, feature analysis is yet another crucial step in machine learning of sequence analysis.

To the best of our knowledge, most of the toolkits or analysis web servers only focus on individual steps listed above, or subsets, but do not include all. Among all these toolkits, BioSeq-Analysis [2] is the only toolkit that has been proposed to automate the steps of feature extraction, predictor construction and performance evaluation. However, BioSeq-Analysis does have its own limitations; it lacks the functionality of feature normalization, clustering, dimension reduction, best model selection,

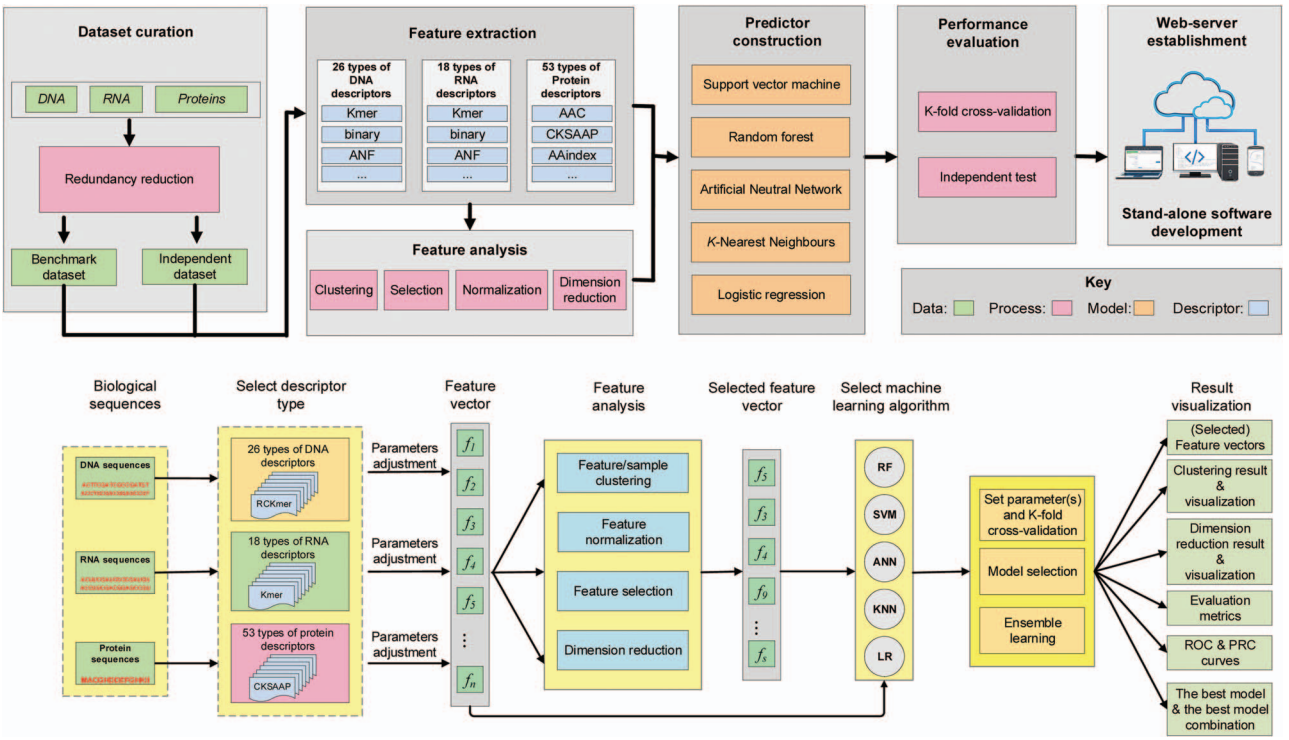

Figure 1. Flowchart of computation methods based on machine-learning algorithms for biological sequences analysis.

ensemble learning and supports only two feature selection algorithms. In addition, BioSeq-Analysis lacks visualization for the feature analysis results. It is in this spirit that we developed *iLearn*, a versatile open-source Python toolkit, which integrates feature calculation, extraction, clustering, feature selection, feature normalization, predictor construction best model selection and ensemble learning for classification and result visualization for DNA, RNA and protein sequences. We additionally note that, while BioSeq-Analysis has 56 descriptors, *iLearn* contains 97 (Table 1). *iLearn* greatly facilitates users' efforts to identify relevant features, conduct feature analysis and construct effective machine-learning-based models without having to retort to different tools for each individual step. Lastly, we developed a user-friendly web server for users to access and use *iLearn*, freely.

Compared with BioSeq-Analysis, the most comprehensive predecessor of *iLearn*, the following advantages (Table 1) of *iLearn* include: (i) more descriptors for DNA, RNA and protein sequences; (ii) enhanced feature analysis functionality for clustering, feature vector normalization, dimension reduction and feature selection; (iii) support of more machine-learning algorithms and more evaluation metrics; (iv) more feature vector output formats; (v) larger number of sequences that can be processed at one submission; (vi) submission tasks can be queried through the *iLearn* web-server; (vii) more abundant graphical display options for results; (viii) one or more machine-learning algorithms can be selected for submission and as a result *iLearn* will return the model with the best performance; and (ix) *iLearn* allows the integration of multiple machine-learning algorithms to achieve the overall best performance (i.e. through ensemble learning).

Implementation

In a previous study, we developed iFeature [37], a versatile open-source Python toolkit, which integrates the functionality of feature calculation and feature analysis. However, iFeature was designed only to focus on analysis of protein and peptide sequences. Due to the demand for use and applicability of iFeature, we decided to extend our work and developed *iLearn*, which integrates the multi-faced functionality of feature extraction/calculation, feature analysis, predictor construction, best descriptor/model selection, ensembling learning, performance evaluation, web-server establishment and stand-alone software packages development for DNA, RNA and protein sequences.

