iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization

Zhen Chen[1,†], Pei Zhao[2,†], Chen Li[3,†], Fuyi Li[3,4,5], Dongxu Xiang[3,4], Yong-Zi Chen[6], Tatsuya Akutsu[7], Roger J. Daly[3], Geoffrey I. Webb[4], Quanzhi Zhao[1,8,*], Lukasz Kurgan[9,*] and Jiangning Song[3,4,*]

[1]Collaborative Innovation Center of Henan Grain Crops, Henan Agricultural University, Zhengzhou 450046, China, [2]State Key Laboratory of Cotton Biology, Institute of Cotton Research of Chinese Academy of Agricultural Sciences (CAAS), Anyang, 455000, China, [3]Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia, [4]Monash Centre for Data Science, Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia, [5]Department of Microbiology and Immunology, The Peter Doherty Institute for Infection and Immunity, The University of Melbourne, Melbourne, Victoria, 3000, Australia, [6]Laboratory of Tumor Cell Biology, Key Laboratory of Cancer Prevention and Therapy, National Clinical Research Center for Cancer, Tianjin Medical University Cancer Institute and Hospital, Tianjin Medical University, Tianjin 300060, China, [7]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan, [8]Key Laboratory of Rice Biology in Henan Province, Henan Agricultural University, Zhengzhou 450046, China, [9]Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA

[†]These authors contributed equally to this work.

[*]To whom the correspondence should be addressed. Tel: +61-3-9902-9304; Email: Jiangning.Song@monash.edu;

Correspondence may also be addressed to Quanzhi Zhao, Tel: +86-0371-56990209; Email: qzzhaoh@henau.edu.cn, and Lukasz Kurgan, Tel: +1-804-827-3986; Email: lkurgan@vcu.edu.

Table of Contents

1. Brief introduction

iLearnPlus is the first machine-learning platform with both graphical- and web-based user interface that enables the construction of automated machine-learning pipelines for computational analysis and predictions using nucleic acid and protein sequences. Four major modules, including *iLearnPlus-Basic*, *iLearnPlus-Estimator*, *iLearnPlus-AutoML*, and *iLearnPlus-LoadModel*, are provided in *iLearnPlus* for biologists and bioinformaticians to conduct customizable sequence-based feature engineering and analysis, machine-learning algorithm construction, performance assessment, statistical analysis, and data visualization, without additional programming. *iLearnPlus* integrates 21 machine-learning algorithms (including 12 conventional classification algorithms, two ensemble-learning frameworks and seven deep-learning approaches) and 19 major sequence encoding schemes (in total 152 feature descriptors), outnumbering all the current web servers and stand-alone tools for biological sequence analysis, to the best of our knowledge. In addition, the friendly GUI (Graphical User Interface) of *iLearnPlus* is available to biologists to conduct their analyses smoothly, significantly increasing the effectiveness and user experience compared to the existing pipelines. *iLearnPlus* is an open-source platform for academic purposes and is available at https://github.com/Superzchen/iLearnPlus/. The *iLearnPlus-Basic* module is online accessible at http://ilearnplus.erc.monash.edu/.

2. Installing and running *iLearnPlus*

Installation

iLearnPlus is an open-source Python-based toolkit, which operates in the Python environment (Python version 3.6 or above) and can run on multiple operating systems (e.g. Windows, Mac, and Linux). Prior to installing and running *iLearnPlus*, all the dependencies should be installed in the Python environment, including PyQt5, qdarkstyle, numpy (1.18.5), pandas (1.0.5), threading, sip, datetime, platform, pickle, copy, scikit-learn (0.23.1), math, scipy (1.5.0), collections, itertools, torch (≥1.3.1), lightgbm (2.3.1), xgboost (1.0.2), matplotlib (3.1.1), seaborn, joblib, warnings, random, multiprocessing, and time. For the sake of convenience, we strongly recommend users install the Anaconda Python environment in their local computers, which can be freely downloaded from https://www.anaconda.com/. The detailed steps of installing these dependencies are provided as follows:

Step 1. Download and install the anaconda platform:

Download from: https://www.anaconda.com/products/individual

Step 2. Install PyTorch:

Please refer to https://pytorch.org/get-started/locally/ for PyTorch installation.

Step 3. Install all other necessary software packages using the following commands:

```
pip3 install lightgbm
pip3 install xgboost
pip3 install qdarkstyle
…
```

Running

To run *iLearnPlus*, go to the installation folder of *iLearnPlus* and run the ‘iLearnPlus.py’ script as follows:

```
python ilearnplus.py
```

Once *iLearnPlus* has started, the interface will show as demonstrated in **Figure S1**.

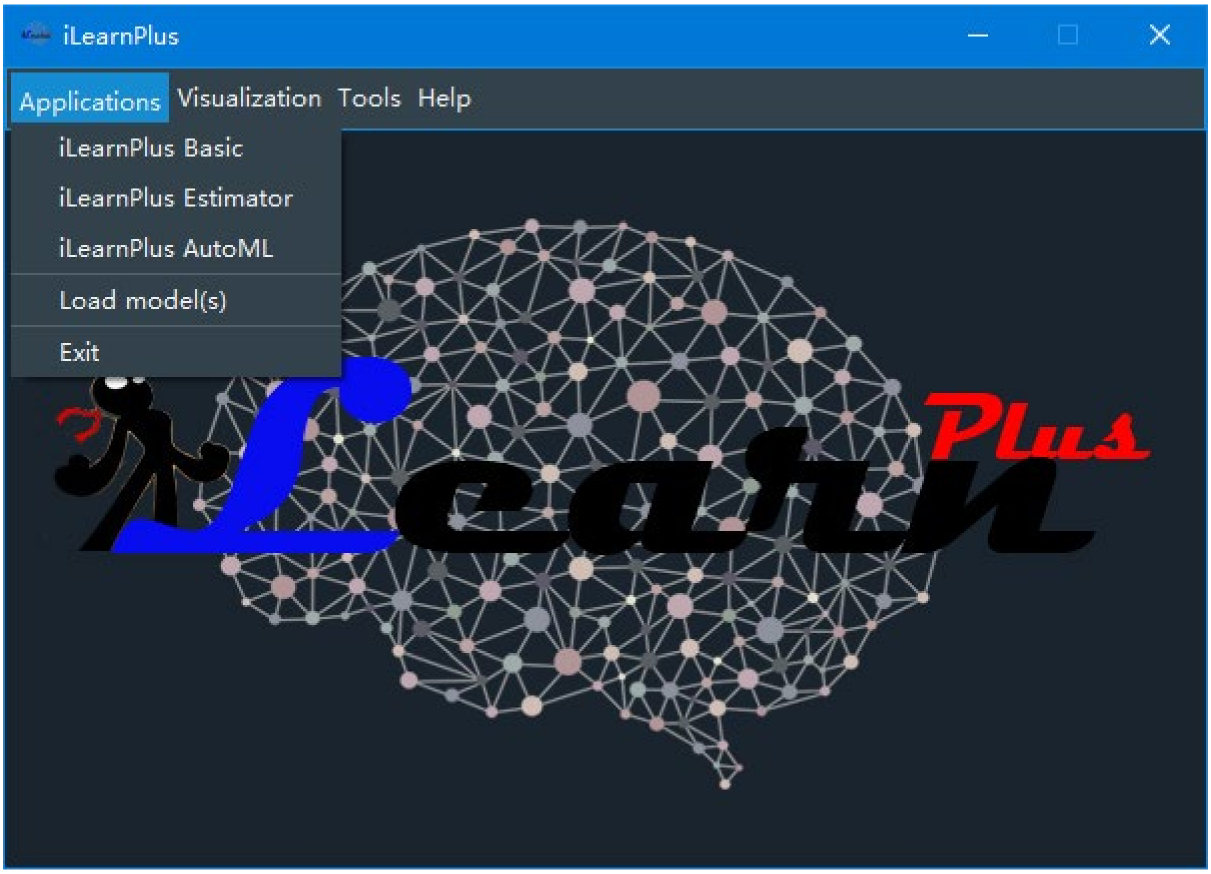


Figure S1. The main interface of the GUI version of *iLearnPlus*.

3. The workflow of *iLearnPlus*

Here we provide a step-by-step user instruction to demonstrate the workflow of *iLearnPlus* toolkit by running the examples provided in the "examples" directory. Five basic functions were designed and implemented in *iLearnPlus*, including feature extraction, feature analysis, predictor construction, and data/result visualization. Using these basic functions, four modules, including *iLearnPlus-Basic*, *iLearnPlus-Estimator* and *iLearnPlus-AutoML*, and *iLearnPlus-LoadModel* were further designed to facilitate sequence-based analysis and predictions on different levels of complexity (**Table S1, Figure S1**).

Table S1. Functions of the four major built-in modules in *iLearnPlus*.

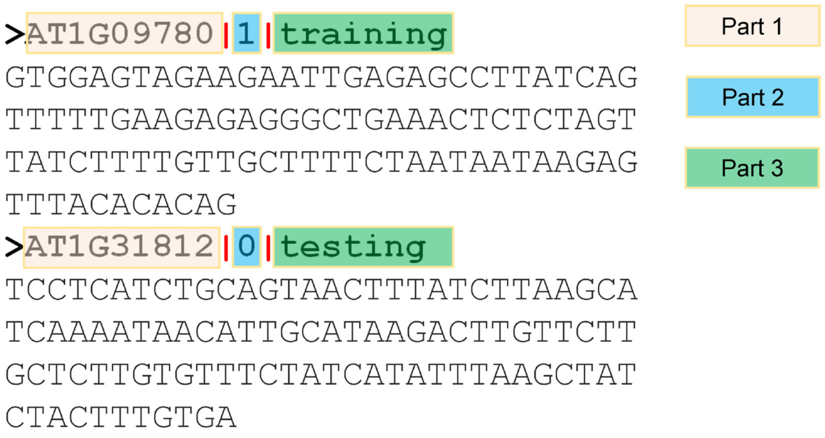
Modules	Function
iLearnPlus-Basic	1) Extraction 152 different types of feature descriptors for DNA, RNA and protein sequences. 2) 20 feature analysis algorithms (ten feature clustering, five feature selection, three dimensionality reduction, and two feature normalization algorithms).

	3)	21 machine-learning algorithms (12 conventional classification algorithms, two ensemble-learning frameworks and seven deep-learning approaches)
	4)	Data visualization (scatter plots for clustering and dimensionality reduction result, histogram and kernel density plot for data distribution, ROC and PRC for performance evaluation)
iLearnPlus-Estimator	1)	Estimation of the prediction ability for the selected descriptors by providing a more flexible way of feature extraction and calculation.
	2)	Data visualization (boxplot for the evaluation metrics of the *K*-fold cross-validation, heatmap for displaying the correlation or *p*-values matrix of the models, ROC and PRC curve for performance evaluation)
	3)	The bootstrap test and student's *t*-test were used to compare the prediction performance difference.
iLearnPlus-AutoML	1)	Automated performance benchmarking of different machine-learning algorithms based on the input features.
	2)	Data visualization (boxplot for the evaluation metrics of the *K*-fold cross-validation, heatmap for displaying the correlation or *p*-values matrix of the models, ROC and PRC curve for performance evaluation)
	3)	The bootstrap test and student's *t*-test were used to compare the prediction performance difference.
iLearnPlus-LoadModel	1)	Performing prediction using the generated models and testing dataset.

4. The input format of *iLearnPlus*

The input of *iLearnPlus* is a set of DNA, RNA or protein sequences in FASTA format with the specially designed header. The FASTA header consists of three parts: part 1, part 2 and part 3, which are separated by the symbol "|" (**Figure S2**). Part 1 is the sequence name while part 2 is the sample category information, which can be filled with any integer. For instance, users may use 1 to indicate the positive samples and 0 to represent the negative samples for a binary classification task, or use 0, 1, 2, … to represent different classes in multiclass classification tasks. Part 3 indicates the role of the sample, for example "training" would indicate that the corresponding sequence would be used as the training set for *K*-fold validation test, and "testing" indicates that the sequence would be used as the independent testing sample for independent testing.

For feature analysis and predictor construction, four file formats are supported, including LIBSVM (1) format, Comma-Separated Values (CSV), Tab Separated Values (TSV), and Waikato Environment for Knowledge Analysis (WEKA) (2) format. For LIBSVM, CSV and TSV format, the first column must be the sample label. Please find the "data" directory of the software for examples of these file formats.



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

Part 1
Part 2
Part 3

Figure S2. An example of the FASTA-formatted input DNA sequences used in *iLearnPlus* for feature descriptor extraction.

5. The *iLearnPlus-Basic* module

The *iLearnPlus-Basic* module aims at simply analysis and prediction using one protein/RNA/DNA descriptor and a machine-learning algorithm of choice. This module is particularly instrumental when interrogating the contributions of a certain type of sequence descriptor or a specific machine-learning algorithm to the prediction performance. All the five basic functions including feature extraction, feature analysis, predictor construction, and data/result visualization. can be implemented through the *iLearnPlus-Basic* module. There are four panels in *iLearnPlus-Basic* module. The “Descriptor” panel is used to extract feature descriptors for DNA, RNA and protein sequences, while the “Cluster / Dimensionality Reduction” and “Feature Normalization / Selection” panels are designed to implement the feature analysis algorithms, and the “Machine-learning” is used to build the prediction model.

Feature descriptor extraction

Each type of feature descriptor can be calculated using the “Descriptor” panel in the *iLearnPlus-Basic* module. Taking the DNA “DAC” descriptor as an example (**Figure S3**):

Step 1: Open the sequences file

Click the “Open” button in “Descriptor” panel and select the DNA sequences file (e.g. “DNA_sequences.txt” in “data” directory of *iLearnPlus* package). The biological sequence type (i.e. DNA, RNA or protein) will then be automatically detected based on the input sequences.

Step 2: Select the feature descriptor and configure the descriptor parameters

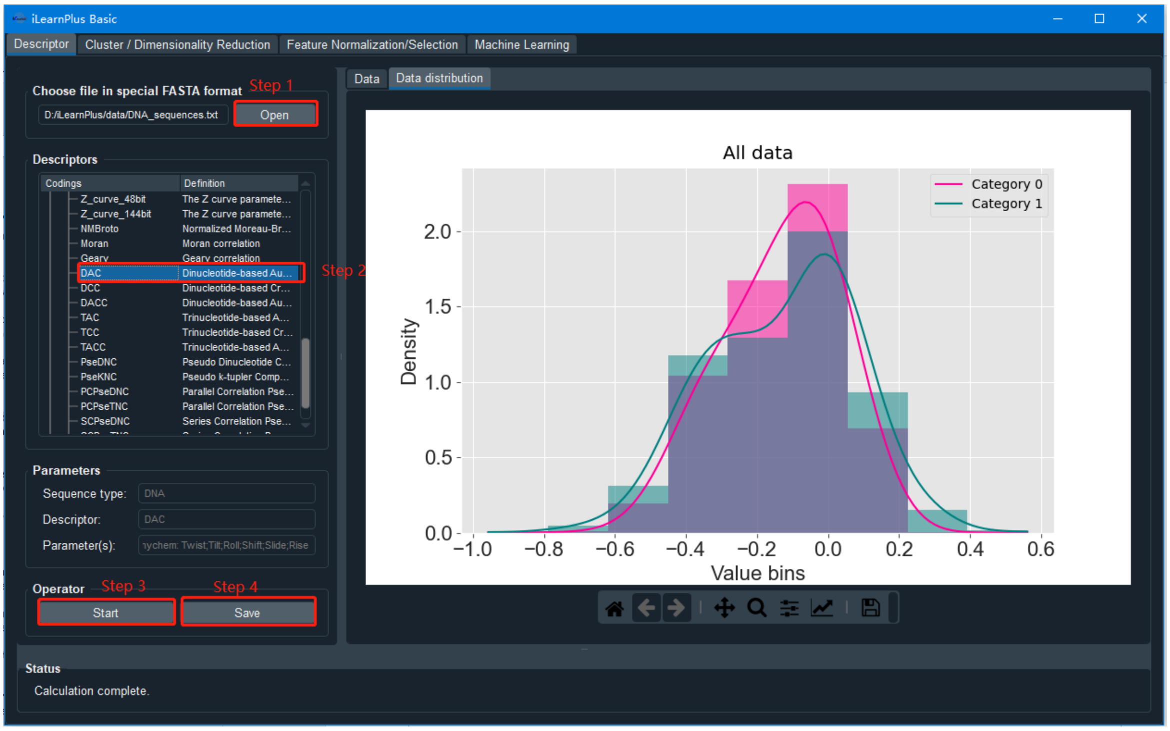
Click the “DAC” descriptor and set the corresponding parameters with the parameter dialog box (the default parameters were used here (**Figure S4**)). The information including sequence type, selected descriptor and the descriptor parameter(s) will be displayed in the “Parameters” area.

