NeuralTE: an accurate approach for Transposable Element superfamily classification with multi-feature fusion

Kang Hu[1,2,3], Minghua Xu[1,3], Xin Gao[4,5] and Jianxin Wang[1,2,3*]

[1]School of Computer Science and Engineering, Central South University, Changsha, 410083, China
[2]Xiangjiang Laboratory, Changsha, 410205, China
[3]Hunan Provincial Key Lab on Bioinformatics, Central South University, Changsha, 410083, China
[4]Computer Science Program, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
[5]Computational Bioscience Research Center, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Abstract

Motivation: *Classifying Transposable Elements (TEs) at the superfamily level offers deeper insights into species variation and evolution. Recent advancements in third-generation sequencing technologies have made a large number of genomes from non-model species becoming available. However, existing TE classification methods suffer from several limitations, including the necessity to train multiple hierarchical classification models, the incapacity to perform classification at the superfamily level, and deficiencies in both accuracy and robustness. Therefore, there is an urgent need for an accurate TE classification method to improve genome annotation.*

Results: *In this study, we develop NeuralTE, a deep learning method designed to classify transposons at the superfamily level. To achieve accurate TE classification, we identify various structural features of transposons, and use different combinations of k-mers for terminal repeats and internal sequences to uncover distinct patterns. Evaluation on all transposons from Repbase shows that NeuralTE outperforms existing deep learning, machine learning, and homology-based methods in classifying TEs. Testing on the transposons from novel species highlights the superior performance of NeuralTE compared to existing methods, achieving an F1-score of 0.8903, a 7.67% improvement over the state-of-the-art method RepeatClassifier. We also conduct TE annotation experiments on rice using different classification tools, and the results show that NeuralTE achieves annotations nearly identical to the gold standard, highlighting its robustness and accuracy in classifying transposons.*

Availability: *NeuralTE is publicly available at https://github.com/CSU-KangHu/NeuralTE.*

Introduction

Transposable Elements (TEs) are mobile DNA sequences within the genome, first identified by Barbara McClintock in her study of genetic instability during maize pigmentation inheritance [1]. These virus-like genomic elements inserted themselves into host genomes millions of years ago in a parasitic manner [2, 3, 4], which have contributed to the evolution of diverse host traits like internal gestation, memory, coloration, and adaptive immunity [5]. The interplay between TE insertion and host regulation resembles a game of hide-and-seek: TEs evolve to evade host defense mechanisms, while hosts counteract with their mutations [6], a dynamic central to genome evolution and speciation [7]. Most of the time, TE insertions are detrimental to the host, as their mobility within the genome can cause genetic instabilities, including mutations and chromosome breakage [8, 9]. However, new TE insertions might generate novel functional proteins [10], enhancing the adaptability of mutant individuals to their environment. As TEs accumulate within host genomes, these elements constitute a significant proportion, ranging from 25% to 75% in mammalian genomes [11]. Accurate identification and annotation of transposons are crucial for understanding their roles in genome stability, evolution, and the regulation of gene expression [12, 13].

According to Wicker's taxonomy [14], TEs are classified into Class I and Class II. Class I elements, also known as retrotransposons, use RNA intermediates for the 'copy-and-paste' mechanism during transposition. Based on their structures, retrotransposons can be classified into five orders: Long Terminal Repeat (LTR) Sequences, Dictyostelium Intermediate Repeat Sequences, Penelope-like Elements, Long Interspersed Nuclear Elements (LINEs), and Short Interspersed Nuclear Elements (SINEs) [15]. These orders can be further divided into a total of 16 distinct superfamilies. On the other hand, Class II elements, known as DNA transposons, employ the 'cut-and-paste' mechanism without RNA intermediates. Class II elements can be divided into Subclasses 1 and 2. Subclass 1 can be classified into TIR (Terminal In-

*Corresponding author: jxwang@mail.csu.edu.cn

verted Repeat) and Crypton orders, while Subclass 2 can be classified into the Helitron and Maverick orders. These orders can further be classified into a total of 12 different superfamilies.

Existing tools employ diverse methods for TE classification, such as Hidden Markov Model (HMM) profiles, homology comparison with known TEs, machine learning, and deep learning methods. Nevertheless, the majority of these tools fail to explore the entire hierarchical classification system and cannot classify TEs up to the superfamily level, such as TEclass [16], REPCLASS [17], and PASTEC [18]. TEclass uses 4-mers and 5-mers as features and support-vector machines (SVMs) for TE classification. To classify unknown TEs, PASTEC employs an HMM method, while REPCLASS utilizes a hybrid approach, incorporating homology-based, structure-based, and target site duplication (TSD) methods. There are several classification methods specialize in classifying specific types of transposons. TE-Learner [19] employs the random forest (RF) method to classify TEs, primarily focusing on classifying LTR retrotransposons. LTRdigest [20], LTRclassifier [21], and LTR_retriever [22] are designed to classify LTR elements, while TIR-learner [23] specializes in classifying TIR elements. TEsorter [24] can classify LTR elements into clades, but its performance in classifying other types of transposons remains unsatisfactory.

TE classification was introduced as a hierarchical classification problem [25], and several tools adopt this approach for the classification of unknown TEs, including DeepTE [26] and ClassifyTE [15]. However, training one or multiple classifiers for each level of hierarchical classification system presents challenges in model training and storage. ClassifyTE is a stacking-based approach that combines various machine learning methods for the hierarchical classification of TEs. However, it fails to classify Retrovirus, Penelope, Helitron, and Crypton elements. DeepTE and TERL [27] are both tools using convolutional neural networks (CNNs) for TE classification. The difference between them lies in the fact that DeepTE encodes sequences based on *k*-mer occurrences, while TERL directly encodes the entire TE sequence. RepeatClassifier is a classification module within the TE annotation tool RepeatModeler2 [28]. It performs homology searches against proteins and sequences in known databases to classify TEs, enabling classification up to the superfamily level. However, it relies on known TEs for homology-based classification, making it challenging to classify novel TEs.

