

K-mer-based machine learning method to classify LTR-retrotransposons in plant genomes

Simon Orozco-Arias^{1,2}, Mariana S. Candamil-Cortés¹, Paula A. Jaimes¹, Johan S. Piña¹, Reinel Tabares-Soto³, Romain Guyot^{3,4} and Gustavo Isaza²

¹ Department of Computer Science, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

² Department of Systems and Informatics, Universidad de Caldas, Manizales, Caldas, Colombia

³ Department of Electronics and Automation, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

⁴ Institut de Recherche pour le Développement, CIRAD, Univ. Montpellier, Montpellier, France

ABSTRACT

Every day more plant genomes are available in public databases and additional massive sequencing projects (i.e., that aim to sequence thousands of individuals) are formulated and released. Nevertheless, there are not enough automatic tools to analyze this large amount of genomic information. LTR retrotransposons are the most frequent repetitive sequences in plant genomes; however, their detection and classification are commonly performed using semi-automatic and time-consuming programs. Despite the availability of several bioinformatic tools that follow different approaches to detect and classify them, none of these tools can individually obtain accurate results. Here, we used Machine Learning algorithms based on k -mer counts to classify LTR retrotransposons from other genomic sequences and into lineages/families with an F1-Score of 95%, contributing to develop a free-alignment and automatic method to analyze these sequences.

Subjects Bioinformatics, Plant Science, Computational Science, Data Mining and Machine Learning, Data Science

Keywords Transposable elements, LTR retrotransposons, Plant genomes, Machine learning, Classification, Free-alignment approach, k -mer based method

INTRODUCTION

The availability of large-scale biological data is changing the way researchers must analyze and find solutions to problems in almost every area of biological sciences. Machine Learning (ML) algorithms can use this data to automatically learn the parameters needed to fit a model to a specific problem (*Shastry & Sanjay, 2020*) in order to predict known labels. This process is called supervised learning (*Zou et al., 2018*). Bioinformatics, which is an intersection between computer sciences, biological sciences, and mathematics (*Orozco-Arias et al., 2017*), plays a central role in storing, analyzing, categorizing, and labeling the huge flow of information generated, for example, by next-generation sequencing (NGS) platforms. Advances in these sequencing technologies have provided a new paradigm in the field of post-genomics (*Rigal & Mathieu, 2011; Chen et al., 2014;*

Submitted 17 February 2021

Accepted 24 April 2021

Published 19 May 2021

Corresponding authors

Simon Orozco-Arias,
simon.orozco.arias@gmail.com

Gustavo Isaza,
gustavo.isaza@ucaldas.edu.co

Academic editor

Gerard Lazo

Additional Information and
Declarations can be found on
page 15

DOI 10.7717/peerj.11456

© Copyright

2021 Orozco-Arias et al.

Distributed under

Creative Commons CC-BY 4.0

OPEN ACCESS

Rishishwar et al., 2017), which focuses on how to store, process, and analyze these data streams in acceptable time frames (*Rigal & Mathieu, 2011; Chen et al., 2014; Rishishwar et al., 2017*).

By automatizing tasks that were done manually, ML is impacting many areas of bioinformatics (*Li et al., 2020*), such as genomics (*Libbrecht & Noble, 2015; Zou et al., 2018; Eraslan et al., 2019*), systems biology (*Larrañaga et al., 2006*), and, specifically, the annotation of transposable elements (TEs) (*Orozco-Arias et al., 2019*). This last task is a current challenge in genomics (*Ou, Chen & Jiang, 2018; Orozco-Arias et al., 2020*). There is a growing interest in these repeated sequences due to their key functional and evolutionary roles on eukaryote genomes (*Hesam & Ali, 2010; Orozco-Arias, Isaza & Guyot, 2019*). Nevertheless, the detection and classification of these sequences remain complex because of their highly repetitive nature, diversity, polymorphism, species specificity, among other factors (*Ou, Chen & Jiang, 2018; Mustafin & Khusnutdinova, 2018*).

Although there is an open debate on how to classify TEs, the most classification approach is based on their lifecycle (i.e., how they move inside the genome). TEs are classified hierarchically (*Orozco-Arias et al., 2019*); first, they are divided into two main classes (e.g., Class I or retrotransposons and Class II or DNA transposons (*Wicker et al., 2007*)) and further divided into orders according to the similarity of their coding domains (*Chaparro et al., 2015; Orozco-Arias, Isaza & Guyot, 2019; Neumann et al., 2019*). In plants, the most frequent elements are, by far, long terminal repeat (LTR) retrotransposons (LTR-RTs, an order from Class I) (*Gao et al., 2012; Grandbastien, 2015a*), which account for 80% of the genome size of species such as wheat, barley, or the rubber tree (*Rahman et al., 2013*).

Indeed, bioinformatic approaches have developed many tools to detect and classify transposable elements, including multiple approaches that group TEs based mainly on their structure, homology, redundancy, or conservation across genomes (*Rawal & Ramaswamy, 2011; Jiang & Ramachandran, 2013; Loureiro et al., 2013; Schietgat et al., 2018*). Nevertheless, the complexity of these elements does not consistently allow for accurate, reliable, and reproducible results across programs for all types of TEs (*Arkhipova, 2017*). In recent years, several machine learning-based methods have been proposed and evaluated, which take advantage of the thousands of sequences available in several datasets, such as Repbase (*Jurka et al., 2005*), RepetDB (*Cornut et al., 2019*), PGSB (*Spannagl et al., 2016a*), and InpactorDB (*Orozco-Arias et al., 2021*).

Several studies have, therefore, proposed the use of ML for TE analysis (reviewed in (*Orozco-Arias et al., 2019*)) to differentiate between LTR-RTs and SINEs (*Ashlock & Datta, 2012*) or autonomous and non-autonomous LTR-RTs in the *Coffea canephora* genome (*Arango-López et al., 2017*), and to improve the accuracy and performance of the classification (*Loureiro et al., 2013; Nakano et al., 2017*). Recently, a random forest algorithm was used to broadly classify LTR-RTs into superfamilies (*Schietgat et al., 2018*), while additional pre-processing techniques and coding schemes allow their deep classification (*Orozco-Arias et al., 2020*). Moreover, several deep neural network (DNN) architectures that perform TE classification have been published so far. For instance, Nakano et al. implemented a fully connected neural network (FNN) (*Nakano et al., 2018*),

da Cruz et al. used a convolutional neural network (CNN) with a 2D representation of the sequences (*da Cruz et al., 2019*), and Yan et al. used a CNN in 1D to classify TEs into superfamilies (*da Cruz et al., 2020; Yan, Bombarely & Li, 2020*).

Despite these efforts, none consider the need to both detect and classify elements at the same time, or specifically aim to classify LTR-RTs to the lineage/family level. In this paper, we evaluate and demonstrate the application of ML algorithms in the binary classification between LTR-RTs and other genomic features, multi-class classification into lineages/families or both (in a single process), as well as the importance of the features used, in order to design a free-alignment method for the annotation of LTR-RT in plant genomes based on k -mer frequencies.

MATERIALS & METHODS

Dataset composition

We used InpactorDB (*Orozco-Arias et al., 2021*), DOI 10.5281/zenodo.4386316 or 10.23708/QCMOUA), which comprises 67,241 LTR retrotransposon sequences, deeply classified into lineages/families, from 195 plant species. This dataset initially contained sequences from Repbase, RepetDB, and PGSB, which were processed using several filters to remove low quality elements (i.e., elements with nested sequences) (*Orozco-Arias et al., 2021*). It also contained LTR-RTs predicted by LTR_STRUC (*McCarthy & McDonald, 2003*) and EDTA (*Ou et al., 2019*). As negative instances, we created a dataset composed of annotated genomic features other than LTR_RT, such as coding sequences (CDS), different types of RNA (e.g., mRNA, tRNA, non-coding RNA, among others), and other types of transposable elements that do not belong to LTR-RTs (e.g., TIR, Helitron, PLEs, DIRs, LINEs, and SINEs) from the same plant species contained in InpactorDB. These additional TE sequences were available in databases such as PGSB PlantsDB (*Spannagl et al., 2016b*), Repbase (v. 20.05, 2017) (*Bao, Kojima & Kohany, 2015*), RepetDB (*Amselem et al., 2019*), Ensembl Plants (*Bolser et al., 2017*), and JGI (Joint Genome Institute) (*Nordberg et al., 2014*) (Supplemental Material 1). This dataset is available in DOI 10.5281/zenodo.4543904.

