A systematic review of the application of Machine Learning in the detection and classification of transposable elements

Simon Orozco-Arias [Corresp., 1, 2], **Gustavo Isaza** [2], **Romain Guyot** [3, 4], **Reinel Tabares-Soto** [4]

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

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

Corresponding Author: Simon Orozco-Arias
Email address: simon.orozco.arias@gmail.com

Background. Transposable elements (TEs) constitute the most common repeated sequences in eukaryotic genomes. Recent studies demonstrated their deep impact on species diversity, adaptation to the environment, and diseases. Although there are many conventional bioinformatics algorithms for detecting and classifying TEs, none have achieved reliable results on different types of TEs. Machine learning (ML) techniques can automatically extract hidden patterns and novel information from labeled or non-labeled data and have been applied to solving several scientific problems.

Methodology. We followed the Systematic Literature Review process, applying the six stages of the review protocol from it, but added a previous stage, which aims to detect the need for a review. Then search equations were formulated and executed in several literature databases. Relevant publications were scanned and used to extract evidence to answer research questions.

Results. Several ML approaches have already been tested on other bioinformatics problems with promising results, yet there are few algorithms and architectures available in literature focused specifically on TEs, despite representing the majority of the nuclear DNA of many organisms. Only 35 papers were found and categorized as relevant in TE or related fields.

Conclusions. ML is a powerful tool that can be used to address many problems. Although ML techniques have been used widely in other biological tasks, their utilization in TE analyses is still limited. Following the Systematic Literature Review process, it was possible to notice that the use of ML for TE analyses (detection and classification) is an open problem, and this new field of research is growing in interest.

1 A systematic review of the application of Machine Learning in the detection and
2 classification of transposable elements

3 Simon Orozco-Arias[1, 2], Gustavo Isaza[2], Romain Guyot[3, 4], Reinel Tabares-Soto[3]

4 [1] Department of Computer Science, Universidad Autónoma de Manizales, Manizales,
5 Colombia

6 [2] Department of Systems and Informatics, Universidad de Caldas, Manizales, Colombia

7 [3] Department of Electronics and Automation, Universidad Autónoma de Manizales,
8 Manizales, Colombia

9 [4] Institut de Recherche pour le Développement, CIRAD, Univ. Montpellier, France

10 Corresponding Author:

11 Simon Orozco-Arias[1,2]

12 Calle 65 No. 26-10, Manizales, Caldas, 170002, Colombia.

13 Email address: simon.orozco.arias@gmail.com

Abstract

15 **Background.** Transposable elements (TEs) constitute the most common repeated
16 sequences in eukaryotic genomes. Recent studies demonstrated their deep impact on
17 species diversity, adaptation to the environment, and diseases. Although there are
18 many conventional bioinformatics algorithms for detecting and classifying TEs, none
19 have achieved reliable results on different types of TEs. Machine learning (ML)
20 techniques can automatically extract hidden patterns and novel information from labeled
21 or non-labeled data and have been applied to solving several scientific problems.

22 **Methodology.** We followed the Systematic Literature Review process, applying the six
23 stages of the review protocol from it, but added a previous stage, which aims to detect
24 the need for a review. Then search equations were formulated and executed in several
25 literature databases. Relevant publications were scanned and used to extract evidence
26 to answer research questions.

27 **Results.** Several ML approaches have already been tested on other bioinformatics
28 problems with promising results, yet there are few algorithms and architectures
29 available in literature focused specifically on TEs, despite representing the majority of
30 the nuclear DNA of many organisms. Only 35 papers were found and categorized as
31 relevant in TE or related fields.

32 **Conclusions**. ML is a powerful tool that can be used to address many problems.
33 Although ML techniques have been used widely in other biological tasks, their utilization
34 in TE analyses is still limited. Following the Systematic Literature Review process, it

35 was possible to notice that the use of ML for TE analyses (detection and classification)
36 is an open problem, and this new field of research is growing in interest.

37 **Keywords:** transposable elements, retrotransposons, detection, classification,
38 bioinformatics, machine learning, deep learning

Introduction

40 Transposable elements (TEs) are genomic units with the ability to move from one locus
41 to another within the genome. TEs have been found in all organisms and comprise the
42 majority of the nuclear DNA content of plant genomes (Orozco-arias et al., 2018), such
43 as in wheat, barley, and maize. In these species, up to 85% of the sequenced DNA is
44 classified into repeated sequences (Choulet et al., 2014), of which TEs represent the
45 most abundant and functionally relevant type (Ventola et al., 2017). Due to the high
46 diversity of TE structures and transposition mechanisms, there are still numerous
47 classification problems and debates on the classification systems (Piégu et al., 2015).
48 TEs in eukaryotes are traditionally classified based on if the reverse transcription is
49 needed for transposition (Class I or retrotransposons) or not (Class II or DNA
50 transposons) (Schietgat et al., 2018). Retrotransposons can be further subclassified into
51 four orders according to structural features and the life cycle of the element.

52 In plants, Long Terminal Repeat retrotransposon (LTR-RT) is the most frequent order
53 (Gao et al., 2012; Grandbastien, 2015) and can contribute up to 80% of the plant
54 genome size (e.g. in wheat, barley or the rubber tree) (Rahman et al., 2013). However,
55 in humans, the non-LTR-RT order is the most common and is related to diseases such
56 as cancer (Tang et al., 2017). Other levels of classification of TEs include sub-classes
57 (only for DNA transposons which are distinguished by the number of DNA strands that
58 are cut during transposition (Wicker et al., 2007)), superfamilies, lineages, and families
59 (de Castro Nunes et al., 2018; Neumann et al., 2019).

60 Although several methods have been developed to detect TEs in genomes, including de
61 novo, structure-based, comparative genomic, and homology-based (reviewed in
62 (Orozco-Arias, Isaza & Guyot, 2019)), there is no single bioinformatics tool achieving
63 reliable results on different types of TEs (Loureiro et al., 2013b). Most of the algorithms
64 available currently use a homology-based approach (Nakano et al., 2018b), but this
65 method can present limited potential due to the vast diversity at the nucleotide level of
66 TEs. Also, the repetitive nature of TEs, as well as their structural polymorphism, species
67 specificity, and high divergence rate even among close relative species (Mustafin &
68 Khusnutdinova, 2018), represent significant obstacles and challenges for their analysis
69 (Ou, Chen & Jiang, 2018). Despite of the complexity, a well-curated detection and
70 classification of TEs is important, due to these elements have key roles into genomes,
71 such as in the chromosomal structure, their interaction with genes, and adaptation and
72 evolution processes (Orozco-Arias, Isaza & Guyot, 2019) and their annotation could
73 provide insights into genomic dynamics (Wheeler et al., 2012).

74 In recent years, machine learning (ML) has been used by life scientists as a system for
75 knowledge discovery (Ma, Zhang & Wang, 2014), achieving promising results. ML can
76 be defined as the process of designing a model that will be calibrated from the training
77 information and a loss function through an optimization algorithm (Mjolsness &
78 DeCoste, 2001). Based on these extracted patterns, algorithms can then predict results
79 from unknown data. Main ML training methods can be classified into supervised
80 learning and unsupervised learning (Ceballos et al., 2019). The goal of supervised
81 learning is to predict a discrete (classification) or continuous (regression) value for each
82 data point by using a provided set of labeled training examples. In unsupervised
83 learning, which is based on clustering algorithms, the goal is to self-learn inherent
84 patterns within the data (Zou et al., 2018). The main objective of ML tasks is to optimize
85 a cost function in terms of a set of parameters for a proposed model. In the optimization
86 process, the proposed model is calibrated. With this aim, the data are randomly split
87 into a minimum of three subsets (named hold-out method): training, validation, and test
88 sets leaving the first set for learning patterns and hyper-parameters, the second set for
89 choosing the best models, and the last set for obtaining more realistic accuracy. On the
90 other hand, k-fold cross-validation randomly splits data into k-folds and then applies
91 hold-out to each subset. (Zou et al., 2018). This process is crucial to avoid overfitting
92 (also called overtraining) or underfitting (undertraining), which both lead to poor
93 predictive performances. Therefore, the algorithm must reach an appropriate balance
94 between model flexibility and the amount of training data. An overly simple model will
95 underfit and fail to let the data 'speak', while an overly flexible model will overfit to
96 spurious patterns in the training data and fail to generalize (Zou et al., 2018).