In this study, we introduce NeuralTE, a deep learning method designed to classify TEs to the superfamily level. Compared to existing methods, NeuralTE identifies more structural features from TEs, enabling accurate classification of novel transposons. Furthermore, NeuralTE simplifies the classification process by training a single model capable of classifying different types of superfamilies. This eliminates the requirement to train multiple models for different levels of the TE hierarchical classification system, reducing both the complexity of model training and the storage demands. NeuralTE uses two core modules to construct features from TE sequences (**Fig. 1**). Firstly, the sequence feature module encodes TSDs, 5-bp ends of terminal sequences, and TE domain information using one-hot encoding. Secondly, the *k*-mer feature module uses *k*-mer occurrences to construct feature matrices from both terminal repeats and internal sequences of TEs. To discover distinct *k*-mer occurrence patterns in terminal repeats and internal sequences of TEs, different combinations of *k*-mers are used. Subsequently, the features built from these two modules are concatenated and inputted into a three-layered convolutional neural network (CNN) to predict the classification labels of TEs.

We first evaluate NeuralTE on all transposons from Repbase. As shown in **Table 1**, the results demonstrate that NeuralTE outperforms all existing deep learning and machine learning methods. However, due to the inclusion of minority samples in the dataset, such as 5S and Merlin, the overall performance of NeuralTE is compromised. These minority samples pose challenges for deep learning-based methods in achieving accurate predictions. Subsequently, we compare different tools for TE classification in novel species, and NeuralTE also outperforms other tools by achieving an F1-score of 89.03% (**Table 3**). Furthermore, when applied to classify transposons in model species, such as rice and maize, NeuralTE demonstrates performance far superior to other tools, aligning with the gold standard for genome annotation (**Figs. 2 and 5**).

Materials and methods

Datasets

Repbase is a fundamental and well-curated reference database for almost all eukaryotic genome sequence analyses [29]. To enable the training and testing of NeuralTE for classifying various transposons, we selected the Repbase database as the gold standard. In this study, we downloaded **Repbase** version 28.06 (https://www.girinst.org/repbase/) for model training and testing.

LTR retrotransposons consist of two highly similar long terminal repeats and an internal sequence. Repbase merges two similar long terminal repeats into one to minimize redundancy. To recover full-length

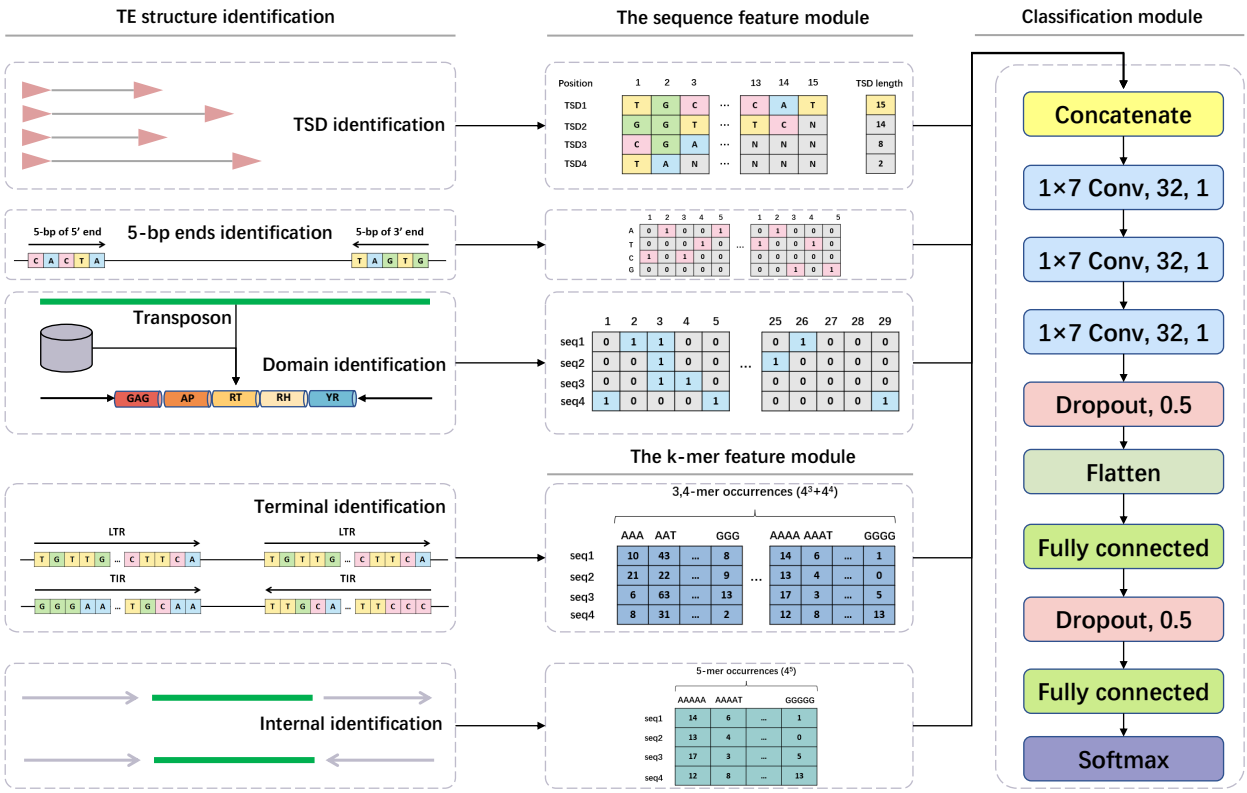

Fig. 1: *NeuralTE model for TE classification. In the TE structure identification, specific features of transposons are identified, including target site duplications (TSDs), the starting and ending 5-bp sequences, protein domain information, and the terminal repeats and internal sequences. In the sequence feature module, the identified sequence features are converted into one-hot encoding. In the k-mer feature module, the feature matrices are constructed by using k-mer occurrences. In the classification module, (1x7 Conv, 32, 1) denotes a convolutional layer comprising 32 kernels, each with a kernel size of 7, and a stride of 1. In CNN, a kernel refers to a convolution matrix, and the stride represents the movement step size of the convolution sliding matrix at each iteration*