For the binary classification task, we randomly selected 10,000 LTR retrotransposon sequences (taken as positive instances) and 10,000 genomic feature sequences for the negative instances. For the classification task into lineages/families, we used only InpactorDB data, while, for the binary plus multi-class classification problem (unified at a single ML process), we filtered the negative instances to retain only sequences longer than 6 Kb. We did this filter in order to reduce the number of sequences from more than 3 million to 34,830 and because the average length of Copia elements in InpactorDB is 5,957.48. In contrast the average length of Gypsy elements is 10,760.57.

As features, we selected k -mer frequencies with $1 \leq k \leq 6$, as recommended in *Orozco-Arias et al. (2020)*, calculating all possible k -mers, and later counting the number of occurrences of them in each sequence. We calculated them for lineage-level classification and the binary plus multi-class classification task. For binary classification, we used the same coding schemes as implemented in *Orozco-Arias et al. (2020)*, such as DAX (*Yu et al., 2015*), EIIP (*Nair & Sreenadhan, 2006*), Complementary (*Akhtar, Epps & Ambikairajah,*

2008), Enthalpy (Kauer & Blöcker, 2003), and Galois (4) (Rosen, 2006). Additionally, two techniques were applied to automatically extract features from the sequences: (i) k -mer frequencies were obtained for each element and (ii) three physical-chemical (PC) properties were extracted, such as average hydrogen bonding energy per base pair (bp), stacking energy (per bp), and solvation energy (per bp), which were calculated by taking the first di-nucleotide and moving in a sliding window of one base at a time (Jaiswal & Krishnamachari, 2019). Moreover, we pre-processed the data by scaling, following the strategy implemented in Tabares-soto et al. (2020), and performed a dimensional reduction through a principal component analysis (PCA) (Wold, Esbensen & Geladi, 1987) with a cumulative variance of 96% and tolerance of $1e-4$.

For the binary classification task, we divided the dataset into a training set (80% of the data) a validation set (10%), and a test set (10%). For multi-class classification into lineages, we used the same partition and additionally, we used k -cross-validation (Komer, Bergstra & Eliasmith, 2014) with $k = 9$ after tuning hyper-parameters in order to test the generalization property of each model.

Machine learning algorithms used

For binary classification between positive (LTR retrotransposons) and negative (other genomic features) instances, we used the same algorithms described in Orozco-Arias et al. (2020); thus, we used Linear Support Vector Classifier (SVC), Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Naive Bayesian Classifier (NB), Multi-Layer Perceptron (MLP), Decision Trees (DT), and Random Forest (RF) and selected the larger F1-Score for different values of a hyper-parameter (as in Orozco-Arias et al. (2020)).

For classification, we used the supervised models that showed the best performance in Orozco-Arias et al. (2020), such as KNN, LR, SVC, and LDA, but we applied hyper-parameter tuning (Table 1) using GridSearchCV from Scikit-learn (Pedregosa et al., 2011), using only sequences from InpactorDB (we did not include negative instances due to the high memory required). We used the F1-Score as a performance metric in all executions since it is not affected much by unbalanced datasets such as LTR-RTs (Orozco-Arias et al., 2020).

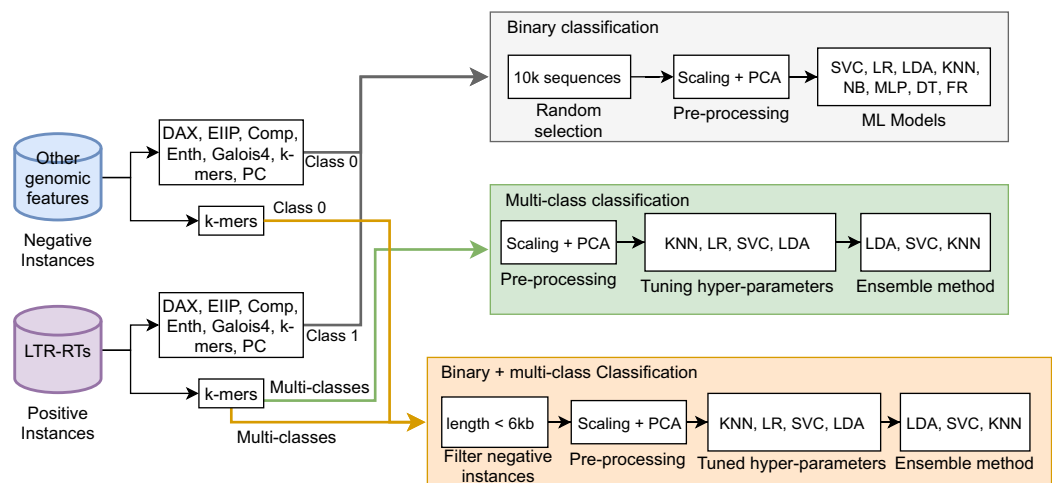
Next, a Stacking Classifier was implemented as an ensemble algorithm, which is a combination of multiple ML models for creating a more complex model (Zhang & Ma, 2012; Müller & Guido, 2016). The stacking classifier comprised LDA, Linear SVC, and KNN algorithms and used Random Forest as a meta-classifier. Similarly, for binary plus multi-class classification, we used a Stacking Classifier with KNN, LDA, and LR algorithms and Random Forest as a meta-classifier. Figure 1 summarizes the three approaches used in this study. The implementation in Python 3 of all algorithms used in this study is available in the Supplemental Material 2 or at https://github.com/simonorozcoarias/MachineLearningInTEs/blob/master/Scripts/binary_plus_multi_clasification.py.

Feature selection

We used the Gradient Boosting algorithm (Friedman, 2002) implemented in Scikit-Learn to determine the importance of each feature using the complete dataset (InpactorDB plus

Table 1 Machine Learning models and hyper-parameters tuned.

Classifier	Parameter	Range
KNN	neighbors	2,20,39,57,76,94,113,131,150
	weights	uniform, distance
	metric	euclidean, manhattan, chebyshev, minkowski, wminkowski, seuclidean, mahalanobis
	algorithm	auto, ball_tree, kd_tree, brute
Linear SVC	C	1×10^i with $i = -4$, to 5
	penalty	l1, l2
	loss	hinge, squared_hinge
	tol	10^{-1} , 10^{-2} , 10^{-4} , 10^{-8}
LR	C	1×10^i with $i = -4$, to 5
	tol	1×10^i with $i = -4$, to 5
	max_iter	1×10^i with $i = 0$, to 6
	penalty	l1, l2, elasticnet, none
LDA	solver	saga, liblinear, newton-cg, lbfgs, sag, saga
	shrinkage	1, 0.1, 0.5, 0.001, 0.0001, 0.00001
	solver	svd, lsqr, eigen
	tol	10^{-1} , 10^{-2} , 10^{-4} , 10^{-8}

**Figure 1** Schematic representation of the different approaches used in this study in the binary and multiclass classification tasks in LTR retrotransposons through Machine Learning.Full-size DOI: [10.7717/peerj.11456/fig-1](https://doi.org/10.7717/peerj.11456/fig-1)

negative instances). Gradient Boosting generates scores for each feature that represent how useful it was in the construction of the boosted decision trees. The more valuable the feature is for making key decisions, the higher its importance score (*Hastie, Tibshirani & Friedman, 2009*). Thus, we extracted features with an importance score greater than 60, 40, 30, 20, 10, which yielded 65, 289, 508, 1,034, and 2,397 features from 5,460 *k*-mers.

We used the following hyper-parameters: `boosting_type='goss'`, `n_estimators=10000`, and

Table 2 Composition of the negative instances dataset.

TE type	Order	Superfamily	Number
Class I (retrotransposons)	Non-LTR	LINEs	767
	Non-LTR	SINEs	551
	PLEs	Penelope	297
	DIRS	DIRS	356
	DIRS	VIPER	1
	DIRS	Ngaro	2
	Unclassified	Unclassified	1,039
	TOTAL		3,013
Class II (DNA transposons)	TIR	Tc1-Mariner	2,326
	TIR	hAT	2,587
	TIR	Merlin	65
	TIR	Transib	119
	TIR	PiggyBac	19
	TIR	PIF – Harbinger	973
	TIR	MuDR	1,016
	TIR	CACTA	57
	TIR	En-Spm	1,265
	TIR	MITE	2,312
	Crypton	Crypton	231
	Helitron	Helitron	740
	Unclassified	Unclassified	22,354
		TOTAL	
Non-TEs	RNA		262,925
	CDS		2,713,028
	Quimeric sequences		3,566
	TOTAL		2,979,519

class_weight='balance'. Finally, we extracted the selected features in order to create new reduced datasets that were used to train the same ensemble algorithm implemented in the binary plus multi-class classification task.

