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 (such as Windows, Mac and Linux). Prior to installing and running *iLearnPlus*, all the dependencies should be installed in the Python environment, including sys, os, re, 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.

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**.

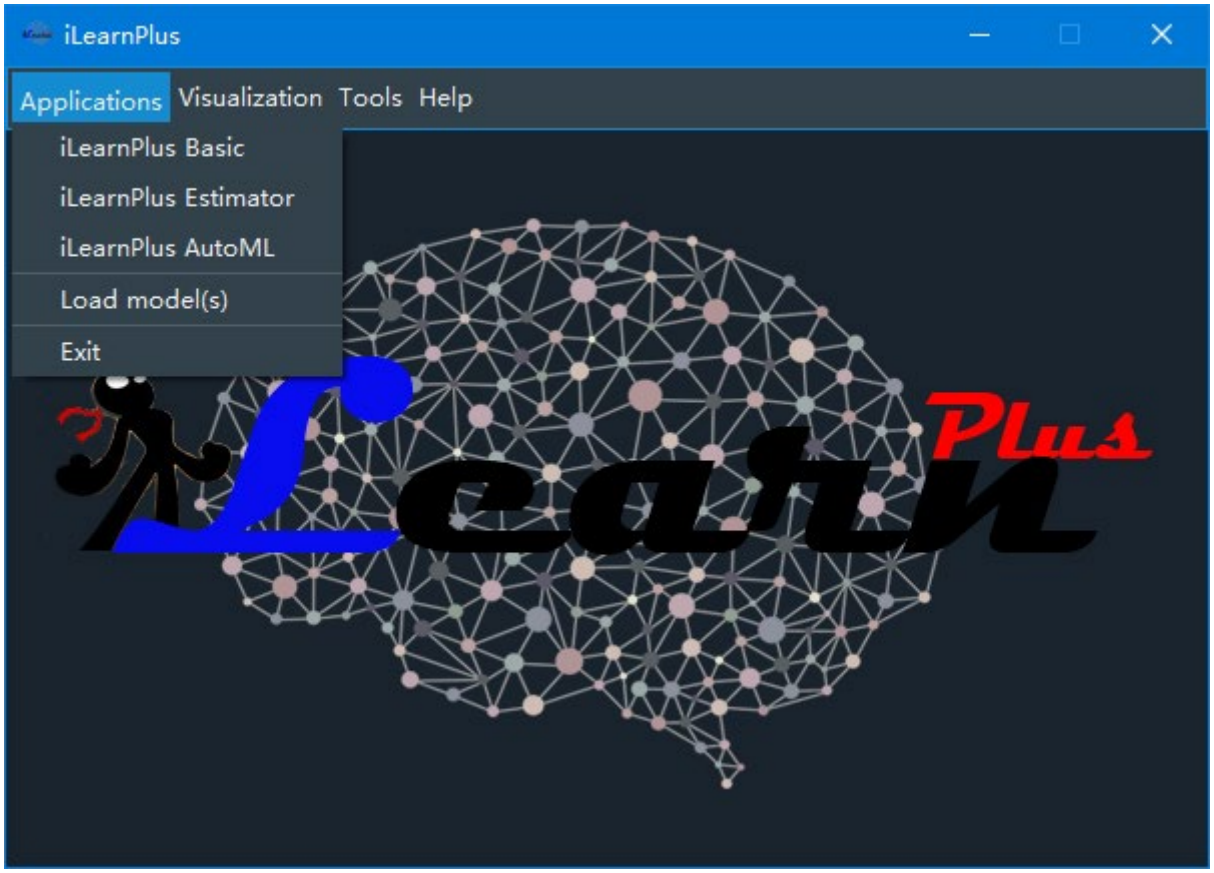


Figure S1. The main interface of the GUI version of *iLearnPlus*.

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.

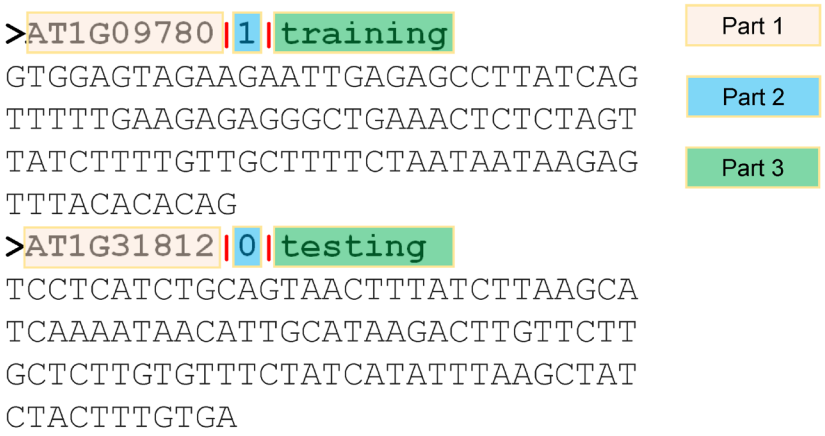


Figure S2. An example of the FASTA-formatted input DNA sequences used in *iLearnPlus* for feature descriptor extraction.

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).

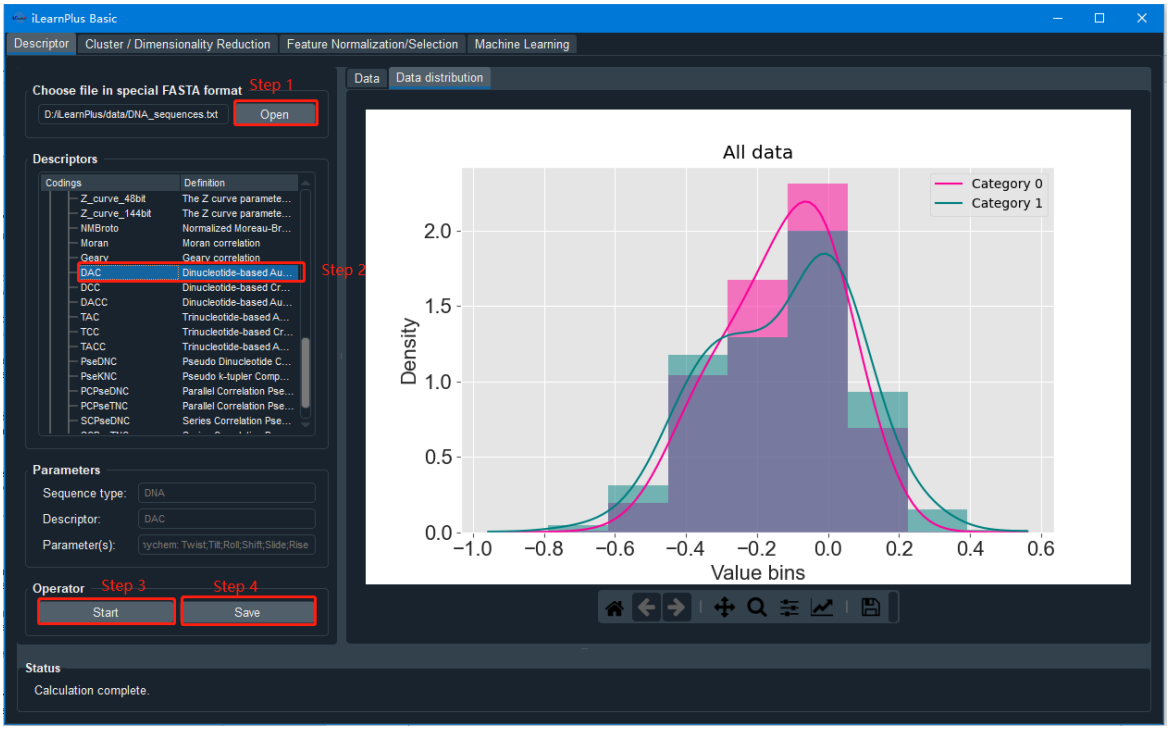
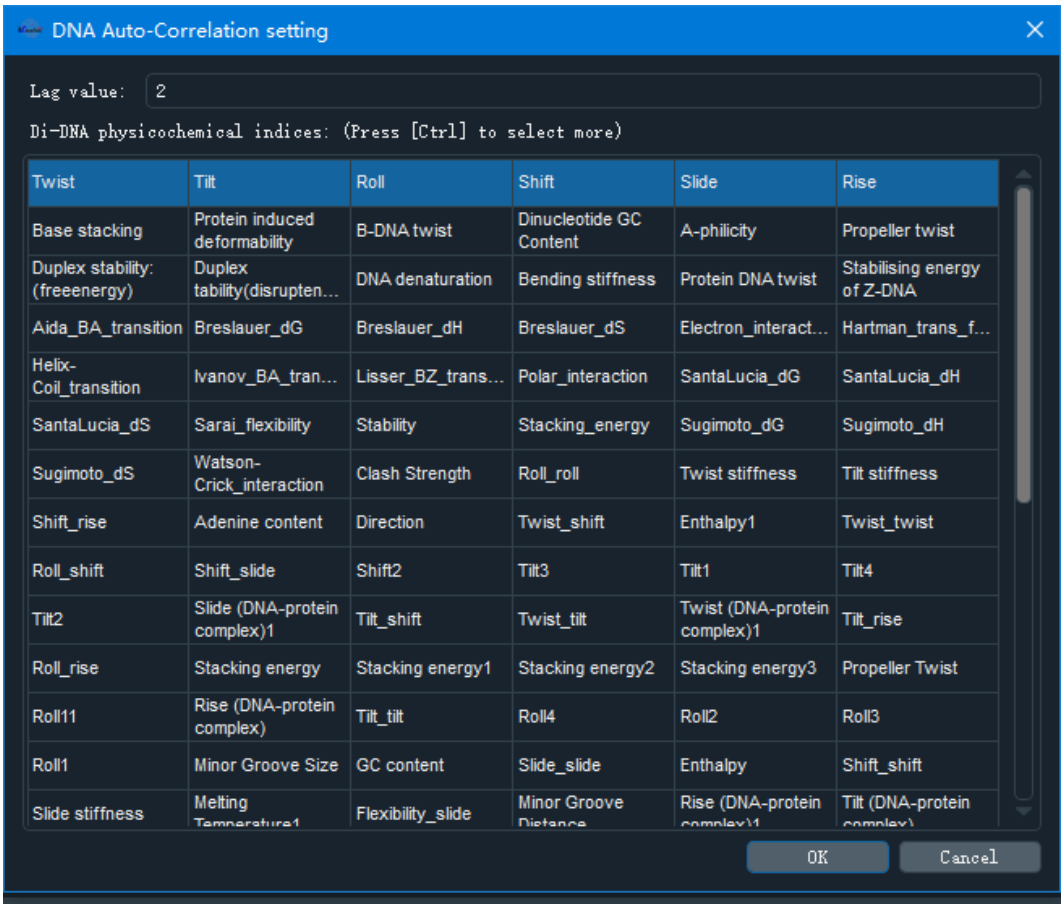