LTRs, we first concatenate the terminal and internal sequences of LTR retrotransposons in Repbase based on sequence headers, resulting in a total of 55,767 sequences from 1,531 species. The hierarchical classification system comprises 28 distinct TE superfamilies. However, since Repbase does not include the Ngaro, VIPER, Maverick, and PiggyBac, we obtain a total of 24 TE superfamilies after converting labels from Repbase to Wicker's taxonomy [14]. Subsequently, for the acquisition of features like target site duplications (TSDs) in transposons, we have downloaded genome assemblies of 493 species and aligned transposons to these assemblies to identify TSD features. Finally, to assess the ability of NeuralTE to identify non-TEs, we have simulated 969 random sequences and extracted 399 non-TEs from Repbase, including tandem repeats, satellites, microsatellites, simple repeats, multi-copy genes, and pseudogenes. The inclusion of TEs and non-TEs leads to the formation of **Dataset 1**, consisting of 42,566 sequences derived from 493 species.

NeuralTE model

As shown in **Fig. 1**, NeuralTE is made up of four modules. The first module identifies specific structures within TE sequences, including TSDs, 5-bp ends of terminal sequences, domain information, terminal repeats, and TE internal sequences. The TE structure information is further fed into feature construction modules. Two modules are tailored to construct distinct features: the sequence feature module constructs features using one-hot encoding from TSDs, 5-bp ends of terminal sequences, and domain information, while the *k*-mer feature module generates features from the sequence of terminal repeats and TE internal sequences through different combinations of *k*-mer occurrences. Subsequently, a CNN-based classification module uses data from both feature modules to predict the TE classification label.

TE structure identification **The connection of LTRs**. In Repbase, LTR elements are split into terminal and internal sequences, such as 'Copia-62_PHord-LTR' and 'Copia-62_PHord-I'. Before identifying TE structures, we concatenate the LTR retrotransposons in Repbase based on the sequence headers to ensure the detection of full-length TEs.

The identification of TSDs. We start by downloading an extensive set of genome assemblies from NCBI (https://www.ncbi.nlm.nih.gov/), including 493 species and containing the majority of sequences in Repbase. To identify TSDs, we conduct BLASTN [30] alignment of TE sequences onto these assemblies, retrieving their copies along with a 20-bp flanking region for each copy. Then, we use regular expressions to search for candidate TSDs within these flanking regions, allowing for a 1-bp mismatch in TSDs longer than 8 bp. To ensure the reliability of identified TSDs, we choose the TSD length that occurs most frequently among all copies. We exclude TE sequences that show substantial differences (> 5bp) between the identified and original boundaries.

The usage of 5-bp ends. Different transposons exhibit distinct terminal motifs. For instance, LTR elements commonly present 2-bp palindromic motifs, notably 5′-TG...CA-3′. Helitron elements lack terminal inverted repeats and are inserted into an AT target site, often displaying a canonical terminal structure of 5′-TC...CTRR-3′ (where 5′-TC...CTAG-3′ predominates). In CACTA transposons, we identifies highly conserved CACT(A/G) motifs within their short terminals. Notably, polyA structures are observed at the 3′-ends of 7SL and L1 transposons. Therefore, we use the 5-bp ends of terminal sequences for TE classification.

The identification of TE domains. Different transposons exhibit diverse domains, such as different types and orders. For instance, distinguishing between DNA transposons and retrotransposons involves identifying the presence of RT (Reverse Transcriptase) or Tase (Transposase) within the TE sequence. Gypsy and Copia, on the other hand, exhibit variations in the order of RT and INT (Integrase) domains. Therefore, we conduct homology searches between the protein database and TE sequences using BLASTX [31], filtering out bad hits with an e-value cutoff of 1e-20. The protein database containing known TE peptides can be downloaded from RepeatMasker (https://www.repeatmasker.org/RMDownload.html). However, alignment results always present fragmentation, complicating the discovery of complete protein domains within TEs. To resolve this problem, we connect fragmented alignments to establish more continuous domain regions within TE families. Subsequently, we generate a table that maps protein domains to the corresponding TE locations, offering a convenient description of TE protein structures.

The identification of terminal repeats. The main difference between LTR (Long Terminal Repeat) and TIR (Terminal Inverted Repeat) elements is that LTRs have long direct repeat sequences spanning from 85 to 5000 base pairs, while TIRs have comparatively shorter terminal inverted repeats, typically ranging from a few base pairs to hundreds of base pairs. To better define the structure of TEs, we use the 'ltrsearch' and 'itrsearch' tools, included in TE Finder 2.30 (https://github.com/urgi-anagen/TE_finder), to determine whether the TEs exhibit long direct repeats or terminal inverted repeats.

The sequence feature module In the sequence feature module, we set the maximum length of TSDs to 15 bp and use one-hot encoding based on the four ATCG nucleotides. TSDs shorter than 15 bp are padded with zeros. Unknown TSDs or those containing N bases are encoded as all ones. Similar encoding is applied to the 5-bp ends. In NeuralTE, we have defined 29 superfamilies to classify 28 categories from the hierarchical classification system and one 'Unknown' category for non-TEs. Since there is a one-to-one correspondence between domains and TE superfamilies, we use vector of length 29 to represent the various types of domains that may occur in a TE sequence, as shown in **Fig. 1**.

The *k*-mer feature module The method of *k*-mer occurrence, used to mining sequence patterns, has been employed in several TE classification approaches [26, 15]. Autonomous LTR and TIR elements have internal sequences that contain the protein domains necessary for their self-replication, typically longer than their terminal sequences. Therefore, in contrast to existing methods, we use different combinations of *k*-mers to discover distinct patterns in the terminal and internal sequences of TEs, respectively. In the *k*-mer feature module, we use *k*-mer occurrences to generate feature matrices from the TE sequences (see **Fig. 1**). Let $x_1, x_2, ..., x_k$ represents a sequence of *k*-mer (x_k being one of the ATCG nucleotides), the number of features are 4^k. Simultaneously, different *k*-mer combinations are used for terminal and internal sequences of TEs. For terminal sequences, we use combinations with k-values of 3 and 4, resulting in feature matrix lengths of $4^3 + 4^4$. In contrast, for internal sequences, we set k to 5, yielding a feature matrix length of 4^5.