RESULTS

Binary classification of LTR retrotransposons and other genomic features

As negative instances, we obtained 2,713,028 coding sequences (CDS), 262,925 RNAs of different types (e.g., ncRNA, mRNA, miRNA, rRNA, snRNA, and tRNA), 37,077 TEs that did not correspond to LTR retrotransposons (i.e., TEs class II, LINEs, SINEs, DIRS, and PLEs), and 3566 quimeric sequences from Repbase, PGSB, and RepetDB (Table 2). These sequences, with the exception of TEs (outside LTR-RTs), were obtained from 47 plant species available in public databases, such as Ensembl Plants and JGI (Supplemental Material 1). We used sequences from InpactorDB as positive instances. Due to the high

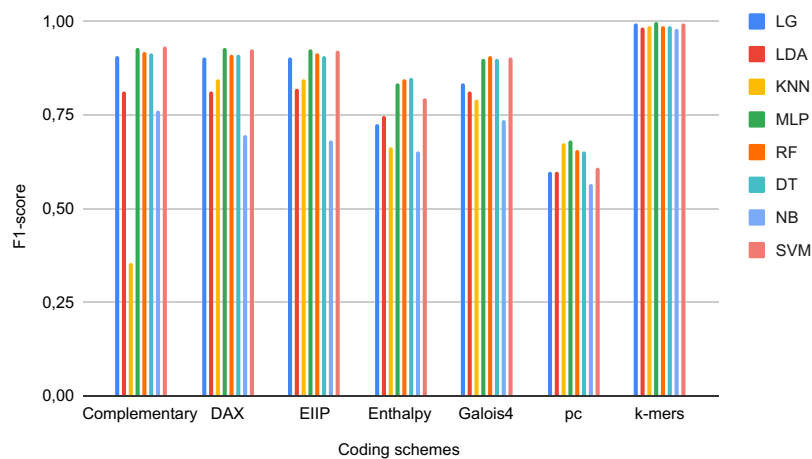


Figure 2 F1-Scores obtained by Machine Learning models using different coding schemes for the test set. [Full-size !\[\]\(fcc3264021d438d9732560e78099f674_img.jpg\) DOI: 10.7717/peerj.11456/fig-2](https://doi.org/10.7717/peerj.11456/fig-2)

imbalance between the two instances (2,979,519 negative vs 67,241 positive), we randomly extracted 10,000 sequences from each class. Then, the DNA sequences were converted to numerical representations using the coding schemes and automated techniques described in [Orozco-Arias et al. \(2020\)](#). Finally, we applied data scaling and a dimensional reduction through PCA ([Tabares-soto et al., 2020](#)). Using this dataset, we trained ML algorithms and determined their performance in terms of the F1-Score of each coding scheme over each ML model ([Fig. 2](#)). For the binary classification task, we obtained F1-Scores up to 97.9% 96.3%, and 95.9% for MLP, SVC, and LR, respectively, in the test dataset using k -mer frequencies as features.

Multi-class classification of LTR retrotransposons into lineages/families

For each of the selected models, a dictionary was created containing the hyper-parameters and the values to be iterated. After training each ML algorithm with GridsearchCV ([Tabares-soto et al., 2020](#)), we determined the parameters that generated the best performance, as shown in [Table 3](#). After tuning the hyper-parameters, each model was retrained to determine its performance. We obtained F1-Scores of 91%, 97%, 96%, and 97% with LR, KNN, LDA, and SVC algorithms, respectively ([Fig. 3](#)).

For the ensemble algorithm, the LR classifier was excluded since it showed the lowest performance ([Fig. 3A](#)). Therefore, the Stacking Classifier was implemented as an ensemble algorithm, composed of LDA, Linear SVC, and KNN algorithms, using Random Forest as meta-classifier. The performance of this ensemble model resulted in a 97% F1-Score, accuracy, recall, precision ([Fig. 4](#)), and 99% in area under ROC (receiver operating characteristic) curve (AUC) ([Fig. S1](#)) for the classification of LTR retrotransposons.

Binary plus multi-class classification task

After obtaining promising results in both the binary and multi-class classification tasks, we proceeded to merge them into a single ML problem. Thus, we included the negative

Table 3 Tuned parameter values.

Classifier	Parameter	Value
KNN	neighbors	2
	weights	distance
	metric	euclidean
	algorithm	auto
Linear SVC	C	0.001
	penalty	l2
	loss	squared_hinge
	tol	0.1
LR	C	0.01
	tol	10
	max_iter	1000
	penalty	l2
	solver	sag
LDA	shrinkage	0.0001
	solver	lsqr
	tol	0.1

instances as another class but deleted the sequences with a length of less than 6 Kb (Table 4). Furthermore, only k -mers frequencies were used as features because of the high performance obtained for the two problems separately. We also used the hyper-parameter values tuned (Table 3) for KNN, LDA, and LR (Fig. 5). Finally, F1-Scores of 95%, 94%, and 84% were obtained using KNN, LDA, and LR, respectively.

We implemented an ensemble method (the same implemented for the classification task) using the three algorithms aforementioned and used RF as a meta-classifier. We obtained an F1-Score of 96% in k -cross validation with $k = 9$ (Fig. 6). This method also obtained 95% of precision and recall and 98% in AUC (Fig. S2). Furthermore, as shown in Fig. 7, the classes with the lowest F1-Scores are Class 8 (Ikeros, Copia) and 16 (Galadriel, Gypsy) since these classes have the lowest number of samples.

Feature selection and evaluation

Using the Gradient Boosting algorithm and the entire dataset (negative instances plus InpactorDB), we obtained the importance of each feature (k -mers frequencies). The number of features is relatively high (5,460). Since the computational cost to process them can be very high, the number of features must be reduced without reducing the performance of the ML algorithm. Figure 8 shows the importance of all features determined by Gradient Boosting.

The results displayed in Fig. 8 demonstrate that some features are not relevant to the binary plus multi-class classification task. We extracted those with an importance score greater or equal to 40, thus, retaining only 289 features out of 5,460 (5.29%). The 10 most important features are: A, T, AAAAAA, ATAT, AGGGGG, CCCCC, TTTTTT, AGCT, GATC, GATGA with importance scores of 199, 179.5, 165, 140, 132.5, 132,

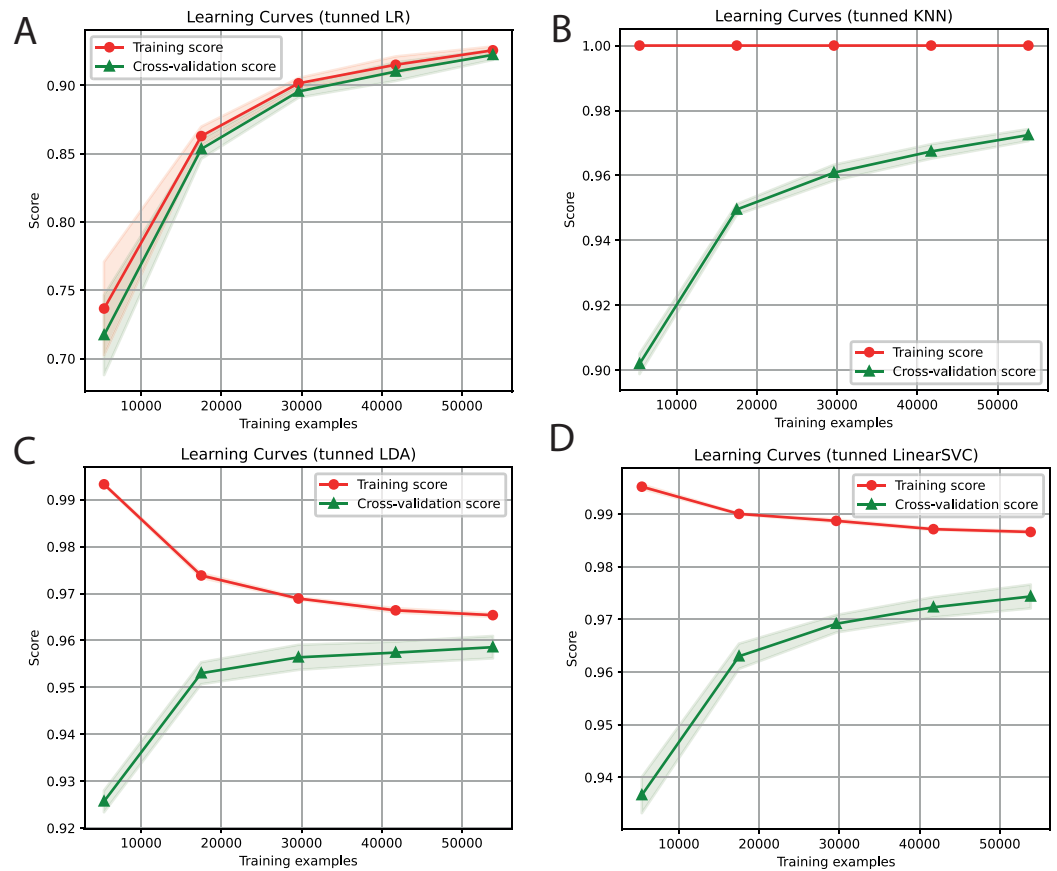


Figure 3 Learning curves of tuned ML algorithms: LR (A), KNN (B), LDA (C), and Linear SVC (D), to classify LTR retrotransposons up to the lineage level. The shadow on the lines indicates the standard variation. [Full-size !\[\]\(fcc3264021d438d9732560e78099f674_img.jpg\) DOI: 10.7717/peerj.11456/fig-3](https://doi.org/10.7717/peerj.11456/fig-3)