Biological sequences representation

iFeature [37] calculates and extracts a comprehensive spectrum of 18 major sequence encoding schemes that encompass 53 different types of feature descriptors. We have embedded all these feature-encoding schemes for protein and peptide sequences in *iLearn*. In addition, *iLearn* also can be used to extract six major encoding schemes, which encompass 26 different types of feature descriptors, for DNA and RNA sequences. A complete list of the six major encoding schemes is summarized in Table 2 and is briefly discussed below. The first group includes nine feature sets, i.e. nucleic acid composition, enhanced nucleic acid composition, di-nucleotide composition, tri-nucleotide composition, *k*-spaced nucleic acid pairs [45], basic kmer, the reverse compliment kmer, the accumulated nucleotide frequency and nucleotide chemical property. The second group is the binary encoding scheme, in which each nucleic acid is represented by a four-dimensional binary vector [45]. The third group includes two types of position-specific tendencies of trinucleotide (i.e. position-specific trinucleotide propensity based on single-strand and position-specific trinucleotide propensity based on single-strand) [46, 47]. The fourth group includes two types of electron–ion interaction pseudopotentials (i.e. electron–ion interaction pseudopotentials value and electron–ion interaction pseudopotentials of trinucleotide)

Table 1. The advantages and key features of *iLearn* compared with BioSeq-Analysis

	iLearn	BioSeq-Analysis
Number of descriptors for DNA	26	20
Number of descriptors for RNA	18	14
Number of descriptors for protein	53	22
Number of clustering algorithms	6	0
Number of feature selection algorithms	5	2
Number of feature vector normalization algorithms	2	0
Number of dimension reduction algorithms	3	0
Number of machine-learning algorithms	5	4
Number of cross-validation methods	2	3
The maximum uploaded sequence at once	2000	1200
Feature vector format	4	2
Evaluation metrics	9	5
Cross-validation and independent test at one submission	Yes	No
User defined cross-validation	Yes	No
Select the model with best performance	Yes	No
Ensemble learning	Yes	No
Query history tasks	Yes	No

[46, 47]. The fifth group includes six types of autocorrelation and cross-covariance feature sets [24, 25]: dinucleotide-based auto covariance, dinucleotide-based cross covariance, dinucleotide-based auto-cross covariance, trinucleotide-based auto covariance, trinucleotide-based cross covariance and trinucleotide-based auto-cross covariance. The sixth group consists of six types of pseudo nucleic acid composition [24, 25]: pseudo dinucleotide composition, pseudo *k*-tupler composition, parallel correlation pseudo dinucleotide composition, parallel correlation pseudo trinucleotide composition, series correlation pseudo dinucleotide composition and series correlation pseudo trinucleotide composition.

Feature analysis

The dimension for some descriptors is extremely high. For example, the AAindex database contains 544 physicochemical properties for each amino acid. For a peptide of 21 residues, the dimensionality of the AAindex descriptors would be 11 424 (i.e. 21∗544). High-dimensional feature sets usually contain noisy and misleading features that are detrimental to the prediction performance and may also increase redundant information. Consequently, machine-learning models trained using high-dimensional initial features often perform poorly. To resolve this problem and reduce the dimension of the resulting feature vectors to a reasonable level, *iLearn* uses five feature selection methods. These include chi-square, mutual information, information gain, Pearson's correlation coefficient and F-score algorithms. By using these feature selection methods, features will be ranked according to their importance, and a user-defined dimension of the features' vectors can be used. Lastly, the interpretability of the descriptors is also very important and feature selection is also used to interpret the relations between the descriptors and the attributes of biological sequences in some studies. Accordingly, in addition to feature selection, *iLearn* also integrates six types of clustering algorithms (i.e. *K*-means clustering [48, 49], Gaussian mixture model, hierarchical clustering [48, 50], mean shift [51], DBSCAN [52] and affinity propagation algorithms [53]) and three types of dimensionality reduction algorithms (i.e. principal component analysis, latent dirichlet allocation and t-distributed stochastic neighbor embedding algorithms). All clustering methods support sample and feature clustering procedures, and clustering and dimensionality reduction results can be visualized in the form of both 2D and 3D scatter diagrams.

Model construction

After the feature extraction process, a powerful machine-learning algorithm is ideally used for constructing the predictors. In *iLearn*, we provide five such options: SVM, RF, ANN, KNN and LR, all commonly used machine-learning algorithms.

SVM is usually the 'first choice' adopted in many computational biology studies. SVM is a supervised learning algorithm, which aims to accurately classify samples by generating optimal hyperplanes based on the feature dimensionality of the training data. A variety of kernels have been developed for SVM, including Gaussian radial basis function (RBF), linear, polynomial and sigmoid kernels, noting users might want to have flexible options of choosing a specific type of kernel function when using in *iLearn*. In addition, *iLearn* also supports automatic parameter optimization, while still permitting users to specify their own kernel functions.

RF is another well-established and widely employed algorithm for bioinformatics applications, not only for classification problems but also regressions and other tasks. The obvious advantage of RF is its interpretability '*if*...*then*...' rules in the decision trees depict how a set of features collaborate to classify the samples. Such rules can potentially provide biologists with intuitive insights and knowledge discovery that would otherwise remain buried in the data [45]. When utilizing the RF algorithm in *iLearn*, one should bear in mind that the number of decision trees is an important parameter and should be specified by the users.