97 The design and implementation of a ML system is a complex process that can be done
98 in three steps: 1) raw data preprocessing (i.e. features selection and extraction, data
99 imputation, etc.), 2) learning or training of the model by using an appropriate ML
100 algorithm or architecture (to calibrate the model), and 3) model evaluation through
101 metrics (Ma, Zhang & Wang, 2014). In some cases, the preprocessing step is very
102 complex and relies on complicated algorithms to automate this task or on experts in the
103 field. The use of deep learning (DL) in ML addresses the issue of selecting the correct
104 data representation or the best features (Eraslan et al., 2019), avoiding the need for an
105 expert in the area. DL has evolved as a sub-discipline of ML through the development of
106 deep artificial neural networks (DNN) (i.e. neural networks with many hidden layers),
107 such as auto-encoders, fully connected neural networks (FNN), convolutional neural
108 networks (CNNs), recurrent neural networks (RNNs), among others (Eraslan et al.,
109 2019). DL has shown successful results in life sciences (Yu, Yu & Pan, 2017),
110 especially in genomics. In this area, it has been used for identifying di□erent types of
111 genomic elements, such as exons, introns, promoters, enhancers, positioned
112 nucleosomes, splice sites, untranslated regions (UTR), etc. (Yue & Wang, 2018).

113 Here, we performed a systematic review of applications of ML algorithms and
114 architectures in TE detection and classification problems. We also discuss other uses of

115 ML and DL in similar tasks that can be extrapolated to TE issues. To our knowledge,
116 this is the first review focused mainly on the use of MLin TEs.

Survey Methodology

118 Simon Orozco-Arias conducted an exhaustive literature review by applying the
119 Systematic Literature Review (SLR) process proposed by (Kitchenham & Charters,
120 2007) and preferred reporting items for systematic reviews and meta-analyses
121 guidelines (Moher et al., 2009) (Figure 1). We followed the six stages of the review
122 protocol used in (Wen et al., 2012) but added a previous stage, which aims to detect the
123 need for a review (Figure 2). First, we searched for other reviews to formulate questions
124 related to the aim of this review, and then we selected the search strategy based on key
125 terms and available databases. In the next step, we defined the exclusion criteria for
126 filtering relevant articles that could contribute to answering the questions from the first
127 stage. For this, we applied several filters in the Quality Assessment Checklist (fourth
128 stage) to choose articles to be included in the following steps. Finally, we performed
129 data extraction and data synthesis to process the information retrieved.

Identification of the need for a review

131 The strategy used was based on the guidelines proposed by (Wen et al., 2012) and
132 used in (Franco-Bedoya et al., 2017) and (Tabares-Soto, Ramos-Pollan & Isaza, 2019).
133 To define the need for a systematic review, we searched through published and
134 available reviews (secondary studies) on the topic of interest. We used equation 1 to
135 search in the literature databases referenced in Table 1.

136 *("transposable element" OR retrotransposon OR transposon) AND ("machine*
137 *learning" OR "deep learning") AND (review OR "systematic review" OR overview*
138 *OR "state of the art" OR "systematic mapping")*   (1)

139 The keywords were selected based on the following: i) type of information that we
140 aimed to retrieve (TEs as well as retrotransposons and transposons classes), ii)
141 techniques addressed in this review (machine learning and deep learning), and iii)
142 keywords related to secondary studies.

143 All of the databases showed results using the search equation (106 results after
144 filtering), yet only one secondary study (Dashti & Masoudi-Nejad, 2010) specifically
145 applied ML in TEs. However, this review published in 2010 focused only on support
146 vector machine (SVM). Given the lack of secondary studies about this topic, we
147 concluded that a systematic review of ML applications in TEs was needed.

Research Questions

149 The main aim of this review is to summarize and clarify trends, metrics, benefits, and
150 possible ML techniques and architectures that have not yet been addressed in the
151 detection and classification (for a graphical representation of the TEs classification, see
152 figure 3) of TEs. With this in mind, we formulated the following questions:

1. Q1. Are ML approaches for TE analyses advantageous compared to bioinformatics approaches? It is important to identify if the application of novel tools like machine learning can contribute to improving current bioinformatics software. This is relevant since it is well-know that current methodologies are still far from yielding confident results in the detection and classification of TEs given their high variability and complexity (Bousios et al., 2012). On the other hand, many papers propose that TEs are involved in key characteristics of genomes, such as chromosome structure, environmental adaptation, and interspecific variability, among others. Therefore, the objective of Q1 is to determine how TEs detection and classification can be improved using ML to understand the dynamics and impacts of these elements better.
2. Q2. Which ML techniques are currently used to detect and classify TEs or other genomic data?
   We were interested in knowing which algorithms or architectures have been tested on TEs or other genomic data, such as long non-coding regions or retrovirus.
3. Q3. What are the best parameters and most used metrics in algorithms and architectures to detect and classify TEs?
   To avoid overfitting, it is important to use a splitting method to reduce dependence on the training data and to determine which type of data is better to use. Thus, we were interested in knowing which current papers addressed this step. Additionally, to compare algorithms and architectures, it is important to define metrics that accurately measure performance. It is also essential to assess if these techniques improve results compared to traditional bioinformatics software.
4. Q4. What are the most used DNA coding schemes in ML tasks?
   Because TEs comprise categorical data (nucleotides), there are many ways to transform this information into numerical data required by ML algorithms. Therefore, we were interested in understanding how this transformation can contribute to improving results and which coding schemes are widely used in this kind of problem.

Search design and study selection

Once we identified the need for a review and formulated the research questions, we developed the search strategy to find research articles, chapter books, conference proceedings, and other review papers in the databases shown in Table 1. Similar to search equation 1, we used general keywords related to a) type of genomic data and b) techniques used. We did not use specific keywords (such as specific algorithms or architectures) and any time limitations, because few relevant results were found on this topic. Major keywords were separated by the "AND" operator, and related keywords were linked using the Boolean operator "OR", as shown in equation 2:

193 **("transposable element" OR retrotransposon OR transposon) AND ("machine**
194 **learning" OR "deep learning")** (2)

195 The literature search using equation 2 retrieved 403 candidate articles of which were
196 eliminated those that were: a) repeated (the same study was found in different
197 databases); b) of different types (books, posters, short articles, letters, and abstracts); c)
198 written in other languages (languages other than English). Then, we performed a fast
199 read process (i.e. title, abstract, and conclusion) to detect papers that could contribute
200 to answering the research questions. For this, we established the following inclusion
201 and exclusion criteria.

202 Inclusion criteria

203 • Application of ML or DL in the detection of TEs (any class)
204 • Application of ML or DL in the classification of TEs (any class)
205 • Description of DNA coding schemes
206 • Use of ML or DL on similar genomic data
207 • Comparison of bioinformatics algorithms to ML or DL techniques
208 • Application of metrics to evaluate ML or DL algorithms for TEs or similar data

209 Exclusion criteria

210 • Do not use any ML or DL techniques
211 • Studies focused only on *in vivo* processes
212 • Studies that do not integrate any of the topics addressed in this review

213 After this selection process (Figure 1), we identified 35 relevant papers that were used
214 to extract and summarize the information.