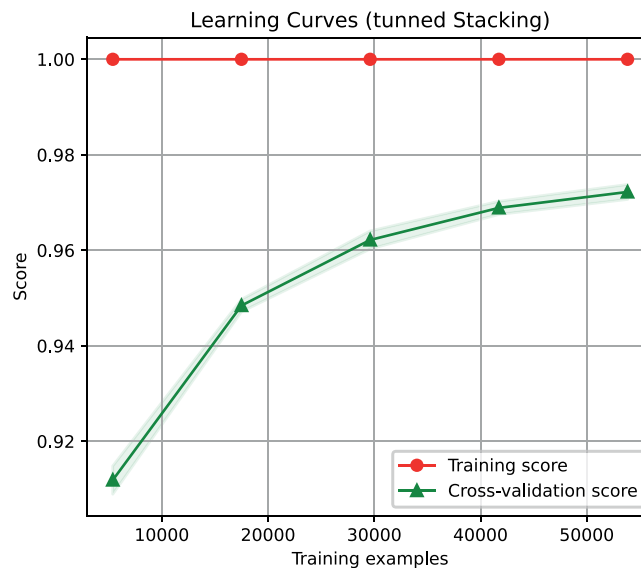


Figure 4 Learning curves of the Stacking classifier algorithm in the classification of LTR retrotransposons up to lineage/family level. The shadow on the line indicates the standard variation. [Full-size !\[\]\(9d188a796ceef961be962a3cd4b57b68_img.jpg\) DOI: 10.7717/peerj.11456/fig-4](https://doi.org/10.7717/peerj.11456/fig-4)

Table 4 Dataset composition used in the binary plus multiclass classification task.

Class	Superfamily	Name	Number of classes
0	–	Other genomic features	34,823
1	Copia	ALE/Retrofit	12,031
3	Copia	Angela	1,458
4	Copia	Bianca	1,827
8	Copia	Ikeros	84
9	Copia	Ivana/Oryco	3,556
12	Copia	Tork/Tar	6,180
13	Copia	SIRE	3,130
		Total Copia	28,266
14	Gypsy	CRM	2,136
16	Gypsy	Galadriel	549
17	Gypsy	Reina	4,532
18	Gypsy	Tekay/DEL	10,396
19	Gypsy	Athila	3,499
20	Gypsy	TAT	17,927
		Total Gypsy	39,039

125.5, 124, 124, 114.5, respectively. Among the 289 selected features, we observed that increasing the length of k decreases the percentage of top selected features with greater importance (Fig. 9). In total, the 289 selected features were composed of 4, 10, 32, 109, 97, 37 of k -mers generated using $k = 1, 2, 3, 4, 5$ and 6, respectively (Supplemental Material 3).

Then, we executed the ensemble method again using the reduced dataset containing the most important features. We also test different importance score thresholds in order to keep different number of features. We used the same pre-processing technique and hyper-parameter values of the previous execution. The results show that reducing the number of features to 1.73% (from 5,460 to 95), 5.29% (289 features), 9.3% (508 features), 18.93% (1,034 features), and 43.9% (2,397 features), did not considerably decrease performance, as indicated by a 93.5%, 95.2%, 95.6%, 95.4%, 95.6% F1-Score (Fig. 10), accuracy, recall, and precision, using as importance score threshold 60, 40, 30, 20, and 10 respectively. We also noted that using 289 features we obtained an 97% in AUC (Fig. S3).

DISCUSSION

Transposable element detection and classification are time-consuming tasks for plant genome annotation projects and involve semi-automatic pipelines and curation by experts. These pipelines generally rely on sequence alignment approaches, which have a significant computational cost in the analysis of large genomes or TE compound genomes. The development of automatic algorithms for TE detection and annotation, without sequence alignment, is now required to process the huge amount of genomic information that is being generated.

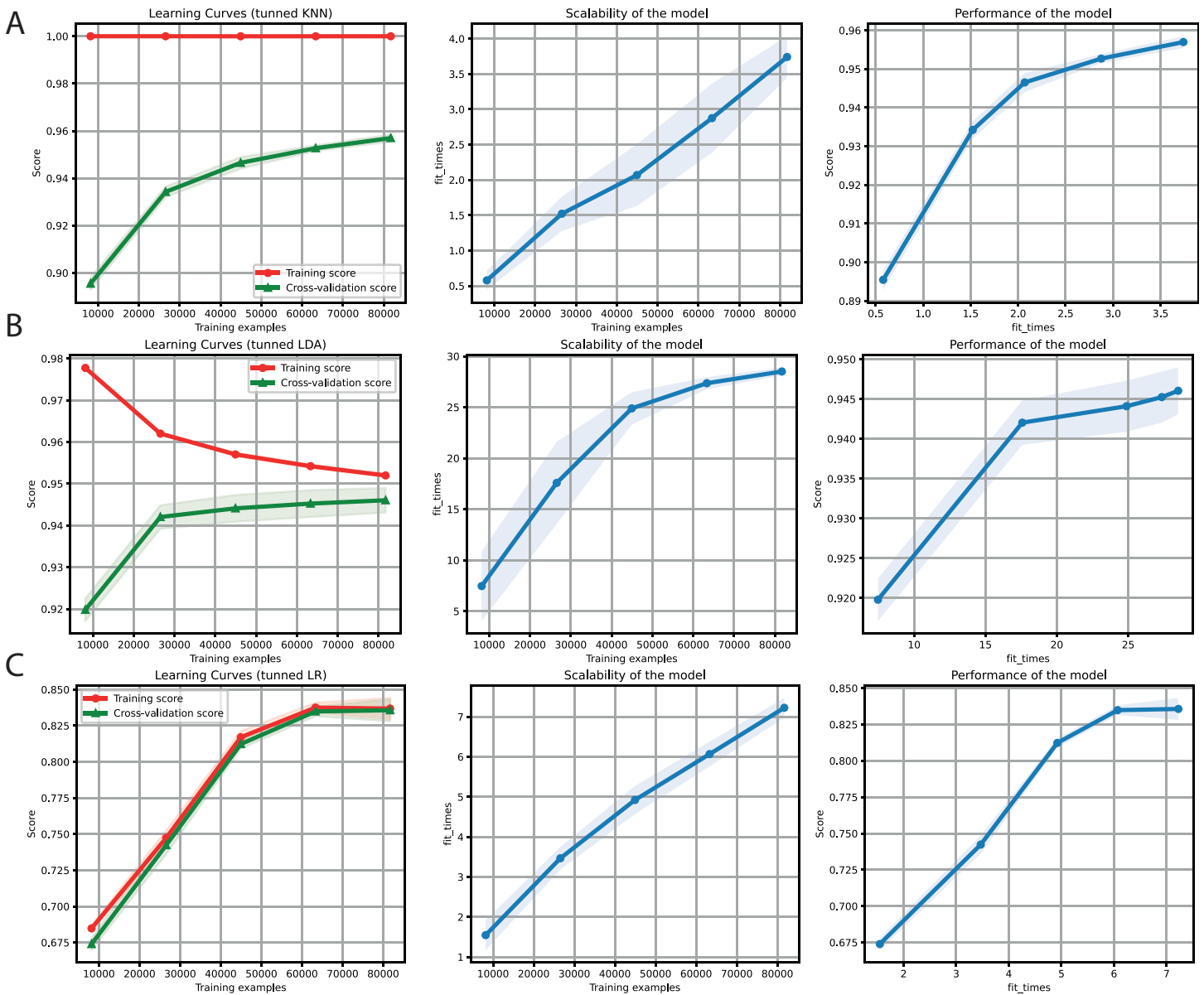


Figure 5 Learning curves for KNN (A), LDA (B), and LR (C) for the binary plus multiclass classification task. The shadow on the lines indicates the standard variation. [Full-size !\[\]\(ba1b80118482ccef74a5d718ca4d7242_img.jpg\) DOI: 10.7717/peerj.11456/fig-5](https://doi.org/10.7717/peerj.11456/fig-5)