An ANN usually contains multiple nodes as input and multiple layers to connect these input nodes. A typical architecture of ANN is composed of three layers, including the input layer, the hidden layer and the output layer. To construct an ANN model, users need to specify the number of hidden layers and the number of neurons in each hidden layer.

The KNN algorithm is a commonly employed unsupervised algorithm that finds the neighbors of a query sample by calculating the similarities/distances among samples belonging to different categories. The class of the query sample is determined by the similarities/distances between the query sample and its neighbors in corresponding categories.

Table 2. List of various descriptors calculated by *iLearn* for DNA and RNA sequences

Descriptor groups	Descriptor	Dimension	DNA or RNA
Nucleic acid composition	Nucleic acid composition (NAC)	4	DNA/RNA
	Enhanced nucleic acid composition (ENAC)	–	DNA/ RNA
	Di-nucleotide composition (DNC)	16	DNA/RNA
	Tri-nucleotide composition (TNC)	64	DNA/RNA
	k-spaced nucleic acid pairs (CKSNNP)	$(k+1)$x16	DNA/RNA
	Basic kmer (Kmer)	–	DNA/RNA
	Reverse compliment kmer (RCKmer)	–	DNA/RNA
	Accumulated nucleotide frequency (ANF)	–	DNA/RNA
	Nucleotide chemical property (NCP)	–	DNA/RNA
Binary	Binary (binary)	–	DNA/RNA
Position-specific tendencies of trinucleotide	Position-specific trinucleotide propensity based on single-strand (PSTNPss)		DNA/RNA
	Position-specific trinucleotide propensity based on single-strand (PSTNPds)		DNA
Electron-ion interaction pseudopotentials	Electron-ion interaction pseudopotentials value (EIIP)		DNA
	Electron-ion interaction pseudopotentials of trinucleotide (PseEIIP)		DNA
Autocorrelation and cross-covariance	Dinucleotide-based auto covariance (DAC)		DNA/RNA
	Dinucleotide-based cross covariance (DCC)		DNA/RNA
	Dinucleotide-based auto-cross covariance (DACC)		DNA/RNA
	Trinucleotide-based auto covariance (TAC)		DNA
	Trinucleotide-based cross covariance (TCC)		DNA
	Trinucleotide-based auto-cross covariance (TACC)		DNA
Pseudo nucleic acid composition	Pseudo dinucleotide composition (PseDNC)		DNA/RNA
	Pseudo k-tupler composition (PseKNC)		DNA/RNA
	Parallel correlation pseudo dinucleotide composition (PCPseDNC)		DNA/RNA
	Parallel correlation pseudo trinucleotide composition (PCPseTNC)		DNA
	Series correlation pseudo dinucleotide composition (SCPseDNC)		DNA/RNA
	Series correlation pseudo trinucleotide composition (SCPseTNC)		DNA

LR is also a machine-learning framework, which can be used to estimate the probability of a binary response based on one or more predictors.

In *iLearn*, the scikit-learn package was used as the implementation of the algorithms mentioned above.

Best model selection and ensemble learning

If more than one machine-learning algorithms were selected at the submission, *iLearn* will use the selected algorithms to build the prediction models, and the model that achieves the best performance will be identified. In addition, the ensemble learning functionality, which further combines the prediction outputs of multiple machine-learning algorithms and makes the final prediction outcomes with the possibility of achieving the overall best performance, is also supported and made available by *iLearn*. For ensemble learning, *iLearn* will evaluate and report the performance for all possible combinations of the selected machine-learning algorithms through a logistic regression approach and return the combination of algorithms that achieves the best performance.

Performance evaluation measures and strategies

Validation examination is an important step prior to applying a predictor [54]. In both the web server and stand-alone software package options, *iLearn* uses the *K*-fold cross-validation test and the independent test to evaluate the performance of constructed predictors. For the *K*-fold cross-validation test, and taking the SVM model in *iLearn* as an example, the data set is divided into *K* roughly equivalent parts; one part is kept as validation data and the remaining *K*-1 parts used as training data. Any or all parts of the training data are used to estimate the parameters (e.g. the gamma and cost in the RBF kernel of the SVM model), while the validation data is used to compute all the performance metrics. The procedure is repeated *K* times, thus using each of the *K* parts as test data once [55]. In parallel with the *K*-fold cross-validation test, the independent test is usually adopted to evaluate the performance of two or more predictors.

We note most of the tasks in the fields are binary classification problems. Seven commonly employed measures are provided and can be calculated by *iLearn*, including sensitivity (Sn), specificity (Sp), accuracy (Acc), Matthews correlation coefficient (MCC), Recall, Precision, F1-score, the area under receiver operating characteristic (ROC) curve (AUROC) and the area under the Precision-Recall (PRC) curve (AUPRC) [2, 16, 26, 27, 56, 57]. Sn, Sp, ACC, MCC, Recall, Precision and F1-score are defined as:

$$Sn = Recall = 1 - \frac{N_{-}^{+}}{N^{+}}, \quad (1)$$

$$Sp = 1 - \frac{N_{+}^{-}}{N^{-}}, \quad (2)$$

$$Acc = 1 - \frac{N_{-}^{+} + N_{+}^{-}}{N^{+} + N^{-}}, \quad (3)$$

$$MCC = \frac{1 - \left(\frac{N_{-}^{+} + N_{+}^{-}}{N^{+} + N^{-}}\right)}{\sqrt{\left(1 + \frac{N_{+}^{-} - N^{+}}{N^{+}}\right)\left(1 + \frac{N_{-}^{+} - N^{-}}{N^{-}}\right)}}, \quad (4)$$

$$Precision = 1 - \frac{N_{+}^{-}}{N^{+} + N_{+}^{-}}, \quad (5)$$

$$F1 - score = 1 - \frac{N_{+}^{-} + N_{-}^{+}}{2 \times N^{+} + N_{+}^{-} + N_{-}^{+}}, \quad (6)$$

where N^{+}, N_{+}^{-}, N^{-} and N_{-}^{+} represent the numbers of true positives, false positives, true negatives and false negatives, respectively. The AUROC value is calculated based on the ROC curve, and takes values between 0 and 1, while the AUPRC value is calculated based on the precision-recall curve (where, the higher the AUROC and AUPRC value, the better the prediction performance).