Step 3: Run the program

Click the “Start” button to calculate the descriptor features. The feature encoding and graphical presentation will be displayed in the “Data” and “Data distribution” panels, respectively. Here, we used the histogram and kernel density plot to display the distribution of the feature encoding.

Step 4: Save the results and image

Click the “Save” button to save the generated feature encodings. *iLearnPlus* supports four formats for saving the calculated features, including LIBSVM, CSV, TSV, and WEKA format, so as to facilitate direct use of the features in the following analysis, prediction model construction and the third-party computational tools, such as scikit-learn and WEKA. All the plots in *iLearnPlus* are generated by the matplotlib library and can be saved to a variety of image formats (e.g. PNG, JPG, PDF, TIFF etc).



Choose file in special FASTA format

D:/iLearnPlus/data/DNA_sequences.txt

Descriptors

Codings	Definition
Z_curve_48bit	The Z curve paramete...
Z_curve_144bit	The Z curve paramete...
NMBroto	Normalized Moreau-Br...
Moran	Moran correlation
Geary	Geary correlation
DAC	Dinucleotide-based Au...
DCC	Dinucleotide-based Cr...
DACC	Dinucleotide-based Au...
TAC	Trinucleotide-based A...
TCC	Trinucleotide-based Cr...
TACC	Trinucleotide-based A...
PseDNC	Pseudo Dinucleotide C...
PseKNC	Pseudo k-tupler Comp...
PCPseDNC	Parallel Correlation Pse...
PCPseTNC	Parallel Correlation Pse...
SCPseDNC	Series Correlation Pse...

Parameters

Sequence type: DNA

Descriptor: DAC

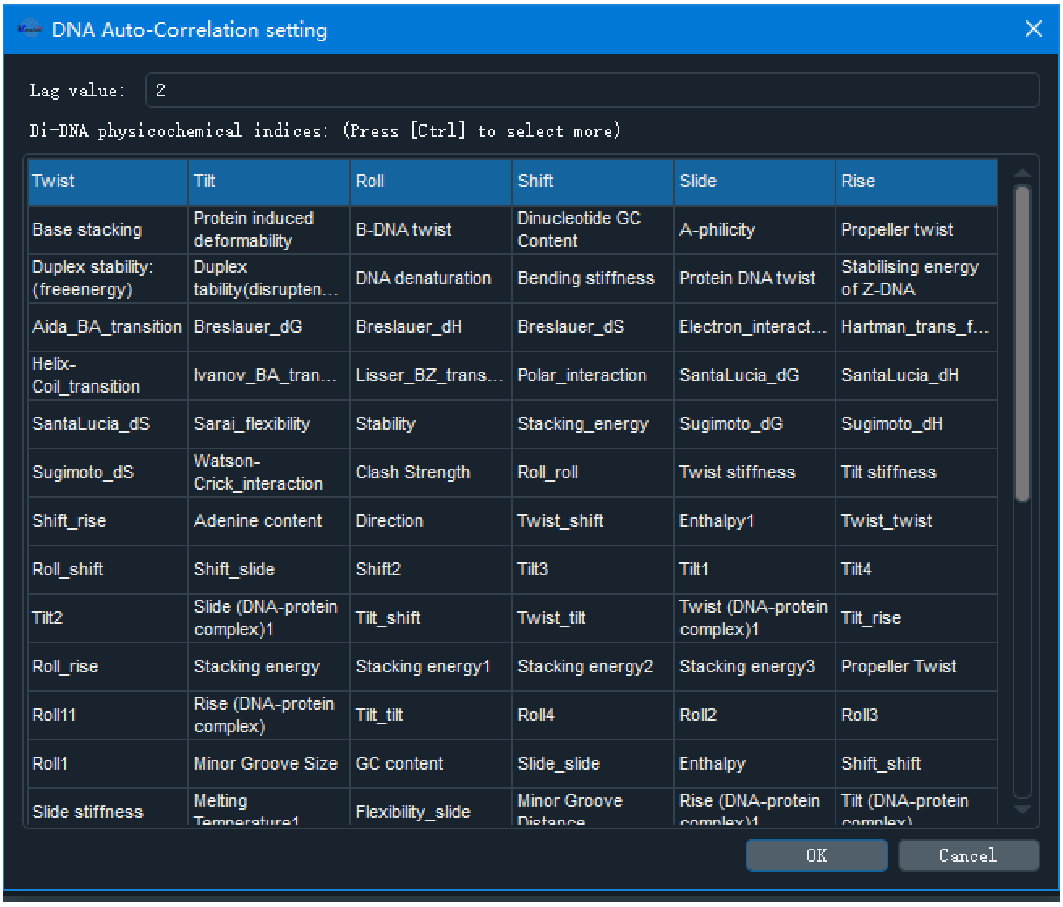
Parameter(s): ıychem: Twist;Tilt;Roll;Shift;Slide;Rise

Operator

Status

Calculation complete.

Figure S3. An example of extracting feature descriptor using the *iLearnPlus-Basic* module.



Lag value: 2

Di-DNA physicochemical indices: (Press [Ctrl] to select more)

Twist	Tilt	Roll	Shift	Slide	Rise
Base stacking	Protein induced deformability	B-DNA twist	Dinucleotide GC Content	A-philicity	Propeller twist
Duplex stability:(freeenergy)	Duplex tability(disrupten...	DNA denaturation	Bending stiffness	Protein DNA twist	Stabilising energy of Z-DNA
Aida_BA_transition	Breslauer_dG	Breslauer_dH	Breslauer_dS	Electron_interact...	Hartman_trans_f...
Helix-Coil_transition	Ivanov_BA_tran...	Lisser_BZ_trans...	Polar_interaction	SantaLucia_dG	SantaLucia_dH
SantaLucia_dS	Sarai_flexibility	Stability	Stacking_energy	Sugimoto_dG	Sugimoto_dH
Sugimoto_dS	Watson-Crick_interaction	Clash Strength	Roll_roll	Twist stiffness	Tilt stiffness
Shift_rise	Adenine content	Direction	Twist_shift	Enthalpy1	Twist_twist
Roll_shift	Shift_slide	Shift2	Tilt3	Tilt1	Tilt4
Tilt2	Slide (DNA-protein complex)1	Tilt_shift	Twist_tilt	Twist (DNA-protein complex)1	Tilt_rise
Roll_rise	Stacking energy	Stacking energy1	Stacking energy2	Stacking energy3	Propeller Twist
Roll11	Rise (DNA-protein complex)	Tilt_tilt	Roll4	Roll2	Roll3
Roll1	Minor Groove Size	GC content	Slide_slide	Enthalpy	Shift_shift
Slide stiffness	Melting Temperature1	Flexibility_slide	Minor Groove Distance	Rise (DNA-protein complex)1	Tilt (DNA-protein complex)

Figure S4. The dialog box for DNA DAC descriptor the parameter setting.

Feature analysis

iLearnPlus provides multiple options to facilitate feature analysis, including ten feature clustering, three dimensionality reduction, two feature normalization and five feature selection approaches (**Table 3** in our paper). In the *iLearnPlus-Basic* module, the "Cluster / Dimensionality Reduction" panel is used to deploy the clustering and dimensionality reduction algorithms; while the "Feature Normalization / Selection" panel is designed to implement the feature normalization and selection function. Taking the clustering as an example:

Step 1: Load data

There are two ways to load the data for analysis: 1) open a coding file and 2) select the data generated from other panels. Here we load the data from file. Click the "Open" button and open the "data.csv" in the "data" directory.

Step 2: Select the analysis algorithm and set the corresponding parameter(s)

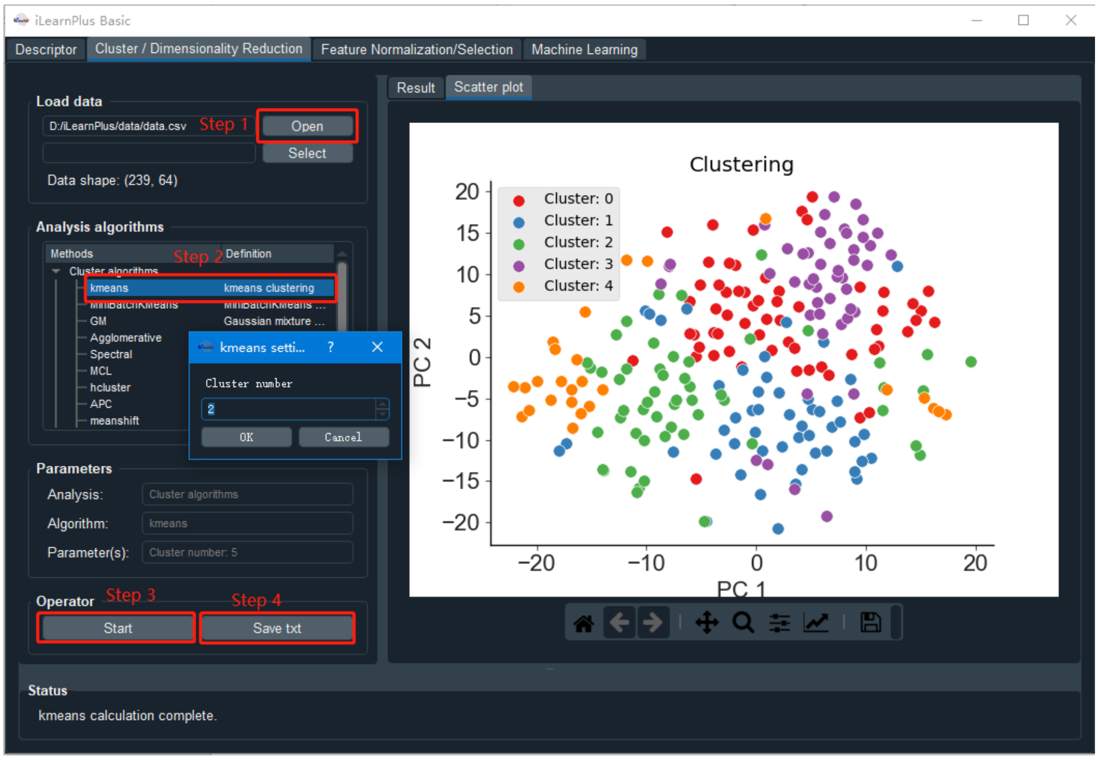
Select "kmeans" clustering algorithm and set the cluster number as 5.

Step 3: Run the program

Click the "Start" button to start the analysis progress. The clustering result and graphical presentation will be displayed in the "Result" and "Scatter plot" panels, respectively. Here, we used the scatter plot to display the clustering result.

Step 4: Save result and image

Click the "Save" button to save the generated clustering results.



D:/iLearnPlus/data/data.csv

Data shape: (239, 64)

Cluster number: 2

Analysis: Cluster algorithms

Algorithm: kmeans

Parameter(s): Cluster number: 5

kmeans calculation complete.

Figure S5. An example of implement the clustering algorithm using *iLearnPlus-Basic* module.

Predictor construction

iLearnPlus offers 12 conventional classification algorithms, two ensemble-learning frameworks, and seven deep-learning approaches. **Figure S6** shows the architectures of the deep-learning approaches. The implementation of these algorithms in *iLearnPlus* is based on four third-party machine-learning platforms, including scikit-learn (3), XGBoost (4), LightGBM (5), and PyTorch (6). Taking the CNN algorithm as the core machine-learning algorithm:

Step 1: Load data.

There are also two ways to load the data for analysis: 1) open a coding file and 2) select the data generated from other panels. Here we load the data from the “Descriptor” panel. At first, open the “m1A_DNA_sequences.txt” in the “data” directory and select the “binary” descriptor. Click “Start” button to calculate the feature encoding. Then, switch to the “Machine-learning” panel and load data by clicking the “Select” button (**Figure S7**). Select “Descriptor data” in the data selection dialog box and click “OK” button.

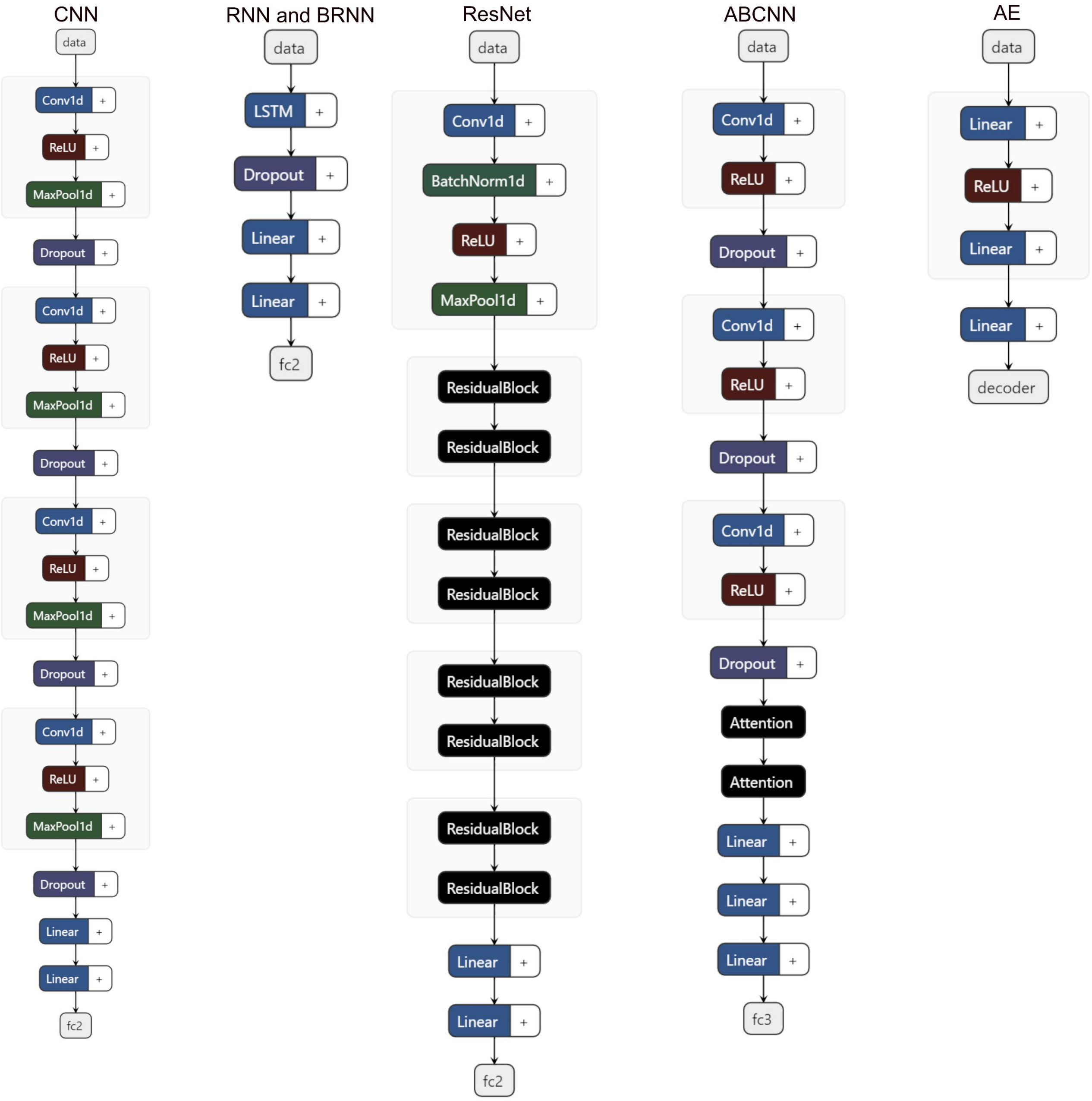
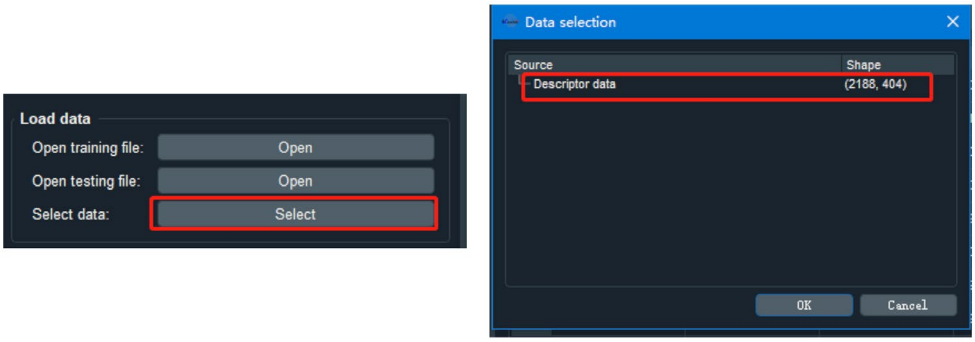


Figure S6. The architectures of the deep-learning approaches employed in *iLearnPlus*.