Classification module In classification module of NeuralTE, we construct a deep CNN model using data from the sequence and *k*-mer feature modules. Our model consists of three convolutional layers with kernel sizes of seven and a stride of one, and does not use any max-pooling layers (**Fig. 1**). During the convolution operation, the vector x is calculated as follows:

$$x_c = Conv(x) = \sum_{i=0}^{l-1} \sum_{j=0}^{k-1} w_j \cdot x_{i+j} \tag{1}$$

where w represents the weight vector of the convolution kernel; l is the length of all vector features; k is the 1-dimension convolution kernel size.

Dropout involves randomly deactivating a certain proportion of neurons in a layer during each training iteration, which can help reduce the interdependency between neurons, force the network to learn more robust features and improve its generalization to unseen data. In NeuralTE, we configure the dropout value to 0.5 after the final convolutional layer. The output from this layer is flattened and served as input to a fully connected layer, where another dropout value of 0.5 is applied. Subsequently, another fully connected layer is used, followed by a softmax output layer to compute probabilities for the classes of input sequences.

ReLU function is used in all three hidden layers and the fully connected layer.

$$x_a = Relu(x_c) = \begin{cases} 0, & if\ x_c < 0 \\ x_c, & otherwise \end{cases} \tag{2}$$

Model parameters are learned on the training set by minimizing the categorical cross-entropy loss function as follows:

$$H(y, \hat{y}) = -\sum_i y_i \cdot \log(\hat{y}_i) \tag{3}$$

where y represents the true class labels; $\hat{y}$ denotes the predicted class probabilities by the model; y_i represents the value of the i-th element in the true label vector; $\hat{y}_i$ represents the i-th element in the predicted probability vector.

We use a batch size of 32, conduct training for 50 epochs, and employ the ADAM optimizer with a learning rate of 0.001.

Computational setup

NeuralTE comprises three steps: TE structure identification, model training, and testing. The TE structure identification step is executed on a server equipped with 40 CPU processors (Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz), while the training and testing steps are carried out on a server with 4*12GB GeForce RTX 2080 Ti GPUs. NeuralTE also supports training and testing on CPU servers.

Results

Evaluation of NeuralTE in classifying TEs at the superfamily level

To evaluate the performance of NeuralTE, we have compared our method with five state-of-the-art methods, including two deep learning-based methods TERL [27] and DeepTE [26], two homology-based methods TEsorter [24] and RepeatClassifier [28], and one machine learning-based method ClassifyTE [15] for TE classification.

We randomly select 80% of transposon sequences from **Dataset 1** for model training and the remaining 20% of sequences for testing. A consistent proportion of transposons from various superfamilies is

Table 1: *Evaluation of performance between different TE classification tools on* ***Dataset 1***

Tools	Precision	Recall	F1-score
TERL	0.5201	0.4804	0.4893
DeepTE	0.7754	0.6268	0.6564
DeepTE†[1]	0.7792	0.6338	0.6624
TEsorter	0.5591	0.3273	0.3531
RepeatClassifier	0.9327	**0.9120**	0.9185
NeuralTE	**0.9714**	0.8967	**0.924**

[1] Applying domains from TEs to correct false classification

Table 2: *Evaluation of performance between NeuralTE and ClassifyTE on* ***Dataset 2***

Tools	Precision	Recall	F1-score
ClassifyTE	0.7431	0.7892	0.7541
NeuralTE	**0.9671**	**0.8957**	**0.9199**

maintained in both the training and testing datasets, and all experimental results are based on macro-averaged metrics. To ensure a fair comparison between NeuralTE and other methods, identical training and testing datasets are used for evaluation. For example, RepeatClassifier relies on the known library provided by RepeatMasker [32] for TE classification, and we substitute the library in RepeatMasker with our training set to assess its performance. An exception is TEsorter, which utilizes its inherent multiple databases for TE classification. Consequently, we evaluate TEsorter using its default databases and parameters.

DeepTE supports post-processing of prediction results by correcting the classification results through searching for specific domains in transposons. We denote the version of DeepTE using post-processing as DeepTE†. As shown in **Table 1**, NeuralTE outperforms TERL, DeepTE, and TEsorter across all evaluation criteria. In comparison to TERL, which encodes the entire TE sequence, DeepTE achieves higher performance by using *k*-mer occurrence encoding. DeepTE†, applying TE domains to correct false classifications, achieves a slight improvement compared to DeepTE. Compared to RepeatClassifier, NeuralTE exhibits lower recall but higher precision, resulting in a slightly higher F1-score (0.9714 versus 0.9327 for precision, 0.8967 versus 0.9120 for recall, and 0.924 versus 0.9185 for F1-score).

Since ClassifyTE fails to classify the Retrovirus, Penelope, Helitron, and Crypton elements, we have filtered out these four types of transposons from **Dataset 1**, resulting in the creation of **Dataset 2**. Subsequently, we perform a random split of the **Dataset 2**, adhering to an 80% training set and 20% testing set principle. As shown in **Table 2**, NeuralTE demonstrates superior performance compared to ClassifyTE (0.9671 versus 0.7431 for precision and 0.8957 versus 0.7892 for recall). In summary, the results show that NeuralTE has superior performance over TERL, DeepTE, TEsorter, and ClassifyTE, and comparable performance with RepeatClassifier in predicting TE classification at the superfamily level. However, due to RepeatClassifier relying on known TEs for homology-based classification, it faces challenges in accurately classifying novel TEs.