For multiclass classification problems, the Acc is used to evaluate the performance, which is defined as [56, 58]:

$$Acc = 1 - \frac{N_{-}^{+}(i) + N_{+}^{-}(i)}{N^{+}(i) + N^{-}(i)}, \quad (7)$$

where $N^{+}(i)$, $N_{+}^{-}(i)$, $N^{-}(i)$ and $N_{-}^{+}(i)$ represent the numbers of the samples in the i-th class, the total number of the samples in the i-th class but predicted as one of the other classes, the total number of the samples not in the i-th class, and the total number of the samples not in the i-th class but predicted as the i-th class, respectively.

Web server

Input

To facilitate users' choices and maximize their convenience, the *iLearn* web server includes three independent web servers, specifically designed for the analysis and machine-learning modeling of DNA, RNA and protein sequences, respectively. For all three, the input is a set of DNA, RNA or protein sequences in a special FASTA format, which can be either uploaded as a single file or copied/passed into the 'TEXT' box. The FASTA header consists of three parts: part 1, part 2 and part 3, which are separated by the symbol '|' (Figure 2). Part 1 is the sequence name. Part 2 is the sample category information, which can be filled with any integer. Usually, users enter 1 to identify the positive samples and −1 or 0 to represent the negative samples for a binary classification task. Alternatively, they may use 0, 1, 2, ... to represent the different classes in multiclass classification tasks. Part 3 indicates the role of the sample, where e.g. 'training' would indicate the corresponding sequence would be used as the training set for *K*-fold validation test and 'testing' would imply the sequence would be used as the independent set for independent testing. It's worth mentioning that part 1 is necessary, while parts 2 and 3 are optional. If the information of part 2 is missing, the fourth step/process (predictor construction) will be skipped. Lastly, it is worth noting the total number of input sequences is limited to 2000, for a single submission.

Output and result visualization

In the results page, the calculated features, feature analysis results and evaluation metrics results are listed in a table. For calculated features and feature analysis results, only the first 10 samples are displayed in the results page, but users can download the complete results as well. *iLearn* provides results in any of four types of output formats (i.e. SVM, Comma-Separated Values (CSV), Tab Separated Values (TSV) and Waikato Environment for Knowledge Analysis (WEKA)). In order to facilitate interpretability of outputs, both the clustering and dimensionality reduction results can be visualized in the form of 2D and 3D scatter diagrams, while the performance of the predictor is displayed by both ROC and PRC curves. Figure 3 shows some examples of the diagrams generated by *iLearn*. For *K*-fold cross-validation test, the ROC or PRC curve for each fold will be displayed in a diagram. The AUROC or AUPRC for each fold and the mean value of AUROC or AUPRC for the *K*-fold will be showed in the legend of the diagram.

Application of *iLearn*

In this section, we apply the *iLearn* stand-alone package to address two practical problems, one with respect to RNA, the other based on protein sequence data; we want to construct machine-learning-based models for RNA 5-methylcytosine (m^5C) site prediction and protein malonylation site prediction. Specifically, we will elaborate on the detailed steps and commands involved in the calculation of different feature encodings and construction of the models and discuss the performance of the constructed predictors. The competitive performance achieved by the *iLearn*-constructed models suggests that *iLearn* is very useful for developing new models with a variety of available feature descriptors and machine-learning algorithms in a user-friendly and cost-effective way. In addition, it also facilitates the down-stream prediction result analysis and visualization with multiple diagrams.

Identification of m^5C sites

m^5C is an important post-transcriptional modification that plays an indispensable role in biological processes [59, 60], including affecting aminoacylation and codon identification, stabilizing the tRNA secondary structure, regulating stress responses and in the proliferation of stem cells [61–64]. Due to the difficulty and expensive costs of identifying m^5C sites with wet laboratory techniques, several predictors have been developed. Zhang *et al.* [65] developed an m5C site predictor named M5C-HPCR by using SVM and with informative nucleotide physicochemical property features (i.e. PseDNC descriptor in *iLearn*), Song *et al.* [66] also proposed an RF-based predictor called PEA-m5C, in which binary, Kmer and nucleotide physicochemical property were employed as the input vector. The state-of-the-art predictors for the prediction of (m^5C) site can be easily reproduced.

Construct SVM-PseDNC predictor

The Met1320 data set was downloaded from [65] as the benchmark data set, which consists of 120 true m^5C sites and 120 negative sites. For establishing SVM-PseDNC predictor, the following commands can be used:

```
# generate the feature vector
python iLearn-nucleotide-Pse.py -file
examples/M5C_sequences.txt -type RNA -method
PseDNC -format svm -out PseDNC_vector.txt
# construct SVM-PseDNC model
python iLearn-ML-SVM.py -train PseDNC_vector.txt
format svm -kernel rbf -auto -out PseDNC
```

```
>AT1G09780|1|training
GTGGAGTAGAAGAATTGAGAGCCTTATCAG
TTTTTGAAGAGAGGGCTGAAACTCTCTAGT
TATCTTTTGTTGCTTTTCTAATAATAAGAG
TTTACACACAG
>AT1G31812|0|testing
TCCTCATCTGCAGTAACTTTATCTTAAGCA
TCAAAATAACATTGCATAAGACTTGTTCTT
GCTCTTGTGTTTCTATCATATTTAAGCTAT
CTACTTTGTGA
```

Part 1

Part 2

Part 3

Figure 2. An example of the FASTA format used in *iLearn*.