Figure S3. An example of extracting feature descriptor using the *iLearnPlus-Basic* module.



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.

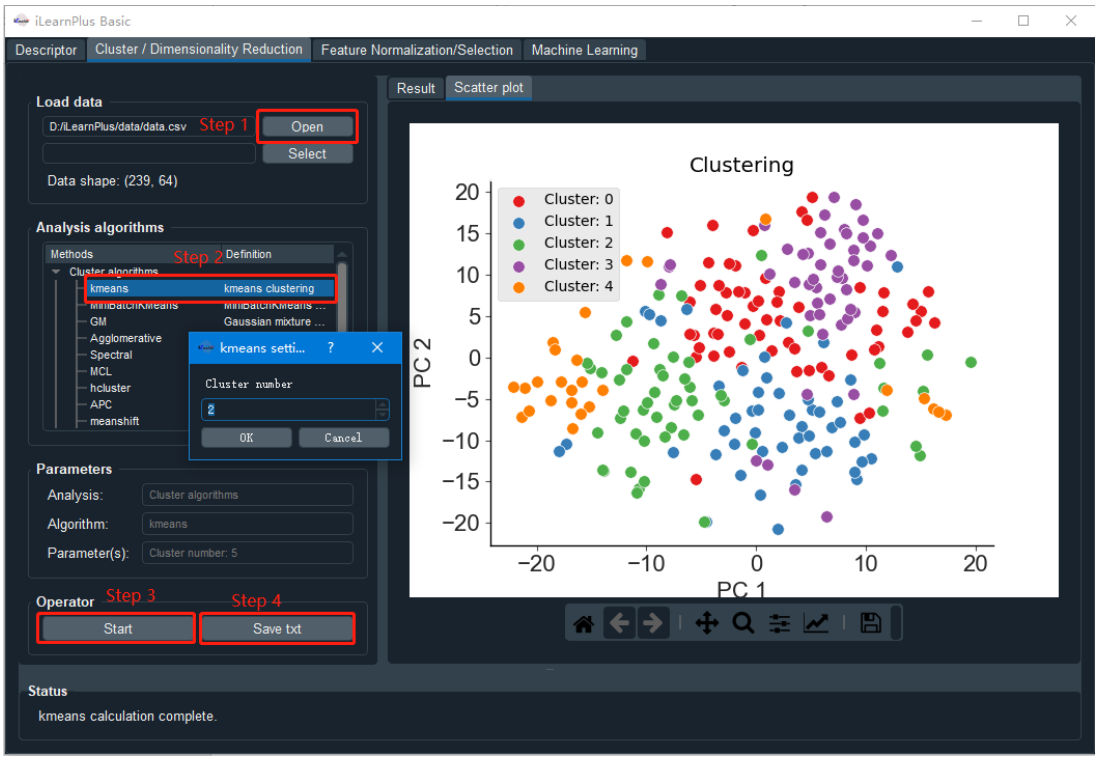


Figure S5. An example of implement the clustering algorithm using *iLearnPlus-Basic* module.

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.

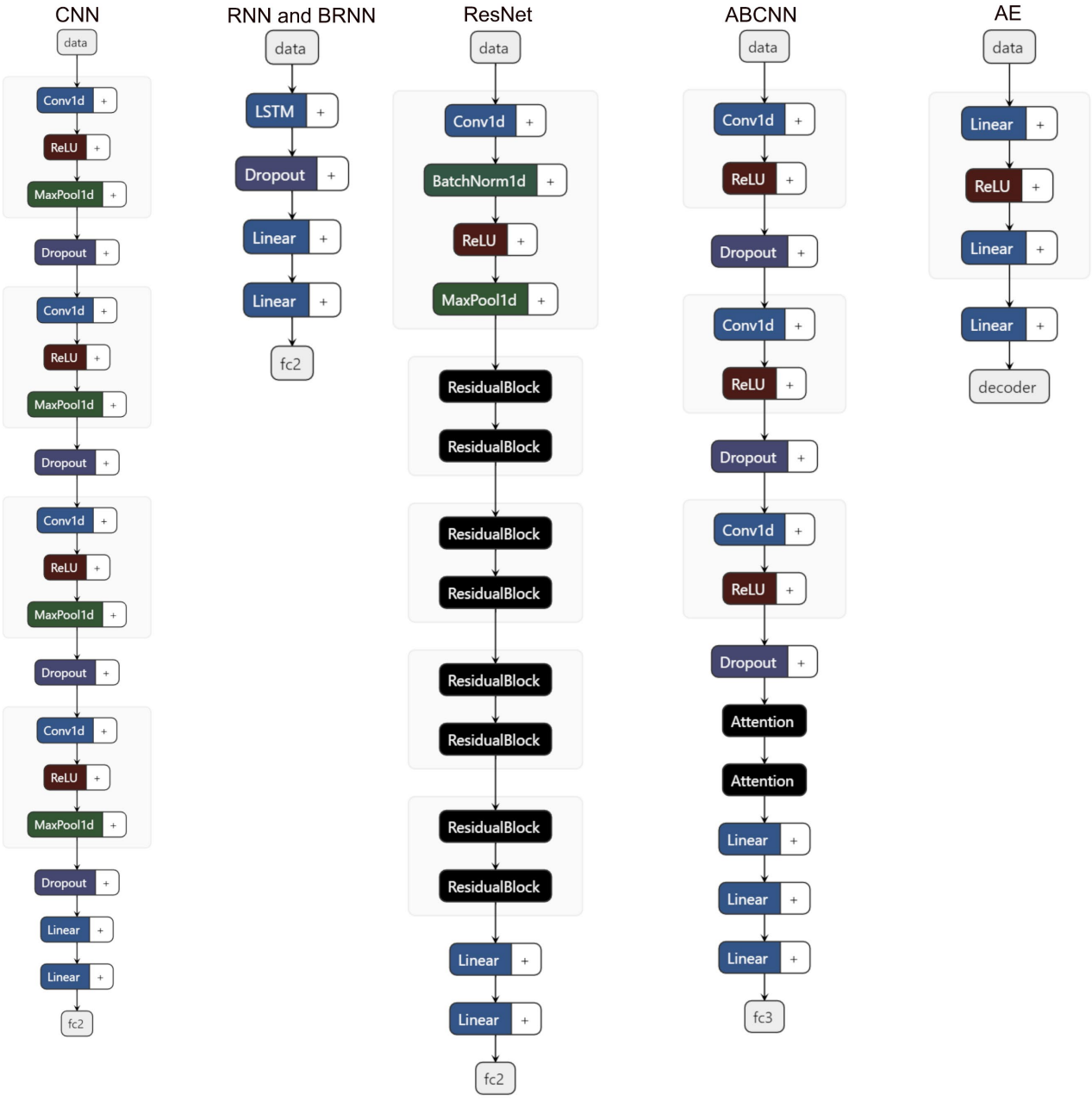


Figure S6. The architectures of the deep-learning approaches employed in *iLearnPlus*.