Table 3: *Evaluation of the predictive performance of different TE classification tools for transposons from novel species (****Dataset 3****)*

Tools	Precision	Recall	F1-score
TERL	0.4768	0.401	0.4209
DeepTE	0.6694	0.5347	0.5608
DeepTE†[1]	0.6777	0.5410	0.5677
TEsorter	0.5534	0.3615	0.3767
RepeatClassifier	0.8956	0.8100	0.8269
NeuralTE	**0.9247**	**0.8747**	**0.8903**

[1] Applying domains from TEs to correct false classification

Evaluation of NeuralTE on prediction of TEs from novel species

To demonstrate the generalization ability of NeuralTE, we have compared NeuralTE with TERL, DeepTE, TEsorter, and RepeatClassifier in predicting transposons from novel species. We randomly select 20% of species from **Dataset 1** as new species for testing, while the remaining 80% of species as known species for training, forming **Dataset 3**. As shown in **Table 3**, NeuralTE demonstrates superior performance compared to TERL, DeepTE, and TEsorter. Compared to TERL, DeepTE achieves higher performance, with a slight improvement when applying TE domains to correct false classifications in DeepTE†. Notably, NeuralTE also achieves a higher performance compared to RepeatClassifier (0.9247 versus 0.8956 for precision and 0.8747 versus 0.8100 for recall). Using more features such as TSDs and 5-bp ends, NeuralTE can achieve higher classification performance for novel transposons.

To evaluate the predictive ability of NeuralTE for individual species, we select rice and maize, known as model species in TE research, for our experiments. We have separated transposons of rice and maize within **Dataset 1** for testing, reserving the remaining species for training. The datasets specific to rice and

Table 4: *Evaluation of the predictive performance of different TE classification tools for eight well-known TIR transposons (**Dataset 7**)*

Tools	Os3378	nDart1-A	Dart	mJing	mPing	Pong	ZB	ZmCCT-CACTA
TERL	✓	✓	✓	×	✓	×	✓	✓
DeepTE	✓	✓	✓	✓	✓	×	✓	✓
DeepTE†[1]	✓	✓	✓	✓	✓	×	✓	✓
TEsorter	✓	×	✓	×	×	✓	✓	✓
RepeatClassifier	✓	✓	✓	×	✓	✓	✓	×
NeuralTE	✓	✓	✓	✓	✓	✓	✓	✓

[1] Applying domains from TEs to correct false classification

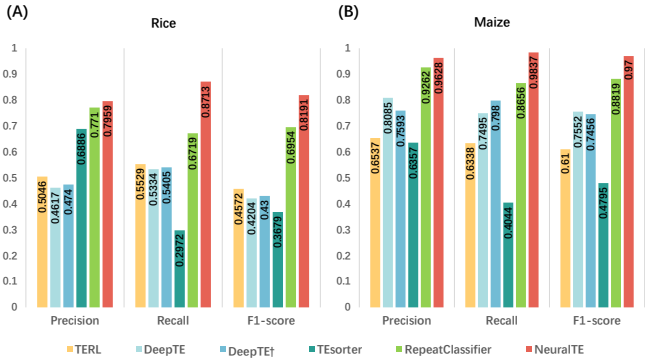

Fig. 2: *Comparison of the predictive performance of different TE classification tools in rice (**Dataset 5**) and maize (**Dataset 6**). The predictive performance in rice (**A**) and maize (**B**). DeepTE†: Applying domains from TEs to correct false classification*

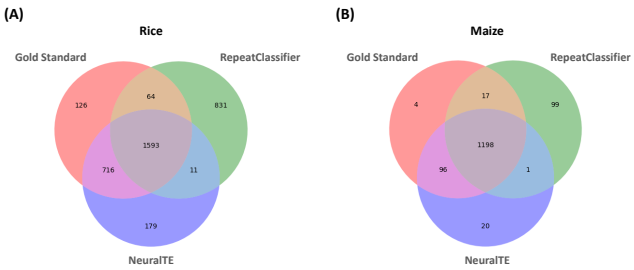

Fig. 3: *Venn diagram showing the number of transposons correctly classified by NeuralTE and RepeatClassifier in rice (**Dataset 5**) and maize (**Dataset 6**). The number of transposons in rice (**A**) and maize (**B**). The central overlapped region depicts the number of correctly classified transposons by both tools. The figure indicates that NeuralTE outperforms RepeatClassifier in correctly classifying more transposons*

maize are denoted as **Dataset 5** and **Dataset 6**, respectively. As shown in **Figs. 2A and B**, NeuralTE outperforms TERL, DeepTE, TEsorter, and RepeatClassifier in both rice and maize, achieving F1-scores of 0.8191 and 0.97, respectively. In contrast, the state-of-the-art method RepeatClassifier only obtains F1-scores of 0.6954 and 0.8819.

We have compared the number of transposons correctly classified by NeuralTE and RepeatClassifier in both rice and maize. As shown in **Fig. 3A**, NeuralTE correctly classifies 2,309 transposons in rice, surpassing RepeatClassifier, which classifies 1,657. Compared to RepeatClassifier, NeuralTE improves by 39.3% in the correct classification of transposons. A similar result is observed in maize, where NeuralTE accurately classifies 1,294 transposons compared to RepeatClassifier's 1,215. In this case, NeuralTE demonstrates a 6.5% improvement over RepeatClassifier in correctly classifying more transposons (**Fig. 3B**).

We have curated eight well-known TIR transposons, including six from rice, one from zebrafish, and one from maize. Each of these transposons has been previously documented in the literature for their significant roles in the functionality and evolution of species. To evaluate the classification performance of different tools on these transposons, we have excluded transposons from rice, zebrafish, and maize species from **Dataset 1**, creating **Dataset 7** as the training set, with these eight transposons serving as the testing set. As shown in **Table 4**, NeuralTE accurately classifies all transposons, whereas other tools exhibit misclassification for certain transposons, notably showing a high error rate for a MITE transposon, mJing. These results show that NeuralTE is more powerful than other tools in predicting the classification of transposons from novel species.