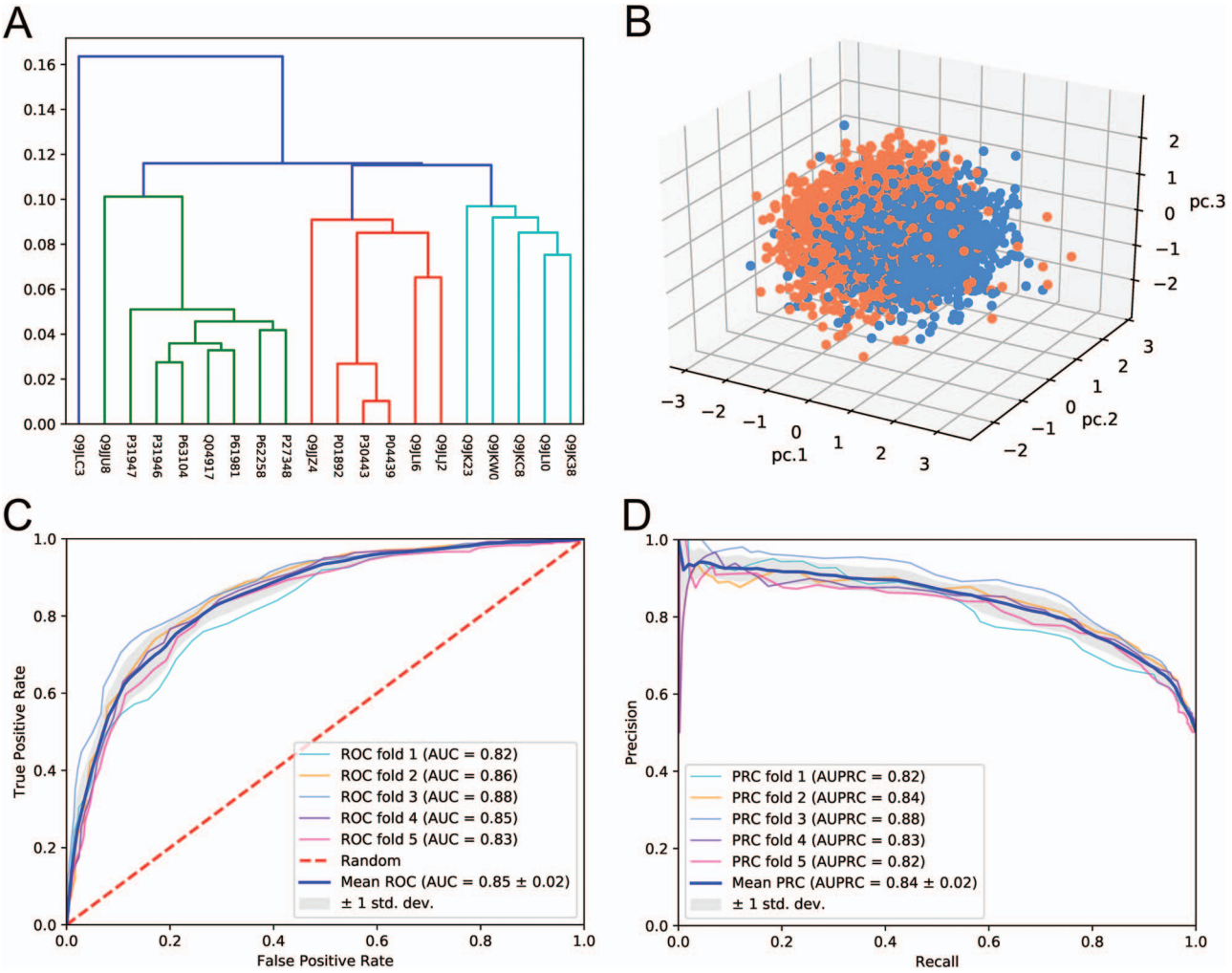

Figure 3. Sample diagrams generated by *iLearn*. (**A**) Hierarchical cluster diagram, (**B**) 3D diagram of the Principal Component Analysis (PCA) analysis, (**C**) and (**D**) are ROC and PRC curves, respectively.

Five files will be generated, 'PseDNC_vector.txt' is the generated feature vector, 'PseDNC_metrics_CV.txt' is the metrics for performance evaluation, 'PseDNC_CV.txt' is the prediction score for each sample, 'PseDNC_ROC_CV.png' and 'PseDNC_PRC_CV.png' are the ROC and PRC curves, respectively. The ROC and PRC curves can be found in Supplementary Figure S1.

Clustering

The sample clustering can be run by the command:

```
# generate the feature vector for clustering
python iLearn-nucleotide-Pse.py --file
examples/M5C_sequences.txt --type RNA --method
PseDNC --format tsv_1 --outPseDNC_vector_for_
 clustering.txt
# feature clustering
python iLearn-clustering.py --file
PseDNC_vector_for_clustering.txt --method kmeans
--nclusters 2 --out PseDNC_clustering_res.txt
```

The samples would be divided into two classes, and the clustering result will be saved in the 'PseDNC_clustering_res.txt'. Supplementary Figure S2 displayed the scatter diagram of the clustering result.