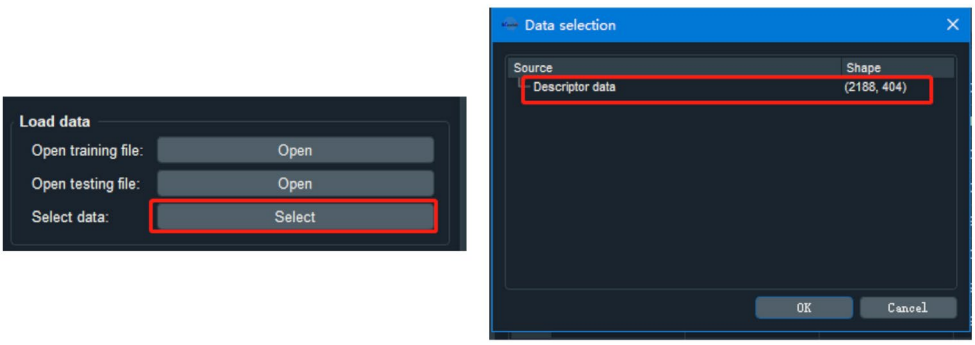


Figure S7. Loading data using the data selection explorer.

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.

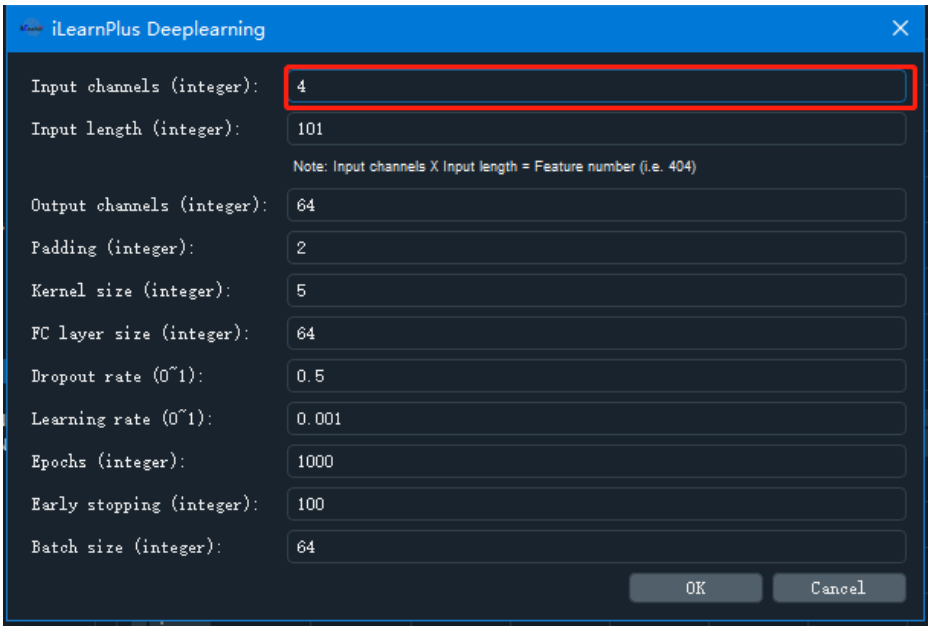


Figure S8. An example of parameter setting for the CNN algorithm.

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**).

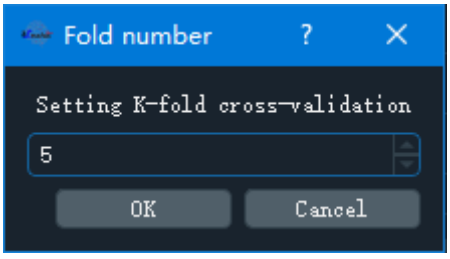


Figure S9. Setting five-fold cross-validation.

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**).

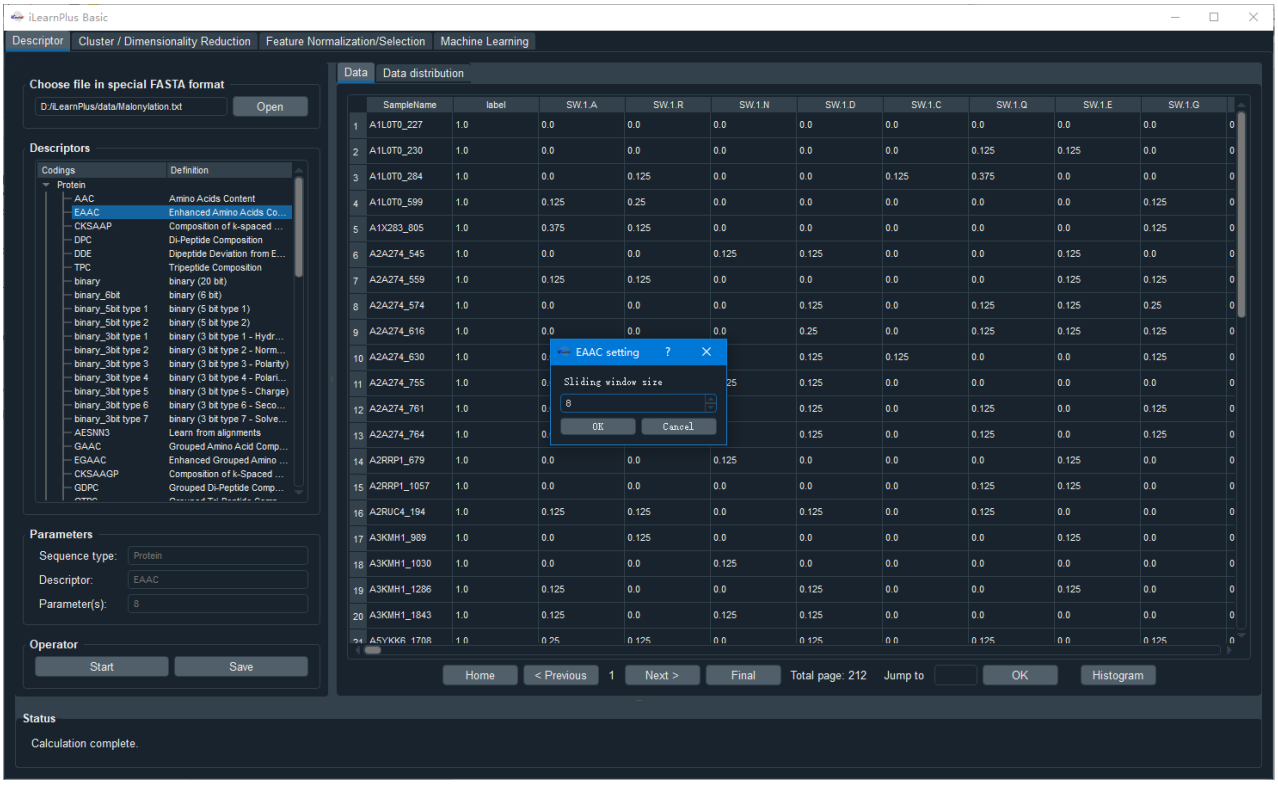


Figure S11. Extracting the EAAC descriptor using "Descriptor" panel in the *iLearnPlus-Basic* module.

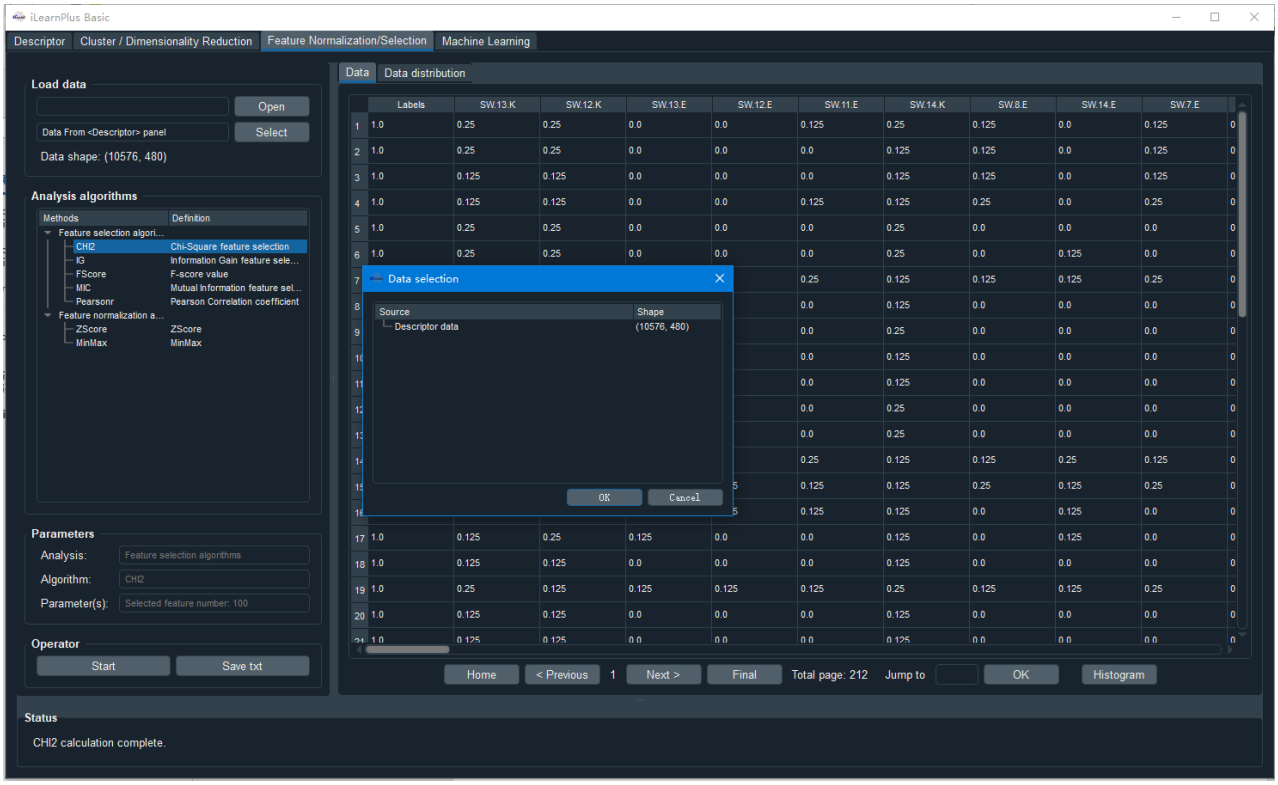


Figure S12. Selecting the top 100 features using "Feature Normalization/Selection" panel in the *iLearnPlus-Basic* module.