The most frequent repeated sequences in plant genomes are LTR retrotransposons (*Gao et al., 2012; Grandbastien, 2015b*) Their transcriptional activities and mobilities can have a profound impact on the structure, composition, and size of genomes, among others (*Orozco-Arias, Isaza & Guyot, 2019*). To detect these elements in assembled genome sequences, the most frequent methods are based on their specific internal structure (e.g., the duplication of their LTR at both ends of the element) and domains, or via homology searches against reference libraries (*Loureiro et al., 2013; Nicolas, Peterlongo & Tempel, 2016*). Nevertheless, these strategies have strong limitations (*Orozco-Arias, Isaza & Guyot, 2019*). Structure-based methods will not accurately detect partial elements,

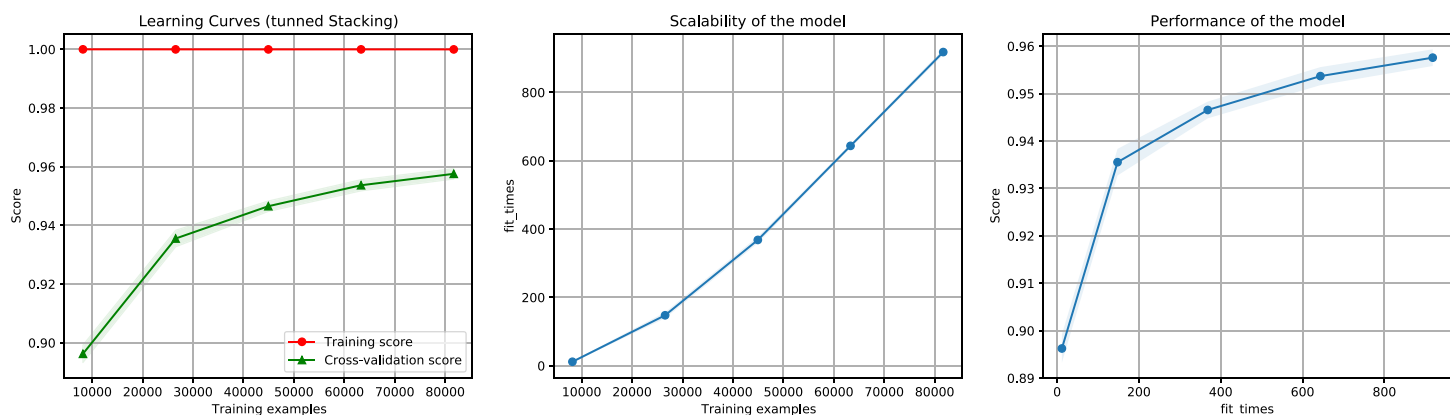


Figure 6 Learning curves for the Staking Classifier (ensemble method) for the binary plus multiclass classification task. The shadow on the lines indicates the standard variation. [Full-size !\[\]\(5f471a71b78d7676bc356df190b88ab4_img.jpg\) DOI: 10.7717/peerj.11456/fig-6](https://doi.org/10.7717/peerj.11456/fig-6)

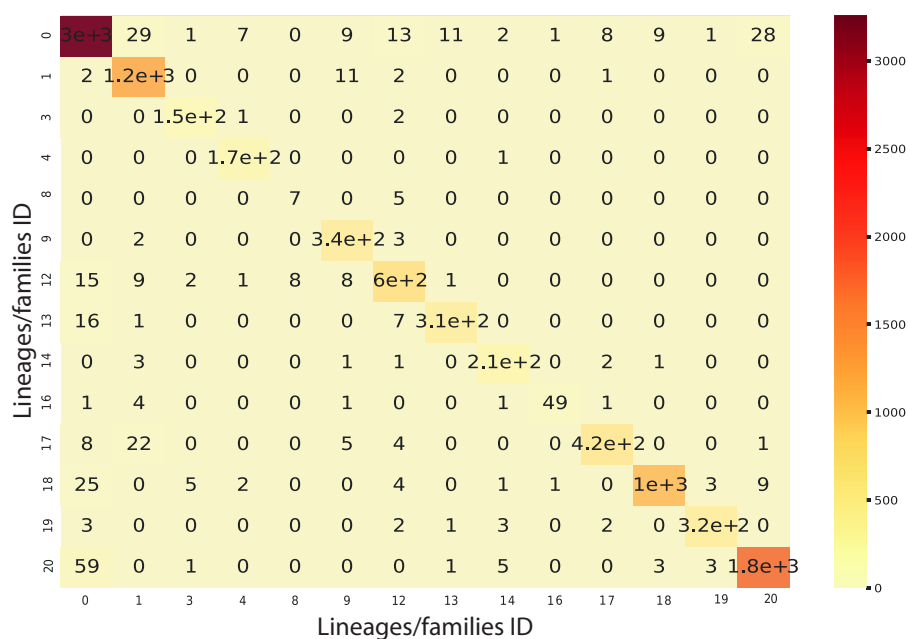


Figure 7 Confusion matrix for the Staking Classifier (ensemble method) used for the binary plus multiclass classification task. Each class (between 0 and 20) corresponds to the negative instances (for class 0) or to a lineage/family otherwise (See Table 4). [Full-size !\[\]\(e6d8ed0e56026ff17854aa495380637d_img.jpg\) DOI: 10.7717/peerj.11456/fig-7](https://doi.org/10.7717/peerj.11456/fig-7)

variations, or non-autonomous elements such as solo-LTR, TR-GAG (Chaparro *et al.*, 2015), or TRIM (Witte *et al.*, 2001). Moreover, homology-based methods can induce low quality annotations if the reference library is incomplete or constructed with distant species of the species to be annotated (Orozco-Arias, Isaza & Guyot, 2019). These problems can be overcome using new methods, such as machine learning methods, that are not based on structure or homology (i.e., sequence alignment) (Nakano *et al.*, 2018; Schietgat *et al.*, 2018; Orozco-Arias *et al.*, 2019; da Cruz *et al.*, 2020). Although some studies have used ML to analyze TEs, none of them have focused on the detection and lineage/family level classification of LTR retrotransposons. In previous studies (Orozco-Arias *et al.*,

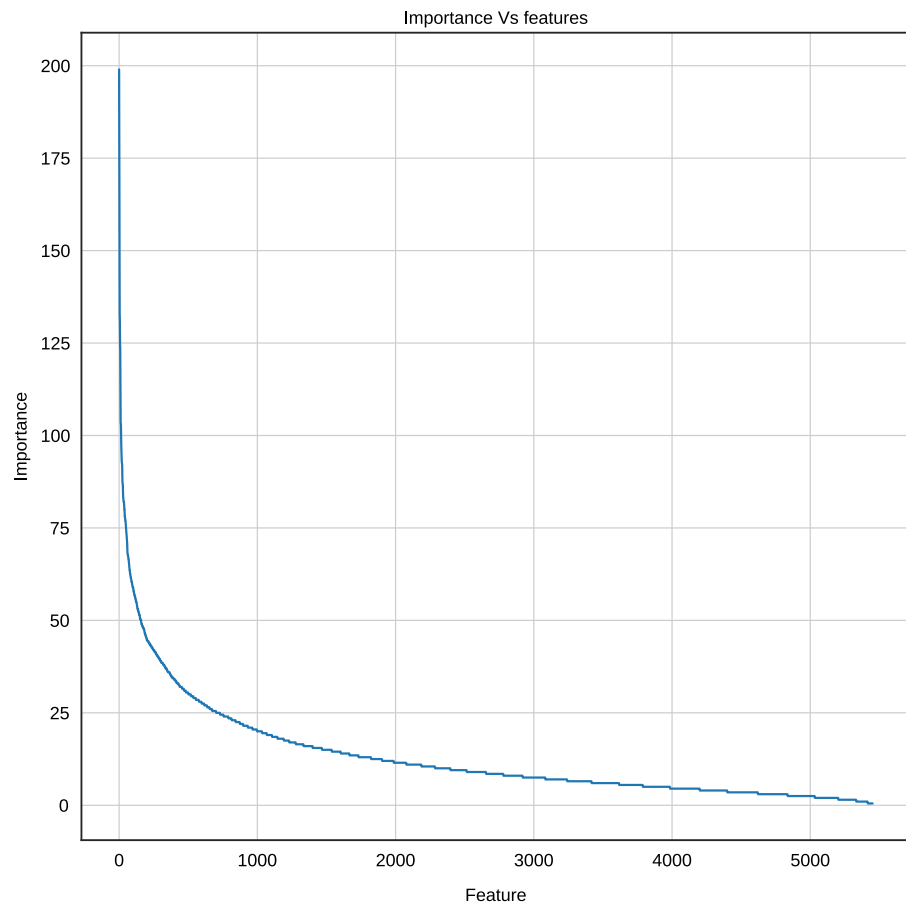


Figure 8 Importance score of the features (k -mer frequencies with $1 \leq k \leq 6$).