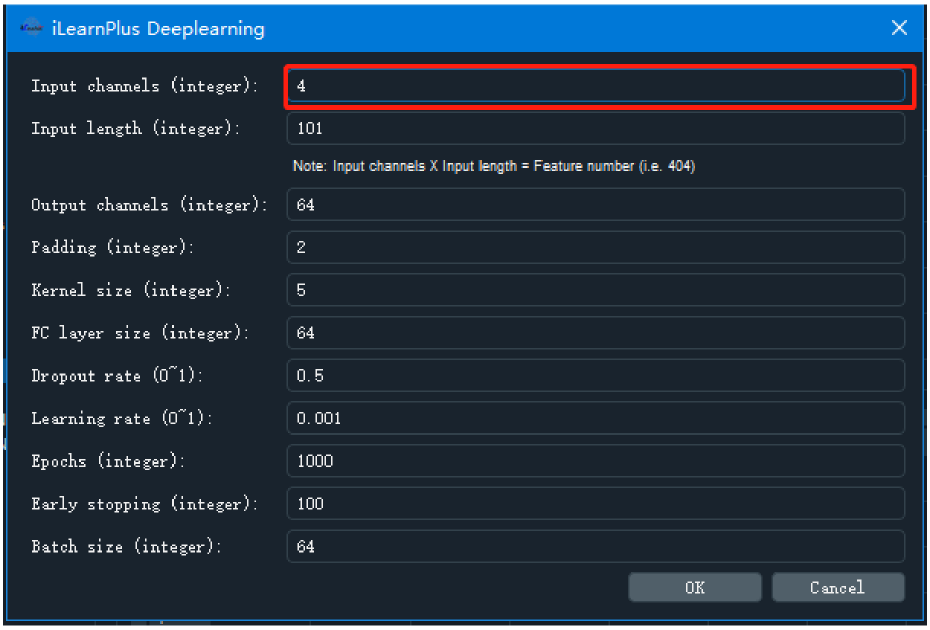


Source	Shape
Descriptor data	(2188, 404)

Figure S7. Loading data using the data selection explorer.

Step 2: Select the machine-learning algorithm and configure the corresponding parameter(s)

Select “Net_1_CNN” and set the “Input channels” as 4 (**Figure S8**). The default values were used for the remaining parameters.



Input channels (integer): 4

Input length (integer): 101

Note: Input channels X Input length = Feature number (i.e. 404)

Output channels (integer): 64

Padding (integer): 2

Kernel size (integer): 5

FC layer size (integer): 64

Dropout rate (0~1): 0.5

Learning rate (0~1): 0.001

Epochs (integer): 1000

Early stopping (integer): 100

Batch size (integer): 64

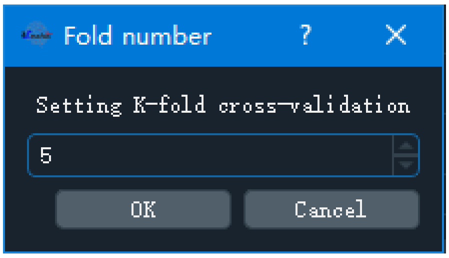
Figure S8. An example of parameter setting for the CNN algorithm.

Step 3: Set K-fold cross-validation

Set the *K* number as 5 (**Figure S9**).

Step 4: Run the program

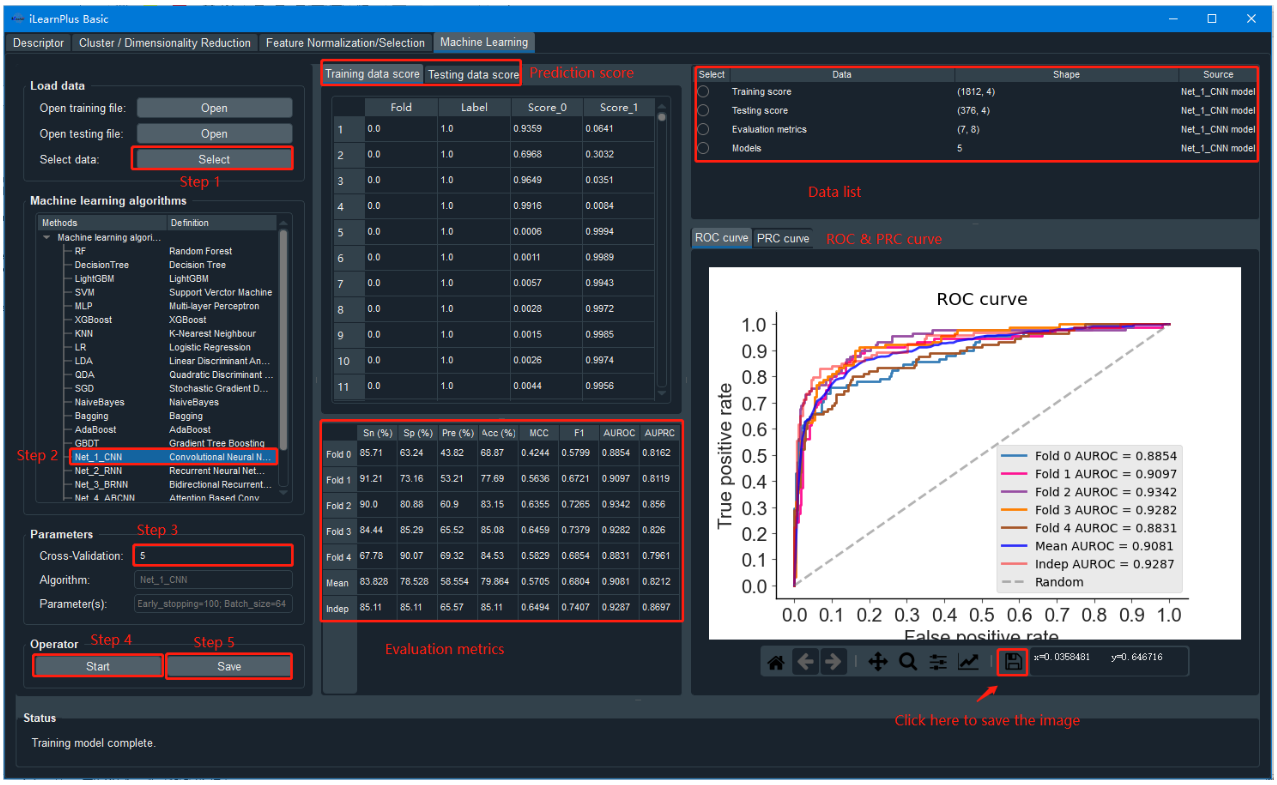
Click the “Start” button to start the analysis progress. The prediction score, evaluation metrics for *K*-fold cross-validation test, independent test, and the ROC and PRC curve will be displayed (**Figure S10**).



Setting K-fold cross-validation

5

Figure S9. Setting five-fold cross-validation.



Methods	Definition
Machine learning algori...	
RF	Random Forest
DecisionTree	Decision Tree
LightGBM	LightGBM
SVM	Support Verctor Machine
MLP	Multi-layer Perceptron
XGBoost	XGBoost
KNN	K-Nearest Neighbour
LR	Logistic Regression
LDA	Linear Discriminant An...
QDA	Quadratic Discriminant ...
SGD	Stochastic Gradient D...
NaiveBayes	NaiveBayes
Bagging	Bagging
AdaBoost	AdaBoost
GBDT	Gradient Tree Boosting
Net_1_CNN	Convolutional Neural N...
Net_2_RNN	Recurrent Neural Net...
Net_3_BRNN	Bidirectional Recurrent...
Net_4_ABCNN	Attention Based Conv

Cross-Validation: 5

Algorithm: Net_1_CNN

Parameter(s): Early_stopping=100; Batch_size=64

	Fold	Label	Score_0	Score_1
1	0.0	1.0	0.9359	0.0641
2	0.0	1.0	0.6968	0.3032
3	0.0	1.0	0.9649	0.0351
4	0.0	1.0	0.9916	0.0084
5	0.0	1.0	0.0006	0.9994
6	0.0	1.0	0.0011	0.9989
7	0.0	1.0	0.0057	0.9943
8	0.0	1.0	0.0028	0.9972
9	0.0	1.0	0.0015	0.9985
10	0.0	1.0	0.0026	0.9974
11	0.0	1.0	0.0044	0.9956

	Sn (%)	Sp (%)	Pre (%)	Acc (%)	MCC	F1	AUROC	AUPRC
Fold 0	85.71	63.24	43.82	68.87	0.4244	0.5799	0.8854	0.8162
Fold 1	91.21	73.16	53.21	77.69	0.5636	0.6721	0.9097	0.8119
Fold 2	90.0	80.88	60.9	83.15	0.6355	0.7265	0.9342	0.856
Fold 3	84.44	85.29	65.52	85.08	0.6459	0.7379	0.9282	0.826
Fold 4	67.78	90.07	69.32	84.53	0.5829	0.6854	0.8831	0.7961
Mean	83.828	78.528	58.554	79.864	0.5705	0.6804	0.9081	0.8212
Indep	85.11	85.11	65.57	85.11	0.6494	0.7407	0.9287	0.8697

Select	Data	Shape	Source
○	Training score	(1812, 4)	Net_1_CNN model
○	Testing score	(376, 4)	Net_1_CNN model
○	Evaluation metrics	(7, 8)	Net_1_CNN model
○	Models	5	Net_1_CNN model

Training model complete.

Figure S10. An example of model construction using CNN algorithm in *iLearnPlus*.

Building machine-learning pipelines

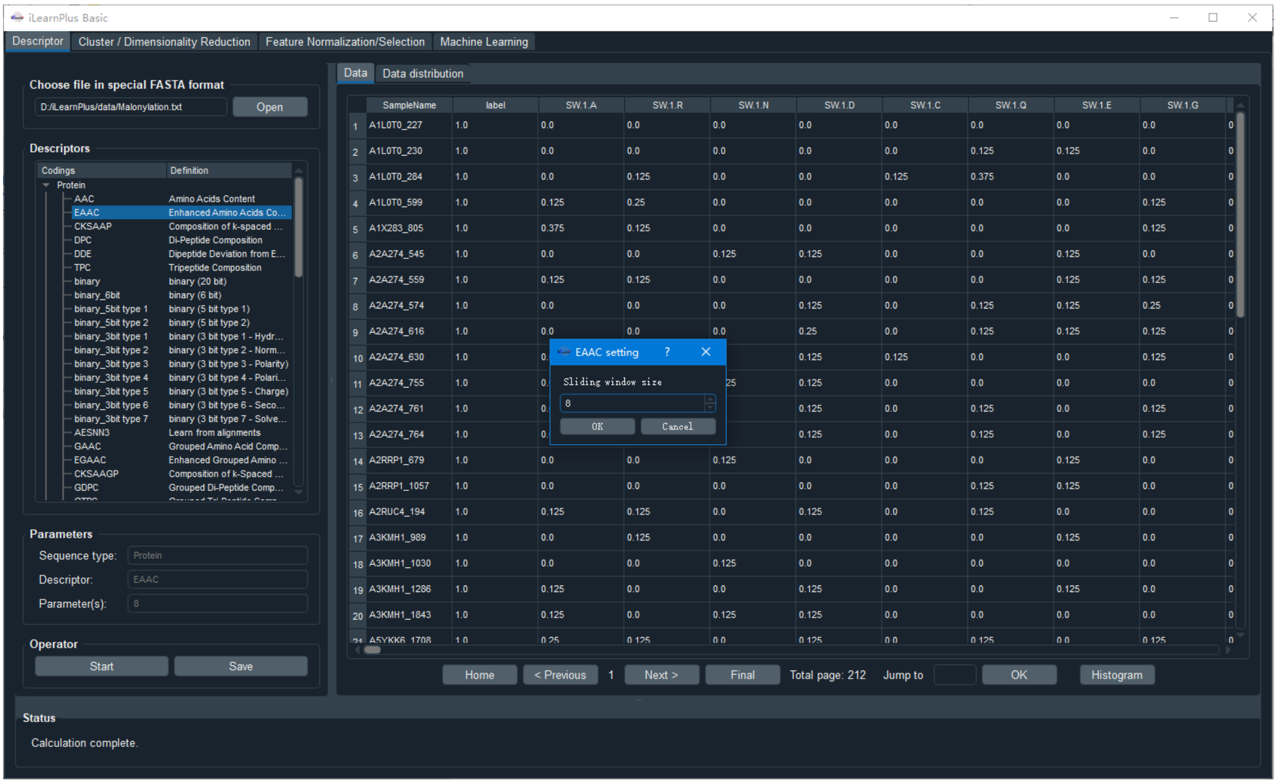
Usually, more than one individual functionality will be used in biological sequence analysis. In this case, the *iLearnPlus-Basic* module allows users to build their own machine-learning pipelines. The output data generated in the previous panel can be used as the input for next panel by using the graphical data selection explorer. Here, we take the malonylation site prediction as an example:

Step 1: Extract the descriptor using the iLearnPlus-Basic panel

In "Descriptor" panel, open the "Malonylation.txt" in "data" directory, select the ENAC descriptor, and set the sliding window size as 8. Click "Start" button to calculate the descriptor (**Figure S11**).

Step 2: Select the top 100 features

Switch to the "Feature Normalization / Selection" panel and load the data which has been generated by the "Descriptor" panel, we can see that the data shape of the generated data is (10578, 480), indicating that there are 10578 samples and the dimension for the feature vector is 480. We used the CHI2 algorithm to select the top 100 features (**Figure S12**).



Choose file in special FASTA format: D:/iLearnPlus/data/Malonylation.txt

Descriptors

Codings	Definition
Protein	
AAC	Amino Acids Content
EAAC	Enhanced Amino Acids Co...
CKSAAP	Composition of k-spaced ...
DPC	Di-Peptide Composition
DDE	Dipeptide Deviation from E...
TPC	Tripeptide Composition
binary	binary (20 bit)
binary_6bit	binary (6 bit)
binary_5bit type 1	binary (5 bit type 1)
binary_5bit type 2	binary (5 bit type 2)
binary_3bit type 1	binary (3 bit type 1 - Hydr...
binary_3bit type 2	binary (3 bit type 2 - Norm...
binary_3bit type 3	binary (3 bit type 3 - Polarity)
binary_3bit type 4	binary (3 bit type 4 - Polari...
binary_3bit type 5	binary (3 bit type 5 - Charge)
binary_3bit type 6	binary (3 bit type 6 - Seco...
binary_3bit type 7	binary (3 bit type 7 - Solve...
AESNN3	Learn from alignments
GAAC	Grouped Amino Acid Comp...
EGAAC	Enhanced Grouped Amino ...
CKSAAGP	Composition of k-Spaced ...
GDPC	Grouped Di-Peptide Comp...