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**).

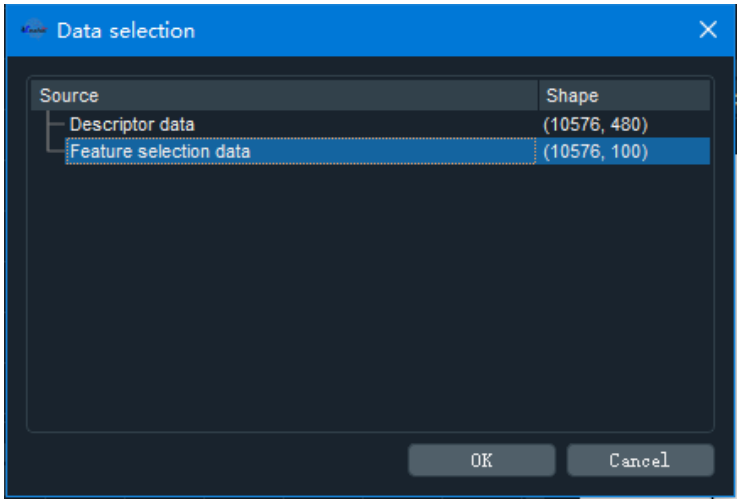


Figure S13. An example of loading data with the data selection dialog box.

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^1A 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.

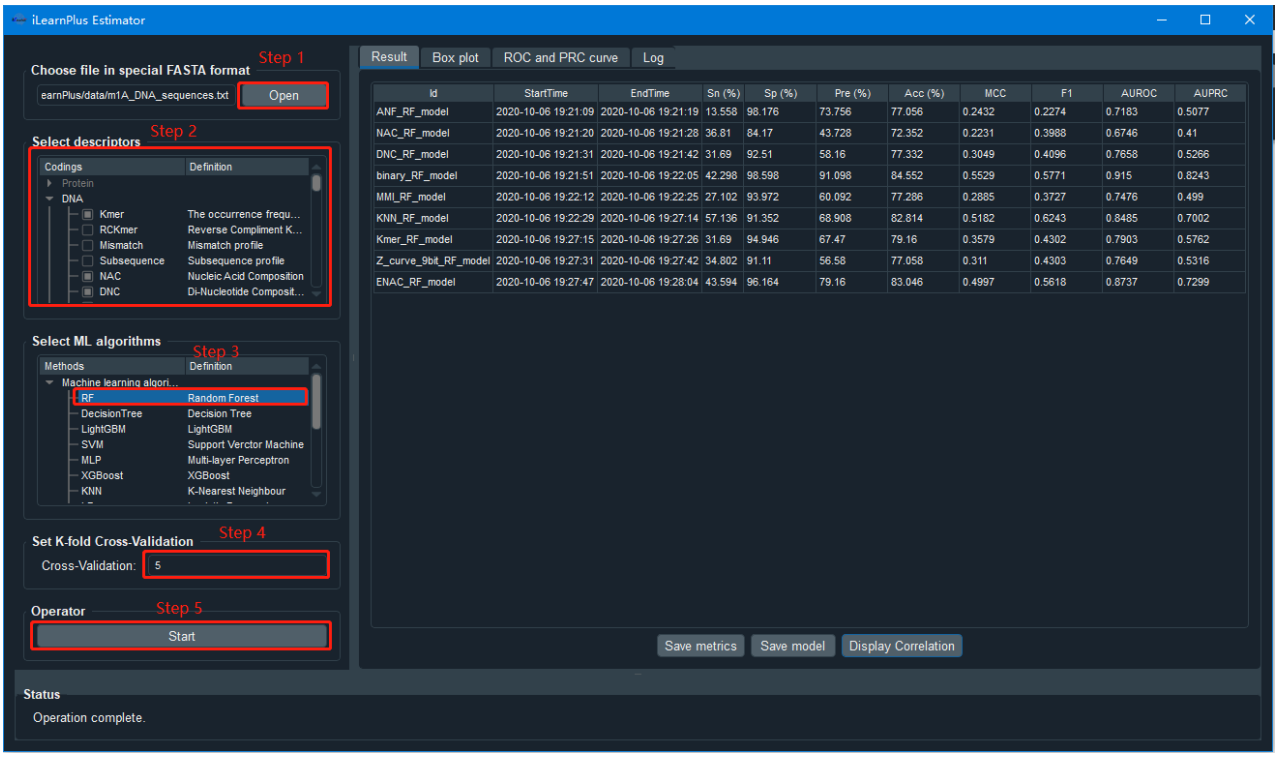


Figure S15. The panel of the *iLearnPlus-Estimator* module.

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**).

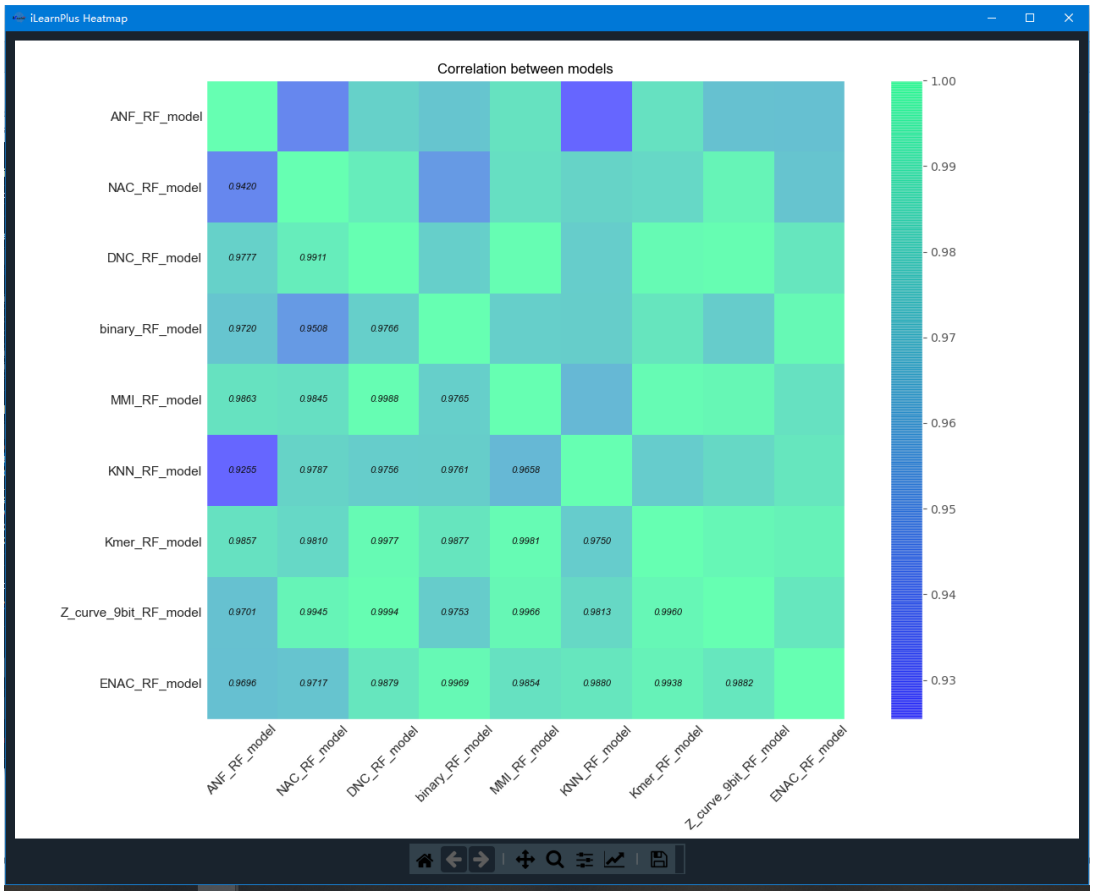


Figure S16. The correlation matrix generated by the *iLearnPlus-Estimator* module.