Feature normalization and selection

```
# feature normalization
python iLearn-feature-normalization.py --file
PseDNC_vector.txt --format svm --method ZScore
--out PseDNC_vector_normalized.txt
# featture selection
python iLearn-feature-selectior.py
--file PseDNC_vector_normalized.txt --
format svm --method CHI2 --num 10 --out
PseDNC_vector_selected.txt
--rank PseDNC_feature_rank.txt
```

The 'ZScore' method was selected to normalize the features and the top 10 features were selected through the 'CHI2' method.

Dimension reduction

```
# dimension reduction
python iLearn-dimension-reduction.py --file
PseDNC_vector_normalized.txt --format
svm --method pca --ncomponents 3 --out
PseDNC_vector_reduced.txt
```

The 3D scatter diagram for dimension reduction is shown in Supplementary Figure S3.

iLearn descriptors estimater

In most of the prediction tasks, we need select the descriptor(s) and machine algorithm(s) with the best performance and *iLearn* package make this easier to implement. For example, we can specify the descriptor types and machine-learning algorithms in the configure file (See section 8.6 of the Supplementary file S1) like this:

```
## Descriptor method
Method=NAC;ENAC;DNC;TNC;CKSNAP;Kmer;RCKmer;ANF;NCP;
binary;DAC;DCC;DACC;PseDNC;PseKNC;PCPseDNC;SCPseDNC
.............
## Model construction (SVM, RF, LR, KNN, MLP)
ML=RF
```

Then run the following command:

```
python iLearn-descriptor-estimater.py
```

All the combinations of the five machine-learning algorithms and 17 descriptors will be tested. Eighty-five different predictors were generated, whose results were shown in Supplementary Figure S4.

Ensemble learning

In some of the prediction task, to obtain the best performance, multiple descriptors and machine-learning algorithms can be integrated to build a combined predictor. Here, four descriptors (i.e. ENAC, NCP, binary and DACC) and all the five machine-learning algorithms were selected to construct the ensemble predictor. *iLearn* will evaluate and report the performance for all possible combinations of the selected machine-learning algorithms through a logistic regression approach and return the combination of algorithms that achieves the best performance by running the following command:

```
python iLearn-auto-pipline.py
```

For the four descriptors, their best performance is RF-DACC with an AUC value of 0.92 (Supplementary Figure S4). However, the AUC is 0.95 for the ensemble model (SVM + RF) with best performance. The corresponding result is shown in Supplementary Figures S4 and S5. These results indicate that *iLearn* is useful for developing new predictors for m^5C prediction, and some predictors generated by *iLearn* even outperformed the existing methods in this field.

Application of iLearn to address the protein malonylation site prediction problem

Protein malonylation is a new type of protein post-translational modification (PTM) and has been implicated in a variety of cellular processes and biological functions [67]. However, its precise roles associated with human diseases remain largely unknown. Accordingly, identification of malonylation sites in target substrates is an initial but crucial step toward elucidation of the molecular mechanisms underlying the regulation of protein malonylation [68]. Here, protein malonylation site prediction is used as an example to demonstrate the application and performance of *iLearn* for addressing sequence-based prediction problems.

The data set was retrieved from [68], which included a benchmark data set and an independent test data set. In the same publication, RF-based predictors were constructed based on different commonly used sequence encoding schemes and their predictive performance of malonylation sites evaluated. The encoding schemes tested include BLOSUM62, CKSAAP, Binary, Z-scales, AAindex, AAC and EAAC. This process was executed using the following command line in *iLearn*:

```
python iLearn-descriptor-estimater.py
```

The configuration file of the detailed descriptors used in *iLearn* is provided in Supplementary Figure S2. The ROC and PRC curves of the seven trained predictors on the benchmark data set and the independent data set are shown in Supplementary Figure S6A and B, respectively. As can be seen, the predictor on the EAAC encoding achieved the best

performance among all seven predictors. More specifically, the EAAC encoding achieved an AUC value of 0.73, which is comparable with the reported AUC value of 0.739 in the original work [68]. These results indicate that *iLearn* can be used as a cost-effective tool to develop predictors to address classification and labeling problems (such as prediction of PTM sites, functional sites and protein family classifications) from sequences information.

Discussion

Several web servers and stand-alone software packages have been developed to calculate various features from biological sequences. Compared with these, *iLearn* has the following advantages: (i) most of the existing web servers and stand-alone software packages can only handle one type of biological sequence, while *iLearn* can extract/calculate a variety of features for DNA, RNA and protein sequences, with the largest number of descriptors for each; (ii) instead of focusing on individual steps, *iLearn* was designed to automatically conduct all main steps for constructing a predictor; (iii) *iLearn* analyzes features by incorporating 16 types of clustering, selection, feature normalization or dimensionality reduction methods, and more abundant graphical display options for the feature analysis result; (iv) various output formats are supported by *iLearn*, so the output can be directly used and processed in other tools; and (v) *iLearn* allows direct construction and use of computational predictors based on machine-learning algorithms, all in one tool.

Conclusions

In this work, we have developed *iLearn*, a comprehensive, flexible and open-source Python toolkit integrating feature calculation, extraction, clustering, feature selection, dimensionality reduction, model construction, best model selection, ensemble learning for classification and result visualization for DNA, RNA and protein sequences. By using *iLearn*, the biological sequence analysis procedures described in Figure 1 can be easily implemented. Five commonly used machine-learning algorithms are supplied and underlying parameters can be automatically optimized for functionality of the predictor construction. We have extensively tested *iLearn* to guarantee correctness of computations. We also applied *iLearn* to train machine-learning models for two different prediction problems (i.e. RNA m^5C site prediction and protein malonylation site prediction) in order to illustrate its utility and performance. Moreover, we purposely designed *iLearn* to ensure workflow efficiency and user friendliness. We anticipate that *iLearn* will be used as a cost-effective and powerful tool that will significantly facilitate automated machine-learning-based modeling, as well as expedite data-driven biomedical knowledge in the post-genomic and data analytics era.