Parameters

Sequence type: Protein

Descriptor: EAAC

Parameter(s): 8

Operator

	SampleName	label	SW.1.A	SW.1.R	SW.1.N	SW.1.D	SW.1.C	SW.1.Q	SW.1.E	SW.1.G	
1	A1L0T0_227	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	A1L0T0_230	1.0	0.0	0.0	0.0	0.0	0.0	0.125	0.125	0.0	0
3	A1L0T0_284	1.0	0.0	0.125	0.0	0.0	0.125	0.375	0.0	0.0	0
4	A1L0T0_599	1.0	0.125	0.25	0.0	0.0	0.0	0.0	0.0	0.125	0
5	A1X283_805	1.0	0.375	0.125	0.0	0.0	0.0	0.0	0.0	0.125	0
6	A2A274_545	1.0	0.0	0.0	0.125	0.125	0.0	0.0	0.125	0.0	0
7	A2A274_559	1.0	0.125	0.125	0.0	0.0	0.0	0.0	0.125	0.125	0
8	A2A274_574	1.0	0.0	0.0	0.0	0.125	0.0	0.125	0.125	0.25	0
9	A2A274_616	1.0	0.0	0.0	0.0	0.25	0.0	0.125	0.125	0.125	0
10	A2A274_630	1.0	[illegible]	[illegible]	[illegible]	0.125	0.125	0.0	0.0	0.125	0
11	A2A274_755	1.0	[illegible]	[illegible]	[illegible]	0.125	0.0	0.0	0.0	0.0	0
12	A2A274_761	1.0	[illegible]	[illegible]	[illegible]	0.125	0.0	0.125	0.0	0.125	0
13	A2A274_764	1.0	[illegible]	[illegible]	[illegible]	0.125	0.0	0.125	0.0	0.125	0
14	A2RRP1_679	1.0	0.0	0.0	0.125	0.0	0.0	0.0	0.125	0.0	0
15	A2RRP1_1057	1.0	0.0	0.0	0.0	0.0	0.0	0.125	0.125	0.0	0
16	A2RUC4_194	1.0	0.125	0.125	0.0	0.125	0.0	0.125	0.0	0.0	0
17	A3KMH1_989	1.0	0.0	0.125	0.0	0.0	0.0	0.0	0.125	0.0	0
18	A3KMH1_1030	1.0	0.0	0.0	0.125	0.0	0.0	0.0	0.0	0.0	0
19	A3KMH1_1286	1.0	0.125	0.0	0.0	0.125	0.0	0.0	0.125	0.0	0
20	A3KMH1_1843	1.0	0.125	0.0	0.125	0.125	0.0	0.0	0.0	0.0	0
21	A5YKK6_1708	1.0	0.25	0.125	0.0	0.125	0.0	0.125	0.0	0.125	0

Total page: 212 Jump to

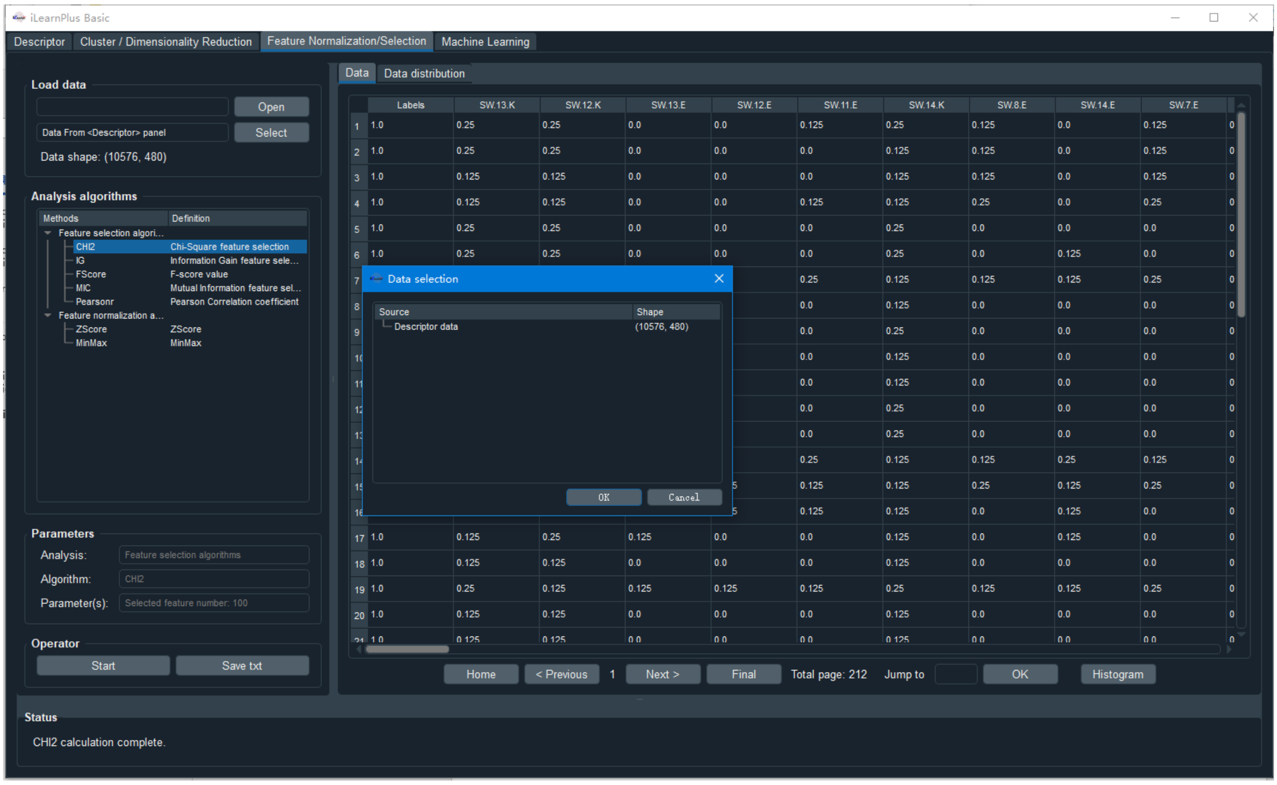
EAAC setting

Sliding window size: 8

Status

Calculation complete.

Figure S11. Extracting the EAAC descriptor using "Descriptor" panel in the *iLearnPlus-Basic* module.



Load data

Data From <Descriptor> panel

Data shape: (10576, 480)

Analysis algorithms

Methods	Definition
Feature selection algori...	
CHI2	Chi-Square feature selection
IG	Information Gain feature sele...
FScore	F-score value
MIC	Mutual Information feature sel...
Pearsonr	Pearson Correlation coefficient
Feature normalization a...	
ZScore	ZScore
MinMax	MinMax

Parameters

Analysis: Feature selection algorithms

Algorithm: CHI2

Parameter(s): Selected feature number: 100

Operator

	Labels	SW.13.K	SW.12.K	SW.13.E	SW.12.E	SW.11.E	SW.14.K	SW.8.E	SW.14.E	SW.7.E	
1	1.0	0.25	0.25	0.0	0.0	0.125	0.25	0.125	0.0	0.125	0
2	1.0	0.25	0.25	0.0	0.0	0.0	0.125	0.125	0.0	0.125	0
3	1.0	0.125	0.125	0.0	0.0	0.0	0.125	0.125	0.0	0.125	0
4	1.0	0.125	0.125	0.0	0.0	0.125	0.125	0.25	0.0	0.25	0
5	1.0	0.25	0.25	0.0	0.0	0.0	0.25	0.0	0.0	0.0	0
6	1.0	0.25	0.25	0.0	0.0	0.0	0.25	0.0	0.125	0.0	0
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.25	0.125	0.125	0.125	0.25	0
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.125	0.0	0.0	0.0	0
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.25	0.0	0.0	0.0	0
10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.125	0.0	0.0	0.0	0
11	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.125	0.0	0.0	0.0	0
12	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.25	0.0	0.0	0.0	0
13	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0	0.25	0.0	0.0	0.0	0
14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.25	0.125	0.125	0.25	0.125	0
15	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.125	0.125	0.25	0.125	0.25	0
16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.125	0.125	0.0	0.125	0.0	0
17	1.0	0.125	0.25	0.125	0.0	0.0	0.125	0.0	0.125	0.0	0
18	1.0	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.0	0.0	0
19	1.0	0.25	0.125	0.125	0.125	0.125	0.25	0.125	0.125	0.25	0
20	1.0	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.0	0.0	0
21	1.0	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.0	0.0	0

Total page: 212 Jump to

Data selection

Source	Shape
Descriptor data	(10576, 480)

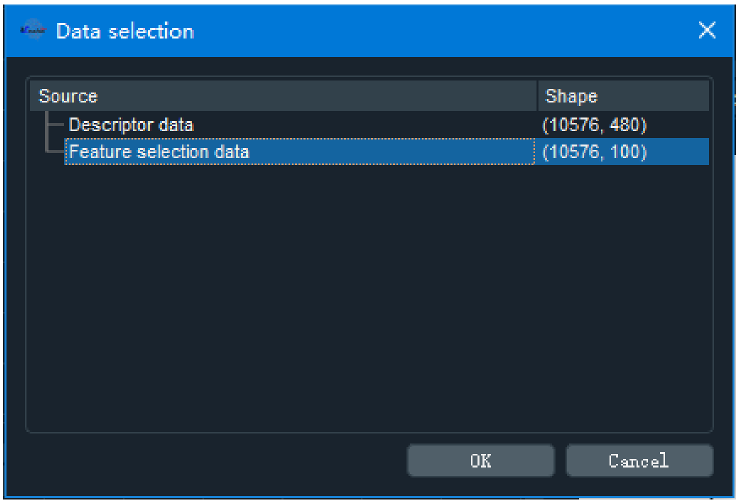
Status

CHI2 calculation complete.

Figure S12. Selecting the top 100 features using "Feature Normalization/Selection" panel in the *iLearnPlus-Basic* module.

Step 3: Build the prediction model using the selected features

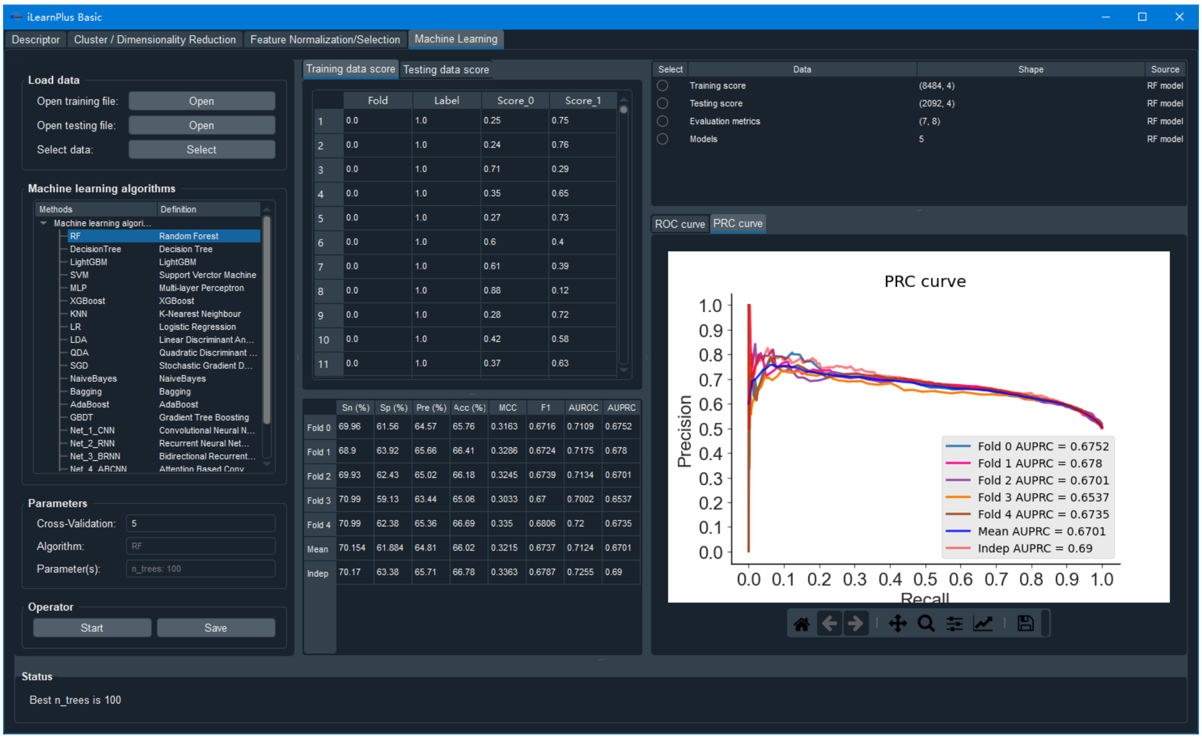
Switch to the “Machine-learning” panel, load the data from “Feature selection data” using the data selection dialog box in the “Machine-learning” panel and the data shape is (10576, 100) (**Figure S13**). Select the RF algorithm and use the default parameters to build the prediction model (**Figure S14**).



Data selection

Source	Shape
Descriptor data	(10576, 480)
Feature selection data	(10576, 100)

Figure S13. An example of loading data with the data selection dialog box.



iLearnPlus Basic

Descriptor
Cluster / Dimensionality Reduction
Feature Normalization/Selection
Machine Learning

Load data

Open training file: Open

Open testing file: Open

Select data: Select

Machine learning algorithms

Methods	Definition
Machine learning algori...	
RF	Random Forest
DecisionTree	Decision Tree
LightGBM	LightGBM
SVM	Support Verctor Machine
MLP	Multi-layer Perceptron
XGBoost	XGBoost
KNN	K-Nearest Neighbour
LR	Logistic Regression
LDA	Linear Discriminant An...
QDA	Quadratic Discriminant ...
SGD	Stochastic Gradient D...
NaiveBayes	NaiveBayes
Bagging	Bagging
AdaBoost	AdaBoost
GBDT	Gradient Tree Boosting
Net_1_CNN	Convolutional Neural N...
Net_2_RNN	Recurrent Neural Net...
Net_3_BRNN	Bidirectional Recurrent...
Net_4_ABCNN	Attention Based Conv

Parameters

Cross-Validation: 5

Algorithm: RF

Parameter(s): n_trees: 100

Operator

Status

Best n_trees is 100

	Fold	Label	Score_0	Score_1
1	0.0	1.0	0.25	0.75
2	0.0	1.0	0.24	0.76
3	0.0	1.0	0.71	0.29
4	0.0	1.0	0.35	0.65
5	0.0	1.0	0.27	0.73
6	0.0	1.0	0.6	0.4
7	0.0	1.0	0.61	0.39
8	0.0	1.0	0.88	0.12
9	0.0	1.0	0.28	0.72
10	0.0	1.0	0.42	0.58
11	0.0	1.0	0.37	0.63

	Sn (%)	Sp (%)	Pre (%)	Acc (%)	MCC	F1	AUROC	AUPRC
Fold 0	69.96	61.56	64.57	65.76	0.3163	0.6716	0.7109	0.6752
Fold 1	68.9	63.92	65.66	66.41	0.3286	0.6724	0.7175	0.678
Fold 2	69.93	62.43	65.02	66.18	0.3245	0.6739	0.7134	0.6701
Fold 3	70.99	59.13	63.44	65.06	0.3033	0.67	0.7002	0.6537
Fold 4	70.99	62.38	65.36	66.69	0.335	0.6806	0.72	0.6735
Mean	70.154	61.884	64.81	66.02	0.3215	0.6737	0.7124	0.6701
Indep	70.17	63.38	65.71	66.78	0.3363	0.6787	0.7255	0.69

Select	Data	Shape	Source
○	Training score	(8484, 4)	RF model
○	Testing score	(2092, 4)	RF model
○	Evaluation metrics	(7, 8)	RF model
○	Models	5	RF model

Figure S14. An example of prediction model construction using RF algorithm in the *iLearnPlus-basic* module.

6. The *iLearnPlus-Estimator* module

The *iLearnPlus-Estimator* module provides a more flexible way of feature extraction and calculation by allowing users to select multiple feature descriptors of interest. For a prediction task, the *iLearnPlus-Estimator* module can select out the descriptor with best performance. Here, we take the m^{1}A site prediction as an example:

Step 1: Load sequence data

Open the “m1A_DNA_sequences.txt” file in “data” directory of the *iLearnPlus* package.

Step 2: Select the descriptors

Here, nine descriptors, including Kmer, NAC, DNC, ANF, ENAC, binary, KNN, Z_curve_9bit and MMI, were selected to evaluate their performance. The default parameters for the descriptors were used.

Step 3: Select machine-learning algorithm

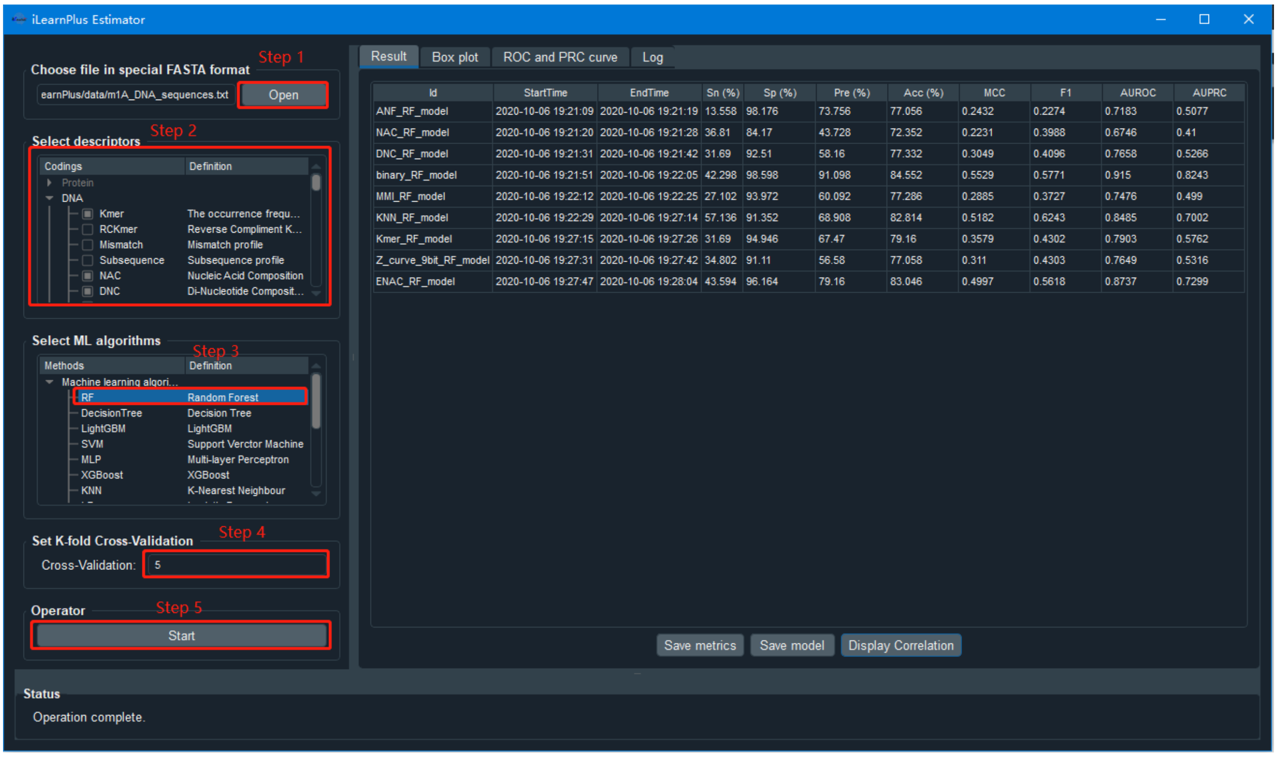
The RF algorithm was selected to build prediction models, and the tree number was set as 1000.