215 **Data extraction and synthesis**

216 In this stage, we first wholly read the selected publications (Table 2) to extract
217 information to answer the research questions. Then, we registered the article into a data
218 extraction card proposed by (Wen et al., 2012) with some adaptations to our study. The
219 card contained information on publication identifier, year, publication name, related
220 research questions, and the information itself.

221 In the final step, we synthesized all of the collected information and obtained evidence
222 to answer the research questions. Interestingly, more than 50% of the selected studies
223 were published between 2017 and 2019 (Figure 4), demonstrating a growing interest in
224 this topic in the last years.

225 We identified 35 relevant publications after the systematic literature review process (see
226 Table 2). These papers were published between 2009 and 2019. Among them, 77%
227 (27) were reported in journals, 17% (6) in conference proceedings, and 6% (2) as book
228 sections (Figure 5A). The selected papers were published in 21 journals, of which ten

229 focused on bioinformatics or computational biology, six on genomics or genetics, and 15
230 on other areas (Figure 5B).

Benefits of ML over bioinformatics (Q1)

232 There is much literature about applications of machine learning in bioinformatics (for
233 example, reviewed in (Larrañaga et al., 2006)), showing improvements in many aspects
234 such as genome annotation (Arango-López et al., 2017). In recent years, much
235 bioinformatics software has been developed to detect TEs (Girgis, 2015) and, although
236 they follow different strategies (such as homology-based, structure-based, de novo, and
237 using comparative genomics), these lack sensitivity and specificity due to the
238 polymorphic structures of TEs (Su, Gu & Peterson, 2019). Loureiro *et al.* (Loureiro et al.,
239 2013a) proved that ML could be used to improve the accuracy of TEs detection by
240 combining results obtained by several conventional software and training a classifier
241 using these results (Schietgat et al., 2018), (Loureiro et al., 2013b). Loureiro's work
242 provided novel evidence for the use of ML in TEs, yet it did not use ML to obtain the
243 predictions, making the results too dependent on traditional algorithms. Using the
244 Random Forest algorithm, Schietgat *et al.* were able to improve results obtained by
245 popular bioinformatics software (which followed a homology-based strategy) such as
246 Censor, RepeatMasker, and LTRDigest (Schietgat et al., 2018) in the detection of LTR
247 retrotransposons. The authors proposed a framework called TE-Learner^{LTR}, which
248 outperformed LTRDigest in recall and RepeatMasker and Censor in terms of precision.

249 ML techniques also obtain better results than traditional methods regarding TEs
250 classification. Using ML, it is possible to classify non-autonomous TEs (specifically
251 derived from LTR retrotransposons) using features other than coding regions (which are
252 commonly used in classification processes), including element length, LTR length, and
253 ORF lengths (Arango-López et al., 2017). ML algorithms can distinguish between
254 retroviral LTRs and SINEs (Short Interspersed Nuclear Elements) by combining a set of
255 methods (Ashlock & Datta, 2012), which is a complicated procedure in bioinformatics.
256 Also, using hierarchical classification, ML-based methods obtain better results than well-
257 known homologous-based methods (specifically, BLASTn algorithm) (Nakano et al.,
258 2017).

259 The advantages of using ML in bioinformatics include the discovery of entirely new
260 information such as arrays of mobile genetic elements, new transposition unit
261 boundaries (Tsafnat et al., 2011), and predicting new long noncoding RNA that are
262 related to cancer (Zhang et al., 2018). Other applications include extracting
263 discriminatory features for automatically determining functional properties of biological
264 sequences (Kamath, De Jong & Shehu, 2014), identifying DNA motifs, which is a
265 difficult task in non-ML applications (Dashti & Masoudi-Nejad, 2010), and automating
266 specific processes like the identification of long non-coding RNAs (Ventola et al., 2017)
267 and the classification of LTR retrotransposons (Arango-López et al., 2017).

268   On the other hand, DL has been applied in biological areas such as genomics (for a
269   review see (Yue & Wang, 2018)) proving to be promising (Yu, Yu & Pan, 2017) due to
270   the flexibility showed by deep neural networks. In (Eraslan et al., 2019), several
271   applications in genomics are discussed such as variant calling, base calling for novel
272   sequencing technologies, denoising ChIP–seq data (chromatin immunoprecipitation
273   followed by sequencing), and enhancing Hi-C data resolution (Chromosome
274   conformation capture followed by pair-end sequencing). Also, some frameworks allow
275   users to use GPUs as a complement for CPUs, achieving a faster execution of DL
276   algorithms (Eraslan et al., 2019). Deep neural networks have also been used to improve
277   the prediction of global enhancers, which was proven to be challenging using other
278   computational tools (Kim et al., 2016).

279   ML and DL fields are supported by multiple companies and industry research groups,
280   which anticipated the great benefits that artificial intelligence can contribute to
281   genomics, human health (Eraslan et al., 2019), and major crops. Several papers using
282   ML or DL techniques reported that TEs are associated with many human diseases
283   (Zhang et al., 2013). For example, cancer-related long noncoding RNAs have higher
284   SINE and LINE numbers than cancer-unrelated long noncoding RNAs (Zhang et al.,
285   2018). Likewise, several types of epithelial cancers acquire somatic insertions of LINE-1
286   as they develop (mentioned in (Tang et al., 2017)). Moreover, the genes that confer
287   antibiotic resistance (called R genes) in bacteria are associated with TEs, and it is
288   possible to detect them through ML (Tsafnat et al., 2011). Finally, although LTR
289   retrotransposons are related to retroviruses such as HIV, ML algorithms can distinguish
290   them from SINEs (Ashlock & Datta, 2012). On the other hand, the human genome is
291   composed of a considerable number of interspersed repeats, such as LINE-1 as one of
292   the most abundant (Tang et al., 2017)), human endogenous retroviral sequences
293   comprising 8-10 percent of the genome, and SINEs contributing with ~11 percent.
294   Meanwhile, protein-coding regions comprise only about 1.5 percent (Ashlock & Datta,
295   2012).

296   Since TEs are under relatively low selection pressure and evolve more rapidly than
297   coding genes (Rawal & Ramaswamy, 2011), they undergo dynamic evolution.
298   Moreover, insertions of other TEs (nested insertion), illegitimate and unequal
299   recombination, cellular gene capture, and inter-chromosomal and tandem duplications
300   (Garbus et al., 2015) make TEs classification and annotation a very complicated task
301   (Bousios et al., 2012). Thus, conventional methods (such as bioinformatics) cannot
302   obtain reliable results in TE detection and classification tasks.