Ablation study

Comparison of different features using all transposons To evaluate the contribution of each feature,

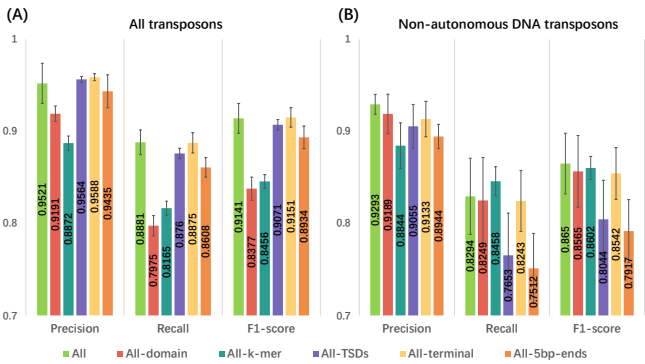

Fig. 4: *The predictive performance of NeuralTE with different features. (**A**) Using all transposons (**Dataset 1**). (**B**) Using non-autonomous DNA transposons (**Dataset 4**). 'All' represents the use of all features, whereas 'All-xxx' indicates the use of all features except for 'xxx'. The results are based on a five-fold cross-validation*

we have implemented six models: All, All-domain, All-*k*-mer, All-TSDs, All-terminal, and All-5bp-ends. 'All' represents the use of all available features, while 'All-xxx' denotes the use of all features except for 'xxx'. As shown in **Fig. 4A**, the results indicate that each feature contributes to the predictive performance of NeuralTE, with the exception of the terminal feature. Notably, domain and *k*-mer features show the most prominent impact on predictive performance. Removing the domain feature reduces the F1-score of model from 0.9141 to 0.8377, underlining the necessity of protein domain in conventional TE classification [17, 33]. Similarly, the removal of the *k*-mer feature results in a decrease in the F1-score of model from 0.9141 to 0.8456. Following in descending order of contribution are 5bp-ends and TSDs.

Comparison of different features using non-autonomous DNA transposons The TE domain and *k*-mer features play a crucial role in classifying autonomous TEs, while other features hold greater significance in the classification of non-autonomous transposons. We have extracted the non-autonomous DNA transposons from **Dataset 1** to form **Dataset 4**, which is then utilized for conducting ablation experiments. As shown in **Fig. 4B**, the result demonstrates that each feature contributes to the predictive performance of NeuralTE. In descending order of contribution, the features are ranked as follows: 5bp-ends, TSDs, terminal, domain, and *k*-mer. Removing the 5bp-ends feature reduces the F1-score of model from 0.8650 to 0.7917, while eliminating the TSDs feature decreases the F1-score from 0.8650 to 0.8044. The results show that 5bp-ends and TSDs features play a crucial role in the classification of non-autonomous DNA transposons.

Using different classification tools for TE annotation

Using different TE classification tools may result in significant differences in TE annotation. To evaluate the genome annotation performance of NeuralTE on novel species, we have compared diverse tools using rice as a model species on **Dataset 5**. As shown in **Fig. 5A**, NeuralTE demonstrates superior predictive performance across all superfamilies compared to RepeatClassifier, DeepTE, TEsorter, and TERL. In the classification of Gypsy and Copia, NeuralTE shows comparable performance with RepeatClassifier. However, on Helitron, Mutator, and other transposons, NeuralTE exhibits higher performance. Additionally, we use RepeatMasker (version 4.1.1) and libraries classified by different tools to conduct genome annotation. As shown in **Fig. 5B**, NeuralTE achieves annotation proportions closest to those of the gold standard. For example, RepeatClassifier, DeepTE†, and TEsorter show a reduced fraction of the genome masked by Helitron elements. DeepTE†, in contrast, results in a higher proportion of the genome masked by Copia and Tc1-Mariner elements. TEsorter displays a significant portion of the genome masked by other types of transposons, while TERL indicates a higher fraction of genome masked by Gypsy elements but a lower proportion masked by Copia elements. Furthermore, the transposon density throughout the genome using different tools is displayed. For simplicity, we showcase Helitron density on chromosome 1 of rice. As shown in **Fig. 5C**, NeuralTE achieves a density distribution nearly identical to that of the gold standard. The results show differences in TE annotation when using different classification tools. Notably, NeuralTE outperforms other tools and achieves annotations consistent with the gold standard.

Annotating novel transposons using NeuralTE

TE families are defined using the '80-80-80' rule [14], which specifies that TE instances are members of the same family if they are longer than 80 bp, and share at least 80% sequence identity over 80% of their length. In this study, we use the '80-80-80' rule to distinguish between known and novel transposons. HiTE [12], a tool for identifying full-length transposons, is proficient in detecting genuine TEs that are absent from databases, particularly in the case of DNA transposons. We use HiTE to identify TEs in rice and maize, clustering them alongside Repbase through CD-HIT-EST [34] and the '80-80-80' rule. Clusters containing any elements from Repbase are designated as known transposons, while the remain-