Step 4: Set K-fold cross-validation

Set the *K* as 5.

Step 5: Runn the program

Click “Start” button to train the models. For each of the selected feature descriptors, the program will extract the feature encoding and build the prediction model automatically one by one.



Choose file in special FASTA format

earnPlus/data/m1A_DNA_sequences.txt

Select descriptors

Codings	Definition
Protein	
DNA	
☒ Kmer	The occurrence frequ...
□ RCKmer	Reverse Compliment K...
□ Mismatch	Mismatch profile
□ Subsequence	Subsequence profile
☒ NAC	Nucleic Acid Composition
☒ DNC	Di-Nucleotide Composit...

Select ML algorithms

Methods	Definition
Machine learning algori...	
RF	Random Forest
DecisionTree	Decision Tree
LightGBM	LightGBM
SVM	Support Verctor Machine
MLP	Multi-layer Perceptron
XGBoost	XGBoost
KNN	K-Nearest Neighbour

Set K-fold Cross-Validation

Cross-Validation: 5

Operator

Id	StartTime	EndTime	Sn (%)	Sp (%)	Pre (%)	Acc (%)	MCC	F1	AUROC	AUPRC
ANF_RF_model	2020-10-06 19:21:09	2020-10-06 19:21:19	13.558	98.176	73.756	77.056	0.2432	0.2274	0.7183	0.5077
NAC_RF_model	2020-10-06 19:21:20	2020-10-06 19:21:28	36.81	84.17	43.728	72.352	0.2231	0.3988	0.6746	0.41
DNC_RF_model	2020-10-06 19:21:31	2020-10-06 19:21:42	31.69	92.51	58.16	77.332	0.3049	0.4096	0.7658	0.5266
binary_RF_model	2020-10-06 19:21:51	2020-10-06 19:22:05	42.298	98.598	91.098	84.552	0.5529	0.5771	0.915	0.8243
MMI_RF_model	2020-10-06 19:22:12	2020-10-06 19:22:25	27.102	93.972	60.092	77.286	0.2885	0.3727	0.7476	0.499
KNN_RF_model	2020-10-06 19:22:29	2020-10-06 19:27:14	57.136	91.352	68.908	82.814	0.5182	0.6243	0.8485	0.7002
Kmer_RF_model	2020-10-06 19:27:15	2020-10-06 19:27:26	31.69	94.946	67.47	79.16	0.3579	0.4302	0.7903	0.5762
Z_curve_9bit_RF_model	2020-10-06 19:27:31	2020-10-06 19:27:42	34.802	91.11	56.58	77.058	0.311	0.4303	0.7649	0.5316
ENAC_RF_model	2020-10-06 19:27:47	2020-10-06 19:28:04	43.594	96.164	79.16	83.046	0.4997	0.5618	0.8737	0.7299

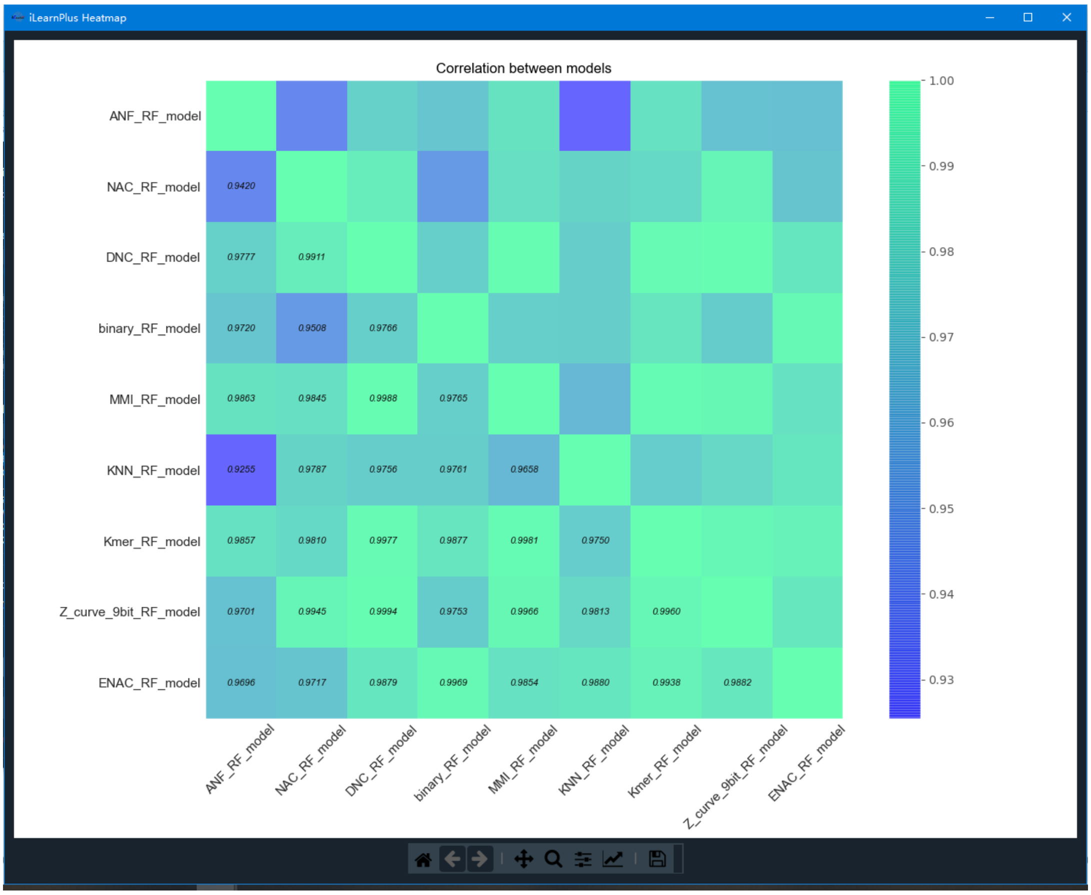
Status

Operation complete.

Figure S15. The panel of the *iLearnPlus-Estimator* module.

Step 6: Display prediction results

1) The evaluation metrics for the nine classifiers were displayed in the table widget (**Figure S15**).
2) The correlation matrix of the nine classifiers was displayed in the form of heatmap (**Figure S16**).
3) Boxplot for the evaluation metrics (**Figure S17**).
4) ROC and PRC curves (**Figure S18**).



	ANF_RF_model	NAC_RF_model	DNC_RF_model	binary_RF_model	MMI_RF_model	KNN_RF_model	Kmer_RF_model	Z_curve_9bit_RF_model	ENAC_RF_model
ANF_RF_model									
NAC_RF_model	0.9420								
DNC_RF_model	0.9777	0.9911							
binary_RF_model	0.9720	0.9508	0.9766						
MMI_RF_model	0.9863	0.9845	0.9988	0.9765					
KNN_RF_model	0.9255	0.9787	0.9756	0.9761	0.9658				
Kmer_RF_model	0.9857	0.9810	0.9977	0.9877	0.9981	0.9750			
Z_curve_9bit_RF_model	0.9701	0.9945	0.9994	0.9753	0.9966	0.9813	0.9960		
ENAC_RF_model	0.9696	0.9717	0.9879	0.9969	0.9854	0.9880	0.9938	0.9882	

Figure S16. The correlation matrix generated by the *iLearnPlus-Estimator* module.

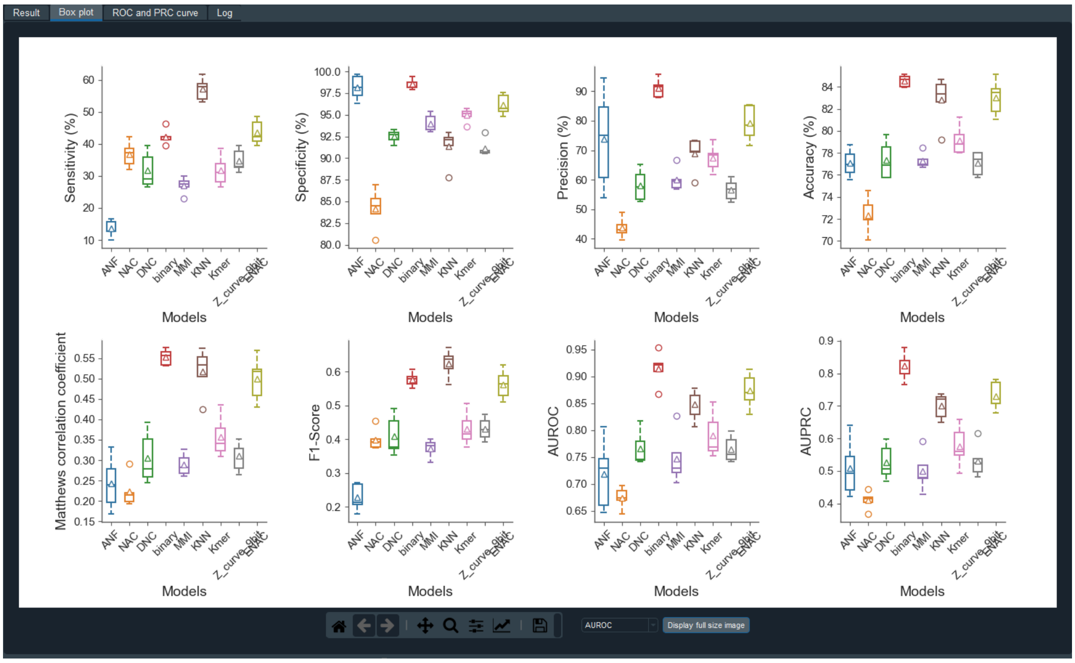


Figure S17. The boxplot generated by the *iLearnPlus-Estimator* module.

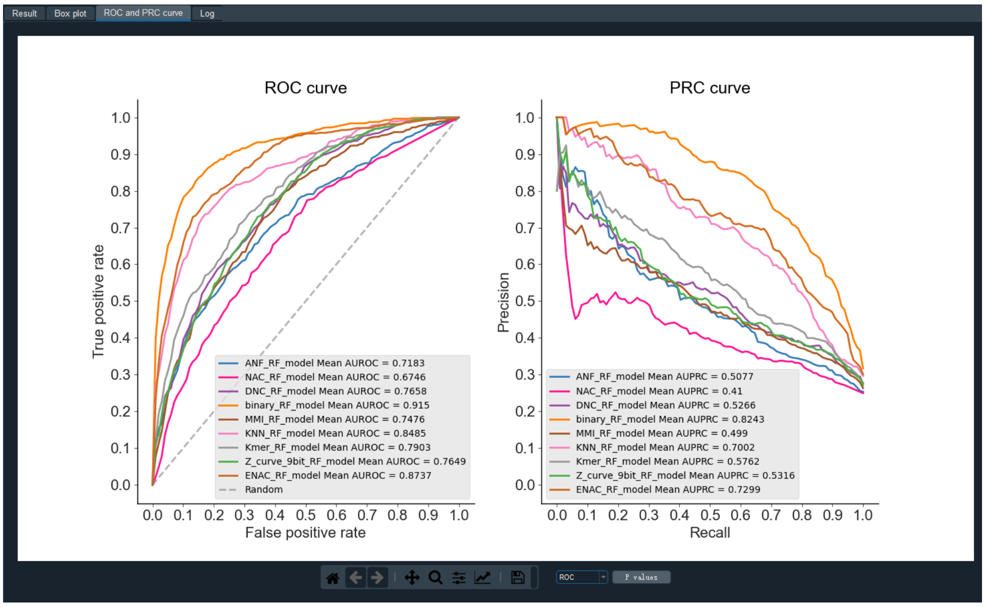
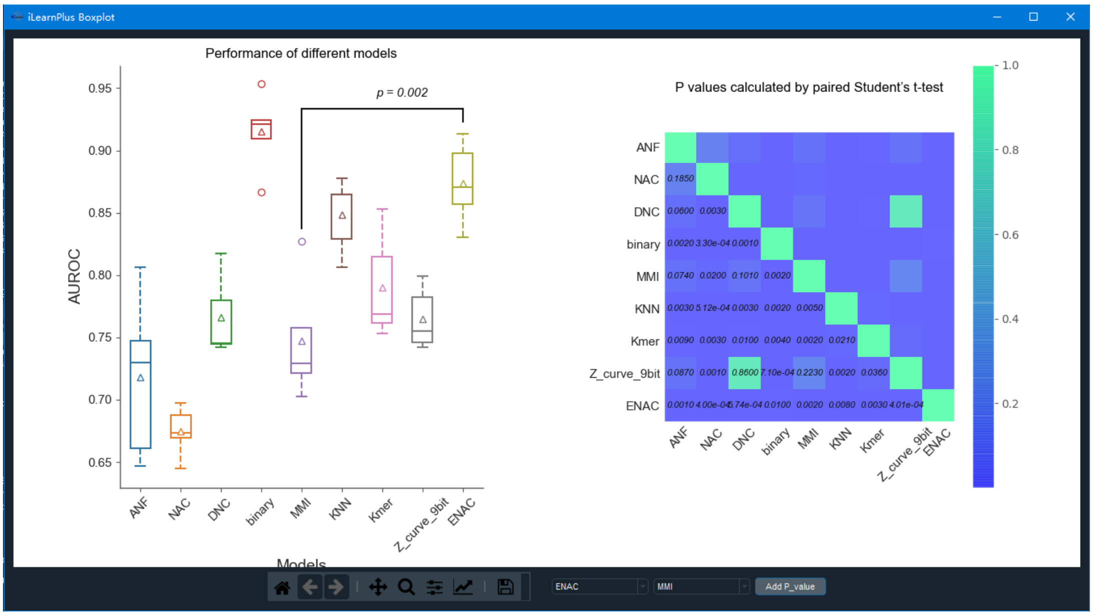

Result
Box plot
ROC and PRC curve
Log
ROC curve
True positive rate
False positive rate
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
ANF_RF_model Mean AUROC = 0.7183
NAC_RF_model Mean AUROC = 0.6746
DNC_RF_model Mean AUROC = 0.7658
binary_RF_model Mean AUROC = 0.915
MMI_RF_model Mean AUROC = 0.7476
KNN_RF_model Mean AUROC = 0.8485
Kmer_RF_model Mean AUROC = 0.7903
Z_curve_9bit_RF_model Mean AUROC = 0.7649
ENAC_RF_model Mean AUROC = 0.8737
Random
PRC curve
Precision
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
ANF_RF_model Mean AUPRC = 0.5077
NAC_RF_model Mean AUPRC = 0.41
DNC_RF_model Mean AUPRC = 0.5266
binary_RF_model Mean AUPRC = 0.8243
MMI_RF_model Mean AUPRC = 0.499
KNN_RF_model Mean AUPRC = 0.7002
Kmer_RF_model Mean AUPRC = 0.5762
Z_curve_9bit_RF_model Mean AUPRC = 0.5316
ENAC_RF_model Mean AUPRC = 0.7299
ROC
P values

Figure S18. The ROC and PRC curves generated by the *iLearnPlus-Estimator* module.