303   In supervised problems, the process of feature extraction or selection is a crucial step
304   for the performance of the entire architecture. In ML, the processes of selection of
305   variables or characteristics must be carried out by a thematic expert. Deep network
306   architectures, on the other hand, allow characteristics to be extracted in a nonlinear and
307   automatic way. The hidden layers of deep neural networks transform these
308   characteristics into intricate patterns relevant to the classification problem (Eraslan et

309 al., 2019). In the specific case of TEs, because they are DNA sequences, the extraction
310 of characteristics is usually a too complex process due to a large amount of information,
311 their unstructured form, and their sequentially. In this case, the deep neural networks
312 provide new features that cannot be extracted manually. For example, convolutional
313 neural networks can discover local patterns in sequential data such as pixels in an
314 image or DNA (Zou et al., 2018). These patterns known as motifs have functions of
315 great importance in the genome as promoters of genes and to be found in the LTR
316 sequences of retrotransposons, in addition, if they are found in different places and
317 under certain frequencies could be very useful to identify or classify TEs. Although
318 motifs are essential for DNA classification problems, it is not enough to find the exact
319 patterns, because DNA can undergo modifications or mutations and because specific
320 motifs can function the same as others even if they do not have the same nucleotides.
321 Another benefit of ML over bioinformatics is the use of labeled data to generate
322 computational models. Currently, a few hundreds of plant genomes are available to train
323 algorithms, but this number will increase significantly in the near future due to massive
324 genome sequencing projects such as the 10K plant project (https://db.cngb.org/10kp/)
325 or the Earth BioGenome Project (https://www.earthbiogenome.org). This large amount
326 of data will help to produce more accurate and reliable software thought ML and DL.
327 Additional to available training data, some plant genomes are very interesting to identify
328 and classify TEs following ML or DL approaches, due to their huge genome and their
329 composition. As examples, sugarcane, maize, and barley have large genomes (3Gb,
330 2Gb, and 5.1 Gb respectively) that are composed mainly by repetitive sequences (up to
331 80%, (Rahman et al., 2013)). On the other hand, the process of supervised training of
332 ML algorithms provides another advantage when training a model that increases true
333 positives and decreases false positives results. To improve this rate of performance, the
334 hyperparameters of the models can be tuned using techniques such as search grid,
335 which does an exhaustive search for the hyperparameters that produce the best
336 accuracy and precision.

337 **ML architectures and algorithms currently used for TEs or similar data (Q2)**

338 ML has been applied in bioinformatics due to a large amount of data that has been
339 generated. Ma and coworkers (Ma, Zhang & Wang, 2014), review the application of ML
340 in topics such as genome assembly, genomic variation detection, genome-wide
341 association studies, and the *in silico* annotation of coding genomic loci. Particular focus
342 has been given to loci that code for proteins, transposable elements, noncoding RNAs,
343 miRNAs and targets, transcription factor binding sites, cis-regulatory elements,
344 enhancer and silencer elements, and mRNA alternative splicing sites. Additionally,
345 many frameworks have been developed to facilitate the implementation of ML
346 algorithms in bioinformatics projects. Tools such as Scikit-learn, Weka, and several
347 packages developed in R (for a complete list, see (Ma, Zhang & Wang, 2014)) allow
348 using ML-based techniques in biological areas.

349 Most of the publications found in this review (Table 3) used supervised learning as the
350 training mechanism (84%, 21 publications). We found only four publications (16%) that
351 used unsupervised learning (Figure 6A), which mainly addressed tasks of features
352 selection and extraction and clustering of motif sequences.

353 Among the supervised learning algorithms used, random forest (RF) and support vector
354 machine (SVM) are the most commonly found in the publications reviewed (Figure 6B).
355 SVM is a widely used classifier, while RF can avoid overfitting and is insensitive to noise
356 (Ashlock & Datta, 2012), two features that are very useful in TE problems due to the
357 high variability and lack of a general structure of TEs. On the other hand, we found only
358 three publications that used HMM and, in the case of (Tsafnat et al., 2011) and (Zamani
359 et al., 2013), HMMs are applied in the preprocessing step for other ML techniques.
360 Hidden Markov Models (HMM) is another well-known technique in bioinformatics. RED,
361 which de novo detects repeats, is the only software found that uses HMM as the
362 primary tool (Girgis, 2015). Finally, 12 of 25 publications used more than one technique
363 to compare results and select the most optimal or to improve the accuracy obtained.

364 Eight publications focused on the detection or classification of TEs, and all of them used
365 supervised techniques. RF, decision trees (DT), and SVM are the most frequent
366 algorithms. The use of preprocessing methods to extract features from DNA or RNA
367 (Figure 7) was a common finding (for a review on the extraction and selection of
368 features from biological sequences see (Kamath, De Jong & Shehu, 2014)). For
369 example, (Loureiro et al., 2013b), (Zamith Santos et al., 2018), and (Abrusan et al.,
370 2009) used k-mer frequencies as features. On the contrary, (Schietgat et al., 2018),
371 (Arango-López et al., 2017), (Loureiro et al., 2013a) used numerical and categorical
372 features mainly based on structures. Other purposes of ML in TEs included detecting
373 the boundaries of mobile elements (Tsafnat et al., 2011), identifying insertion sites of
374 somatic LINEs insertions (Tang et al., 2017), and other mobile elements at the genome
375 level (Rawal & Ramaswamy, 2011), detecting aneuploidy in patients with cancer
376 through LINEs (Douville et al., 2018), detecting conserved motifs in TIR elements (Su,
377 Gu & Peterson, 2019), distinguishing endogenous retroviral LTRs from SINEs (Ashlock
378 & Datta, 2012), and comparing multiple studies on transposon insertion sequencing
379 (Hubbard et al., 2019). The last study used an unsupervised algorithm based on
380 principal component analysis (PCA) to reduce the feature dimensions and improve the
381 clustering analysis.

382 We also found publications that applied ML to other genomic data than TEs (these
383 publications can be found in Table 2). Long non-coding RNAs (lncRNAs) are gaining
384 attention because of critical biological functions suggested by recent studies (for a
385 review see (Mercer, Dinger & Mattick, 2009)). Some of the ML applications found
386 included the detection of cancer-related lncRNA (Zhang et al., 2018), the discrimination
387 of circular RNAs from other lncRNAs (Chen et al., 2018), and selection of the most
388 informative features of lncRNA (Ventola et al., 2017). Other applications in the RNA field
389 address the identification and clustering of RNA structure motifs (Smith et al., 2017),

390 prediction of microRNA precursors (Ding, Zhou & Guan, 2010), prediction of piRNA
391 (Brayet et al., 2014), and classification of small non-coding RNAs (Ryvkin et al., 2014).
392 Although TEs are DNA molecules, the techniques applied to lncRNA could be
393 extrapolated to TEs since they are composed of long non-coding regions containing
394 motifs. For example, LTR retrotransposons have two highly similar characteristic Long
395 Terminal Repeats (LTR) that usually contain Short Inverted Repeat (SIR) motifs TG-5'
396 and 3'-CA at both ends (Mascagni et al., 2015), (Yin et al., 2013), as well as one to
397 three AT-rich regions with one or two TATA-boxes and a polyadenylation signal
398 (AATAAA motif) (Benachenhou et al., 2013), (Gao et al., 2012). Consequently, the
399 approaches implemented in (Zamani et al., 2013) can be beneficial for predicting
400 patterns inside TEs, yielding better results in classification processes.

401 Regarding DL, we found five articles that addressed TE classification (Nakano et al.,
402 2018b), the detection of long intergenic non-coding RNA (lincRNA) using different
403 coding schemes and outperforming SVM results (Yu, Yu & Pan, 2017), the use of DL to
404 predict enhancers based on chromatin features (Kim et al., 2016), and the use of
405 convolutional neural networks to classify TEs (da Cruz et al., 2019). Lastly, the fifth
406 article reviewed the applications of DL in genomics (Eraslan et al., 2019).

407 The first ideas on deep artificial neural networks (DNN) were discussed in the 1990s,
408 although mature concepts on the subject matter appeared in the 2000s (Yu, Yu & Pan,
409 2017). Auto-encoders, which can perform non-linear dimensionality reduction by training
410 a multilayer neural network with a small central layer to reconstruct high-dimensional
411 input vectors (Zou et al., 2018), have been used by Yu and coworkers (Yu, Yu & Pan,
412 2017) with a setting of two layers. They demonstrated better results than SVM in the
413 prediction of lncRNA. Moreover, other publications have reported better results from DL
414 compared to conventional ML techniques. In (Eraslan et al., 2019), the authors
415 reviewed several DL architectures used in genomics (Table 4), showing improved
416 predictive performance over ML methods, including Logistic Regression, Decision Trees
417 or Random Forest. Since no publications were found addressing TE detection, the
418 application of auto-encoders can be a novel way to predict TEs with long non-coding
419 regions (such as LTR retrotransposons).