Full-size  DOI: 10.7717/peerj.11456/fig-8

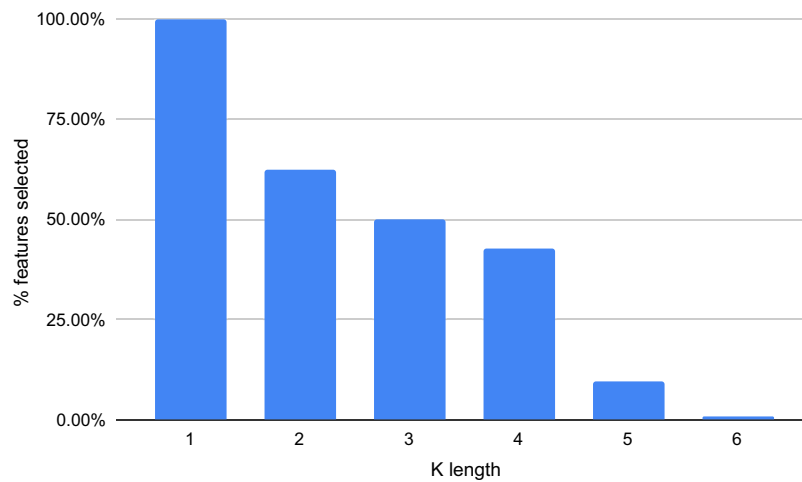


Figure 9 Percentage of features selected based on each value of k (between 1 and 6).

Full-size  DOI: 10.7717/peerj.11456/fig-9

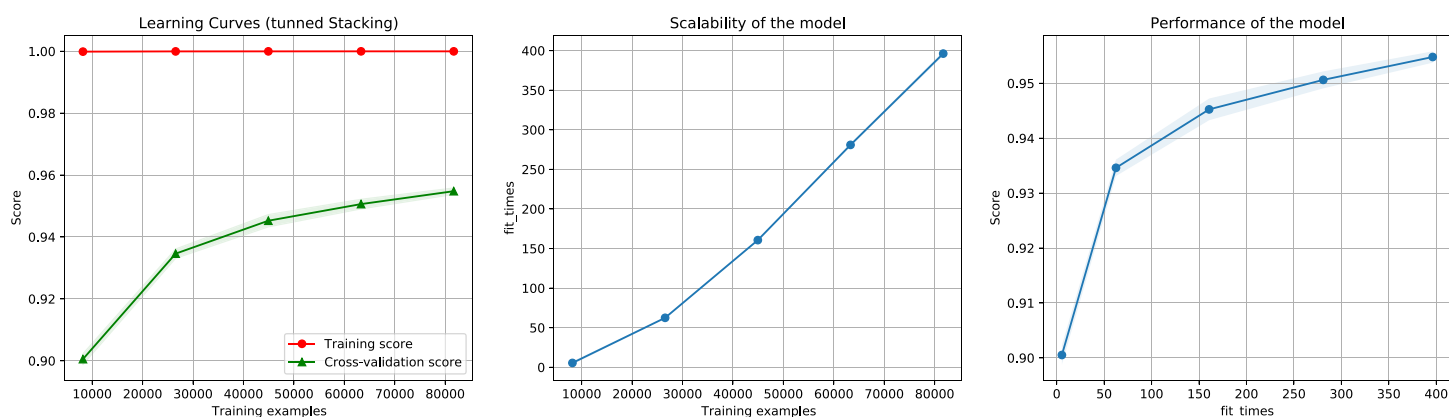


Figure 10 Learning curves for the Staking Classifier (ensemble method) for the binary plus multiclass classification task using only 289 selected features. The shadow in the lines indicates the standard variation. [Full-size !\[\]\(1679558f37f6db0dd8360a2a7e913e90_img.jpg\) DOI: 10.7717/peerj.11456/fig-10](https://doi.org/10.7717/peerj.11456/fig-10)

2020, 2021), we have shown that the classification of these elements can be very efficient with ML approaches but that coding schemes and feature extraction techniques can deeply influence the performance of such algorithms.

Here, we have shown the possibility of both binary and multi-class classification through ML with different approaches. First, by using a negative dataset (i.e., sequences of genomic features excluding LTR-RTs), we obtained an F1-Score up to 97.9% in the binary classification task. In the detailed lineage/family level classification task, we obtained a performance of 97% by using the same datasets as in [Orozco-Arias et al. \(2020\)](#) but including more elements from InpactorDB. Finally, we unified both tasks into a single process, obtaining a 96% F1-Score. These results were achieved using the k -mer frequencies of sequences. K -mers are frequently used in bioinformatics in the areas of quality control of generated sequences ([Mapleson et al., 2017](#)), metagenomics ([Breitwieser, Baker & Salzberg, 2018](#)), de novo assembly of genomes and transcriptomes ([Zerbino & Birney, 2008](#); [Simpson et al., 2009](#)), genome size estimation ([Sun et al., 2018](#)), and de novo detection of transposable elements ([Price, Jones & Pevzner, 2005](#)). In addition to detection, k -mers have also demonstrated their usefulness in higher hierarchical classification approaches (at the superfamily level). Nakano and colleagues ([Nakano et al., 2018](#)) exploited the characteristics of k -mer frequencies (with $k = 2,3,4$) as features to train a deep neural network, and [Yan, Bombarely & Li \(2020\)](#) demonstrated interesting statistical differences using k -mer frequencies with k ranging from 3 to 7. Here, we found that the combination of certain k -mer frequencies with different values of k (from 1 to 6) added reliability to the binary plus multi-class classification task, requiring only 289 features (out of the initial 5,460 features) to obtain an F1-Score of 95%. We also demonstrated that, among all possible k -mers generated, larger k values reduce the number of features with high importance scores. The specific k -mer that contributes the most to the binary and multi-class classification characteristics must be analyzed in detail to understand its involvement in the composition and structure of the elements of each lineage. This work highlights the feasibility of designing and implementing ML-based tools that automate the complex process of annotating LTR retrotransposons in plant genomes. As future work,

we propose the implementation of benchmarking to compare this method with well-established methods such as those based on homology and structure in various plant genomes.

CONCLUSIONS

Massive sequencing projects require automatic tools to analyze large amounts of genomic information in a fast yet accurate, reliable, and reproducible manner. The binary classification of LTR-RTs and other genomic features and lineage-level classification of them in plant genomes can be performed using ML-based and ensemble methods, demonstrating good performance (up to 96% F1-Score). This task can be performed with only 289 *k*-mer frequencies, allowing low computational resources and time. These results can be used in the design and implementation of automatic and alignment-free tools to solve the issue of processing the increasing number of available plant genomes.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Simon Orozco-Arias is supported by a Ph.D. grant from the Ministry of Science, Technology and Innovation (Minciencias) of Colombia, Grant Call 785/2017. The authors and publication fees were supported by Universidad Autónoma de Manizales, Manizales, Colombia under project 589-089. This work was supported by Ecos-Nord N°C21MA01 and STICAMSUD 21-STIC-13. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:

Ministry of Science, Technology and Innovation (Minciencias) of Colombia: 785/2017.

Universidad Autónoma de Manizales, Manizales, Colombia: 589-089.

Ecos-Nord: N°C21MA01.

STICAMSUC 21-STIC-13.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Simon Orozco-Arias conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Mariana S. Candamil-Cortés performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Paula A. Jaimes performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

- Johan S. Piña performed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Reinel Tabares-Soto conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.
- Romain Guyot conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.
- Gustavo Isaza conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The implementation in Python 3 of all algorithms is available at GitHub: https://github.com/simonorozcoarias/MachineLearningInTEs/blob/master/Scripts/binary_plus_multi_clasification.py.

The positive instances dataset (LTR-Retrotransposons) is available at Zenodo: Orozco-Arias Simon, Jaimes Paula A., Candamil Mariana, Jiménez-Varón Cristian Felipe, Tabares-Soto Reinel, Isaza Gustavo, & Guyot Romain. (2021). InpactorDB: A Plant classified lineage-level LTR retrotransposon reference library for free-alignment methods based on Machine Learning (Version V3) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.4386316>.

Negative instances datasets (other genomic features) are available at Zenodo: Simon Orozco Arias, Mariana S. Candamil, Paula A. Jaimes, Johan S. Piña, Reinel Tabares-Soto, Romain Guyot, & Gustavo Isaza. (2021). Negative instances for detect LTR-Retrotransposons using Machine Learning (Version 1) [Data set]. Zenodo. DOI [10.5281/zenodo.4543904](https://doi.org/10.5281/zenodo.4543904).

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj.11456#supplemental-information>.