In addition, *iLearnPlus* offers two statistical tests for users to compare the prediction performance difference. The student's *t*-test is used to statistically compare the means of any two performance evaluation measures (e.g. *Sn*, *Sp*, *Acc*, *MCC* etc.) obtained via the *K*-fold cross-validation test (**Figure S19**); while a bootstrap test was used to assess the significance of performance difference between all pairs in the ROC or PRC curve (**Figure S20**).



P values calculated by paired Student's t-test

	ANF	NAC	DNC	binary	MMI	KNN	Kmer	Z_curve_9bit	ENAC
ANF									
NAC	0.1850								
DNC	0.0600	0.0030							
binary	0.0020	3.30e-04	0.0010						
MMI	0.0740	0.0200	0.1010	0.0020					
KNN	0.0030	5.12e-04	0.0030	0.0020	0.0050				
Kmer	0.0090	0.0030	0.0100	0.0040	0.0020	0.0210			
Z_curve_9bit	0.0870	0.0010	0.8600	7.10e-04	0.2230	0.0020	0.0360		
ENAC	0.0010	4.00e-04	5.74e-04	0.0100	0.0020	0.0080	0.0030	4.01e-04	

Figure S19. The paired *p*-values calculated by student's *t*-test method in the *iLearnPlus-Estimator* module.

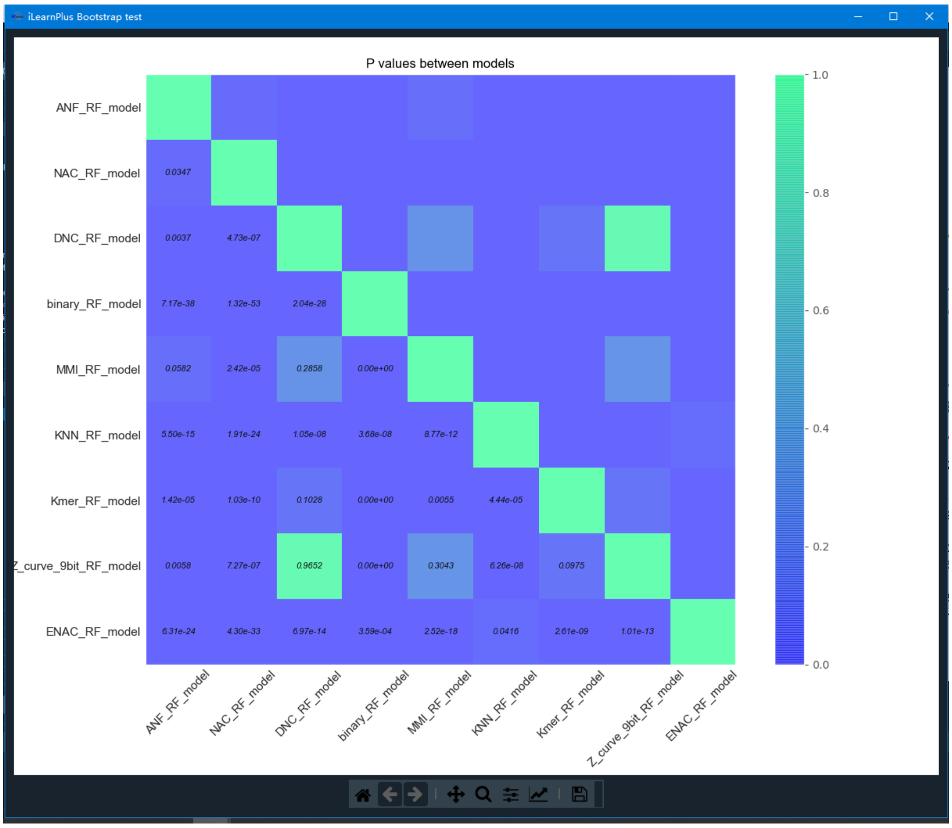

iLearnPlus Bootstrap test
P values between models
ANF_RF_model
NAC_RF_model 0.0347
DNC_RF_model 0.0037 4.73e-07
binary_RF_model 7.17e-38 1.32e-53 2.04e-28
MMI_RF_model 0.0582 2.42e-05 0.2858 0.00e+00
KNN_RF_model 5.50e-15 1.91e-24 1.05e-08 3.68e-08 8.77e-12
Kmer_RF_model 1.42e-05 1.03e-10 0.1028 0.00e+00 0.0055 4.44e-05
Z_curve_9bit_RF_model 0.0058 7.27e-07 0.9652 0.00e+00 0.3043 6.26e-08 0.0975
ENAC_RF_model 6.31e-24 4.30e-33 6.97e-14 3.59e-04 2.52e-18 0.0416 2.61e-09 1.01e-13
ANF_RF_model
NAC_RF_model
DNC_RF_model
binary_RF_model
MMI_RF_model
KNN_RF_model
Kmer_RF_model
Z_curve_9bit_RF_model
ENAC_RF_model
1.0
0.8
0.6
0.4
0.2
0.0

Figure S20. The paired *p*-values calculated by the bootstrap method in the *iLearnPlus-Estimator* module.

7. The *iLearnPlus-AutoML* module

The *iLearnPlus-AutoML* module focuses on automated performance benchmarking of different machine-learning algorithms based on the input features. The usage of the *iLearnPlus-AutoML* module is similar with the usage of the *iLearnPlus-Estimator* module. The difference of the two models is that the input of the *iLearnPlus-Estimator* module only contains biological sequences; while the input of the *iLearnPlus-AutoML* module is the feature encoding information in CSV, TSV, LIBSVM or WEKA format. Combining the *iLearnPlus-Estimator* and *iLearnPlus-AutoML* modules, users can evaluate and compare the prediction performance using all the selected feature descriptors and machine-learning algorithms in an efficient manner.

8. The *iLearnPlus-LoadModel* module

All the generated models can be saved in the three aforementioned modules and users can upload their models and testing dataset to perform prediction directly via the *iLearnPlus-LoadModel* module.

Step 1: Load models

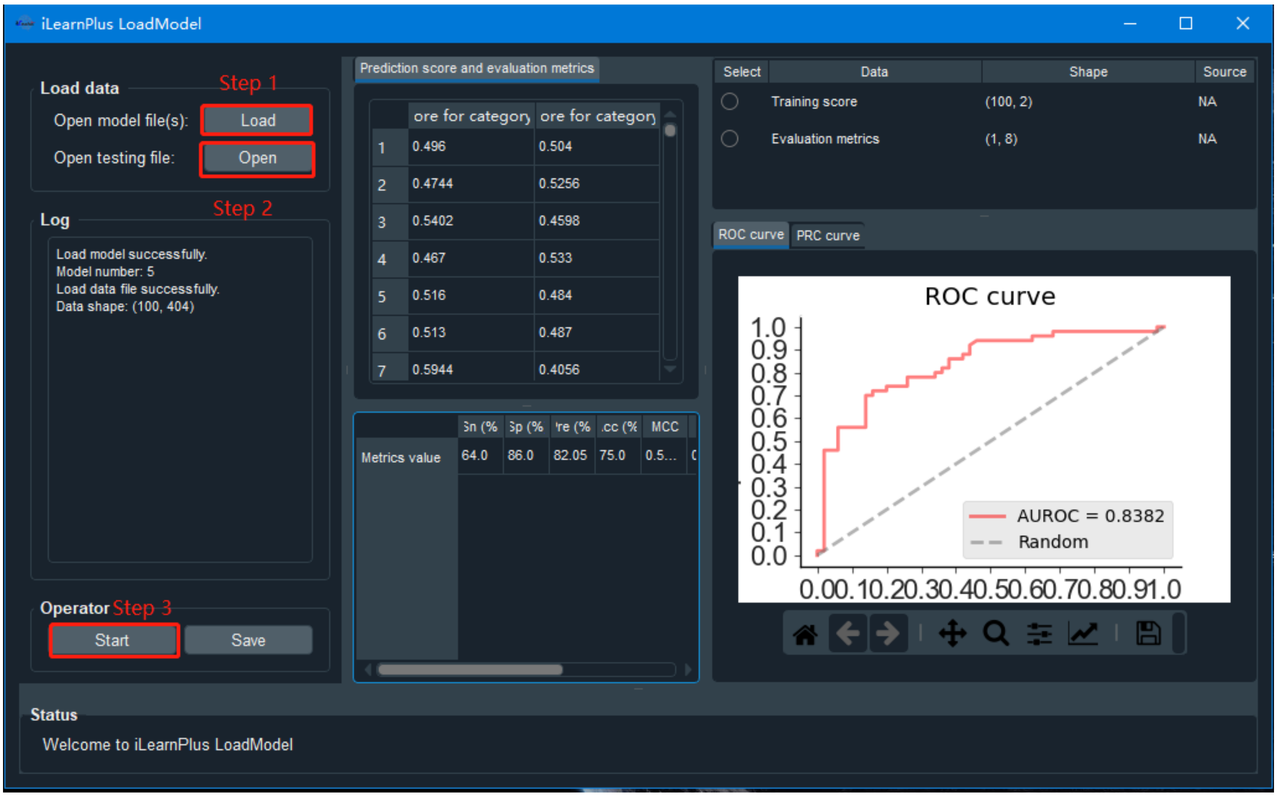
Click "Load" button and select the models in the "models" directory, one or more modules can be selected at one time.

Step 2: Load testing file

Click "Open" button and select the "binary_ind.csv" file in the "data" directory of the *iLearnPlus* package.

Step 3: Run the program

Click "Start" button to predict the testing file using the loaded models. If multiple models are loaded, the average prediction score of the models will be displayed. In addition, the evaluation metrics, ROC and PRC curves will also be displayed (**Figure S21**).



```
Load model successfully.
Model number: 5
Load data file successfully.
Data shape: (100, 404)
```

	ore for category	ore for categon
1	0.496	0.504
2	0.4744	0.5256
3	0.5402	0.4598
4	0.467	0.533
5	0.516	0.484
6	0.513	0.487
7	0.5944	0.4056

	Sn (%	Sp (%	're (%	.cc (%	MCC	
Metrics value	64.0	86.0	82.05	75.0	0.5...	(

Select	Data	Shape	Source
○	Training score	(100, 2)	NA
○	Evaluation metrics	(1, 8)	NA

Welcome to iLearnPlus LoadModel

Figure S21. The interface of the *iLearnPlus-LoadModel* module.

9. Other functions

For the users' convenience, *iLearnPlus* also supplies some additional applications, including "Plot ROC curve", "Plot PRC curve", "Boxplot", "Heatmap", "Scatter plot", "Distribution visualization", "File format transformation" and "Merge coding files into one". These applications aim to facilitate plotting with user-defined data and file operations.

10. Performance evaluation strategy in *iLearnPlus*

As described in our paper, *iLearnPlus* supports both the binary classification task and multiclass classification task. For binary classification problems, nine frequently-used measures are supported by *iLearnPlus*, including Sensitivity (*Sn*), Specificity (*Sp*), Accuracy (*Acc*), Matthew correlation coefficient (*MCC*), *Recall*, *Precision*, *F1-score*, the Area Under ROC curve (*AUROC*) and the Area Under the PRC curve (*AUPRC*). *Sn*, *Sp*, *Acc*, *MCC*, *Recall*, *Precision* and *F1-score* are defined as:

$$Sn = Recall = \frac{TP}{TP+FN},$$

$$Sp = \frac{TN}{TN+FP},$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \; ,$$

$$MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

$$Precision = \frac{TP}{TP+FP} \; ,$$

$$F1 = 2\times\frac{Precision\times Recall}{Precision+Recall},$$

where *TP*, *FP*, *TN* and *FN* represent the numbers of true positives, false positives, true negatives and false negatives, respectively. The *AUROC* and *AUCPRC* values, ranging between 0 and 1, are calculated based on the Receiver-Operating-Characteristic (ROC) curve and the Precision-Recall curve, respectively. The higher the *AUROC* and *AUPRC* values, the better the predictive performance of the model.

For multi-class classification tasks, the *Acc* is commonly used to evaluate the performance, which is defined as:

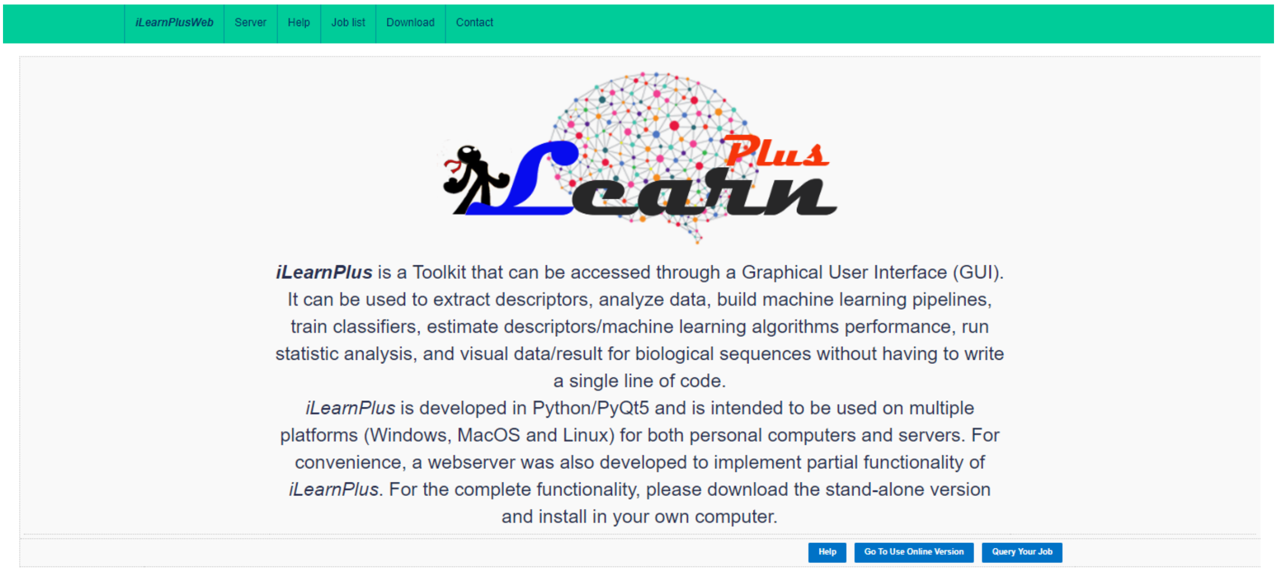
$$Acc = \frac{TP(i)+TN(i)}{TP(i)+TN(i)+FP(i)+FN(i)} \ ,$$

where *TP(i)*, *FP(i)*, *TN(i)* and *FN(i)* represent the numbers of the samples (molecules) predicted correctly to be in the *i*-th class, the total number of the samples in the *i*-th class that are predicted as one of the other classes, the total number of the samples predicted correctly not to be in the *i*-th class, and the total number of the samples not in the *i*-th class that are predicted as the *i*-th class, respectively.