420 One of the most useful characteristics of DL architectures is that DNN can automatically
421 learn non-linear features since each layer uses multiple linear models, and the outputs
422 are transformed by non-linear activation functions, such as sigmoid functions or
423 rectified-linear unit (ReLU) (Eraslan et al., 2019). This process could facilitate
424 classification tasks that include, for instance, distinguishing superfamilies of LTR
425 retrotransposons (*Copia* and *Gypsy)*. Furthermore, (Nakano et al., 2018b) used these
426 advantages to improve the hierarchical classification of TEs (Figure 8) through fully
427 connected neural networks and using k-mer frequencies as features.

428 Advances in DL can be attributed to the use of frameworks, which facilitate the
429 implementation of crucial operations required to build and train neural networks, such as
430 Keras (Chollet & others, 2015), tensorFlow (Abadi et al., 2016), Theano (Bergstra et al.,

431 2011) or Pytorch (Paszke et al., 2017). These operations include matrix multiplication,
432 convolution, and automatic differentiation (Eraslan et al., 2019), allowing users to
433 specify their models more easily and quickly. Another advantage is that users do not
434 need to parallelize their codes since frameworks like tensorFlow can do it automatically.

435 Convolutional neural networks (CNN) have been widely used in genomics. Most of the
436 publications shown in Table 4 take advantage of the ability of CNNs to extract high-level
437 features directly from sequences (in most cases, using one-hot codification). These
438 features were then passed to other layers (i.e. fully connected layers) to calculate the
439 final results. Recently, CNNs have been applied to the classification of TEs and have
440 shown better results than conventional bioinformatics software such as PASTEC and
441 REPCLASS. Accordingly, benefits can be gained from the use of this kind of neural
442 networks (da Cruz et al., 2019) (Figure 9). Since TEs display different structural
443 features, specific motifs, and promoters, CNNs can find features that are not calculable
444 with conventional methods. This ability can provide useful information to researchers
445 interested in understanding the diversity and characteristics of TEs, as well as
446 improving the detection and classification of these elements. Other architectures such
447 as recurrent neural networks (RNNs) have been applied in distinct tasks in genomics,
448 such as the prediction of binding sequences. A crucial feature of this kind of DNNs is
449 the implementation of memory. An application of RNNs in TEs is the identification of
450 boundaries, which, in most orders, are composed of short duplications at both ends
451 (Target Site Duplications –TSD-) and, in some orders, (i.e. LTR retrotransposons and
452 TIR DNA transposons) of non-coding repeat regions (long terminal repeats for LTR
453 retrotransposons and terminal inverted repeat for TIR DNA transposons). On the other
454 hand, since nucleotides from TEs are used as input, it is likely to have more variables
455 than individuals, leading to overfitting in the training steps. Thus, auto-encoders can be
456 used to reduce the number of features in a non-linear way, helping to overcome this
457 issue.

458 To summarize, the ML techniques already used in TEs are mainly RF, DT, and SVM.
459 Although most publications use supervised learning, some papers can be found using
460 unsupervised learning, mainly for extracting and selecting characteristics. Only two
461 articles were found that applied DL (one publication used FNN and the other CNN) to
462 the classification of TE, but they aimed to predict TE orders. Therefore, more research
463 is needed on DL approaches.

Parameters and metrics applied in algorithms and architectures (Q3)

465 To ensure that ML architectures do not exclusively learn patterns of the training dataset,
466 there are several techniques used to split information into different datasets, such as the
467 hold-out and k-cross-validation methods. These methods should be used in problems
468 with information of any kind. Particularly with data of genomic origin, k-cross-validation
469 seems to be the most popular (Ma, Zhang & Wang, 2014). Using k=10, different studies
470 demonstrated high accuracies for long non-coding RNAs (Chen et al., 2018; Zhang et
471 al., 2018), for selecting features for classification of biological sequences (Kamath, De

472 Jong & Shehu, 2014), analyzing insertion sites of somatic LINEs in ovarian cancer
473 (Tang et al., 2017), and improving the classification of TEs (Loureiro et al., 2013b;
474 Nakano et al., 2018b). Meanwhile, k=5 was used by (Segal et al., 2018) to infer the
475 importance of specific genes for growth under laboratory conditions.

476 In ML tasks, it is essential to have curated datasets (Loureiro et al., 2013a). The quality
477 of TE and other genomic datasets is complex to evaluate, and their nature can influence
478 the final results. Databases such as PGSB contain genomic TE sequences from many
479 species, while repetDB or RepBase comprise consensus sequences of TEs. Ashlock
480 and Datta (2012) proposed that, although consensus sequences have been used to
481 train several ML algorithms (i.e. TEclass (Abrusan et al., 2009) and REPCLASS
482 (Feschotte et al., 2009)), this type of datasets caused poor results. The authors also
483 recommended taking this into account for ML projects in genomics.

484 A key aspect in the field of artificial intelligence is the calculation of metrics that
485 represent the performance of the algorithms and architectures. Classification or
486 detection tasks mostly rely on defining two classes, positive and negative. Accordingly,
487 the predicted results are marked as true positives (TP) if they were classified as positive
488 and are contained in the positive class, while as false negatives (FN) if they were
489 rejected but did not belong to the negative class. Also, candidates that appear in the
490 negative set that were classified as positives are counted as false positives (FP), and all
491 others are classified as true negatives (TN) (Tsafnat et al., 2011). Most metrics are
492 based on the frequencies of these markers (Table 5).

493 The most popular metrics in ML projects are accuracy, sensitivity, specificity, precision,
494 recall, F-score, and ROC curves (Ma, Zhang & Wang, 2014). However, these are not
495 appropriate in every case, especially when the positive and negative data sets are
496 unbalanced. For example, ROC curves are not used in TE classification, because only a
497 small portion of the genome contains certain TE superfamilies. In this case, it is more
498 interesting to recognize positive results than predict negative candidates correctly
499 through precision recall-curves (PRC) (Schietgat et al., 2018). Also, instead of using
500 accuracy, AUC and PRC are used for the feature construction and selection of
501 classification of biological sequences (Kamath, De Jong & Shehu, 2014) and the
502 identification of long non-coding RNA (Ventola et al., 2017).

503 In hierarchical classification problems, there is no consensus on which metrics should
504 be used (Zamith Santos et al., 2018), although a set of evaluation measures have been
505 proposed such as hierarchical Precision (hP), hierarchical Recall (hR), and hierarchical
506 F-measure (hF) (Figure 10) (Nakano et al., 2017). Since TE classification systems
507 follow a hierarchical topology, these metrics can contribute to improving the
508 measurement of algorithms and architectures to classify TEs.

509 In brief, because the detection and classification of TE can be covered from different
510 approaches (binary problems, multi classes or hierarchical classification), multiple
511 metrics can be applied. However, it is necessary to use those metrics that are not

512 affected by unbalanced data sets, which is a problem linked to these types of data.
513 Although in genomics, the k-cross-validation method is the most common, the k-value
514 depends on the size of the training dataset. The papers evaluated in this review used
515 values of k=5 and 10. On the other hand, it was found that the nature of the data
516 (genomic sequences or consensus) affects the performance of ML algorithms, so some
517 authors recommend the use of genomic sequences.

518 **Most used DNA coding schemes (Q4)**

519 One of the most critical tasks in ML algorithms is correct data representation. In contrast
520 to other datasets, DNA nucleotide sequences are human-readable characters, C, T, A,
521 and G, so it is necessary to encode them in a machine-readable form (Yu, Yu & Pan,
522 2017). Table 6 shows coding schemes that can be applied to representing nucleotides
523 by different approaches. Using deep neural networks, (Yu, Yu & Pan, 2017)
524 demonstrated that the complementary scheme had the best performance, while the
525 other schemes achieved similar predictive rates.