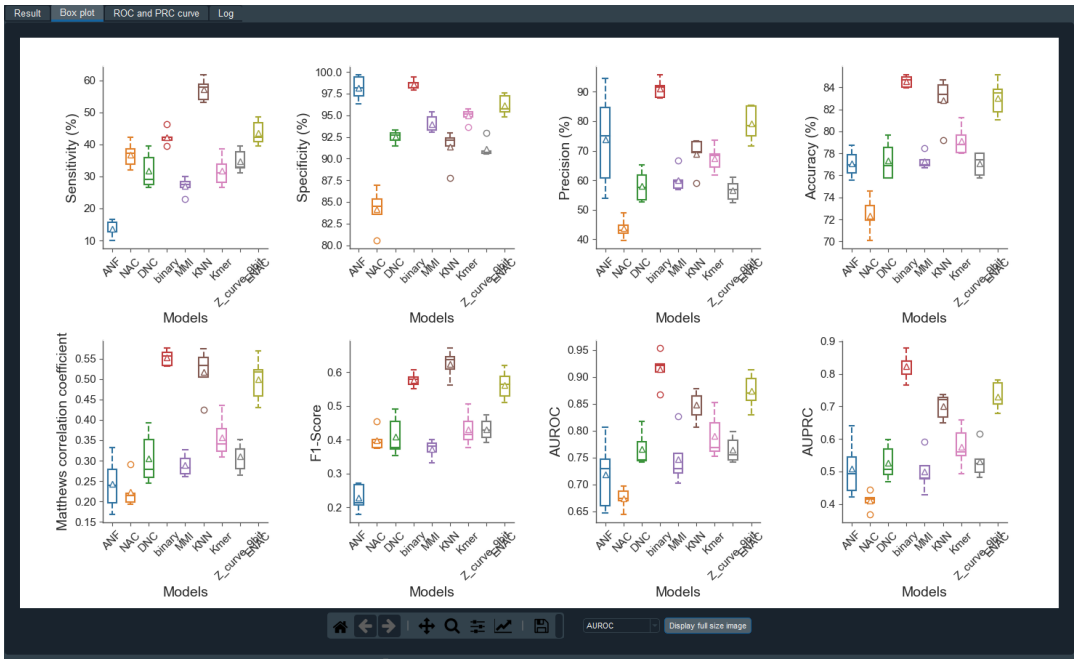


Figure S17. The boxplot generated by the *iLearnPlus-Estimator* module.

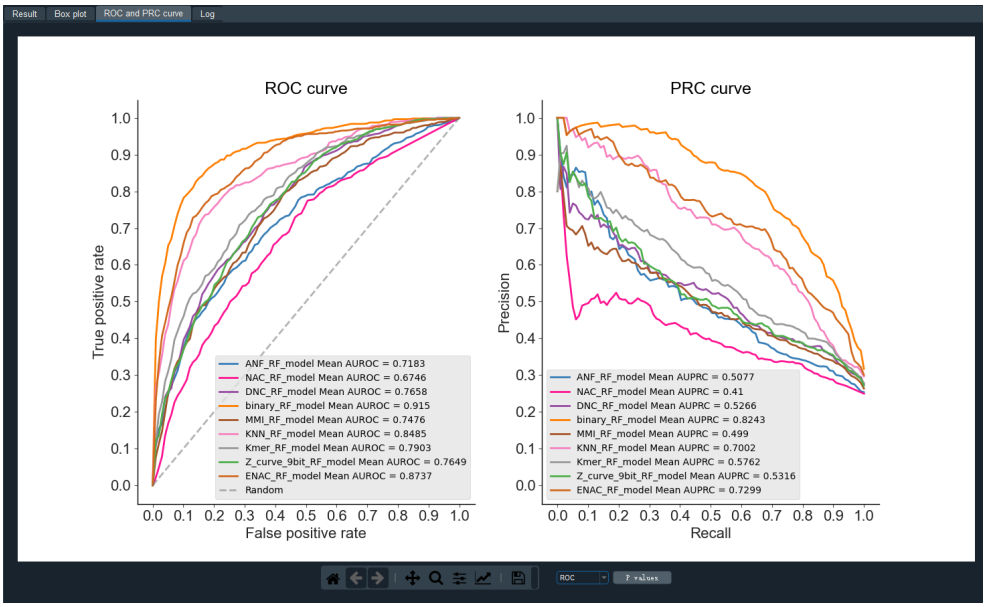


Figure S18. The ROC and PRC curves generated by the *iLearnPlus-Estimator* module.

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**).

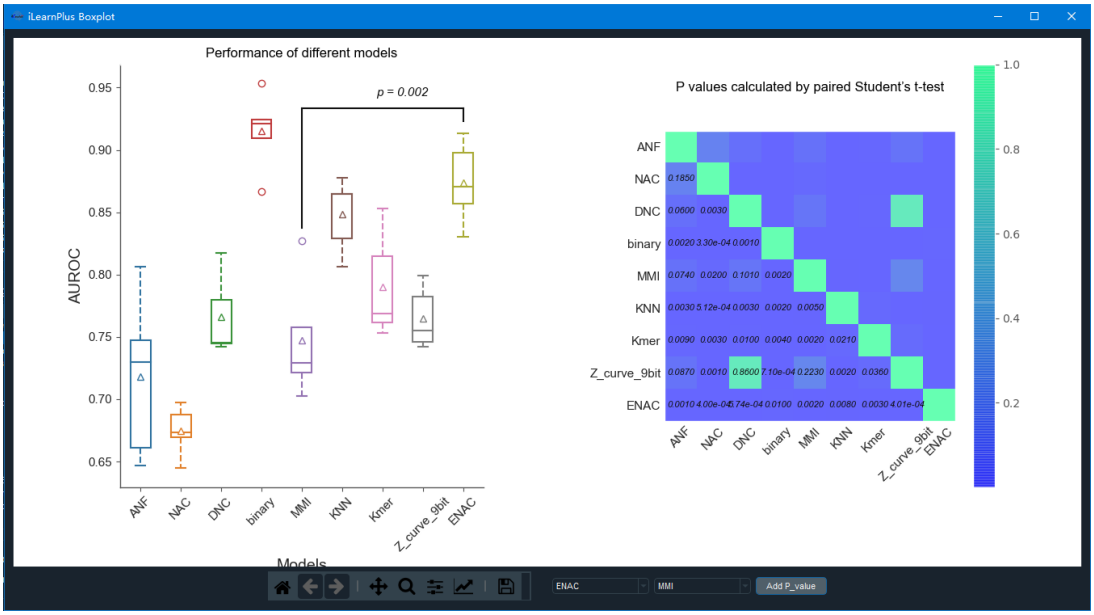


Figure S19. The paired *p*-values calculated by student's *t*-test method in the *iLearnPlus-Estimator* module.

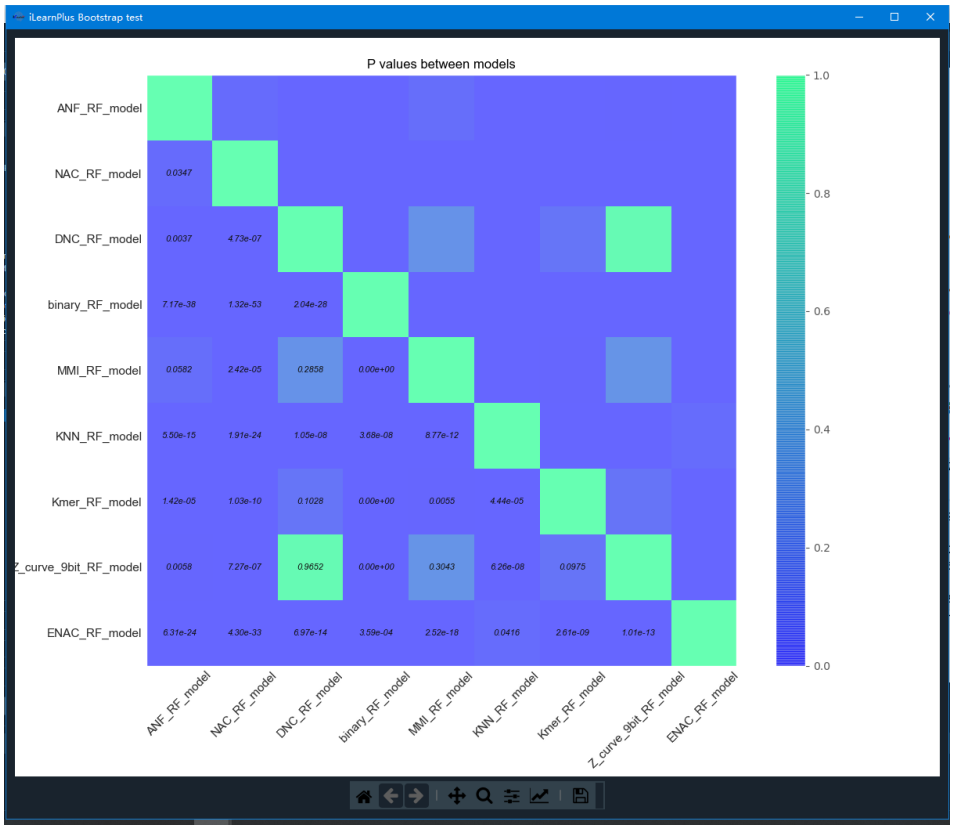


Figure S20. The paired *p*-values calculated by the bootstrap method in the *iLearnPlus-Estimator* module.