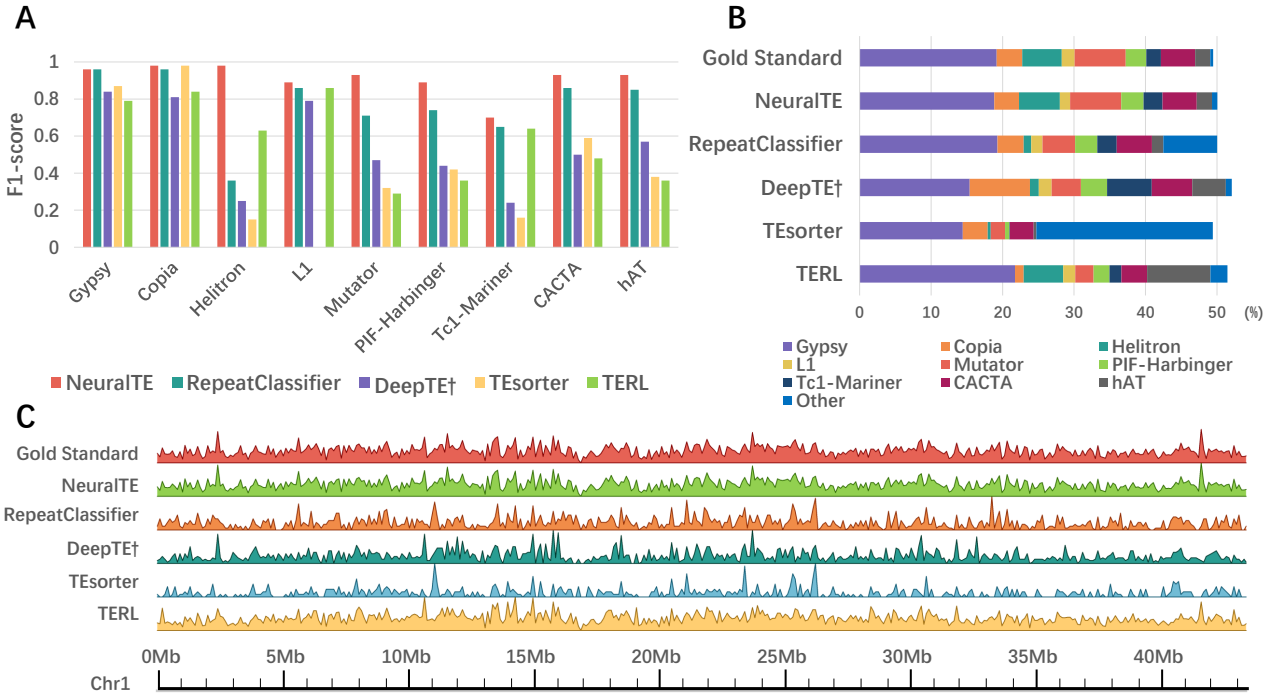

Fig. 5: *Differential analysis of rice TE classification using various tools (**Dataset 5**). (**A**) Evaluation of the predictive performance of different tools across distinct TE superfamilies using the F1-score metric. TEsorter lacks the ability to predict L1 transposons, resulting in the absence of values. (**B**) Comparison of the percent of genome masked by different superfamilies classified using different tools. (**C**) Comparison of TE density profiles derived from different tools (showcasing Helitron elements in the figure)*

Table 5: *Comparative classification of novel transposons identified by HiTE using NeuralTE and RepeatClassifier. Misclassifications are represented by the labels 'Other' and 'Unknown'. RC: RepeatClassifier*

Species	Metrics	Tools	CACTA	hAT	Mutator	PIF-Harbinger	Tc1-Mariner	Other	Unknown
rice	TE count	NeuralTE	127	119	469	180	82	7	4
		RC	114	109	296	100	33	15	321
	Coverage (%)	NeuralTE	4.14	1.30	6.75	3.07	1.84	2.99	0.04
		RC	3.91	0.94	3.30	2.41	1.57	4.19	3.73
maize	TE count	NeuralTE	143	376	122	313	40	15	7
		RC	121	346	86	157	15	44	247
	Coverage (%)	NeuralTE	5.76	1.00	0.62	1.06	0.37	5.39	2.21
		RC	2.36	1.03	0.50	0.68	0.08	10.13	0.72

ing clusters are classified as novel. We identifies a total of 988 and 1,016 novel transposons in rice and maize, respectively. These transposons are recognized as DNA transposons as they originate from the TIR identification module of HiTE, and their credibility can be confirmed by examining multiple sequence alignments. These novel transposons exhibit distinct TE structures, such as terminal inverted repeats (TIRs) and target site duplications (TSDs), with the surrounding regions outside the TE structure appearing to be random (https://github.com/CSU-KangHu/TE_annotation).

To explore the characteristics of these novel DNA transposons, we use NeuralTE and RepeatClassifier for their respective classifications. As shown in **Table 5**, NeuralTE classifies all novel DNA transposons primarily into five superfamilies, including CACTA, hAT, Mutator, PIF-Harbinger, and Tc1-Mariner. Since all transposons are DNA transposons, the remaining classifications are considered as misclassification, such as 'Other' and 'Unknown'. Compared to RepeatClassifier, NeuralTE shows a lower rate of misclassifications, with only 11 misclassifications in rice, whereas RepeatClassifier yields 336 misclassifications (**Table 5**). Incorrect classifications can propagate throughout the genome annotation, potentially leading to misleading results. The genome coverage of misclassification transposons for NeuralTE is 3.03%, compared to 7.92% for RepeatClassifier. Similar results are observed in maize, where NeuralTE obtains 22 misclassifications, while RepeatClassifier generates 291. The genome coverage of misclassification transposons for NeuralTE is 7.6%, while RepeatClassifier is 10.85%. This further highlights the superior classification capability of NeuralTE compared to RepeatClassifier for novel transposons.

Discussion

In this study, we develop NeuralTE, a deep learning method to classify transposons into distinct superfamilies. Evaluation of NeuralTE on all transposons from Repbase reveals its superior performance in classifying TEs at the superfamily level compared to existing TE classification methods.

Recent advancements in third-generation sequencing technologies have made a large number of genomes from non-model species becoming available. Our tests on transposons from these novel species demonstrate that NeuralTE outperforms all existing methods in classifying TEs at the superfamily level, achieving an F1-score of 89.03%. In contrast, RepeatClassifier experiences significant performance degradation when classifying transposons from novel species. NeuralTE consistently achieves high performance by using multiple features to classify unknown TEs. Moreover, when applied to classify transposons in individual species, such as rice and maize, NeuralTE demonstrates performance far superior to other tools, aligning with the gold standard for genome annotation. Given that the novel transposons identified by HiTE can be confirmed by examining multiple sequence alignments, when used to classify those novel transposons, NeuralTE successfully classifies the majority of transposons into different superfamilies, whereas RepeatClassifier produces a high proportion of misclassifications.

Existing TE classification methods suffer from several limitations, including the necessity to train multiple hierarchical classification models, the incapacity to perform classification at the superfamily level, and deficiencies in both accuracy and robustness. To solve these problems, NeuralTE makes full use of various TE structure features to enhance the predictive performance of the model. Rather than classifying TEs layer by layer according to the hierarchical classification system, NeuralTE trains a single model to directly classify TEs up to the superfamily level. The results show that NeuralTE achieves higher performance and robustness. In conclusion, NeuralTE demonstrates significant promise as a reliable tool for TE classification. We anticipate that the tool developed in this study will contribute to genome variation research, potentially benefiting applications like crop breeding.

References

[1] Barbara McClintock. "Controlling elements and the gene". In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 21. Cold Spring Harbor Laboratory Press. 1956, pp. 197–216.