526 In some cases, input sequences need to be first transformed into k-mer counts (Zamith
527 Santos et al., 2018). For example, for distinguishing between Endogenous Retroviral
528 LTRs from SINEs, the dinucleotide (2-mer) "TT" appears more frequently in LTRs than
529 in SINEs and LTRs have more TAs, TGs, As, and Gs before their first C than SINEs.
530 These k-mer features add more information than the raw DNA sequences in the
531 classification process (Ashlock & Datta, 2012). Additionally, k-mer frequencies of k =
532 2,3,4 have been used (Nakano et al., 2017, 2018a) for TE classification through DL.
533 Other examples that apply k-mer occurrences to sequence transformation are the
534 prediction of DNA promoter regions, cis sites, HS sites, splice sites, among others
535 (reviewed in (Kamath, De Jong & Shehu, 2014)).

536 Recently, an innovative way to convert sequences into numerical representations was
537 proposed by (Jaiswal & Krishnamachari, 2019). The authors considered three
538 physicochemical properties, namely, average hydrogen bonding energy per base pair
539 (bp), stacking energy (per bp), and solvation energy (per bp), which are calculated by
540 taking the first di-nucleotide and then moving a sliding window, one base at a time.
541 Accordingly, a classification task was performed using this process. Although this
542 classification was carried out on the *Saccharomyces cerevisiae* genome, it can be
543 extrapolated to other species with distinct types of TEs.

544 To summarize, the use of k-mers frequencies seems to be more common and get better
545 results in ML algorithms. On the other hand, in DL architectures, they mainly use the
546 one-hot coding scheme because the extraction is carried out automatically.
547 Interestingly, the problem of TEs detection could be addressed using the physico-
548 chemical characteristics of the di-nucleotides.

549 # Conclusions and future work

550 ML is a powerful tool that can extract patterns, novel information, and relations from
551 labeled data (supervised learning) or unlabeled data (unsupervised learning). These
552 artificial intelligence approaches improve complex tasks and automate processes that
553 would otherwise be done manually. Although ML and DL fields have been applied in
554 areas such as genomics, human health, face recognition, and many others, the use of
555 ML and DL in TEs is still limited. This is especially true for deep neural networks such
556 as convolutional neural networks, which could provide opportunities to extract features
557 that are undetected by conventional bioinformatics methods. Although TE detection and
558 classification are very complex tasks because of the variability of these elements, there
559 are databases with thousands of TE sequences that have been recently released.
560 These databases can contribute training sets for obtaining better and generalized
561 models to improve the accuracy and reliability of the results. TEs are associated with
562 many aspects in humans (such as diseases) and plants (like intra and inter-species
563 diversity, adaptation to the environment, among others). Therefore, a broader
564 understanding of these elements can provide better knowledge of our genomes as well
565 as about important crops. Accordingly, this can lead to faster and reliable diagnostic and
566 clinical treatments in diseases like cancer and more resistant and productive crops.
567 Unquestionably, ML and DL can support novel methods to detect, classify, and analyze
568 repeated sequences. To date, there are few publications on the application of DL in
569 TEs, so the door is open to proposing innovative methodologies and architectures.

570 Taking into account this systematic review of literature, we propose the following ideas
571 as future work:

572 - To use Autoencoders to increase the size of the training datasets (data augmentation)
573 on the TEs already classified and validated by the bioinformatics community, in order to
574 obtain a better generalization of the ML and DL algorithms.

575 - To use sets of simultaneous classifiers (SVM, RF, DT, LR, among others) in order to
576 generate separation frontiers of classes more adapted to the nature of this type of
577 problem and thus be able to increase the percentages of precision in the detection and
578 classification of TEs.

579 - To train new artificial neural network architectures using transfer learning techniques
580 from the results obtained by the neural networks proposed in the literature.

581 - To use techniques of selection of characteristics (Random Forest) or reduction of
582 dimensionality (PCA) in order to diminish the databases' complexity and to increase the
583 percentages of precision in the detection and classification of TEs.

584 **Acknowledgements**

585 The authors thanks the Universidad Autónoma de Manizales, Manizales, Colombia, for
586 support and covering publication fees under project 589-089 and the LMI BIO-INCA for
587 supporting Romain Guyot.

588 **References**

589 Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G,
590   Isard M, others. 2016. Tensorflow: A system for large-scale machine learning. In:
591   *12th USENIX Symposium on Operating Systems Design and Implementation*
592   *(OSDI 16)*. 265–283.

593 Abrusan G, Grundmann N, DeMester L, Makalowski W. 2009. TEclass--a tool for
594   automated classification of unknown eukaryotic transposable elements.
595   *Bioinformatics* 25:1329–1330. DOI: 10.1093/bioinformatics/btp084.

596 Arango-López J, Orozco-Arias S, Salazar JA, Guyot R. 2017. Application of Data Mining
597   Algorithms to Classify Biological Data: The Coffea canephora Genome Case. In:
598   156–170. DOI: 10.1007/978-3-319-66562-7_12.

599 Ashlock W, Datta S. 2012. Distinguishing endogenous retroviral LTRs from SINE
600   elements using features extracted from evolved side effect machines. *IEEE/ACM*
601   *Transactions on Computational Biology and Bioinformatics* 9:1676–1689. DOI:
602   10.1109/TCBB.2012.116.

603 Benachenhou F, Sperber GO, Bongcam-Rudloff E, Andersson G, Boeke JD, Blomberg
604   J. 2013. Conserved structure and inferred evolutionary history of long terminal
605   repeats (LTRs). *MOBILE DNA* 4. DOI: 10.1186/1759-8753-4-5.

606 Bergstra J, Bastien F, Breuleux O, Lamblin P, Pascanu R, Delalleau O, Desjardins G,
607   Warde-Farley D, Goodfellow I, Bergeron A, others. 2011. Theano: Deep learning
608   on gpus with python. In: *NIPS 2011, BigLearning Workshop, Granada, Spain*. 1–48.

609 Bousios A, Minga E, Kalitsou N, Pantermali M, Tsaballa A, Darzentas N. 2012.
610   MASiVEdb: the Sirevirus Plant Retrotransposon Database. *BMC Genomics* 13:158.
611   DOI: 10.1186/1471-2164-13-158.

612 Brayet J, Zehraoui F, Jeanson-Leh L, Israeli D, Tahi F. 2014. Towards a piRNA
613   prediction using multiple kernel fusion and support vector machine. *Bioinformatics*
614   *(Oxford, England)* 30:i364-70. DOI: 10.1093/bioinformatics/btu441.

615 de Castro Nunes R, Orozco-Arias S, Crouzillat D, Mueller LA, Strickler SR, Descombes
616   P, Fournier C, Moine D, de Kochko A, Yuyama PM, Vanzela ALL, Guyot R. 2018.
617   Structure and Distribution of Centromeric Retrotransposons at Diploid and
618   Allotetraploid Coffea Centromeric and Pericentromeric Regions. *Frontiers in Plant*
619   *Science*. DOI: 10.3389/fpls.2018.00175.

620 Ceballos D, López-Álvarez D, Isaza G, Tabares-Soto R, Orozco-Arias S, Ferrin C.
621   2019. A Machine Learning-based Pipeline for the Classification of CTX-M in
622   Metagenomics Samples. *Processes* 7:235. DOI: 10.3390/pr7040235.

623 Chen L, Zhang Y-H, Huang G, Pan X, Wang S, Huang T, Cai Y-D. 2018. Discriminating
624   cirRNAs from other lncRNAs using a hierarchical extreme learning machine (H-
625   ELM) algorithm with feature selection. *Molecular Genetics and Genomics* 293:137–
626   149. DOI: 10.1007/s00438-017-1372-7.