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 predicted 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**).

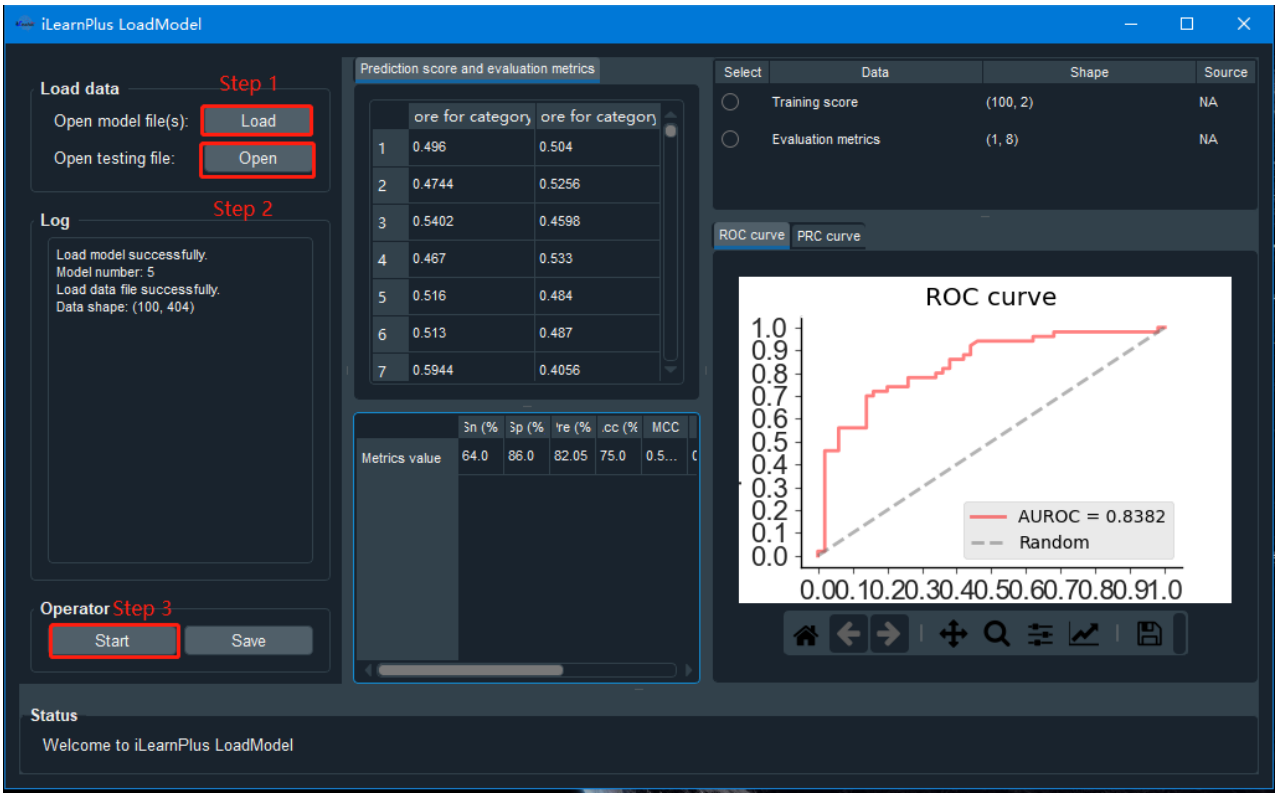


Figure S21. The interface of the *iLearnPlus-LoadModel* module.

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:

$$Sn = Recall = \frac{TP}{TP+FN}, \quad (2)$$

$$Sp = \frac{TN}{TN+FP}, \quad (3)$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}, \quad (4)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \quad (5)$$

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

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

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 are, the better prediction performance the model has.

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

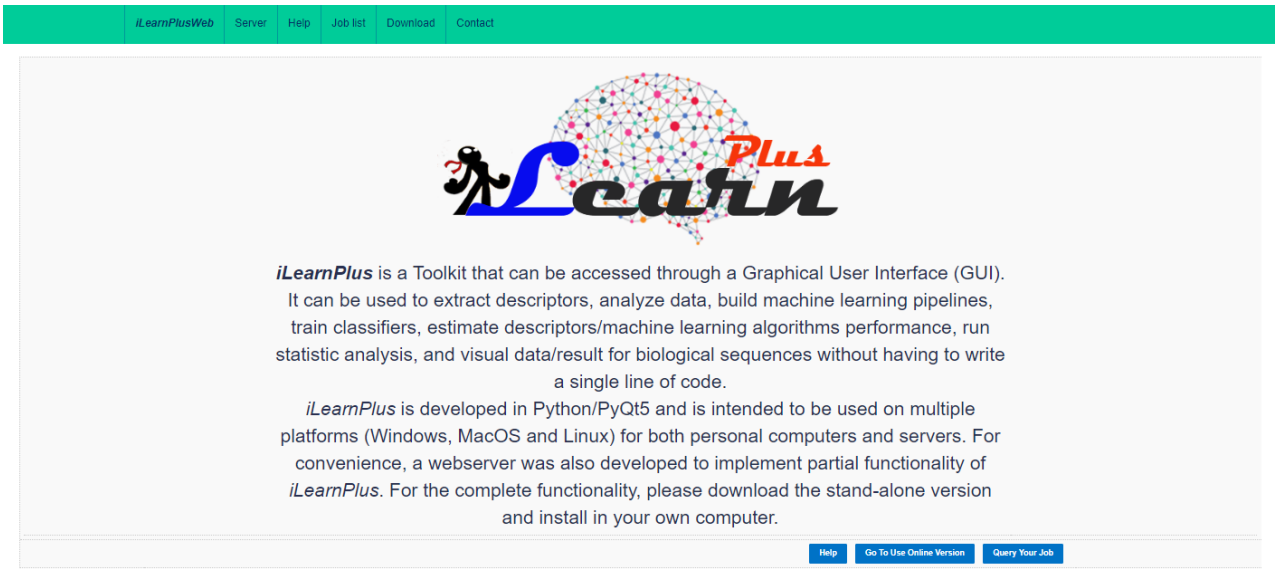
$$Acc = \frac{TP(i)+TN(i)}{TP(i)+TN(i)+FP(i)+FN(i)} ,$$

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.

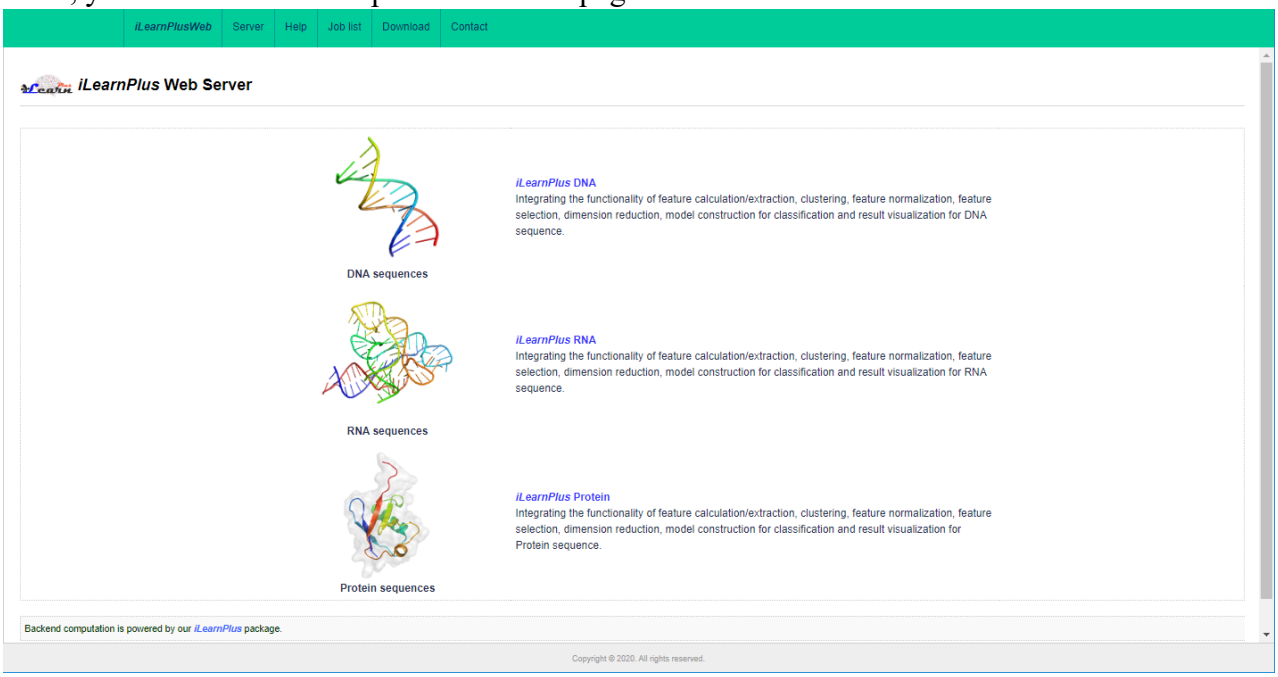
11. Online web server

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:

