BioData Mining

SOFTWARE

Open Access

SeqHIVE: a Python package to convert the biological sequences to informative vectors for sequence property predictions

Xin Feng[1,2,3], Cheng Gao[1], Sudan Bai[4], Jiaxin Zheng[2,5], Cuinan Yu[2,5], Kewei Li[2,5*], Lan Huang[2,5], Bo Han[7], Tao You[7], Jun Zhang[7] and Fengfeng Zhou[2,5,6*]

*Correspondence:
Kewei Li
kwbb1997@gmail.com
Fengfeng Zhou
FengfengZhou@gmail.com;
ffzhou@jlu.edu.cn

Full list of author information is available at the end of the article

Abstract

Sequence-based analysis and prediction form a cornerstone of bioinformatics investigations of the sequence-structure-function paradigm across DNA, RNA, and proteins. The exponential growth in sequence data necessitates sequence encoding methods and advanced predictive models. This study introduces a comprehensive machine learning and deep learning platform SeqHIVE, which aims to streamline the development of prediction pipelines for nucleic acid and protein sequences. SeqHIVE is implemented as a graphical user interface (GUI)–driven platform, enabling intuitive construction, evaluation, and visualization of sequence-based predictive models. SeqHIVE integrates a broad spectrum of 19 biological sequence encoding algorithms, 5 feature selection algorithms, and 26 classifiers for the biological sequence prediction tasks. It automates the processes of sequence-based feature extraction, model construction, predictive performance assessment, statistical analysis, and data visualization. SeqHIVE is engineered to serve both expert bioinformatics researchers with extensive customizable options, and biologists through a user-friendly interface and an intuitive design process. The utility of SeqHIVE is exemplified through a case study on iLearnPlus DNA locus prediction task. The tool SeqHIVE, the documentation, and its source code are freely available at: https://healthinformaticslab.org/supp/. and is also hosted on the GitHub repository: https://github.com/EienKune/SeqHIVE.

Keywords Bioinformatics, Feature engineering, Feature extraction, Python, Biological sequence

Introduction

The rapid advancements in high-throughput sequencing technologies have led to the production of vast quantities of biological sequence data in recent years [1]. Efficient analyses of these datasets have become a computational challenge for the interdisciplinary researchers [2]. Predictions of biological sequences with specific properties have traditionally relied on numerical feature engineering algorithms [3–6]. These algorithms are often referred to as feature extraction algorithms, and play a pivotal role in

transforming sequence data into a format acceptable to machine learning and deep learning algorithms.

Numerous bioinformatics tools have been developed to streamline the extraction of numerical features from biological sequences. For instance, a collaborative repository KIPOI offers predictive models for genomic analyses, including feature extraction capabilities [7]. Another tool iFeature [8] and its subsequent package iLearn [9] facilitate feature extraction and selection from protein and peptide sequences. These tools have been applied to various bioinformatics prediction tasks of circRNA genes [10], protein's modification sites [11], and drug-target interaction [12], etc. Some other feature extraction algorithms are detailed in Table 1, with data sourced from the article on MathFeature [13]. The computational efficiency and usability of these tools could be further enhanced with advanced programming techniques.

This study consolidates the advantages of the existing sequence feature extraction algorithms into the proposed platform SeqHIVE. SeqHIVE provides a Graphical User Interface (GUI) for conducting biological feature extraction and prediction tasks conveniently and comprehensively. The proposed platform consists of three main modules, i.e., feature engineering, feature selection, and prediction. The intermediate data generated in each main module are given in both numerical and visualized forms. The GUI-based platform ensures the user-friendly analysis procedures for both wet-lab biologists and interdisciplinary computer scientists. The main contributions of the proposed platform SeqHIVE could be summarized as follows.

- SeqHIVE provides a fully GUI-driven and modular workflow that enables users to construct, evaluate, and visualize complete sequence-based prediction pipelines

Table 1 Feature extraction tools for the biological sequences. We summarized the application scope, the numbers of mathematical descriptor (MD) features and the deep learning (DL)-embedded latent feature descriptors for each tool in the columns "Application", "MD" and "DL", respectively. The column "Total" gave the total number of sequence descriptors provided by this tool. The literature for each tool was given in the last column "Ref"

Tool	Application	MD	DL	Total	Ref
MathFeature	DNA + RNA + Protein	20	17	37	[13]
PROFEAT	Protein	0	2	2	[14]
PseAAC	Protein	0	2	2	[15]
Propy	Protein	0	5	5	[16]
PseKNC-general	DNA + RNA	0	5	5	[17]
SPiCE	Protein	0	4	4	[18]
ProtrWeb	Protein	0	5	5	[19]
ProFET	Protein	2	3	5	[20]
Pse-in-One	DNA + RNA + Protein	0	5	5	[21]
RepDNA	DNA	0	5	5	[22]
Rcpi	Protein	0	3	3	[23]
RepRNA	RNA	0	5	5	[24]
BioSeq-analysis	DNA + RNA + Protein	0	9	9	[25]
iFeature	Protein	1	4	5	[8]
Seq2Feature	DNA + Protein	0	0	0	[26]
PyBioMed	DNA + Protein	0	7	7	[27]
PyFeat	DNA + RNA + Protein	1	8	9	[28]
iLearn	DNA + RNA + Protein	2	13	15	[9]

without writing scripts or command-line instructions, thereby significantly lowering the technical barrier for non-expert users.

- SeqHIVE integrates both classical machine learning and deep learning classifiers within a unified framework, allowing flexible model selection, hyperparameter tuning, and direct performance comparison through an intuitive graphical interface.
- In contrast to most existing platforms, SeqHIVE supports advanced deep learning architectures, including bidirectional recurrent networks and attention-based models, which enhance the ability to capture long-range dependencies and contextual information in biological sequences.
- SeqHIVE offers a comprehensive collection of 19 sequence feature encoding algorithms, 5 feature selection algorithms, and 26 classification models, providing flexible combinations of mathematical descriptors and learning models to construct accurate predictors on user-customized datasets.

Materials and methods

Overall architecture of SeqHIVE

After loading the biological sequences as input to the proposed platform SeqHIVE, users are presented with the opportunity to interactively explore its three core modules via the graphical user interface (GUI): feature engineering, feature selection, and prediction, as depicted in Fig. 1.

This study focuses on binary classification tasks involving DNA, RNA, and protein sequences. SeqHIVE is engineered to provide seamless navigation across modules, facilitated by easy bi-directional transitions with the click of a button. This user-friendly architecture not only promotes exploratory analysis but also ensures that all intermediate results can be conveniently exported for further investigation as needed by users.

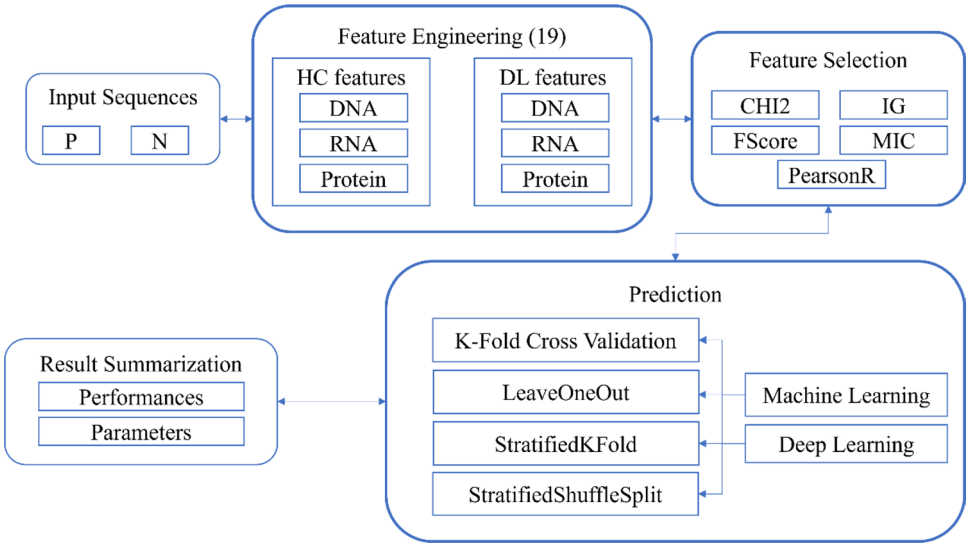


Fig. 1 Architecture of the SeqHIVE Platform. Users can effortlessly navigate between modules by simply clicking a button, allowing for bi-directional connections between any two steps

Feature encoding

Most of the existing sequence property predictors could not handle the biological sequences directly, and they usually extracted numerical descriptors from biological sequences via the feature encoding algorithms [6, 28].