627 Chollet F, others. 2015. Keras.

628 Choulet F, Alberti A, Theil S, Glover N, Barbe V, Daron J, Pingault L, Sourdille P,

Couloux A, Paux E, others. 2014. Structural and functional partitioning of bread wheat chromosome 3B. *Science* 345:1249721.

da Cruz MHP, Saito PTM, Paschoal AR, Bugatti PH. 2019. Classification of Transposable Elements by Convolutional Neural Networks. In: *Lecture Notes in Computer Science*. Springer International Publishing, 157–168. DOI: 10.1201/9781420049138.

Dashti TH, Masoudi-Nejad A. 2010. Mining Biological Repetitive Sequences Using Support Vector Machines and Fuzzy SVM. *IRANIAN JOURNAL OF CHEMISTRY & CHEMICAL ENGINEERING-INTERNATIONAL ENGLISH EDITION* 29:1–17.

Ding J, Zhou S, Guan J. 2010. MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC Bioinformatics* 11:S11. DOI: 10.1186/1471-2105-11-S11-S11.

Douville C, Springer S, Kinde I, Cohen JD, Hruban RH, Lennon AM, Papadopoulos N, Kinzler KW, Vogelstein B, Karchin R. 2018. Detection of aneuploidy in patients with cancer through amplification of long interspersed nucleotide elements (LINEs). *Proceedings of the National Academy of Sciences of the United States of America* 115:1871–1876. DOI: 10.1073/pnas.1717846115.

Eraslan G, Avsec Ž, Gagneur J, Theis FJ. 2019. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*. DOI: 10.1038/s41576-019-0122-6.

Feschotte C, Keswani U, Ranganathan N, Guibotsy ML, Levine D. 2009. Exploring repetitive DNA landscapes using REPCLASS, a tool that automates the classification of transposable elements in eukaryotic genomes. *Genome biology and evolution* 1:205–20. DOI: 10.1093/gbe/evp023.

Franco-Bedoya O, Ameller D, Costal D, Franch X. 2017. Open source software ecosystems: A Systematic mapping. *Information and Software Technology* 91:160–185. DOI: 10.1016/j.infsof.2017.07.007.

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:e48595. DOI: 10.1371/journal.pone.0048595.

Garbus I, Romero JR, Valarik M, Vanžurová H, Karafiátová M, Cáccamo M, Doležel J, Tranquilli G, Helguera M, Echenique V. 2015. Characterization of repetitive DNA landscape in wheat homeologous group 4 chromosomes. *BMC Genomics* 16:375. DOI: 10.1186/s12864-015-1579-0.

Girgis HZ. 2015. Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. *BMC Bioinformatics* 16:227. DOI: 10.1186/s12859-015-0654-5.

Grandbastien M-A. 2015. LTR retrotransposons, handy hitchhikers of plant regulation and stress response. *Biochimica et biophysica acta* 1849:403–16. DOI: 10.1016/j.bbagrm.2014.07.017.

669 Hubbard TP, D'Gama JD, Billings G, Davis BM, Waldor MK. 2019. Unsupervised
670   Learning Approach for Comparing Multiple Transposon Insertion Sequencing
671   Studies. *mSphere* 4. DOI: 10.1128/mSphere.00031-19.

672 Jaiswal AK, Krishnamachari A. 2019. Physicochemical property based computational
673   scheme for classifying DNA sequence elements of Saccharomyces cerevisiae.
674   *Computational Biology and Chemistry* 79:193–201. DOI:
675   https://doi.org/10.1016/j.compbiolchem.2018.12.014.

676 Kamath U, De Jong K, Shehu A. 2014. Effective automated feature construction and
677   selection for classification of biological sequences. *PloS one* 9:e99982. DOI:
678   10.1371/journal.pone.0099982.

679 Kim SG, Harwani M, Grama A, Chaterji S. 2016. EP-DNN: A Deep Neural Network-
680   Based Global Enhancer Prediction Algorithm. *SCIENTIFIC REPORTS* 6. DOI:
681   10.1038/srep38433.

682 Kitchenham B, Charters S. 2007. *Guidelines for performing Systematic Literature*
683   *Reviews in Software Engineering*.

684 Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armañanzas
685   R, Santafé G, Pérez A, Robles V. 2006. Machine learning in bioinformatics.
686   *Briefings in Bioinformatics* 7:86–112. DOI: 10.1093/bib/bbk007.

687 Loureiro T, Camacho R, Vieira J, Fonseca NA. 2013a. Boosting the Detection of
688   Transposable Elements Using Machine Learning. In: 85–91. DOI: 10.1007/978-3-
689   319-00578-2_12.

690 Loureiro T, Camacho R, Vieira J, Fonseca NA. 2013b. Improving the performance of
691   Transposable Elements detection tools. *Journal of integrative bioinformatics*
692   10:231. DOI: 10.2390/biecoll-jib-2013-231.

693 Ma C, Zhang HH, Wang X. 2014. Machine learning for Big Data analytics in plants.
694   *Trends in Plant Science* 19:798–808. DOI: 10.1016/j.tplants.2014.08.004.

695 Mascagni F, Barghini E, Giordani T, Rieseberg LH, Cavallini A, Natali L. 2015.
696   Repetitive DNA and Plant Domestication: Variation in Copy Number and Proximity
697   to Genes of LTR-Retrotransposons among Wild and Cultivated Sunflower
698   (Helianthus annuus) Genotypes. *Genome biology and evolution* 7:3368–82. DOI:
699   10.1093/gbe/evv230.

700 Mercer TR, Dinger ME, Mattick JS. 2009. Long non-coding RNAs: insights into
701   functions. *Nature reviews genetics* 10:155.

702 Mjolsness E, DeCoste D. 2001. Machine learning for science: state of the art and future
703   prospects. *science* 293:2051–2055.

704 Moher D, Liberati A, Tetzlaff J, Altman DG. 2009. Preferred reporting items for
705   systematic reviews and meta-analyses: the PRISMA statement. *Annals of internal*
706   *medicine* 151:264–269.

707 Mustafin RN, Khusnutdinova EK. 2018. The Role of Transposons in Epigenetic

708 Regulation of Ontogenesis. *Russian Journal of Developmental Biology* 49:61–78.
709 DOI: 10.1134/S1062360418020066.

710 Nakano FK, Martiello Mastelini S, Barbon S, Cerri R. 2018a. Stacking methods for
711 hierarchical classification. In: *Proceedings - 16th IEEE International Conference on*
712 *Machine Learning and Applications, ICMLA 2017*. 289–296. DOI:
713 10.1109/ICMLA.2017.0-145.

714 Nakano FK, Mastelini SM, Barbon S, Cerri R. 2018b. Improving Hierarchical
715 Classification of Transposable Elements using Deep Neural Networks. In:
716 *Proceedings of the International Joint Conference on Neural Networks*. Rio de
717 Janeiro, Brazil,.

718 Nakano FK, Pinto WJ, Pappa GL, Cerri R. 2017. Top-down strategies for hierarchical
719 classification of transposable elements with neural networks. In: *Proceedings of the*
720 *International Joint Conference on Neural Networks*. Anchorage, AK, United states,
721 2539–2546.

722 Neumann P, Novák P, Hoštáková N, MacAs J. 2019. Systematic survey of plant LTR-
723 retrotransposons elucidates phylogenetic relationships of their polyprotein domains
724 and provides a reference for element classification. *Mobile DNA* 10:1–17. DOI:
725 10.1186/s13100-018-0144-1.

726 Orozco-Arias S, Isaza G, Guyot R. 2019. Retrotransposons in Plant Genomes:
727 Structure, Identification, and Classification through Bioinformatics and Machine
728 Learning. *International Journal of Molecular Sciences* 20:3837. DOI:
729 10.3390/ijms20153837.