REFERENCES

- Akhtar M, Epps J, Ambikairajah E. 2008.** Signal processing in sequence analysis: advances in eukaryotic gene prediction. *IEEE Journal of Selected Topics in Signal Processing* 2(3):310–321 DOI [10.1109/JSTSP.2008.923854](https://doi.org/10.1109/JSTSP.2008.923854).
- Amselem J, Cornut G, Choisne N, Alaux M, Alfama-Depauw F, Jamilloux V, Maumus F, Letellier T, Luyten I, Pommier C, Adam-Blondon AF, Quesneville H. 2019.** RepetDB: a unified resource for transposable element references. *Mobile DNA* 10(1):4–11 DOI [10.1186/s13100-019-0150-y](https://doi.org/10.1186/s13100-019-0150-y).
- Arango-López J, Orozco-Arias S, Salazar JA, Guyot R, Arango-Lopez J, Orozco-Arias S, Salazar JA, Guyot R. 2017.** Application of Data Mining Algorithms to Classify Biological Data: The Coffea canephora Genome Case. In: *Communications in Computer and Information Science*, Cali, Colombia, 156–170.
- Arkhipova IR. 2017.** Using bioinformatic and phylogenetic approaches to classify transposable elements and understand their complex evolutionary histories. *Mobile DNA* 8(1):19 DOI [10.1186/s13100-017-0103-2](https://doi.org/10.1186/s13100-017-0103-2).

- Ashlock W, Datta S. 2012. Distinguishing endogenous retroviral LTRs from SINE elements using features extracted from evolved side effect machines. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9(6):1676–1689 DOI 10.1109/TCBB.2012.116.
- Bao W, Kojima KK, Kohany O. 2015. Repbase update, a database of repetitive elements in eukaryotic genomes. *Mobile DNA* 6(1):4–9 DOI 10.1186/s13100-015-0041-9.
- Bolser DM, Staines DM, Perry E, Kersey PJ. 2017. Ensembl plants: integrating tools for visualizing, mining, and analyzing plant genomic data. *Methods in Molecular Biology* 1533:1–31 DOI 10.1007/978-1-4939-6658-5_1.
- Breitwieser FP, Baker DN, Salzberg SL. 2018. KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. *Genome Biology* 19(1):1–10 DOI 10.1186/s13059-018-1568-0.
- Chaparro C, Gayraud T, De Souza RF, Domingues DS, Akaffou SS, Vanzela ALL, De Kochko A, Rigoreau M, Cruzillat D, Hamon S, Hamon P, Guyot R, Laforga Vanzela AL, de Kochko A, Rigoreau M, Cruzillat D, Hamon S, Hamon P, Guyot R. 2015. Terminal-repeat retrotransposons with GAG domain in plant genomes: a new testimony on the complex world of transposable elements. *Genome Biology and Evolution* 7(2):493–504 DOI 10.1093/gbe/evv001.
- Chen W, Feng PM, Deng EZ, Lin H, Chou KC. 2014. iTIS-PseTNC: a sequence-based predictor for identifying translation initiation site in human genes using pseudo trinucleotide composition. *Analytical Biochemistry* 462:76–83 DOI 10.1016/j.ab.2014.06.022.
- Cornut G, Choisine N, Alaux M, Alfama-Depauw F, Jamilloux V, Maumus F, Letellier T, Luyten I, Pommier C, Adam-Blondon A-F, Quesneville H. 2019. RepetDB: a unified resource for transposable element references. *Mobile DNA* 10:6.
- da Cruz MHP, Domingues DS, Saito PTM, Paschoal AR, Bugatti PH. 2020. TERL: classification of transposable elements by convolutional neural networks. *Briefings in Bioinformatics* 19(10):1329 DOI 10.1093/bib/bbaa185.
- da Cruz MHP, Saito PTM, Paschoal AR, Bugatti PH. 2019. Classification of transposable elements by convolutional neural networks. In: Rutkowski L, Scherer R, Korytkowski M, Pedrycz W, Tadeusiewicz R, Zurada J, eds. *Artificial Intelligence and Soft Computing. ICAISC 2019. Lecture Notes in Computer Science*. Vol. 11509. Cham: Springer DOI 10.1007/978-3-030-20915-5_15.
- Eraslan G, Avsec Ž, Gagneur J, Theis FJ. 2019. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics* 20(7):389–403 DOI 10.1038/s41576-019-0122-6.
- Friedman JH. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38(4):367–378 DOI 10.1016/S0167-9473(01)00065-2.
- Gao D, Jimenez-Lopez JC, Iwata A, Gill N, Jackson SA. 2012. Functional and structural divergence of an unusual LTR retrotransposon family in plants. *PLOS ONE* 7(10):e48595 DOI 10.1371/journal.pone.0048595.
- Grandbastien M-A. 2015a. LTR retrotransposons, handy hitchhikers of plant regulation and stress response. *Biochimica et Biophysica Acta* 1849(4):403–416 DOI 10.1016/j.bbagr.2014.07.017.
- Grandbastien M-AA. 2015b. LTR retrotransposons, handy hitchhikers of plant regulation and stress response. *Biochimica et Biophysica Acta - Gene Regulatory Mechanisms* 1849(4):403–416 DOI 10.1016/j.bbagr.2014.07.017.
- Hastie T, Tibshirani R, Friedman J. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Berlin: Springer Science & Business Media.
- Hesam TD, Ali M-N. 2010. Mining biological repetitive sequences using support vector machines and fuzzy SVM. *Iranian Journal of Chemistry and Chemical Engineering* 29:1–17.

- Jaiswal AK, Krishnamachari A. 2019. Physicochemical property based computational scheme for classifying DNA sequence elements of *Saccharomyces cerevisiae*. *Computational Biology and Chemistry* 79(37):193–201 DOI 10.1016/j.compbiolchem.2018.12.014.
- Jiang S-Y, Ramachandran S. 2013. Genome-wide survey and comparative analysis of LTR retrotransposons and their captured genes in rice and sorghum. *PLOS ONE* 8(7):e71118 DOI 10.1371/journal.pone.0071118.
- Jurka J, Kapitonov VV, Pavlicek A, Klonowski P, Kohany O, Walichiewicz J. 2005. Repbase Update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research* 110(1–4):462–467 DOI 10.1159/000084979.
- Kauer G, Blöcker H. 2003. Applying signal theory to the analysis of biomolecules. *Bioinformatics* 19(16):2016–2021 DOI 10.1093/bioinformatics/btg273.
- Komer B, Bergstra J, Eliasmith C. 2014. Hyperopt-Sklearn: automatic hyperparameter configuration for Scikit-Learn. *Scipy* 2014:33–39 DOI 10.25080/issn.2575-9752.
- Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armañanzas R, Santafé G, Pérez A, Robles V. 2006. Machine learning in bioinformatics. *Briefings in Bioinformatics* 7(1):86–112 DOI 10.1093/bib/bbk007.
- Li H, Tian S, Li Y, Fang Q, Tan R, Pan Y, Huang C, Xu Y, Gao X. 2020. Modern deep learning in bioinformatics. *Journal of Molecular Cell Biology* 12(11):823–827 DOI 10.1093/jmcb/mjaa030.
- Libbrecht MW, Noble WS. 2015. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* 16(6):321–332 DOI 10.1038/nrg3920.
- Loureiro T, Camacho R, Vieira J, Fonseca NA. 2013. Improving the performance of transposable elements detection tools. *Journal of Integrative Bioinformatics* 10(3):231 DOI 10.1515/jib-2013-231.
- Mapleson D, Garcia Accinelli G, Kettleborough G, Wright J, Clavijo BJ. 2017. KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. *Bioinformatics* 33:574–576.
- McCarthy EM, McDonald JF. 2003. LTR STRUC: a novel search and identification program for LTR retrotransposons. *Bioinformatics* 19(3):362–367 DOI 10.1093/bioinformatics/btf878.
- Müller AC, Guido S. 2016. *Introduction to machine learning with Python: a guide for data scientists*. Newton: O'Reilly Media, Inc.
- Mustafin RN, Khusnutdinova EK. 2018. The role of transposons in epigenetic regulation of ontogenesis. *Russian Journal of Developmental Biology* 49(2):61–78 DOI 10.1134/S1062360418020066.
- Nair AS, Sreenadhan SP. 2006. A coding measure scheme employing electron-ion interaction pseudopotential (EIIP). *Bioinformation* 1:197.
- Nakano FK, Mastelini SM, Barbon S, Cerri R. 2018. Improving hierarchical classification of transposable elements using deep neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*. Piscataway: IEEE.
- Nakano FK, Pinto WJ, Pappa GL, Cerri R. 2017. Top-down strategies for hierarchical classification of transposable elements with neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*, Anchorage: United States, 2539–2546.
- Neumann P, Novák P, Hošťáková N, MacAs J. 2019. Systematic survey of plant LTR-retrotransposons elucidates phylogenetic relationships of their polyprotein domains and provides a reference for element classification. *Mobile DNA* 10(1):e1000732 DOI 10.1186/s13100-018-0144-1.