SeqHIVE offers 19 feature encoding algorithms to extract features from biological sequences. Among these, four deep learning (DL)-based feature encoding algorithms are capable of processing all three types of biological sequences - DNA, RNA, and Protein. In contrast, mathematical descriptor features have limited applicability to DNA, RNA, and Protein sequences. Table 2 provides detailed information on the feature encoding algorithms applicable to each of the three types of biological sequences.

The Byte-Pair Encoding (BPE) algorithm was originally developed to address out-of-vocabulary (OOV) terms in various natural language processing (NLP) tasks [29]. BPE aggregates the most frequently occurring consecutive byte pairs to form new subwords. We utilize BPE to encode consecutive bases with independent significances into a single token.

The WordPiece tokenizer was proposed in the original pre-trained BERT model [30], and serves as another encoding strategy for biological sequences. It segments text into manageable and frequently occurring pieces, enhancing model performance on unseen data.

SentencePiece is a language-independent subword tokenizer designed for neural-based text processing [31]. Its primary advantage is its ability to operate directly on raw text without requiring pre-tokenization, and this feature makes SentencePiece highly versatile across different languages and scripts.

Lastly, WindowsSplit is a straightforward encoding method where sequences are simply divided into fixed-size windows [32]. This approach is particularly useful for tasks where local sequence information is crucial, as it ensures uniform coverage across the sequence.

However, fixed-window segmentation methods such as WindowsSplit exhibit inherent limitations when applied to biological sequence encoding [32]. By relying on predefined window sizes and fixed segmentation rules, these approaches have limited flexibility in accommodating biological functional units of varying lengths, such as regulatory elements, conserved motifs, or protein domains. As a result, important sequence patterns may be fragmented or incompletely represented when their characteristic lengths do not align with the predefined window configuration.

In contrast, data-driven subword tokenization methods such as BPE, WordPiece, and SentencePiece are able to adaptively learn variable-length sequence fragments from the underlying data distribution [29–31]. These fragments often correspond to frequently occurring and potentially meaningful sequence patterns. Such multi-scale representations enable more flexible modeling of contextual dependencies within sequences and facilitate the capture of latent biological semantics, thereby improving model expressiveness and generalization performance in downstream prediction tasks.

Table 2 Feature engineering algorithms for the biological sequences

	DL	Math
DNA	WindowsSplit, SentencePiece, WordPiece, BPE	NAC, Mismatch, RCKmer, Kmer, RCKmer+Mismatch
RNA	WindowsSplit, SentencePiece, WordPiece, BPE	Binary, NCP, ENAC, Kmer, ENAC+Binary
Protein	WindowsSplit, SentencePiece, WordPiece, BPE	AAC, EAAC, DPC, Kmer, AAC + DPC

For DNA sequences, five specific feature engineering algorithms have been designed: nucleic acid composition (NAC) [33], frequency of a k-mer with mismatches (Mismatch) [34], frequency of a k-mer and its reverse complementary form (RCKmer) [9], simple k-mer frequency (Kmer) [35], and frequency of a k-mer and its reverse complementary form with mismatches (RCKmer+Mismatch) [36]. These algorithms collectively enhance the representation of DNA sequences for computational analysis.

RNA sequences are processed using five slightly different feature engineering algorithms: nucleotide chemical property (NCP) [36], enhanced nucleic acid composition (ENAC) [37], binary encoding (Binary) [38], the combination of binary and ENAC encoding (ENAC+Binary) [9], and k-mer frequency (Kmer) [35]. This suite of algorithms is tailored to capture the unique structural and functional characteristics of RNA.

Peptide sequences are encoded with five feature engineering algorithms: amino acid composition (AAC) [39], enhanced amino acid composition (EAAC) [37], dipeptide composition (DPC) [40], the combination of k-mer composition and AAC (AAC + DPC) [41], and k-mer frequency (Kmer) [35]. These methods provide a comprehensive analysis of peptide sequences, emphasizing the importance of amino acid interactions and sequence patterns.

Prediction

The proposed platform SeqHIVE supports binary prediction models using both machine learning and deep learning algorithms. Biological sequence prediction tasks are typically formulated as binary classification problems for specific types of sequence elements, such as DNA replication origins [42] and protein phosphorylation [37].

SeqHIVE accommodates these prediction tasks with nine traditional machine learning classifiers: AdaBoost (Adaptive Boosting), Bagging (Bagging Classifier), Decision Tree, GaussianNB (Gaussian Naïve Bayes), KNeighbors (K Nearest Neighbors), LGBM (Light Gradient Boosting Machine), Logistic Regression (LR), Random Forest (RF), and SVM (Support Vector Machine). These algorithms are integrated from Python libraries such as scikit-learn, XGBoost, LightGBM, and libsvm. By default, the primary parameters of these algorithms are set to their optimal values, but they can also be manually adjusted through SeqHIVE's graphical user interface (GUI).

Furthermore, SeqHIVE offers neural network-based classifiers for these tasks. It provides a standard Multi-Layer Perceptron (MLP) for binary classification and implements both Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) in their original forms. Simple attention and multihead attention mechanisms are also available for both CNN and RNN classifiers. Additionally, two variants of RNNs, LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), are implemented. An encoder-decoder network is included to map features to output class labels. The deep learning models available are EncoderDecoder_Attention, CNN, CNN_Simple_Attention, CNN_Multihead_Attention, RNN, RNN(bi), RNN_Simple_Attention, RNN_Simple_Attention(bi), RNN_Multihead_Attention, RNN_Multihead_Attention(bi), GRU, GRU(bi), LSTM, LSTM(bi), MLP(down), MLP(up), and MLP(up2down). The suffixes "Attention" and "MultiheadAttention" indicate the inclusion of attention layers and multihead attention layers, respectively. The suffix "bi" indicates that the network is bidirectional. The suffixes "down", "up", and "up2down" refer to the architectural design of the MLP layers in terms of increasing or decreasing the number of neurons. "down" implies

a decreasing sequence of neurons, compacting the information as it progresses through the network. "up" involves an increasing sequence, and expands the information processing capacity. "up2down" combines these approaches, initially increasing and subsequently decreasing the number of neurons to refine the feature extraction.

In the SeqHIVE platform, users have the capability to adjust the hyperparameters of each classifier prior to the initiation of model training via the graphical user interface (GUI). The evaluation of the classification procedure can be conducted using either k-fold cross-validation (KFCV) or the leave-one-out (LeaveOneOut) method. Users can opt for either a stratified or random distribution of samples. After the training process, the platform enables users to save the configuration of the model that has achieved the highest accuracy, and thus preserves the most effective model settings for future applications.

Feature selection

Feature selection is a crucial step in removing redundant and irrelevant features to enhance the predictive power of models [43, 44]. This process is particularly beneficial for deep learning (DL)-based classification models, which can see significant improvements through effective feature selection [45]. The SeqHIVE platform incorporates five feature selection algorithms, namely chi-squared (CHI2), F1-score (F1), information gain (IG), maximum information coefficient (MIC), and Pearson correlation coefficient (Pearsonr). These feature selection methods are implemented using the Python library, sklearn.