Availability

Availability: http://ilearn.erc.monash.edu/.

Key Points

- With the explosive growth of biological sequences generated in the post-genomic era, one of the most challenging problems in bioinformatics and computational biology is to computationally characterize sequences, structures and functions in an efficient, accurate and high-throughput manner.
- We propose *iLearn*, which is an integrated platform and meta-learner for feature engineering and machine-learning analysis and modeling of DNA, RNA and protein sequence data.
- Seven major steps, including feature extraction, clustering, selection, normalization, dimensionality reduction, predictor construction and result visualization for the DNA, RNA and protein, can be completed automatically based on the input data set in FASTA format.
- The web server and stand-alone program of *iLearn* are available. *iLearn* allows the integration of multiple descriptors or multiple machine-learning algorithms to achieve the overall best performance (i.e. through ensemble learning).
- Four output formats are supported by *iLearn*, so the output can be directly used and processed in other tools.

Funding

This work was supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (APP1127948, APP1144652 and APP490989), the Young Scientists Fund of the National Natural Science Foundation of China (31701142), the Australian Research Council (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), a Major Inter-Disciplinary Research project awarded by Monash University, and the Collaborative Research Program of Institute for Chemical Research, Kyoto University (2018-28). T.M.L. and A.L.'s work was supported in part by the Informatics Institute of the School of Medicine at the University of Alabama at Birmingham. R.J.D. and J.L. are NHMRC principal research fellows.

References

1. Toronen P, Medlar A, Holm L. PANNZER2: a rapid functional annotation web server. *Nucleic Acids Res* 2018;**46**:W84–8.
2. Liu B. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Brief Bioinform* 2017, doi: 10.1093/bib/bbx165.
3. Chou PY, Fasman GD. Prediction of the secondary structure of proteins from their amino acid sequence. *Adv Enzymol Relat Areas Mol Biol* 1978;**47**:45–148.
4. Chen J, Guo M, Wang X, *et al*. A comprehensive review and comparison of different computational methods for protein remote homology detection. *Brief Bioinform* 2018;**19**:231–44.
5. Yan K, Xu Y, Fang X, *et al*. Protein fold recognition based on sparse representation based classification. *Artif Intell Med* 2017;**79**:1–8.
6. Cao DS, Xiao N, Xu QS, *et al*. Rcpi: R/bioconductor package to generate various descriptors of proteins, compounds and their interactions. *Bioinformatics* 2015;**31**:279–81.
7. Shen J, Zhang J, Luo X, *et al*. Predicting protein–protein interactions based only on sequences information. *Proc Natl Acad Sci U S A* 2007;**104**:4337–41.

8. Chou KC, Shen HB. Cell-PLoc: a package of web servers for predicting subcellular localization of proteins in various organisms. *Nat Protoc* 2008;**3**:153–62.
9. Bhasin M, Raghava GP. Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J Biol Chem* 2004;**279**:23262–6.
10. Rottig M, Rausch C, Kohlbacher O. Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families. *PLoS Comput Biol* 2010;**6**:e1000636.
11. Song J, Tan H, Shen H, *et al*. Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics* 2010;**26**:752–60.
12. Chen Z, Zhou Y, Song J, *et al*. hCKSAAP_UbSite: improved prediction of human ubiquitination sites by exploiting amino acid pattern and properties. *Biochim Biophys Acta* 1834;**2013**:1461–7.
13. Chen Z, Zhou Y, Zhang Z, *et al*. Towards more accurate prediction of ubiquitination sites: a comprehensive review of current methods, tools and features. *Brief Bioinform* 2015;**16**:640–57.
14. Chen Z, Chen YZ, Wang XF, *et al*. Prediction of ubiquitination sites by using the composition of k-spaced amino acid pairs. *PLoS One* 2011;**6**:e22930.
15. Liu B, Fang L, Liu F, *et al*. Identification of real microRNA precursors with a pseudo structure status composition approach. *PLoS One* 2015;**10**:e0121501.
16. Liu B, Fang L, Long R, *et al*. iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics* 2016;**32**:362–9.
17. Chou CH, Shrestha S, Yang CD, *et al*. miRTarBase update 2018: a resource for experimentally validated microRNA-target interactions. *Nucleic Acids Res* 2018;**46**:D296–302.
18. Yan J, Friedrich S, Kurgan L. A comprehensive comparative review of sequence-based predictors of DNA- and RNA-binding residues. *Brief Bioinform* 2016;**17**:88–105.
19. Zhang J, Liu B. PSFM-DBT: identifying DNA-binding proteins by combing position specific frequency matrix and distance-bigram transformation. *Int J Mol Sci* 2017;**18**:1856.
20. Liu B, Wang S, Long R, *et al*. iRSpot-EL: identify recombination spots with an ensemble learning approach. *Bioinformatics* 2017;**33**:35–41.
21. Lin H, Deng EZ, Ding H, *et al*. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res* 2014;**42**:12961–72.
22. Guo SH, Deng EZ, Xu LQ, *et al*. iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics* 2014;**30**:1522–9.
23. Chen W, Feng PM, Lin H, *et al*. iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic Acids Res* 2013;**41**:e68.
24. Liu B, Liu F, Wang X, *et al*. Pse-in-one: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res* 2015;**43**: W65–71.
25. Liu B, Liu F, Fang L, *et al*. repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics* 2015;**31**:1307–9.
26. Song J, Li F, Takemoto K, *et al*. PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. *J Theor Biol* 2018;**443**:125–37.
27. Song J, Wang Y, Li F, *et al*. iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Brief Bioinform* 2018. doi: 10.1093/bib/bby028.
28. Chou KC. Some remarks on protein attribute prediction and pseudo amino acid composition. *J Theor Biol* 2011;**273**: 236–47.
29. Li ZR, Lin HH, Han LY, *et al*. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res* 2006;**34**:W32–7.
30. Rao HB, Zhu F, Yang GB, *et al*. Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res* 2011;**39**:W385–90.
31. Shen HB, Chou KC. PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. *Anal Biochem* 2008;**373**:386–8.
32. Cao DS, Xu QS, Liang YZ. propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics* 2013;**29**:960–2.
33. Du P, Gu S, Jiao Y. PseAAC-general: fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. *Int J Mol Sci* 2014;**15**:3495–506.
34. Xiao N, Cao DS, Zhu MF, *et al*. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics* 2015;**31**: 1857–9.
35. Zuo Y, Li Y, Chen Y, *et al*. PseKRAAC: a flexible web server for generating pseudo K-tuple reduced amino acids composition. *Bioinformatics* 2017;**33**:122–4.
36. Wang J, Yang B, Revote J, *et al*. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* 2017;**33**: 2756–8.
37. Chen Z, Zhao P, Li F, *et al*. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 2018;**34**: 2499–502.
38. Larranaga P, Calvo B, Santana R, *et al*. Machine learning in bioinformatics. *Brief Bioinform* 2006;**7**:86–112.
39. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. *Nat Rev Genet* 2015;**16**:321–32.
40. Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995;**20**:273–97.
41. Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32.
42. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 1943;**5**: 115–33.
43. Altman NS. An Introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 1992;**46**:175–85.
44. Freedman AD. Statistical models: theory and practice. *Technometrics* 2006;**48**:315.
45. Zhou Y, Zeng P, Li YH, *et al*. SRAMP: prediction of mammalian N6-methyladenosine (m6A) sites based on sequence-derived features. *Nucleic Acids Res* 2016;**44**:e91.
46. He W, Jia C, Duan Y *et al*. 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. *BMC Syst Biol* 2018;**12**:44.
47. He W, Jia C, Zou Q. 4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction. *Bioinformatics* 2018;**35**:593–601.