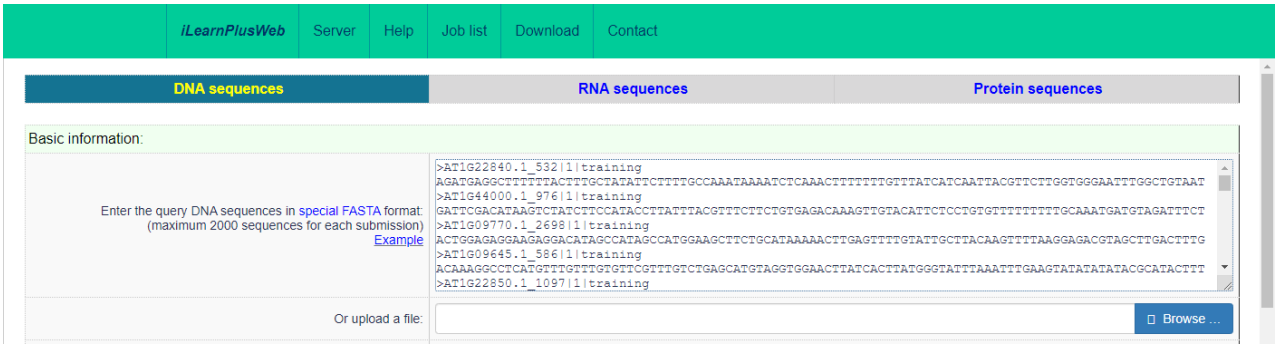
Type “http://ilearnplus.erc.monash.edu” on your browser, and click the “Go To Use Online Version” button.



Then, you will see the descriptor calculation page.

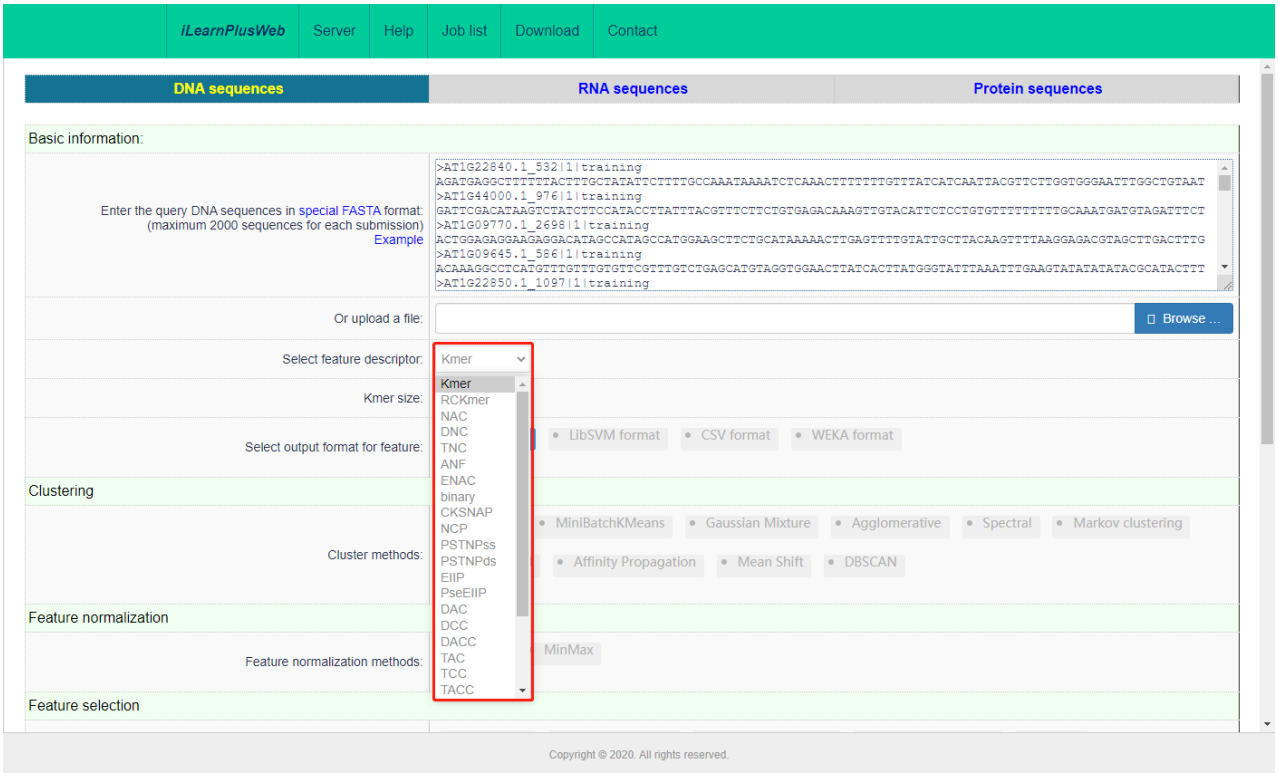


Step 1: Paste sequences or upload a sequence file.

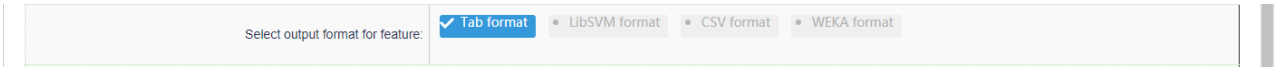


Note: Paste your protein (or peptide) sequences in the text area or upload a file that includes the sequences. The biological sequences must be in a specified 'FASTA' format. *iLearnPlus* is designed to accept no more than 2000 sequences at one time.

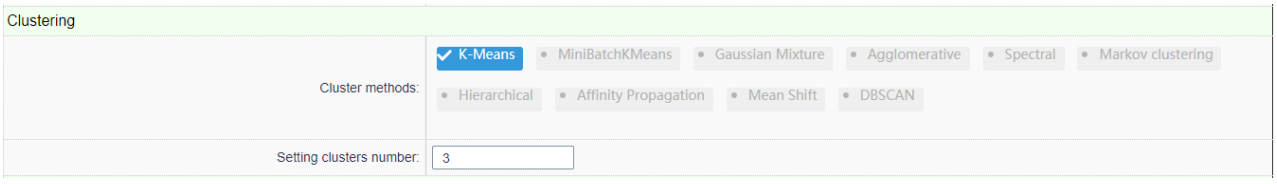
Step 2: Select the descriptor type.



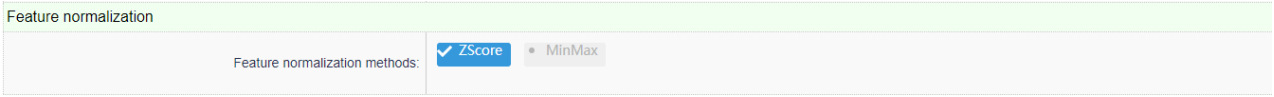
Step 3: Select the output format.



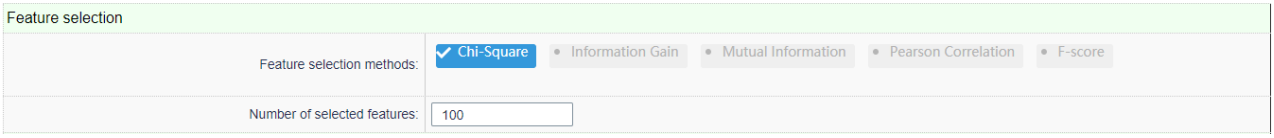
Step 4: Select the clustering method (optional).



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

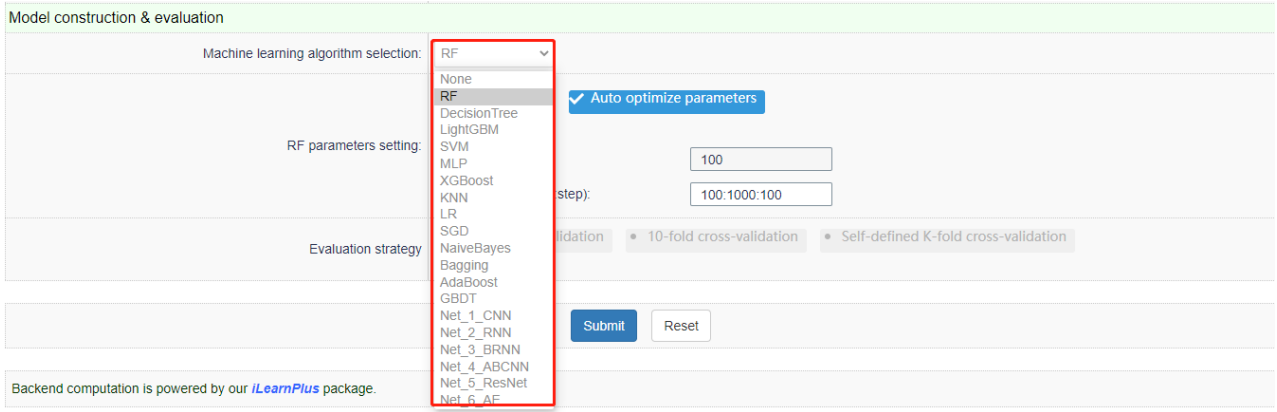


Step 6: Select the feature selection method (optional).