Data visualization and statistical analysis

Visualization and statistical analyses are essential for exploring and identifying hidden patterns in the features extracted from biological sequences. The SeqHIVE platform offers these capabilities within its various data analysis modules. Upon loading the dataset and extracting features from the sequences, the distributions of sequence lengths and class percentages are visualized using box plots and pie charts, respectively. The features selected in the feature selection module are displayed in a tabular format. In the classification module, the prediction models are visualized using both receiver operating characteristic (ROC) curves and precision-recall (PRC) curves. Additionally, the values of the area under the ROC curve (AUROC) and the area under the PRC curve (AUPRC) are also depicted in the plots.

Overall process of software operations

SeqHIVE streamlines the model construction and result analysis for property prediction tasks involving DNA, RNA, and protein sequences. It employs a coherent workflow consisting of four modules: Encoding, Feature Selection (supporting non-DL encoding methods), Model Selection, and Result, as illustrated in Fig. 2. This structured approach enables efficient processing from data input through to performance evaluation.

1. Encoding: This module facilitates the initial step by enabling users to load and convert DNA/RNA/protein sequences under investigation into numerical feature vectors using embedded feature engineering algorithms.
2. Feature Selection: This module leverages five popular algorithms to selectively filter the numerical features derived from the Encoding module. The primary objective of

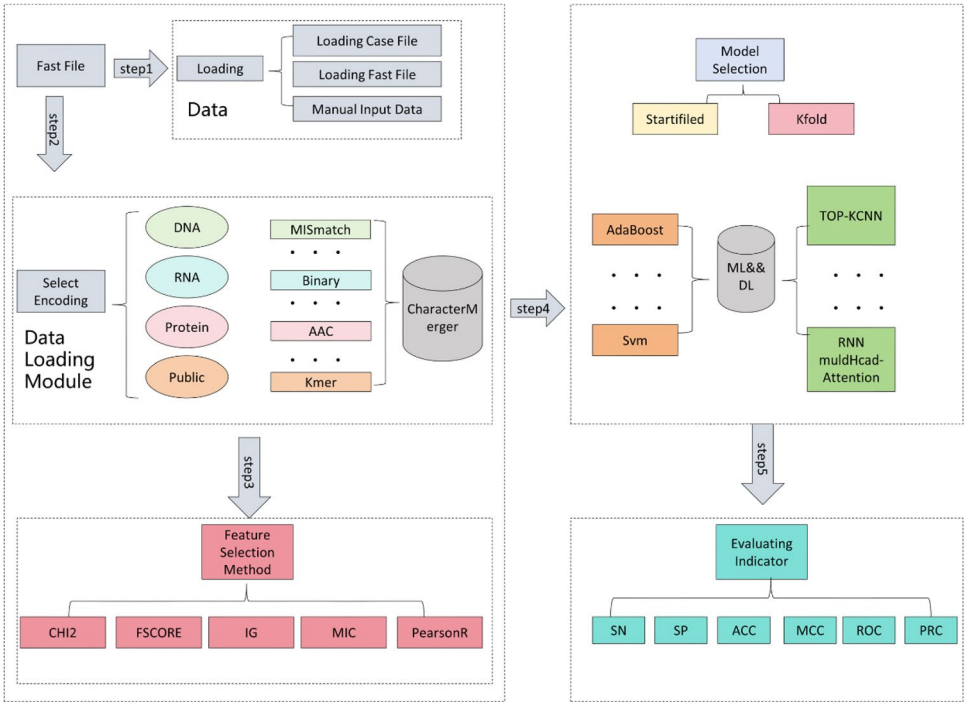


Fig. 2 The flow chart of the major SeqHIVE modules

this module is to identify and select features with the highest relevance rankings, and to ensure that the selected features contribute effectively to the model's predictive accuracy.

3. Model Selection: Supports eight deep learning classifiers and nine machine learning classifiers. Users can select their preferred classifiers for the specific prediction tasks.
4. Result: This module interacts closely with the Model Selection module to evaluate the performance of the chosen classifiers, ensuring the results are robust and reliable.

From a user-oriented perspective, SeqHIVE follows a streamlined interactive workflow consisting of sequence encoding, optional feature selection, model selection, and result analysis.

Taking DNA sequence analysis as an example, users first upload a prepared FASTA file through the "Upload FASTA file" button in the Encoding module. After confirmation, SeqHIVE automatically identifies the sequence type and displays basic dataset statistics in the "Sample Statistics" panel, including sample size, sequence length distribution, and training–testing partitioning, corresponding to Step 1 in Fig. 2.

Users then select either deep learning–based or non–deep learning–based encoding strategies (Step 2 in Fig. 2). For deep learning–based encoding, in addition to the conventional fixed-window segmentation method (WindowsSplit), SeqHIVE provides dynamic feature encoding strategies, including BPE, WordPiece, and SentencePiece, which adaptively learn variable-length sequence fragments. For non–deep learning–based encoding, users may apply a single feature engineering method or adopt feature fusion encoding to integrate multiple sequence descriptors into a unified representation (e.g., RCKmer combined with Mismatch).

When non–deep learning encoding strategies are adopted, users may proceed to the Feature Selection module (Step 3 in Fig. 2), where one of five filtering methods can be

selected. Users specify the desired number of features and initiate feature selection, after which the selected features are displayed in the interface. This module can be manually skipped. When deep learning–based encoding methods are used, the feature selection module is intentionally bypassed.

In the Model Selection module, users choose appropriate machine learning or deep learning models based on the selected encoding strategy, adjust parameters, configure validation schemes, and initiate training (Step 4 in Fig. 2). Deep learning–based encoding methods are paired with deep learning models, whereas non–deep learning encoding methods are paired with conventional machine learning models.

During training, performance metrics and visualizations are displayed in real time, including commonly used evaluation metrics in gene sequence prediction studies (Sn, Sp, Pre, Acc, MCC, F1, AUROC, and AUPRC) for each validation fold, along with ROC and PRC curves. After clicking "Next," SeqHIVE presents a summarized view of the model performance and additional details, corresponding to Step 5 in Fig. 2.

Throughout the workflow, users may adjust parameters, repeat experiments, or return to previous steps using the "Previous" button, enabling flexible exploration and iterative optimization of the analysis process.

Performance evaluation metrics

This work utilizes both the k-fold cross-validation (KFCV) strategy and the independent validation (IV) strategy to evaluate classification performance. The KFCV strategy involves randomly dividing a dataset into k subsets. Each subset is used sequentially as the test set while the remaining (k-1) subsets serve as the training sets. The IV strategy employs an independent test set to evaluate the model trained via the KFCV strategy.

SeqHIVE generates a confusion matrix to aid in the evaluation of the trained prediction model. Seven standard metrics are calculated for model evaluation. In the context of binary classification, SeqHIVE calculates sensitivity (Sn), specificity (Sp), accuracy (Acc), Matthew's Correlation Coefficient (MCC), F1-score (F1), and the areas under the receiver operating characteristic (ROC) curve and the precision-recall curve (PRC). These metrics are derived from the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The effectiveness of a model is indicated by high values of the areas under the ROC and PRC curves. The binary classification model is evaluated based on the accuracy of each class label and the overall accuracy.