11. Online web server

The *iLearnPlus* server is freely accessible at http://ilearnplus.erc.monash.edu/, which resides on the Nectar (The National eResearch Collaboration Tools and Resources) infrastructure and managed by the eResearch Centre at Monash University. The *iLearnPlus* server was implemented based on the open-source web platform LAMP (Linux-Apache-MySQL-PHP) and is equipped with 16 cores, 64 GB memory and a 2 TB hard disk. The server has been tested across five commonly used browsers, including Internet Explorer (version ≥7.0), Microsoft Edge, Mozilla Firefox, Google Chrome and Safari. Considering the computational burden, the web server only contains the *iLearnPlus-Basic* model for the analysis and machine-learning modeling of DNA, RNA and protein sequences. The step-by-step of usage instructions is as follows:

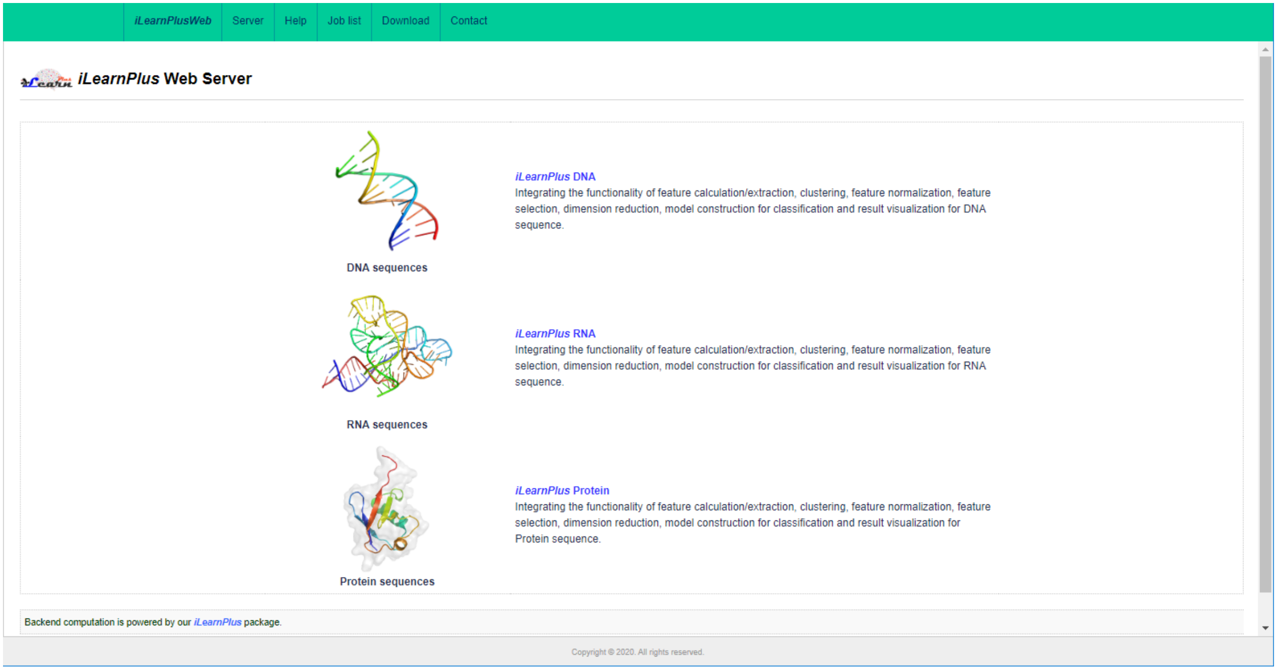
Type "http://ilearnplus.erc.monash.edu" on your browser, and click the "Go To Use Online Version" button.



iLearnPlus is a Toolkit that can be accessed through a Graphical User Interface (GUI). It can be used to extract descriptors, analyze data, build machine learning pipelines, train classifiers, estimate descriptors/machine learning algorithms performance, run statistic analysis, and visual data/result for biological sequences without having to write a single line of code.

iLearnPlus is developed in Python/PyQt5 and is intended to be used on multiple platforms (Windows, MacOS and Linux) for both personal computers and servers. For convenience, a webserver was also developed to implement partial functionality of *iLearnPlus*. For the complete functionality, please download the stand-alone version and install in your own computer.

Then, you will see the descriptor calculation page.



iLearnPlus Web Server

iLearnPlus DNA
Integrating the functionality of feature calculation/extraction, clustering, feature normalization, feature selection, dimension reduction, model construction for classification and result visualization for DNA sequence.

DNA sequences

iLearnPlus RNA
Integrating the functionality of feature calculation/extraction, clustering, feature normalization, feature selection, dimension reduction, model construction for classification and result visualization for RNA sequence.

RNA sequences

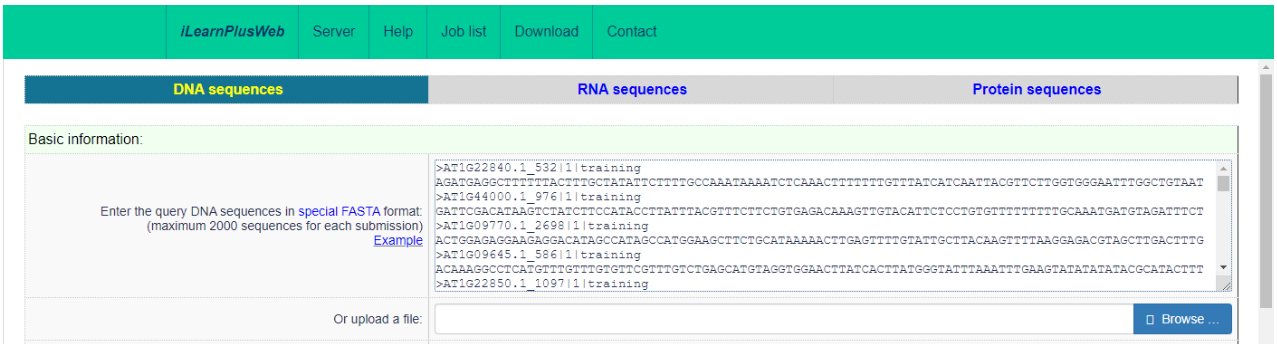
iLearnPlus Protein
Integrating the functionality of feature calculation/extraction, clustering, feature normalization, feature selection, dimension reduction, model construction for classification and result visualization for Protein sequence.

Protein sequences

Backend computation is powered by our *iLearnPlus* package.

Copyright © 2020. All rights reserved.

Step 1: Paste sequences or upload a sequence file.



Basic information:

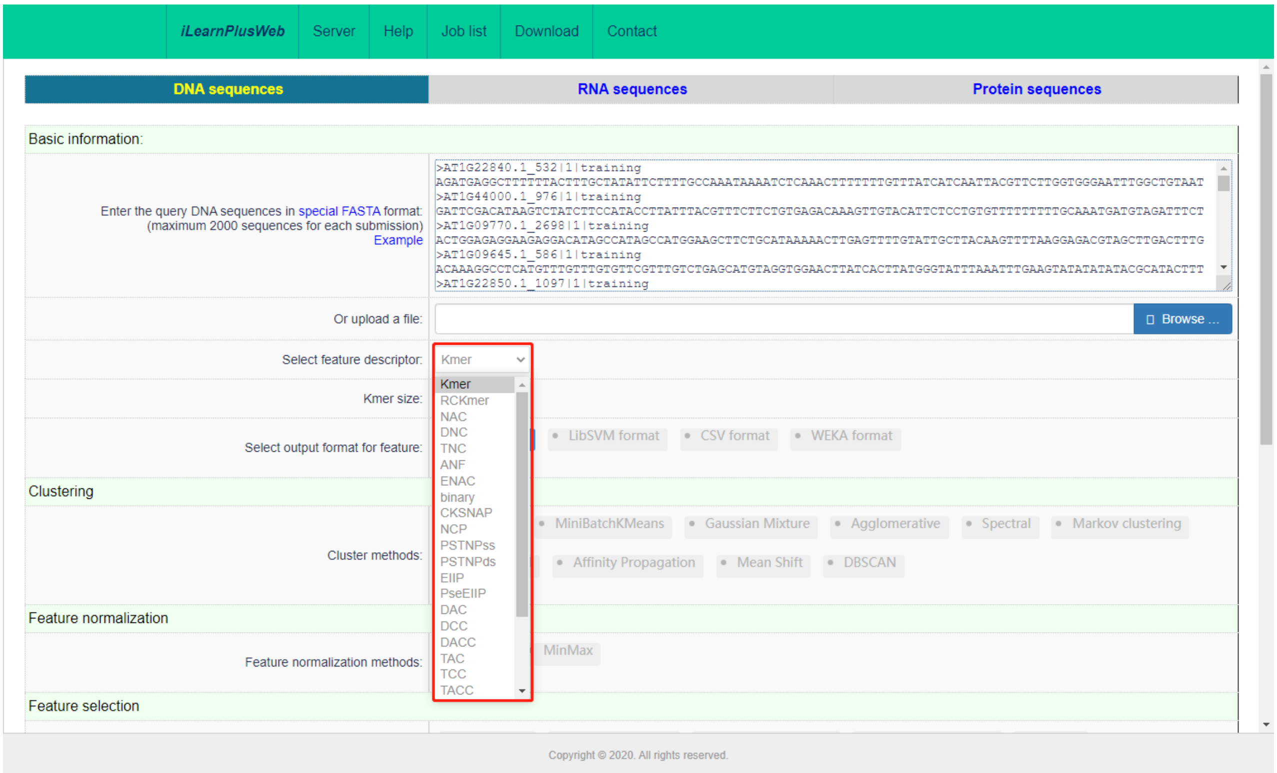
Enter the query DNA sequences in special FASTA format:
(maximum 2000 sequences for each submission)
Example

```
>AT1G22840.1_532|1|training
AGATGAGGCTTTTTTACTTTGCTATATTCTTTTGCCAAATAAAATCTCAAACTTTTTTTGTTTATCATCAATTACGTTCTTGGTGGGAATTTGGCTGTAAT
>AT1G44000.1_976|1|training
GATTCGACATAAGTCTATCTTCCATACCTTATTTACGTTTCTTCTGTGAGACAAAGTTGTACATTCTCCTGTGTTTTTTTTTGCAAATGATGTAGATTTCT
>AT1G09770.1_2698|1|training
ACTGGAGAGGAAGAGGACATAGCCATAGCCATGGAAGCTTCTGCATAAAAACTTGAGTTTTGTATTGCTTACAAGTTTTAAGGAGACGTAGCTTGACTTTG
>AT1G09645.1_586|1|training
ACAAAGGCCTCATGTTTGTTTGTGTTCGTTTGTCTGAGCATGTAGGTGGAACTTATCACTTATGGGTATTTAAATTTGAAGTATATATATACGCATACTTT
>AT1G22850.1_1097|1|training
```

Or upload a file:

Note: Paste your protein (or peptide) sequences in the text area or upload a file that includes the sequences. The biological sequences must be in a specified 'FASTA' format. *iLearnPlus* is designed to accept no more than 2000 sequences at one time.

Step 2: Select the descriptor type.



Basic information:

Enter the query DNA sequences in special FASTA format: (maximum 2000 sequences for each submission) Example

```
>AT1G22840.1_532|1|training
AGATGAGGCTTTTTTACTTTGCTATATTCTTTTGCCAAATAAAATCTCAAACTTTTTTTGTTTATCATCAATTACGTTCTTGGTGGGAATTTGGCTGTAAT
>AT1G44000.1_976|1|training
GATTCGACATAAGTCTATCTTCCATACCTTATTTACGTTTCTTCTGTGAGACAAAGTTGTACATTCTCCTGTGTTTTTTTTTGCAAATGATGTAGATTTCT
>AT1G09770.1_2698|1|training
ACTGGAGAGGAAGAGGACATAGCCATAGCCATGGAAGCTTCTGCATAAAAACTTGAGTTTTGTATTGCTTACAAGTTTTAAGGAGACGTAGCTTGACTTTG
>AT1G09645.1_586|1|training
ACAAAGGCCTCATGTTTGTTTGTGTTCGTTTGTCTGAGCATGTAGGTGGAACTTATCACTTATGGGTATTTAAATTTGAAGTATATATATACGCATACTTT
>AT1G22850.1_1097|1|training
```

Or upload a file: Browse ...

Select feature descriptor: Kmer

Kmer
RCKmer
NAC
DNC
TNC
ANF
ENAC
binary
CKSNAP
NCP
PSTNPss
PSTNPds
EIIP
PseEIIP
DAC
DCC
DACC
TAC
TCC
TACC

Kmer size:

Select output format for feature: LibSVM format • CSV format • WEKA format

Clustering

Cluster methods: MiniBatchKMeans • Gaussian Mixture • Agglomerative • Spectral • Markov clustering • Affinity Propagation • Mean Shift • DBSCAN

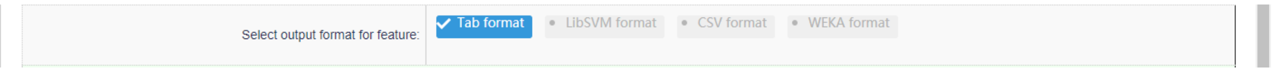
Feature normalization

Feature normalization methods: MinMax

Feature selection

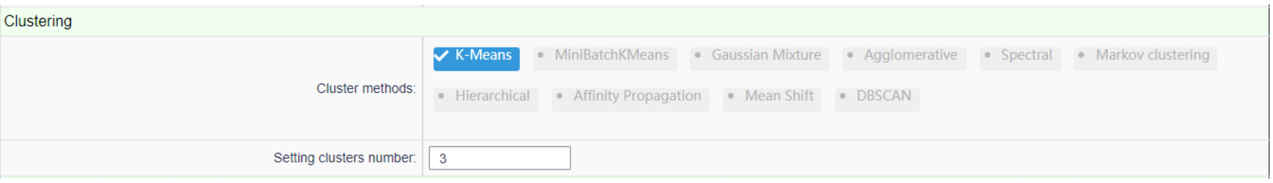
Copyright © 2020. All rights reserved.

Step 3: Select the output format.



Select output format for feature: ✓ Tab format • LibSVM format • CSV format • WEKA format

Step 4: Select the clustering method (optional).

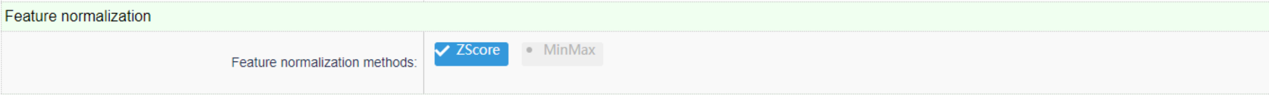


Clustering

Cluster methods: ✓ K-Means • MiniBatchKMeans • Gaussian Mixture • Agglomerative • Spectral • Markov clustering • Hierarchical • Affinity Propagation • Mean Shift • DBSCAN

Setting clusters number: 3

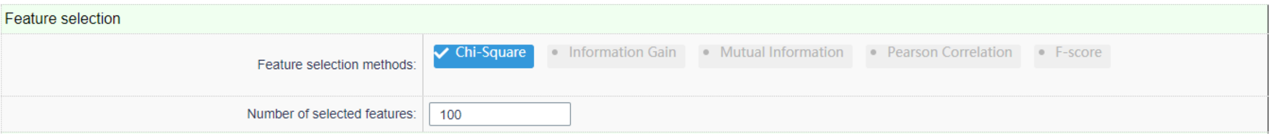
Step 5: Select the feature normalization method (optional).



Feature normalization

Feature normalization methods: ✓ ZScore • MinMax

Step 6: Select the feature selection method (optional).

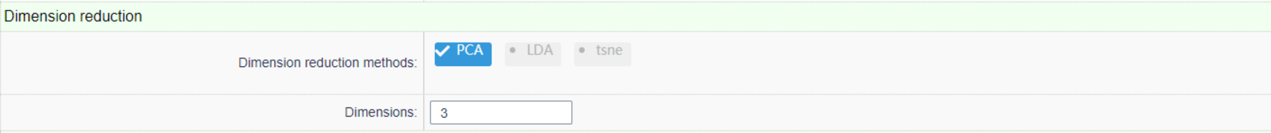


Feature selection

Feature selection methods: ✓ Chi-Square • Information Gain • Mutual Information • Pearson Correlation • F-score

Number of selected features: 100

Step 7: Select the dimension reduction method (optional).

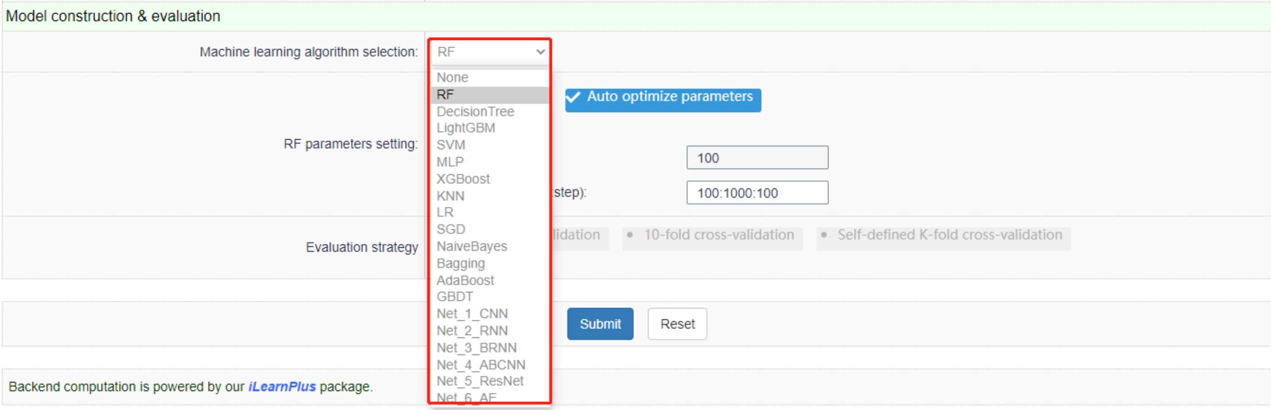


Dimension reduction

Dimension reduction methods: ☑ PCA ○ LDA ○ tsne

Dimensions: 3

Step 8: Select the machine-learning algorithm (optional).