[2] Hua-Hao Zhang et al. "Horizontal transfer and evolution of transposable elements in vertebrates". In: *Nature communications* 11.1 (2020), p. 1362.

[3] Amandine BONNET. "Transposable Elements: Parasites that Shape Genome Evolution". In: *Function and Evolution of Repeated DNA Sequences* (2023), p. 117.

[4] Clément Gilbert et al. "A role for host–parasite interactions in the horizontal transfer of transposons across phyla". In: *Nature* 464.7293 (2010), pp. 1347–1350.

[5] Alexander Hayward and Clément Gilbert. "Transposable elements". In: *Current Biology* 32.17 (2022), R904–R909.

[6] Frank MJ Jacobs et al. "An evolutionary arms race between KRAB zinc-finger genes ZNF91/93 and SVA/L1 retrotransposons". In: *Nature* 516.7530 (2014), pp. 242–245.

[7] Tuğçe Aktaş. "Transposons: catch them if you can". In: *Nature Reviews Molecular Cell Biology* 23.5 (2022), pp. 306–306.

[8] Nunzia Colonna Romano and Laura Fanti. "Transposable elements: major players in shaping genomic and evolutionary patterns". In: *Cells* 11.6 (2022), p. 1048.

[9] Yun-Ji Kim, Jungnam Lee, and Kyudong Han. "Transposable elements: no more'Junk DNA'". In: *Genomics & informatics* 10.4 (2012), p. 226.

[10] Rachel L Cosby et al. "Recurrent evolution of vertebrate transcription factors by transposase capture". In: *Science* 371.6531 (2021), eabc6405.

[11] Austin B Osmanski et al. "Insights into mammalian TE diversity through the curation of 248 genome assemblies". In: *Science* 380.6643 (2023), eabn1430.

[12] Kang Hu et al. "HiTE: An accurate dynamic boundary adjustment approach for full-length Transposable Elements detection and annotation in Genome Assemblies". In: *bioRxiv* (2023), pp. 2023–05.

[13] Mourdas Mohamed et al. "TrEMOLO: accurate transposable element allele frequency estimation using long-read sequencing data combining assembly and mapping-based approaches". In: *Genome Biology* 24.1 (2023), pp. 1–20.

[14] Thomas Wicker et al. "A unified classification system for eukaryotic transposable elements". In: *Nature reviews genetics* 8.12 (2007), pp. 973–982.

[15] Manisha Panta et al. "ClassifyTE: a stacking-based prediction of hierarchical classification of transposable elements". In: *Bioinformatics* 37.17 (2021), pp. 2529–2536.

[16] György Abrusán et al. "TEclass—a tool for automated classification of unknown eukaryotic transposable elements". In: *Bioinformatics* 25.10 (2009), pp. 1329–1330.

[17] Cédric Feschotte et al. "Exploring repetitive DNA landscapes using REPCLASS, a tool that automates the classification of transposable elements in eukaryotic genomes". In: *Genome biology and evolution* 1 (2009), pp. 205–220.

[18] Claire Hoede et al. "PASTEC: an automatic transposable element classification tool". In: *PloS one* 9.5 (2014), e91929.

[19] Leander Schietgat et al. "A machine learning based framework to identify and classify long terminal repeat retrotransposons". In: *PLoS computational biology* 14.4 (2018), e1006097.

[20] Sascha Steinbiss et al. "Fine-grained annotation and classification of de novo predicted LTR retrotransposons". In: *Nucleic acids research* 37.21 (2009), pp. 7002–7013.

[21] Cecile Monat et al. "LTRclassifier: a website for fast structural LTR retrotransposons classification in plants". In: *Mobile genetic elements* 6.6 (2016), e1241050.

[22] Shujun Ou and Ning Jiang. "LTR_retriever: a highly accurate and sensitive program for identification of long terminal repeat retrotransposons". In: *Plant physiology* 176.2 (2018), pp. 1410–1422.

[23] Weijia Su, Xun Gu, and Thomas Peterson. "TIR-Learner, a new ensemble method for TIR transposable element annotation, provides evidence for abundant new transposable elements in the maize genome". In: *Molecular plant* 12.3 (2019), pp. 447–460.

[24] Ren-Gang Zhang et al. "TEsorter: an accurate and fast method to classify LTR-retrotransposons in plant genomes". In: *Horticulture Research* 9 (2022), uhac017.

[25] Felipe Kenji Nakano et al. "Top-down strategies for hierarchical classification of transposable elements with neural networks". In: *2017 International joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 2539–2546.

[26] Haidong Yan, Aureliano Bombarely, and Song Li. "DeepTE: a computational method for de novo classification of transposons with convolutional neural network". In: *Bioinformatics* 36.15 (2020), pp. 4269–4275.

[27] Murilo Horacio Pereira da Cruz et al. "TERL: classification of transposable elements by convolutional neural networks". In: *Briefings in bioinformatics* 22.3 (2021), bbaa185.

[28] Jullien M Flynn et al. "RepeatModeler2 for automated genomic discovery of transposable element families". In: *Proceedings of the National Academy of Sciences* 117.17 (2020), pp. 9451–9457.

[29] Weidong Bao, Kenji K Kojima, and Oleksiy Kohany. "Repbase Update, a database of repetitive elements in eukaryotic genomes". In: *Mobile Dna* 6 (2015), pp. 1–6.

[30] GenBank BlastN. "BLAST: basic local alignment search tool". In: *NUTRITIONAL AND PHYSIOLOGICAL DISORDERS IN HORTICULTURAL CROPS* (2019).

[31] Scott McGinnis and Thomas L Madden. "BLAST: at the core of a powerful and diverse set of sequence analysis tools". In: *Nucleic acids research* 32.suppl_2 (2004), W20–W25.

[32] AFA Smit, R Hubley, and P Green. *RepeatMasker Open-4.0. 2013–2015*. 2015.

[33] Kevin Riehl et al. "TransposonUltimate: software for transposon classification, annotation and detection". In: *Nucleic Acids Research* 50.11 (2022), e64–e64.

[34] Weizhong Li and Adam Godzik. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22.13 (2006), pp. 1658–1659.