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{2}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

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

$$\mathrm{AUC} = \frac{\sum_{i \in \ \mathrm{PositiveClass}} \mathrm{rank_i} \ - \ \frac{P(1+P)}{2}}{P \ \times \ N} \tag{6}$$

$$\mathrm{F1-Score} = \frac{2 \times \ \mathrm{Precision} \times \ \mathrm{Recall}}{\mathrm{Precision} + \ \mathrm{Recall}} \ = \frac{2TP}{2 \ \times \ \ TP \ + \ FP \ + \ FN} \tag{7}$$

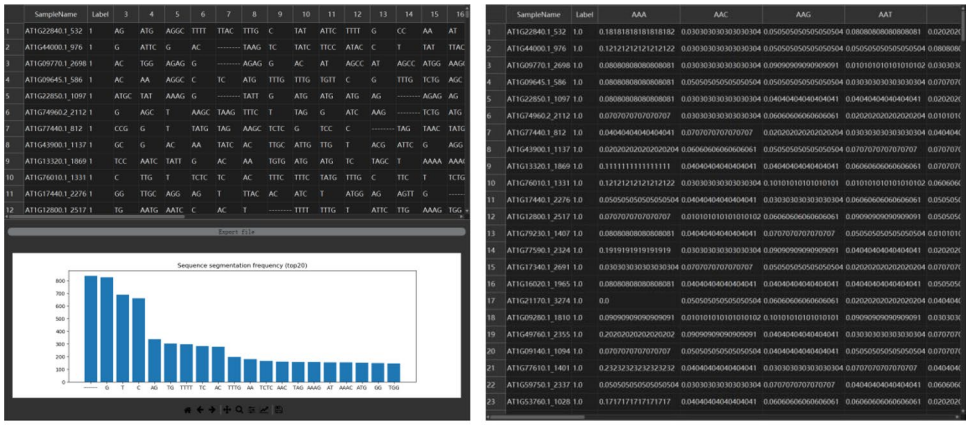
Implementation and testing details

SeqHIVE uses the test case given in iLearnPlus [37], where the training set contains 200 DNA base sequences, and the test set contains 100 DNA base sequences. The kmer code sliding window is set to 3, and the 5-fold cross-validation is used. The performance indicators of the AdaBoost algorithm under the independent verification set are: sn = 0.88, sp = 0.7, Acc = 0.79, Mcc = 0.5896, F1 = 0.8344, using the NAC encoding method, the AdaBoost algorithm under the 5-fold cross-validation is independent.

Results and discussion

Feature encoding module

This module provides two core feature encoding functionalities: feature dynamic encoding(Fig. 3(a)) and feature fusion encoding (Fig. 3(b)). Feature dynamic encoding adapts to the variability in sequence structures, and allows the model to capture inherent sequence-specific patterns. While the feature fusion encoding combines multiple types of sequence encoding data into a unified representation, and enhances the model's ability to integrate diverse biological signals and improve predictive performance. Together, these encoding strategies provide a robust foundation for complex biological sequence analyses within SeqHIVE, and enables tailored and comprehensive sequence feature extraction.



(a) (b)

Fig. 3 Feature encoding module of SeqHIVE. (**a**) Feature dynamic encoding using the BPE tokenizer. Each row corresponds to one DNA sample and its tokenized sequence representation. The first column indicates the sample name, the second column denotes the class label, and the remaining columns present the variable-length tokens generated from sequence segmentation. The bar chart below shows the top 20 most frequent segmented tokens across all samples. The x-axis represents the tokens ranked in descending order of frequency, and the y-axis indicates their occurrence counts. (**b**) Feature fusion encoding using the RCKmer+Mismatch method. Each row represents one DNA sample. The first column shows the sample name, the second column indicates the class label, and the remaining columns correspond to the concatenated feature vectors generated by the RCKmer and Mismatch methods. Specifically, the RCKmer part provides normalized k-mer frequencies (including reverse complementary merging), whereas the Mismatch part provides mismatch-based k-mer occurrence counts. These two feature sets are combined into a unified representation for downstream modeling

SeqHIVE loads the example data through the button of "Load example FASTA". Figure 3 (a) and (b) provide the choices of dynamic encoding BPE and fusion encoding RCKmer+Mismatch, respectively. The right panels of Fig. 3 (a) and (b) show the encoded results.

Feature selection module

The feature selection module is available when fusion coding or machine learning-based encoding methods are applied within this software. Five widely used feature selection techniques are supported: CHI2, Information Gain (IG), F-Score, Maximum Information Coefficient (MIC), and Pearson correlation coefficient (Pearsonr). Users can specify the number of top features K to retain by setting the "Selected Feature Number". The software then filters the top K features based on the selected method, and optimizes the feature set for subsequent model construction.

For example, when CHI2 is chosen as the feature selection method and K = 5, with the RCKmer + Mismatch coding scheme, the top five selected features are TAT, TTA, TTT, ATT, and TGT, as presented in the software SeqHIVE (Fig. 4 (b)). These selected

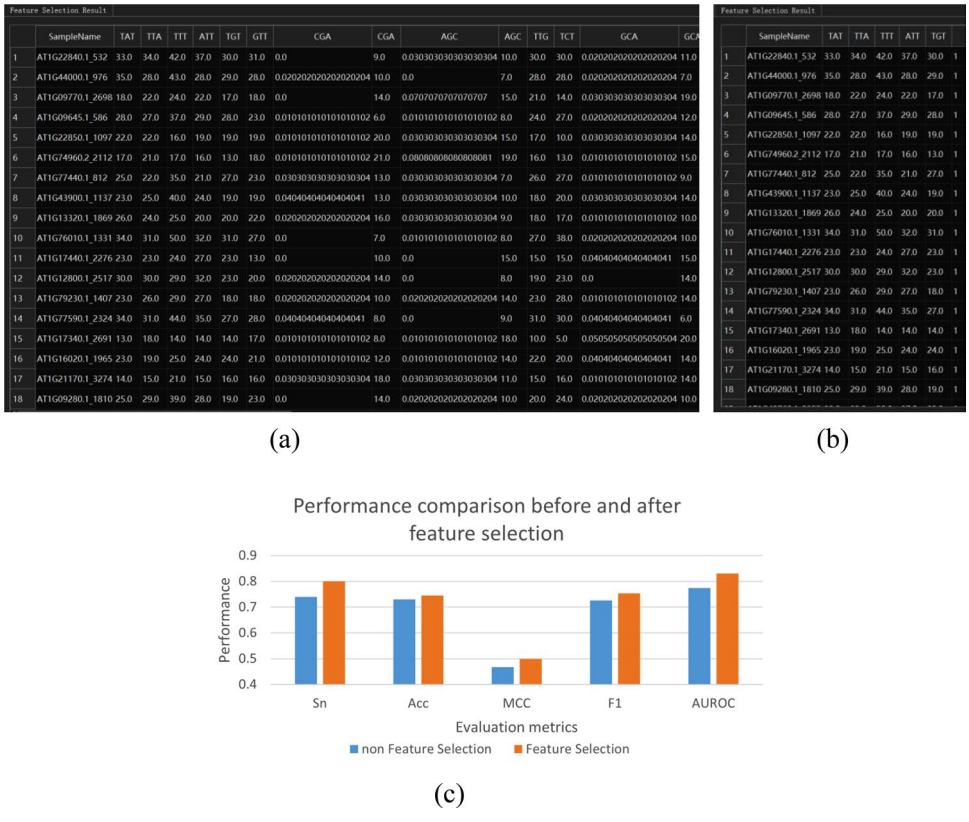


Fig. 4 (**a**) Interface without feature selection. By setting K = 100, which exceeds the total number of available features, all features are retained and displayed. Each row corresponds to one DNA sample. The first column indicates the sample name, the last column denotes the class label, and the intermediate columns represent feature values generated by the encoding method. (**b**) Interface after applying CHI2 feature selection with K = 5. Only the top five ranked features (TAT, TTA, TTT, ATT, and TGT) are retained and displayed. The table structure is the same as in (**a**), where each row represents one DNA sample, the first column indicates the sample name, the last column denotes the class label, and the intermediate columns correspond to the selected feature values. (**c**) Performance comparison before and after feature selection. The x-axis represents evaluation metrics (Sn, Acc, MCC, F1, and AUROC), and the y-axis denotes performance scores

features provide a refined input set, and enhance model accuracy and interpretability in predictive tasks.