Model construction & evaluation

Machine learning algorithm selection: RF

RF parameters setting: ☑ Auto optimize parameters

100

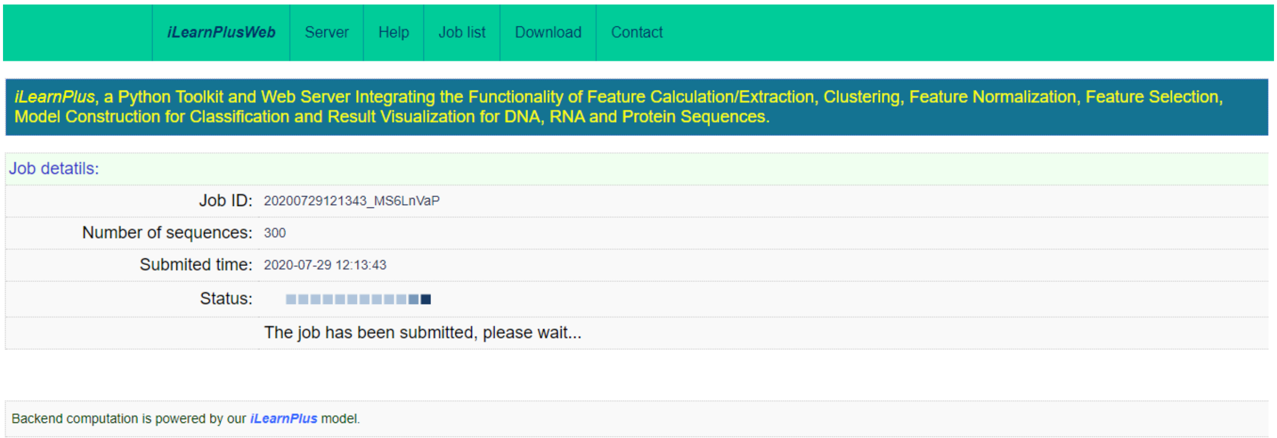
step): 100:1000:100

Evaluation strategy ○ lidation ○ 10-fold cross-validation ○ Self-defined K-fold cross-validation

Backend computation is powered by our *iLearnPlus* package.

Finally, click the 'Submit' button to calculate the descriptors and run the selected clustering, feature selection and machine-learning algorithms.

Step 9. Wait for prediction results.

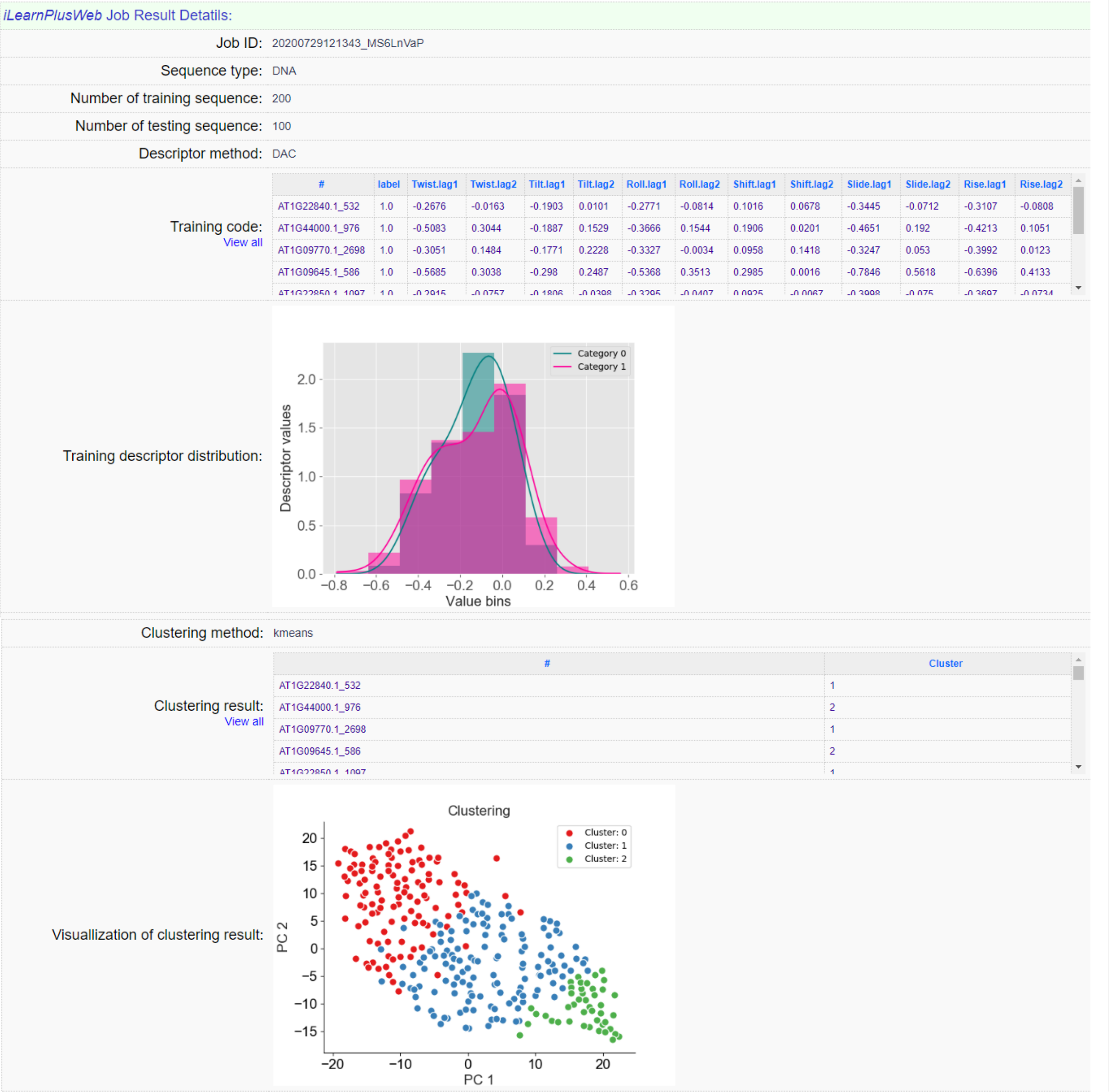


iLearnPlus, a Python Toolkit and Web Server Integrating the Functionality of Feature Calculation/Extraction, Clustering, Feature Normalization, Feature Selection, Model Construction for Classification and Result Visualization for DNA, RNA and Protein Sequences.

Job detatils:

Job ID:	20200729121343_MS6LnVaP
Number of sequences:	300
Submited time:	2020-07-29 12:13:43
Status:	
	The job has been submitted, please wait...

Backend computation is powered by our *iLearnPlus* model.



iLearnPlusWeb Job Result Details:

Job ID: 20200729121343_MS6LnVaP

Sequence type: DNA

Number of training sequence: 200

Number of testing sequence: 100

Descriptor method: DAC

Training code:
View all

#	label	Twist.lag1	Twist.lag2	Tilt.lag1	Tilt.lag2	Roll.lag1	Roll.lag2	Shift.lag1	Shift.lag2	Slide.lag1	Slide.lag2	Rise.lag1	Rise.lag2
AT1G22840.1_532	1.0	-0.2676	-0.0163	-0.1903	0.0101	-0.2771	-0.0814	0.1016	0.0678	-0.3445	-0.0712	-0.3107	-0.0808
AT1G44000.1_976	1.0	-0.5083	0.3044	-0.1887	0.1529	-0.3666	0.1544	0.1906	0.0201	-0.4651	0.192	-0.4213	0.1051
AT1G09770.1_2698	1.0	-0.3051	0.1484	-0.1771	0.2228	-0.3327	-0.0034	0.0958	0.1418	-0.3247	0.053	-0.3992	0.0123
AT1G09645.1_586	1.0	-0.5685	0.3038	-0.298	0.2487	-0.5368	0.3513	0.2985	0.0016	-0.7846	0.5618	-0.6396	0.4133
AT1G22850.1_1097	1.0	-0.2915	-0.0757	-0.1806	-0.0398	-0.3295	-0.0407	0.0925	-0.0067	-0.3998	-0.075	-0.3697	-0.0734

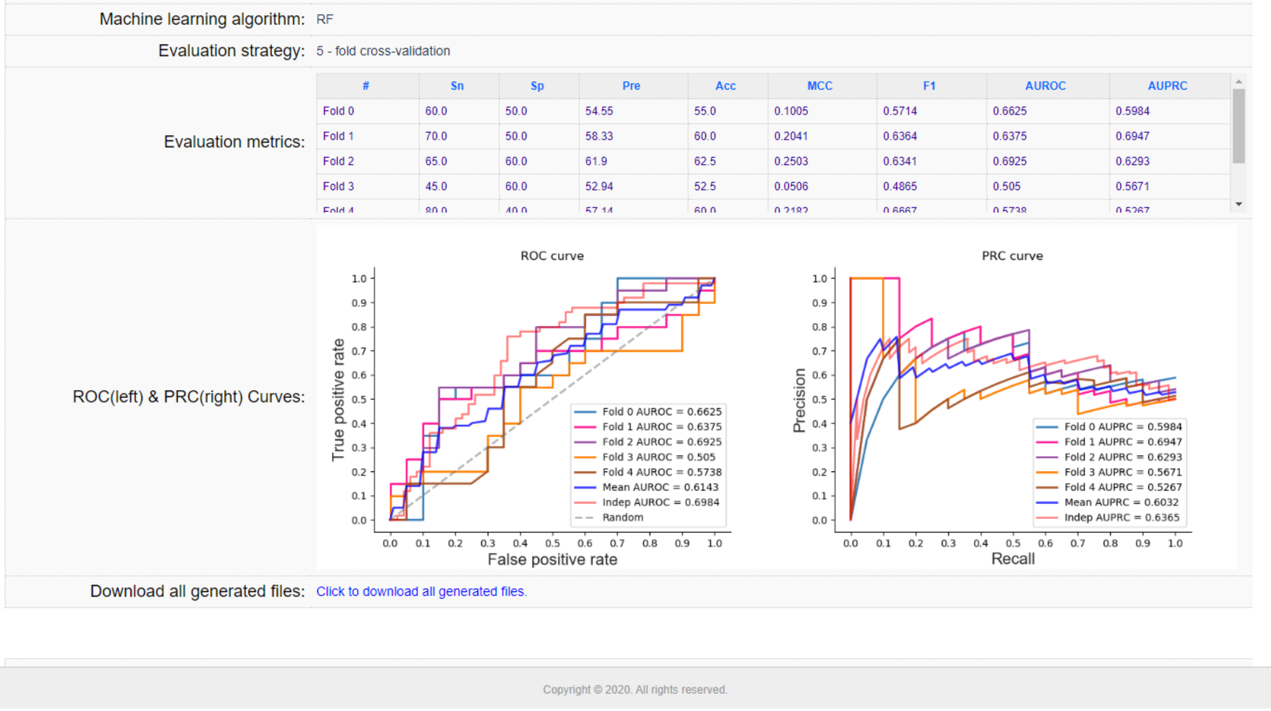
Training descriptor distribution:

Clustering method: kmeans

Clustering result:
View all

#	Cluster
AT1G22840.1_532	1
AT1G44000.1_976	2
AT1G09770.1_2698	1
AT1G09645.1_586	2
AT1G22850.1_1097	1

Visuallization of clustering result:



Machine learning algorithm: RF

Evaluation strategy: 5 - fold cross-validation

Evaluation metrics:

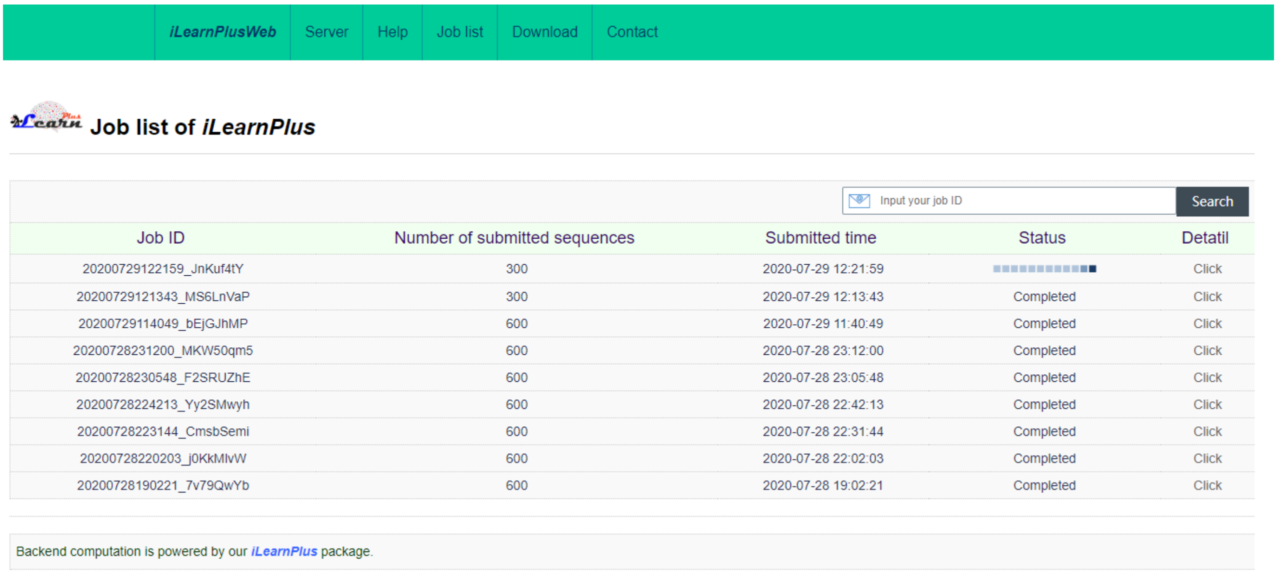
#	Sn	Sp	Pre	Acc	MCC	F1	AUROC	AUPRC
Fold 0	60.0	50.0	54.55	55.0	0.1005	0.5714	0.6625	0.5984
Fold 1	70.0	50.0	58.33	60.0	0.2041	0.6364	0.6375	0.6947
Fold 2	65.0	60.0	61.9	62.5	0.2503	0.6341	0.6925	0.6293
Fold 3	45.0	60.0	52.94	52.5	0.0506	0.4865	0.505	0.5671
Fold 4	80.0	40.0	57.14	60.0	0.2182	0.6667	0.5738	0.5267

ROC(left) & PRC(right) Curves:

Download all generated files: Click to download all generated files.

Copyright © 2020. All rights reserved.

Step 10: Query your result.



Job list of *iLearnPlus*

Job ID	Number of submitted sequences	Submitted time	Status	Detatil
20200729122159_JnKuf4tY	300	2020-07-29 12:21:59	▪▪▪▪▪▪▪▪▪▪▪■	Click
20200729121343_MS6LnVaP	300	2020-07-29 12:13:43	Completed	Click
20200729114049_bEjGJhMP	600	2020-07-29 11:40:49	Completed	Click
20200728231200_MKW50qm5	600	2020-07-28 23:12:00	Completed	Click
20200728230548_F2SRUZhE	600	2020-07-28 23:05:48	Completed	Click
20200728224213_Yy2SMwyh	600	2020-07-28 22:42:13	Completed	Click
20200728223144_CmsbSemi	600	2020-07-28 22:31:44	Completed	Click
20200728220203_j0KkMlvW	600	2020-07-28 22:02:03	Completed	Click
20200728190221_7v79QwYb	600	2020-07-28 19:02:21	Completed	Click

Backend computation is powered by our *iLearnPlus* package.

After a few seconds, the result should display in the result page. For each job, the server will generate a job ID, and the results will be stored for a week. Within a week, you can query your result by searching your job ID.

12. Summary

In summary, *iLearnPlus* has been extensively benchmarked to guarantee the reliability and precision of the computations and has been specifically designed to ensure the workflow efficiency.

To the best of our knowledge, this is the first both GUI (Graphical User Interface)- and web-based software platform for building automated machine-learning pipelines, facilitating user-friendly and in-depth analysis, modeling and prediction using DNA, RNA and protein sequence data. We will regularly maintain and update the analysis, clustering and machine-learning algorithms to enable high-qulaity interactive analysis and machine-learning-based modeling in the future. It is anticipated that *iLearnPlus* will be widely used as a powerful tool for analyzing, predicting and visualizing the nucleic acid and protein sequences.

References

1. Chang, C.C. and Lin, C.J. (2011) LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, Article 27.
2. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009) The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, **11**, 10–18.
3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. *et al.* (2011) Scikit-learn: Machine Learning in Python. *J Mach Learn Res*, **12**, 2825–2830.
4. Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, San Francisco, California, USA, pp. 785–794.
5. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y. (2017) LightGBM: a highly efficient gradient boosting decision tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., Long Beach, California, USA, pp. 3149–3157.
6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. *et al.* (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche-Buc, F., Fox, E. and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, Vol. 32. pp. 8024–8035.