- Nicolas J, Peterlongo P, Tempel S. 2016. Finding and characterizing repeats in plant genomes. In: Edwards D, ed. *Plant Bioinformatics*. New York: Springer, 293–337.
- Nordberg H, Cantor M, Dusheyko S, Hua S, Poliakov A, Shabalov I, Smirnova T, Grigoriev IV, Dubchak I. 2014. The genome portal of the Department of Energy Joint Genome Institute: 2014 updates. *Nucleic Acids Research* 42(D1):26–31 DOI 10.1093/nar/gkt1069.
- Orozco-Arias S, Isaza G, Guyot R. 2019. Retrotransposons in plant genomes: structure, identification, and classification through bioinformatics and machine learning. *International Journal of Molecular Sciences* 20(15):1–29 DOI 10.3390/ijms20153837.
- Orozco-Arias S, Isaza G, Guyot R, Tabares-soto R. 2019. A systematic review of the application of machine learning in the detection and classification of transposable elements. *PeerJ* 7(10):18311 DOI 10.7717/peerj.8311.
- Orozco-Arias S, Jaimes PA, Candamil MS, Jiménez-Varón CF, Tabares-soto R, Isaza G, Guyot R. 2021. InpactorDB: a classified lineage-level plant LTR retrotransposon reference library for free-alignment methods based on machine learning. *MDPI Genes* 12(2):17 DOI 10.3390/genes12020190.
- Orozco-Arias S, Piña JS, Tabares-soto R, Castillo-Ossa LF, Guyot R, Isaza G. 2020. Measuring performance metrics of machine learning algorithms for detecting and classifying transposable elements. *Processes* 8(6):1–20 DOI 10.3390/pr8060638.
- Orozco-Arias S, Tabares-Soto R, Ceballos D, Guyot R. 2017. Parallel programming in biological sciences, taking advantage of supercomputing in genomics. In: Solano A, Ordoñez H, eds. *Advances in Computing*. Zurich: Springer, 627–643.
- Ou S, Chen J, Jiang N. 2018. Assessing genome assembly quality using the LTR Assembly Index (LAI). *Nucleic Acids Research* 29:1–11 DOI 10.1093/nar/gky730.
- Ou S, Su W, Liao Y, Chougule K, Agda JRA, Hellinga AJ, Lugo CSB, Elliott TA, Ware D, Peterson T, Jiang N, Hirsch CN, Hufford MB. 2019. Benchmarking transposable element annotation methods for creation of a streamlined, comprehensive pipeline. *Genome Biology* 20(1):275 DOI 10.1186/s13059-019-1905-y.
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. 2011. Scikit-learn: machine learning in python. *Journal of Machine Learning Research* 12:2825–2830.
- Price AL, Jones NC, Pevzner PA. 2005. De novo identification of repeat families in large genomes. *Bioinformatics* 21(Suppl. 1):351–358 DOI 10.1093/bioinformatics/bti1018.
- Rahman AYA, Usharraj AO, Misra BB, Thottathil GP, Jayasekaran K, Feng Y, Hou S, Ong SY, Ng FL, Lee LS, Tan HS, Sakaff MKLM, Teh BS, Khoo B, Badai SS, Aziz NA, Yuryev A, Knudsen B, Dionne-Laporte A, Mchunu NP, Yu Q, Langston BJ, Freitas TAK, Young AG, Chen R, Wang L, Najimudin N, Saito JA, Alam M. 2013. Draft genome sequence of the rubber tree *Hevea brasiliensis*. *BMC Genomics* 14:75.
- Rawal K, Ramaswamy R. 2011. Genome-wide analysis of mobile genetic element insertion sites. *Nucleic Acids Research* 39(16):6864–6878 DOI 10.1093/nar/gkr337.
- Rigal M, Mathieu O. 2011. A mille-feuille of silencing: epigenetic control of transposable elements. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1809(8):452–458 DOI 10.1016/j.bbagr.2011.04.001.
- Rishishwar L, Wang L, Clayton EA, Mariño-Ramírez L, McDonald JF, Jordan IK. 2017. Population and clinical genetics of human transposable elements in the (post) genomic era. *Mobile Genetic Elements* 7(1):1–20 DOI 10.1080/2159256X.2017.1280116.

- Rosen GL. 2006.** *Signal processing for biologically-inspired gradient source localization and DNA sequence analysis*. Atlanta: Georgia Institute of Technology.
- Schietgat L, Vens C, Cerri R, Fischer CN, Costa E, Ramon J, Carareto CMA, Blockeel H. 2018.** A machine learning based framework to identify and classify long terminal repeat retrotransposons. *PLOS Computational Biology* **14(4)**:e1006097
DOI [10.1371/journal.pcbi.1006097](https://doi.org/10.1371/journal.pcbi.1006097).
- Shastri KA, Sanjay HA. 2020.** Machine learning for bioinformatics. In: *Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications*, New York: Springer, 25–39.
- Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol I. 2009.** ABySS: a parallel assembler for short read sequence data. *Genome Research* **19(6)**:1117–1123
DOI [10.1101/gr.089532.108](https://doi.org/10.1101/gr.089532.108).
- Spannagl M, Bader K, Pfeifer M, Nussbaumer T, Mayer KFX. 2016a.** PGSB/MIPS plant genome information resources and concepts for the analysis of complex grass genomes. *Methods in Molecular Biology* **1374**:165–186 DOI [10.1007/978-1-4939-3167-5_8](https://doi.org/10.1007/978-1-4939-3167-5_8).
- Spannagl M, Nussbaumer T, Bader KC, Martis MM, Seidel M, Kugler KG, Gundlach H, Mayer KFX. 2016b.** PGSB plantsDB: Updates to the database framework for comparative plant genome research. *Nucleic Acids Research* **44(D1)**:D1141–D1147 DOI [10.1093/nar/gkv1130](https://doi.org/10.1093/nar/gkv1130).
- Sun H, Ding J, Piednoël M, Schneeberger K. 2018.** findGSE: estimating genome size variation within human and Arabidopsis using k-mer frequencies. *Bioinformatics* **34(4)**:550–557
DOI [10.1093/bioinformatics/btx637](https://doi.org/10.1093/bioinformatics/btx637).
- Tabares-soto R, Orozco-Arias S, Romero-Cano V, Segovia Bucheli V, Rodríguez-Sotelo JL, Jiménez-Varón CF. 2020.** A comparative study of machine learning and deep learning algorithms to classify cancer types based on microarray gene expression. *PeerJ Computer Science* **6(1)**:1–22 DOI [10.7717/peerj-cs.270](https://doi.org/10.7717/peerj-cs.270).
- Wicker T, Sabot F, Hua-Van A, Bennetzen JL, Capy P, Chalhoub B, Flavell A, Leroy P, Morgante M, Panaud O, Paux E, SanMiguel P, Schulman AH. 2007.** A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics* **8(12)**:973–982
DOI [10.1038/nrg2165](https://doi.org/10.1038/nrg2165).
- Witte C-P, Le QH, Bureau T, Kumar A. 2001.** Terminal-repeat retrotransposons in miniature (TRIM) are involved in restructuring plant genomes. *Proceedings of The National Academy of Sciences of The United States of America* **98(24)**:13778–13783 DOI [10.1073/pnas.241341898](https://doi.org/10.1073/pnas.241341898).
- Wold S, Esbensen K, Geladi P. 1987.** Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **2(1–3)**:37–52 DOI [10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).
- Yan H, Bombarely A, Li S. 2020.** DeepTE: a computational method for de novo classification of transposons with convolutional neural network. *Bioinformatics*
DOI [10.1101/2020.01.27.921874](https://doi.org/10.1101/2020.01.27.921874).
- Yu N, Guo X, Gu F, Pan Y. 2015.** DNA AS X: an information-coding-based model to improve the sensitivity in comparative gene analysis. In: *International Symposium on Bioinformatics Research and Applications*. 366–377.
- Zerbino DR, Birney E. 2008.** Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* **18(5)**:821–829 DOI [10.1101/gr.074492.107](https://doi.org/10.1101/gr.074492.107).
- Zhang C, Ma Y. 2012.** *Ensemble machine learning: methods and applications*. Boston: Springer.
- Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. 2018.** A primer on deep learning in genomics. *Nature Genetics* **51(1)**:12–18 DOI [10.1038/s41588-018-0295-5](https://doi.org/10.1038/s41588-018-0295-5).