48. Jain AK, Murty MN, Flynn PJ. Data clustering: a review. *ACM Comput Surv* 1999;**31**:264–323.
49. Rokach L, Maimon O. Clustering methods. In: Maimon O, Rokach L (eds). *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2005, 321–52.
50. Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recognit Lett* 2010;**31**:651–66.
51. Cheng YZ. Mean shift, mode seeking, and clustering. *IEEE Trans Pattern Anal Mach Intell* 1995;**17**:790–9.
52. Ester M, Kriegel H-P, Sander J, *et al*. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon: AAAI Press, **1996**, pp. 226–31.
53. Frey BJ, Dueck D. Clustering by passing messages between data points. *Science* 2007;**315**:972–6.
54. Chou KC, Shen HB. Recent progress in protein subcellular location prediction. *Anal Biochem* 2007;**370**:1–16.
55. Lopez Y, Dehzangi A, Lal SP, *et al*. SucStruct: prediction of succinylated lysine residues by using structural properties of amino acids. *Anal Biochem* 2017;**527**: 24–32.
56. Liu B, Yang F, Huang DS, *et al*. iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics* 2018; **34**:33–40.
57. Li F, Li C, Marquez-Lago TT, *et al*. Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. *Bioinformatics* 2018;**34**:4223–31.
58. Feng PM, Chen W, Lin H, *et al*. iHSP-PseRAAAC: identifying the heat shock protein families using pseudo reduced amino acid alphabet composition. *Anal Biochem* 2013;**442**: 118–25.
59. Xuan JJ, Sun WJ, Lin PH, *et al*. RMBase v2.0: deciphering the map of RNA modifications from epitranscriptome sequencing data. *Nucleic Acids Res* 2018;**46**:D327–34.
60. 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.
61. David R, Burgess A, Parker B, *et al*. Transcriptome-wide mapping of RNA 5-Methylcytosine in Arabidopsis mRNAs and noncoding RNAs. *Plant Cell* 2017; **29**:445–60.
62. Agris PF. Bringing order to translation: the contributions of transfer RNA anticodon-domain modifications. *EMBO Rep* 2008;**9**:629–35.
63. Alexandrov A, Chernyakov I, Gu W, *et al*. Rapid tRNA decay can result from lack of nonessential modifications. *Mol Cell* 2006;**21**:87–96.
64. Motorin Y, Lyko F, Helm M. 5-methylcytosine in RNA: detection, enzymatic formation and biological functions. *Nucleic Acids Res* 2010;**38**:1415–30.
65. Zhang M, Xu Y, Li L, *et al*. Accurate RNA 5-methylcytosine site prediction based on heuristic physical–chemical properties reduction and classifier ensemble. *Anal Biochem* 2018;**550**: 41–8.
66. Song J, Zhai J, Bian E, *et al*. Transcriptome-wide annotation of m(5)C RNA modifications using machine learning. *Front Plant Sci* 2018;**9**:519.
67. Du Y, Cai T, Li T, *et al*. Lysine malonylation is elevated in type 2 diabetic mouse models and enriched in metabolic associated proteins. *Mol Cell Proteomics* 2015;**14**: 227–36.
68. Chen Z, He N, Huang Y, *et al*. Integration of a deep learning classifier with a random forest approach for predicting malonylation sites. *Genomics Proteomics Bioinformatics* 2018;**16**:451–9.