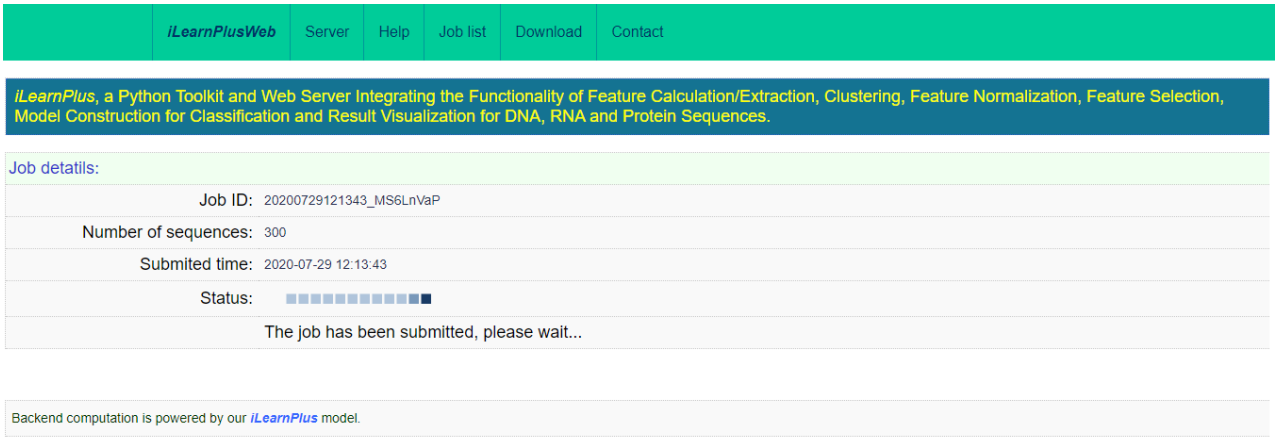
Step 7: Select the dimension reduction method (optional).

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



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

Step 9. Wait for prediction results.



iLearnPlusWeb Job Result Detatils:

Job ID: 20200729121343_MS6LnVaP

Sequence type: DNA

Number of training sequence: 200

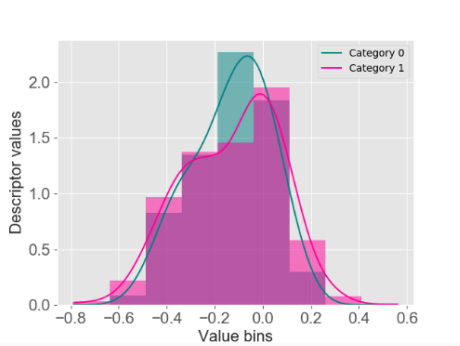
Number of testing sequence: 100

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:

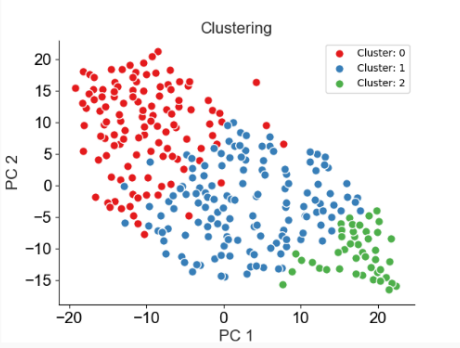


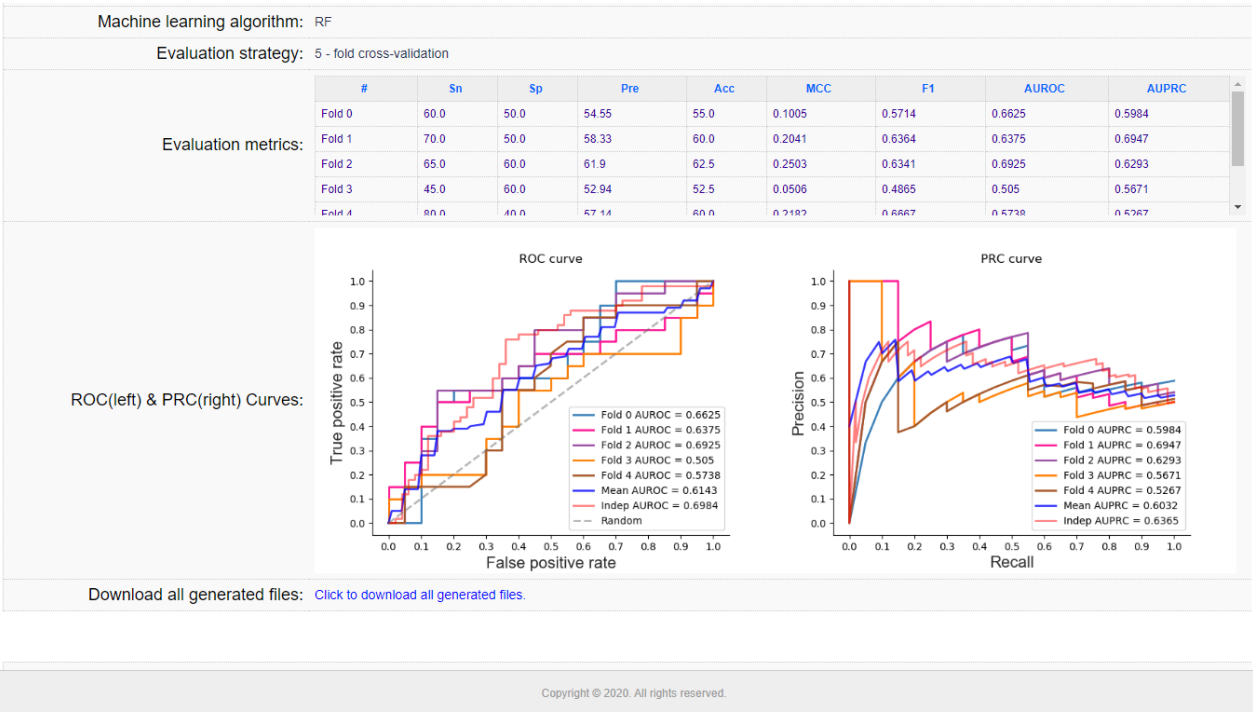
Clustering method: kmeans

Clustering result:
View all

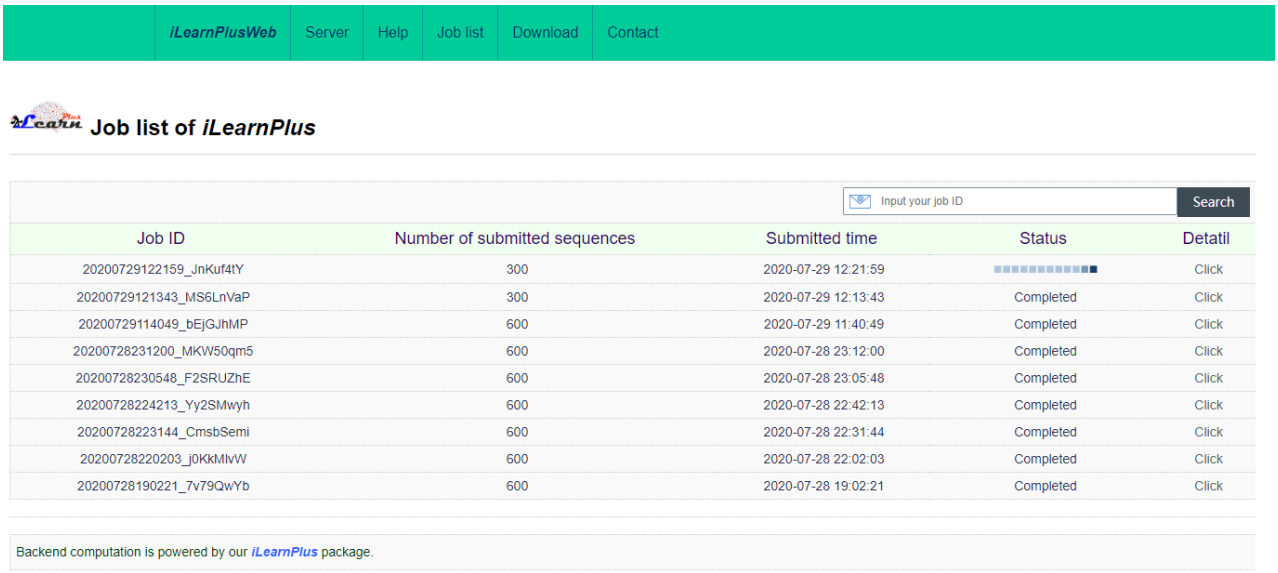
#	Cluster
AT1G22840.1_532	1
AT1G44000.1_976	2
AT1G09770.1_2698	1
AT1G09645.1_586	2
AT1G22850.1_1097	1

Visuallization of clustering result:





Step 10: Query your result.



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 accuracy of computations and was deliberately designed to ensure workflow efficiency. To the best of our

knowledge, this is the first both GUI (Graphical User Interface)- and web-based platform for building automated machine-learning pipelines, facilitating smooth analysis and predictions using DNA, RNA and protein sequences. We will integrate more analysis, clustering and machine-learning algorithms to enable interactive analysis and machine-learning-based modeling in the future work. 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.