Optimization of predictive models

After encoding the sequences into numerical features, users have the flexibility to perform feature selection based on their chosen encoding methods and preferences. Following feature selection, users can partition the dataset to suit their analysis needs. SeqHIVE provides four data partitioning options to accommodate various dataset sizes and formats: KFold, stratifiedKFold, LeaveOneOut, and stratifiedShuffleSplit.

SeqHIVE recommends stratifiedKFold the default method, and sets cross-validation to 5 folds to train multiple models. The software supports 17 deep learning models and 9 machine learning models. Users may select the most suitable models for their chosen encoding methods and desired outcomes.

The feature dynamic coding module leverages the Byte-Pair Encoding (BPE) and WordPiece algorithms and provides 17 deep learning models: Attention, CNN, CNN_Simple_Attention, CNN_Multihead_Attention, RNN, RNN(bi), RNN_Simple_Attention, RNN_Simple_Attention(bi), RNN_Multihead_Attention, RNN_Multihead_Attention(bi), GRU, GRU(bi), LSTM, LSTM(bi), MLP(down), MLP(up), and MLP(up2down). Users can choose different models based on their encoding approach. Results from different encoding-method combinations are presented in Fig. 5 (a)-(d). For

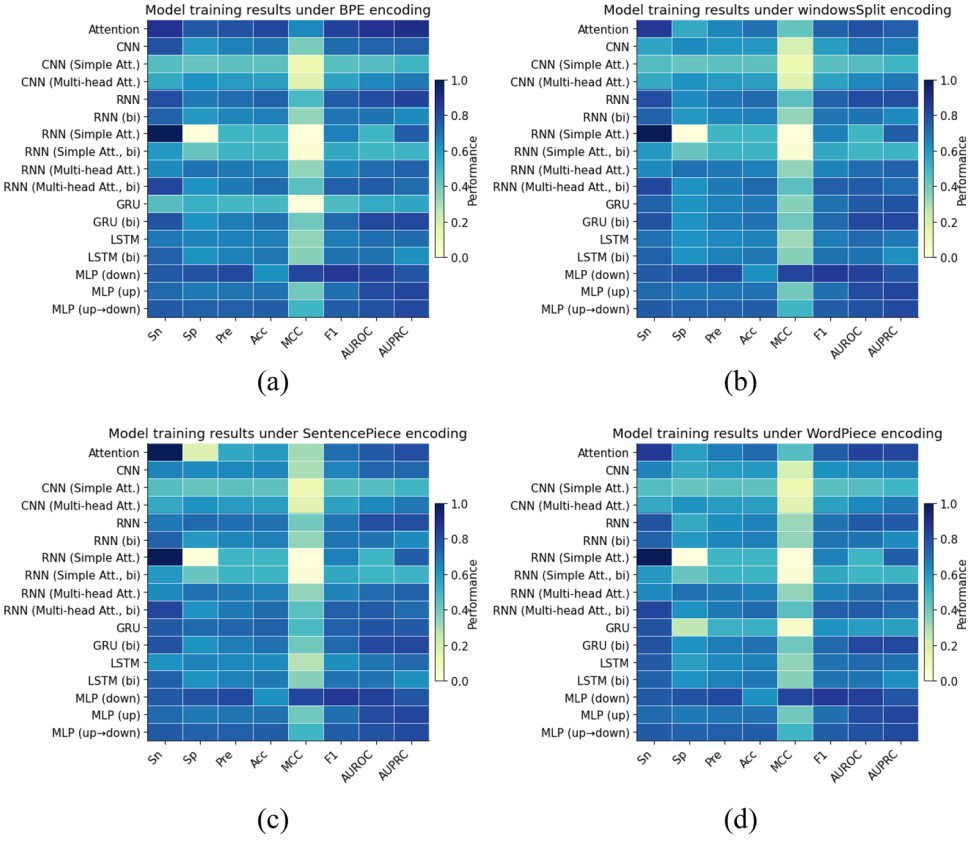


Fig. 5 Performance evaluation of feature dynamic encoding strategies. The heatmaps illustrate the performance of different models under (**a**) BPE, (**b**) windowsSplit, (**c**) SentencePiece, and (**d**) WordPiece encoding strategies. Rows correspond to prediction models and columns represent evaluation metrics, including Sn, Sp, Precision (Pre), Accuracy (Acc), MCC, F1, AUROC, and AUPRC. Darker colors indicate higher performance values

the test dataset provided, the combination of BPE encoding with the Attention model achieved the highest AUROC, reaching 0.8812.

Feature fusion encoding module integrates several classical encoding methods. The DNA dataset used in this study primarily employs encoding methods such as Kmer, RCKmer, Mismatch, NAC, and RCKmer + Mismatch. Model training results for each encoding method are detailed in Fig. 6 (a)-(e). For the case dataset provided, the fusion encoding method Kmer paired with RandomForest achieves the highest AUROC of 0.8614. Additionally, other encoding-method combinations demonstrated strong

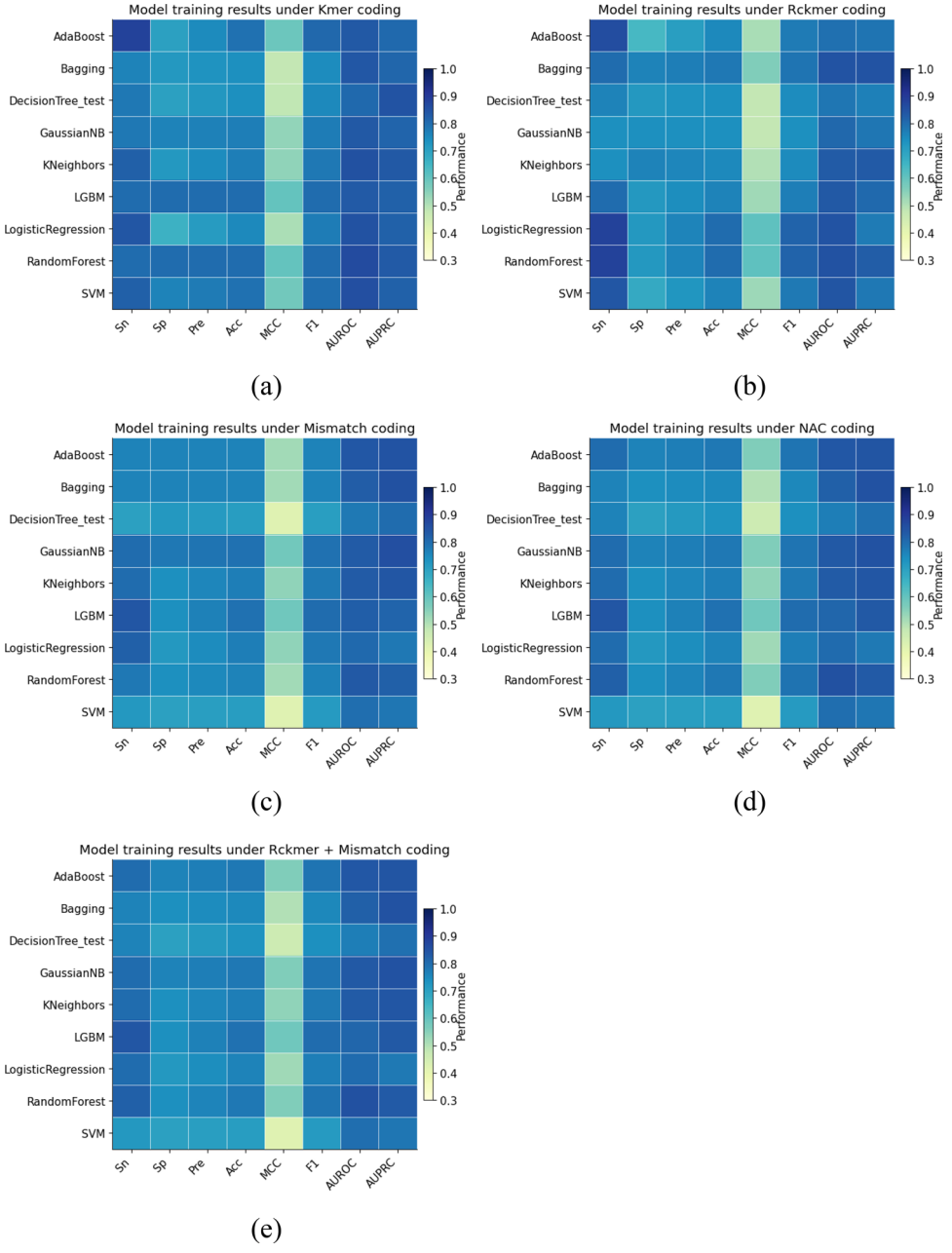


Fig. 6 Performance evaluation of feature fusion encoding strategies. The heatmaps illustrate the performance of different models under (**a**) Kmer, (**b**) Rckmer, (**c**) Mismatch, (**d**) NAC, and (**e**) Rckmer+Mismatch encoding strategies. Rows correspond to prediction models and columns represent evaluation metrics, including Sn, Sp, Precision (Pre), Accuracy (Acc), MCC, F1, AUROC, and AUPRC. Darker colors indicate higher performance values

performance, with most yielding AUROC values above 0.8, indicating robust predictive accuracy across models.

Performance comparisons with existing studies

This study introduces SeqHIVE, an advanced machine learning and deep learning platform tailored for the processing of biological sequences, including DNA, RNA, and protein data. SeqHIVE emphasizes user-friendly workflows and robust classification methods, and it's designed to enhance bioinformatics research efficiency. To validate the platform's effectiveness, we conducted detailed case studies on DNA, RNA, and protein sequences, as described in the experimental section. Due to inherent differences among biological sequence types and the varying scope of existing tools, in this study, we selected domain-specific datasets relevant to real-world applications for DNA, RNA, and protein tasks, respectively. For each sequence type, evaluation strategies consistent with commonly adopted practices in the corresponding research community were applied, enabling fair and meaningful comparisons with existing tools, models, and methods where overlapping functionality exists.

An overview of the benchmark datasets, evaluation strategies, and comparison targets used in each case study is summarized in Table 3.

In the case study of DNA analysis [46], we employed the high-quality sigma70 promoter DNA benchmark dataset, containing 741 positive and 1,400 negative samples. Using the Kmer method for feature selection and Support Vector Machine (SVM) for classification, we achieved high predictive accuracy through five-fold cross-validation. SeqHIVE performance outperformed that of MathFeature, and achieves an accuracy (Acc) of 0.7775, a Matthews correlation coefficient (MCC) of 0.4958, and an F1 of 0.8361, compared to MathFeature's values of 0.8595, 0.6852, and 0.7872, respectively. SeqHIVE demonstrates a higher F1-score, indicating a more balanced trade-off between precision and recall. This observation suggests that SeqHIVE may be more effective in identifying positive samples, which can be particularly advantageous in application scenarios where correct detection of positive instances is of primary interest. Although MathFeature achieves higher Acc and MCC values, the results suggest that SeqHIVE provides competitive and complementary performance, particularly in scenarios where F1-score is a critical evaluation metric.

For the RNA-based prediction task [47], we utilized a benchmark dataset comprising 18,000 lncRNA and 18,000 mRNA samples, and partitioned the dataset into 80% for training and 20% for testing. The Kmer method facilitated feature extraction, and the Random Forest (RF) served as the classifier. This combination achieved an accuracy of 0.8018 and an F1 of 0.6085, and outperformed several established deep learning models, including lncRNAnet [48] (Acc: 0.7290, F1: 0.7260) and LncADeep [49] (Acc: 0.8000, F1: 0.7690). These results indicate that SeqHIVE's machine learning capabilities can surpass

Table 3 Benchmark datasets, evaluation strategies, and comparison targets for DNA, RNA, and protein sequence prediction tasks

Sequence	Benchmark Dataset	Evaluation Strategy	Compared Tools/Models
DNA	sigma70 promoter	5-fold CV	MathFeature
RNA	lncRNA / mRNA	80/20 split	lncRNAnet, LncADeep
Protein	PVP dataset	Independent test	PVPred, PVP-SVM, PVPred-scm

even complex deep learning approaches in certain RNA classification tasks, underscoring its flexibility and efficacy.

Using the phage virion protein (**PVP)** dataset [50], we evaluated SeqHIVE's logistic regression (LR) model for classifying protein sequences. The dataset includes 500 training sequences (250 PVP and 250 non-PVP) and 126 testing sequences (63 PVP and 63 non-PVP). The **Kmer** method facilitated feature extraction, and demonstrated efficiency and accuracy in processing protein data. The LR classifier achieved an accuracy of 0.7300 and an MCC of 0.5181, and this model outperformed existing methods such as PVPred [51] (Acc: 0.7300, MCC: 0.5050), PVP-SVM [52] (MCC: 0.5050), and PVPred-scm [53] (Acc: 0.7140, MCC: 0.4320). These results emphasize SeqHIVE's robust capabilities in protein sequence detection.

A quantitative performance comparison between SeqHIVE and existing tools, models, and methods across the three case studies is summarized in Table 4, providing an intuitive overview of the comparative results.

These comparisons demonstrate SeqHIVE's advantages across diverse application scenarios, and offer empirical support for its potential to drive new insights in bioinformatics. The release of SeqHIVE is anticipated to significantly benefit the field, enabling more efficient analysis and fostering advances in related research.

The SeqHIVE platform was rigorously evaluated through the case studies on DNA, RNA, and protein data. Each classification task utilized datasets relevant to real-world applications: the sigma70 promoter dataset for gene expression studies, the lncRNA dataset for gene regulation and disease research, and the PVP dataset for antimicrobial protein research. SeqHIVE leverages its 19 feature encoding algorithms and 26 classifiers to enhance model accuracy and robustness. It should be noted that the experiments designed for DNA, RNA, and protein tasks are intended as independent validations of SeqHIVE within their respective biological contexts. Each case study follows benchmarking practices that are widely accepted in the corresponding research domain and among existing tools, models, and methods, rather than aiming for direct cross-domain performance comparison. These studies validate SeqHIVE's flexibility and performance across omics data, offering valuable insights and advancing bioinformatics applications.

Visualization capabilities

SeqHIVE's visualization tools provide critical insights at every stage of data processing and allow users to interpret results both numerically and graphically (Fig. 7). Users can download all tabular and graphical outputs directly from the platform's panel.

Table 4 Performance comparison of SeqHIVE with existing tools, models, and methods on DNA, RNA, and protein benchmark datasets

Sequence	Tools/Models	Acc	MCC	F1
DNA	SeqHIVE	0.7775	0.4958	0.8361
	MathFeature	0.8595	0.6852	0.7872
RNA	SeqHIVE	0.8018	/	0.6085
	lncRNAnet	0.7290	/	0.7260
	LncADeep	0.8000	/	0.7690
Protein	SeqHIVE	0.7300	0.5181	/
	PVPred	0.7300	0.5050	/
	PVP-SVM	/	0.5050	/
	PVPred-scm	0.7140	0.4320	/

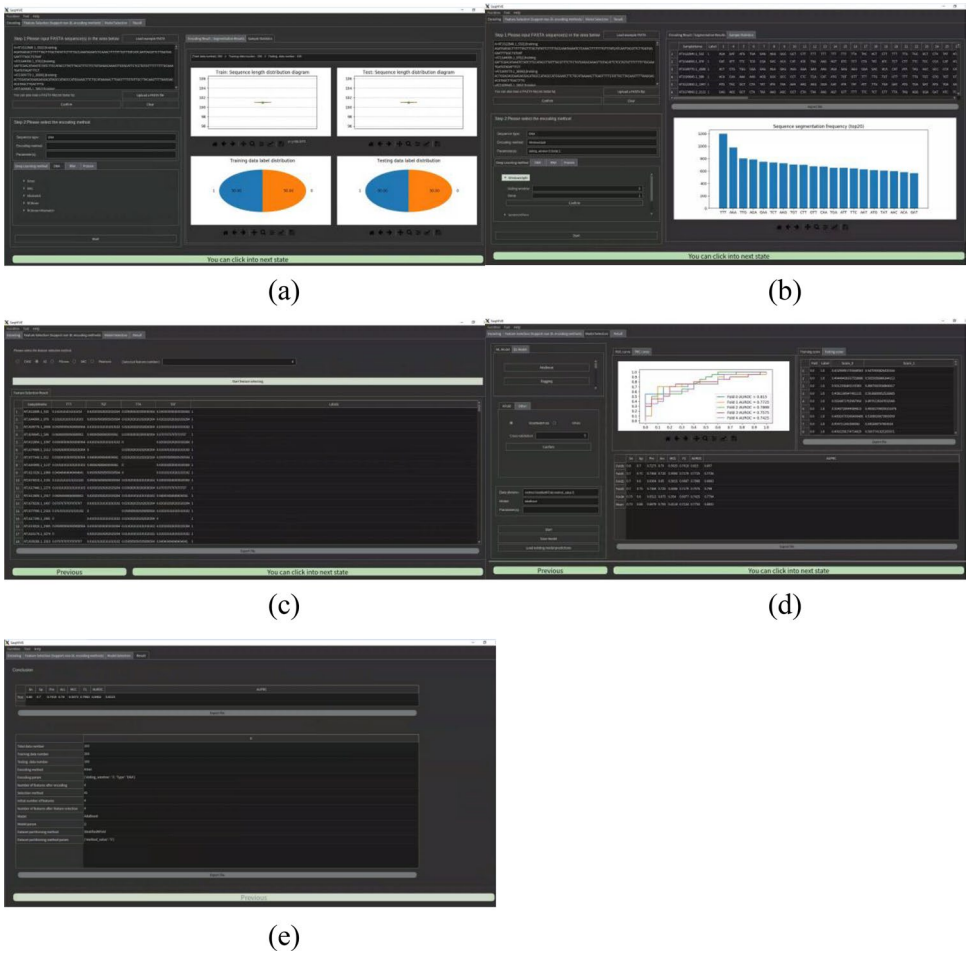


Fig. 7 Visualization capabilities of SeqHIVE. (**a**) Coding statistics. (**b**) Segmentation frequencies. (**c**) The selected features. (**d**) Performance metrics of the generated predictive models. (**e**) Embedding visualization toolkit

The encoding module visualizes the distributions of sequence length and class labels (Fig. 7(a)), as well as the frequencies of the top 20 segments generated during encoding (Fig. 7(b)), facilitating data inspection and encoding assessment. The feature selection module lists the selected features together with their detailed information in a sorted table (Fig. 7(c)), enabling transparent feature inspection. The model selection module visualizes the performance metrics of the generated models for both machine learning and deep learning approaches (Fig. 7(d)). The result module displays the performance of the trained model on the test set with the corresponding parameters (Fig. 7(e)). In addition, the embedding visualization toolkit enables intermediate embedding representations learned during training to be projected for clustering and exploratory analysis of sample distributions.

Conclusions

This study presents SeqHIVE, a versatile machine learning and deep learning platform tailored for analyzing biological sequences, including DNA, RNA, and protein data. SeqHIVE integrates 19 feature encoding algorithms, 5 feature selection techniques, and 26 classifiers, providing a robust toolkit for bioinformatics researchers. The platform's user-friendly interface and modular design simplify complex sequence analysis

workflows, from encoding and feature selection to model training and performance evaluation. In case studies on DNA, RNA, and protein data, SeqHIVE demonstrated high predictive accuracy, and confirmed its potential as a powerful tool for various bioinformatics applications.

However, SeqHIVE has certain limitations. Its performance relies on the quality and quantity of training data, which can be a constraint in biological datasets that are limited or imbalanced. Regarding class imbalance, SeqHIVE currently adopts stratified sampling strategies during data partitioning to preserve class distributions across training and validation sets. Future versions of SeqHIVE will consider incorporating additional imbalance-aware techniques, such as cost-sensitive learning and data resampling strategies.

Additionally, while SeqHIVE offers a range of deep learning models, the interpretability of these models remains a challenge, particularly for clinical applications where model transparency is essential. Future enhancements may focus on expanding SeqHIVE's capabilities for handling diverse data types and improving model interpretability to support its application in translational bioinformatics research.

In terms of future extensions, SeqHIVE is designed with extensibility in mind. Planned developments include support for multi-class classification and regression tasks, as well as enhanced scalability through potential integration with cloud-based or high-performance computing (HPC) environments to accommodate large-scale biological datasets.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13040-026-00534-4.

Supplementary Material 1

Acknowledgements

This work is supported by the National Natural Science Foundation of China Mathematics Tianyuan Fund Project (12326377), the Natural Science Foundation of Jilin Province (YDZJ202301ZYTS401).

Author contributions

Xin Feng: Conceived the study, Designed and performed the experiment, and wrote the original draft. Cheng Gao: Co-designed the experiment, conducted core experiments, and participated in drafting the original manuscript. Sudan Bai: Design experiments, Test experiments, Edited the manuscript. Jiaxin Zheng: Data Collection and Visualization Interface. Cuinan Yu: Visualization Interface. Kewei Li: Visualization Interface. Lan Huang: Designed and performed the experiments. Bo Han: Data Collection. Tao You: Visualization Interface. Jun Zhang: Data Collection. FengFeng Zhou: Conceived the whole study, Edited the manuscript, Evaluation, Case studies, and Manuscript.

Data availability

The tool SeqHIVE, the documentation, and its source code are freely available at: https://healthinformaticslab.org/supp/ and is also hosted on the GitHub repository: https://github.com/EienKune/SeqHIVE.

Declarations

Competing interests

The authors declare no competing interests.

Author details

[1]School of Artificial Intelligence, Jilin University of Chemical Technology, Jilin 130000, PR China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China
[3]Department of Epidemiology and Biostatistics, School of Public Health, Jilin University, Changchun 130012, China
[4]College of Information and Control Engineering, Jilin University of Chemical Technology, Jilin 132000, China
[5]College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China
[6]School of Biology and Engineering, Guizhou Medical University, Guiyang, Guizhou 550025, China
[7]Jilin City Hospital of Chemical Industry, Jilin 130000, P.R. China

Received: 23 December 2025 / Accepted: 20 February 2026
Published online: 11 April 2026

References

1. Yang B, Yang Y, Su X. Deep structure integrative representation of multi-omics data for cancer subtyping[J]. Bioinformatics. 2022;38(13):3337–42.
2. Andrews TD, Jeelall Y, Talaulikar D, et al. DeepSNVMiner: a sequence analysis tool to detect emergent, rare mutations in subsets of cell populations[J]. PeerJ. 20164: e2074.
3. Peng X, Wang X, Guo Y, et al. RBP-TSTL is a two-stage transfer learning framework for genome-scale prediction of RNA-binding proteins[J]. Brief Bioinform. 2022;23(4):bbac215.
4. Feng X, Ma Z, Yu C, et al. MRNDR: multihead attention-based recommendation network for drug repurposing[J]. J Chem Inf Model. 2024;64(7):2654–69.
5. Yan K, Lv H, Guo Y, et al. TPpred-ATMV: therapeutic peptide prediction by adaptive multi-view tensor learning model[J]. Bioinformatics. 2022;38(10):2712–8.
6. Zhang Y, Zhu G, Li K, et al. HLAB: learning the BiLSTM features from the ProtBert-encoded proteins for the class I HLA-peptide binding prediction[J]. Brief Bioinform. 2022;23(5):bbac173.
7. Avsec Ž, Kreuzhuber R, Israeli J, et al. The Kipoi repository accelerates community exchange and reuse of predictive models for genomics[J]. Nat Biotechnol. 2019;37(6):592–600.
8. Chen Z, Zhao P, Li F, et al. iFeature: a python package and web server for features extraction and selection from protein and peptide sequences[J]. Bioinformatics. 2018;34(14):2499–502.
9. Chen Z, Zhao P, Li F, et al. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data[J]. Brief Bioinform. 2020;21(3):1047–57.
10. Niu M, Wang C, Chen Y, et al. CircRNA identification and feature interpretability analysis[J]. BMC Biol. 2024;22(1):44.
11. Meng L, Chan WS, Huang L, et al. Mini-review: recent advances in post-translational modification site prediction based on deep learning[J]. Comput Struct Biotechnol J. 2022;20:3522–32.
12. Ma J, Li C, Zhang Y, et al. MULGA, a unified multi-view graph autoencoder-based approach for identifying drug–protein interaction and drug repositioning[J]. Bioinformatics. 2023;39(9):btad524.
13. Bonidia RP, Domingues DS, Sanches DS, et al. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors[J]. Brief Bioinform. 2022;23(1):bbab434.
14. Li ZR, Lin HH, Han LY, et al. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence[J]. Nucleic Acids Res. 2006;34(suppl2):W32–7.
15. Shen HB, Chou KC. PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition[J]. Anal Biochem. 2008;373(2):386–8.
16. Cao DS, Xu QS, Liang YZ. propy: a tool to generate various modes of Chou's PseAAC[J]. Bioinformatics. 2013;29(7):960–2.
17. Chen W, Zhang X, Brooker J, et al. PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions[J]. Bioinformatics. 2015;31(1):119–20.
18. Bayley A. A summary of current DNA methods for herb and spice identification[J]. J AOAC Int. 2019;102(2):386–9.
19. Xiao N, Cao DS, Zhu MF, et al. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences[J]. Bioinformatics. 2015;31(11):1857–9.
20. Ofer D, Linial M, ProFET. Feature engineering captures high-level protein functions[J]. Bioinformatics. 2015;31(21):3429–36.
21. Liu B, Liu F, Wang X, et al. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences[J]. Nucleic Acids Res. 2015;43(W1):W65–71.
22. Liu B, Liu F, Fang L, et al. repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects[J]. Bioinformatics. 2015;31(8):1307–9.
23. Cao DS, Xiao N, Xu QS, et al. Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions[J]. Bioinformatics. 2015;31(2):279–81.
24. Bin Liu F, Liu L, Fang X, Wang, Kuo-Chen Chou. reprna: a web server for generating various feature vectors of rna sequences. Mol Genet Genomics. 2016;291(1):473–81.
25. Liu B. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches[J]. Brief Bioinform. 2019;20(4):1280–94.
26. Nikam R, Gromiha MM. Seq2Feature: a comprehensive web-based feature extraction tool for biological sequence data[J]. Bioinformatics. 2019;35(22):4797–9.
27. Dong J, Yao ZJ, Zhang L, et al. PyBioMed: a python library for various molecular representations of chemicals, proteins and DNAs and their interactions[J]. J Cheminform. 2018;10:1–11.
28. Muhammod R, Ahmed S, Md Farid D, et al. PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences[J]. Bioinformatics. 2019;35(19):3831–3.
29. Bostrom K, Durrett G. Byte pair encoding is suboptimal for language model pretraining[J]. arXiv preprint arXiv:2004.03720. 2020.
30. Song X, Salcianu A, Song Y, et al. Fast wordpiece tokenization[J]. arXiv preprint arXiv:2012.15524. 2020.
31. Choo S, Kim W. A study on the evaluation of tokenizer performance in natural language processing[J]. Appl Artif Intell. 2023;37(1):2175112.
32. Becker F, Li ZL. Towards a local split window method over land surfaces[J]. Int J Remote Sens. 1990;11(3):369–93.
33. Wu Y. Google's neural machine translation system: Bridging the gap between human and machine translation[J]. arXiv preprint arXiv:1609.08144. 2016.
34. Kudo T, Sentencepiece. A simple and language independent subword tokenizer and detokenizer for neural text processing[J]. arXiv preprint arXiv:1808.06226. 2018.
35. Liu B, Gao X, Zhang H. BioSeq-Analysis2. 0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches[J]. Nucleic Acids Res. 2019;47(20):e127–127.
36. Noble WS, Kuehn S, Thurman R, et al. Predicting the in vivo signature of human gene regulatory sequences[J]. Bioinformatics. 2005;21(suppl1):i338–43.
37. Chen Z, Zhao P, Li C, et al. iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization[J]. Nucleic Acids Res. 2021;49(10):e60–60.
38. Chen W, Tran H, Liang Z, et al. Identification and analysis of the N6-methyladenosine in the Saccharomyces cerevisiae transcriptome[J]. Sci Rep. 2015;5(1):13859.

39. Chen Z, Zhou Y, Song J, et al. hCKSAAP_UbSite: improved prediction of human ubiquitination sites by exploiting amino acid pattern and properties[J]. Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics. 2013;1834(8): 1461–7.
40. Bhasin M, Raghava GPS. Classification of nuclear receptors based on amino acid composition and dipeptide composition[J]. J Biol Chem. 2004;279(22):23262–6.
41. Zhou C, Wang C, Liu H, et al. Identification and analysis of adenine N 6-methylation sites in the rice genome[J]. Nat plants. 2018;4(8):554–63.
42. Saravanan V, Gautham N. Harnessing computational biology for exact linear B-cell epitope prediction: a novel amino acid composition-based feature descriptor[J]. OMICS. 2015;19(10):648–58.
43. Lou C, Zhao J, Shi R, et al. sefOri: selecting the best-engineered sequence features to predict DNA replication origins[J]. Bioinformatics. 2020;36(1):49–55.
44. Ayati M, Yılmaz S, Chance MR, et al. Functional characterization of co-phosphorylation networks[J]. Bioinformatics. 2022;38(15):3785–93.
45. Gao S, Wang P, Feng Y, et al. RIFS2D: A two-dimensional version of a randomly restarted incremental feature selection algorithm with an application for detecting low-ranked biomarkers[J]. Comput Biol Med. 2021;133:104405.
46. Lin H, Liang ZY, Tang H, et al. Identifying sigma70 promoters with novel pseudo nucleotide composition[J]. IEEE/ACM Trans Comput Biol Bioinf. 2017;16(4):1316–21.
47. Meng J, Kang Q, Chang Z, et al. PIncRNA-HDeep: plant long noncoding RNA prediction using hybrid deep learning based on two encoding styles[J]. BMC Bioinformatics. 2021;22(Suppl 3):242.
48. Baek J, Lee B, Kwon S, et al. LncRNAnet: long non-coding RNA identification using deep learning[J]. Bioinformatics. 2018;34(22):3889–97.
49. Yang C, Yang L, Zhou M, et al. LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning[J]. Bioinformatics. 2018;34(22):3825–34.
50. Charoenkwan P, Nantasenamat C, Hasan MM, et al. Meta-iPVP: a sequence-based meta-predictor for improving the prediction of phage virion proteins using effective feature representation[J]. J Comput Aided Mol Des. 2020;34(10):1105–16.
51. Ding H, Feng PM, Chen W, et al. Identification of bacteriophage virion proteins by the ANOVA feature selection and analysis[J]. Mol Biosyst. 2014;10(8):2229–35.
52. Manavalan B, Shin TH, Lee G. PVP-SVM: sequence-based prediction of phage virion proteins using a support vector machine[J]. Front Microbiol. 2018;9:476.
53. Charoenkwan P, Kanthawong S, Schaduangrat N, et al. PVPred-SCM: improved prediction and analysis of phage virion proteins using a scoring card method[J]. Cells. 2020;9(2):353.

Publisher's note