730 Orozco-arias S, Liu J, Id RT, Ceballos D, Silva D, Id D, Ming R, Guyot R. 2018.
731 Inpactor, Integrated and Parallel Analyzer and Classifier of LTR Retrotransposons
732 and Its Application for Pineapple LTR Retrotransposons Diversity and Dynamics.
733 *Biology*. DOI: 10.3390/biology7020032.

734 Ou S, Chen J, Jiang N. 2018. Assessing genome assembly quality using the LTR
735 Assembly Index (LAI). *Nucleic Acids Research*:1–11. DOI: 10.1093/nar/gky730.

736 Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A,
737 Antiga L, Lerer A. 2017. Automatic differentiation in pytorch.

738 Piégu B, Bire S, Arensburger P, Bigot Y. 2015. A survey of transposable element
739 classification systems - A call for a fundamental update to meet the challenge of
740 their diversity and complexity. *Molecular Phylogenetics and Evolution* 86:90–109.
741 DOI: 10.1016/j.ympev.2015.03.009.

742 Rahman AYA, Usharraj AO, Misra BB, Thottathil GP, Jayasekaran K, Feng Y, Hou S,
743 Ong SY, Ng FL, Lee LS, others. 2013. Draft genome sequence of the rubber tree
744 Hevea brasiliensis. *BMC genomics* 14:75.

745 Rawal K, Ramaswamy R. 2011. Genome-wide analysis of mobile genetic element
746 insertion sites. *Nucleic acids research* 39:6864–78. DOI: 10.1093/nar/gkr337.

747 Ryvkin P, Leung YY, Ungar LH, Gregory BD, Wang L-S. 2014. Using machine learning

748   and high-throughput RNA sequencing to classify the precursors of small non-
749   coding RNAs. *Methods* 67:28–35. DOI:
750   https://doi.org/10.1016/j.ymeth.2013.10.002.

751   Schietgat L, Vens C, Cerri R, Fischer CN, Costa E, Ramon J, Carareto CMA, Blockeel
752   H. 2018. A machine learning based framework to identify and classify long terminal
753   repeat retrotransposons. *PLoS computational biology* 14:e1006097. DOI:
754   10.1371/journal.pcbi.1006097.

755   Segal ES, Gritsenko V, Levitan A, Yadav B, Dror N, Steenwyk JL, Silberberg Y, Mielich
756   K, Rokas A, Gow NAR, Kunze R, Sharan R, Berman J. 2018. Gene Essentiality
757   Analyzed by In Vivo Transposon Mutagenesis and Machine Learning in a Stable
758   Haploid Isolate of Candida albicans. *mBio* 9. DOI: 10.1128/mBio.02048-18.

759   Smith MA, Seemann SE, Quek XC, Mattick JS. 2017. DotAligner: identification and
760   clustering of RNA structure motifs. *Genome Biology* 18:244. DOI: 10.1186/s13059-
761   017-1371-3.

762   Su W, Gu X, Peterson T. 2019. TIR-Learner, a New Ensemble Method for TIR
763   Transposable Element Annotation, Provides Evidence for Abundant New
764   Transposable Elements in the Maize Genome. *Molecular Plant* 12:447–460. DOI:
765   10.1016/j.molp.2019.02.008.

766   Tabares-Soto R, Ramos-Pollan R, Isaza G. 2019. Deep Learning Applied to
767   Steganalysis of Digital Images: A Systematic Review. *IEEE Access* 4:1–1. DOI:
768   10.1109/ACCESS.2019.2918086.

769   Tang Z, Steranka JP, Ma S, Grivainis M, Rodić N, Huang CRL, Shih I-M, Wang T-L,
770   Boeke JD, Fenyö D, Burns KH. 2017. Human transposon insertion profiling:
771   Analysis, visualization and identification of somatic LINE-1 insertions in ovarian
772   cancer. *Proceedings of the National Academy of Sciences* 114:E733–E740. DOI:
773   10.1073/pnas.1619797114.

774   Tsafnat G, Setzermann P, Partridge SR, Grimm D. 2011. Computational inference of
775   difficult word boundaries in DNA languages. In: *ACM International Conference
776   Proceeding Series*. Barcelona, Spain, IEEE; Kyranova Ltd, Center for
777   TeleInFrastruktur (.

778   Ventola GMM, Noviello TMR, D'Aniello S, Spagnuolo A, Ceccarelli M, Cerulo L. 2017.
779   Identification of long non-coding transcripts with feature selection: a comparative
780   study. *BMC Bioinformatics* 18:187. DOI: 10.1186/s12859-017-1594-z.

781   Wen J, Li S, Lin Z, Hu Y, Huang C. 2012. Systematic literature review of machine
782   learning based software development effort estimation models. *Information and
783   Software Technology* 54:41–59. DOI: 10.1016/j.infsof.2011.09.002.

784   Wheeler TJ, Clements J, Eddy SR, Hubley R, Jones TA, Jurka J, Smit AFA, Finn RD.
785   2012. Dfam: a database of repetitive DNA based on profile hidden Markov models.
786   *Nucleic acids research* 41:D70--D82.

787   Wicker T, Sabot F, Hua-Van A, Bennetzen JL, Capy P, Chalhoub B, Flavell A, Leroy P,

788   Morgante M, Panaud O, Paux E, SanMiguel P, Schulman AH. 2007. A unified
789   classification system for eukaryotic transposable elements. *Nature Reviews*
790   *Genetics* 8:973–982. DOI: 10.1038/nrg2165.

791   Yin H, Liu J, Xu Y, Liu X, Zhang S, Ma J, Du J. 2013. TARE1, a mutated Copia-like LTR
792   retrotransposon followed by recent massive amplification in tomato. *PloS one*
793   8:e68587. DOI: 10.1371/journal.pone.0068587.

794   Yu N, Yu Z, Pan Y. 2017. A deep learning method for lincRNA detection using auto-
795   encoder algorithm. *BMC Bioinformatics* 18:511. DOI: 10.1186/s12859-017-1922-3.

796   Yue T, Wang H. 2018. Deep Learning for Genomics: A Concise Overview. :1–40.

797   Zamani N, Russell P, Lantz H, Hoeppner MP, Meadows JR, Vijay N, Mauceli E, di
798   Palma F, Lindblad-Toh K, Jern P, Grabherr MG. 2013. Unsupervised genome-wide
799   recognition of local relationship patterns. *BMC Genomics* 14:347. DOI:
800   10.1186/1471-2164-14-347.

801   Zamith Santos B, Trindade Pereira G, Kenji Nakano F, Cerri R. 2018. Strategies for
802   selection of positive and negative instances in the hierarchical classification of
803   transposable elements. In: *Proceedings - 2018 Brazilian Conference on Intelligent*
804   *Systems, BRACIS 2018*. 420–425. DOI: 10.1109/BRACIS.2018.00079.

805   Zhang Y, Babaian A, Gagnier L, Mager DL. 2013. Visualized Computational Predictions
806   of Transcriptional Effects by Intronic Endogenous Retroviruses. *PLoS ONE*
807   8:e71971. DOI: 10.1371/journal.pone.0071971.

808   Zhang X, Wang J, Li J, Chen W, Liu C. 2018. CRlncRC: a machine learning-based
809   method for cancer-related long noncoding RNA identification using integrated
810   features. *BMC Medical Genomics* 11:120. DOI: 10.1186/s12920-018-0436-9.

811   Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. 2018. A primer on deep
812   learning in genomics. *Nature Genetics*. DOI: 10.1038/s41588-018-0295-5.

813

Figure 1

PRISMA flow diagram

PRISMA flow chart for search and article screening process. From: Moher et al. (2009).
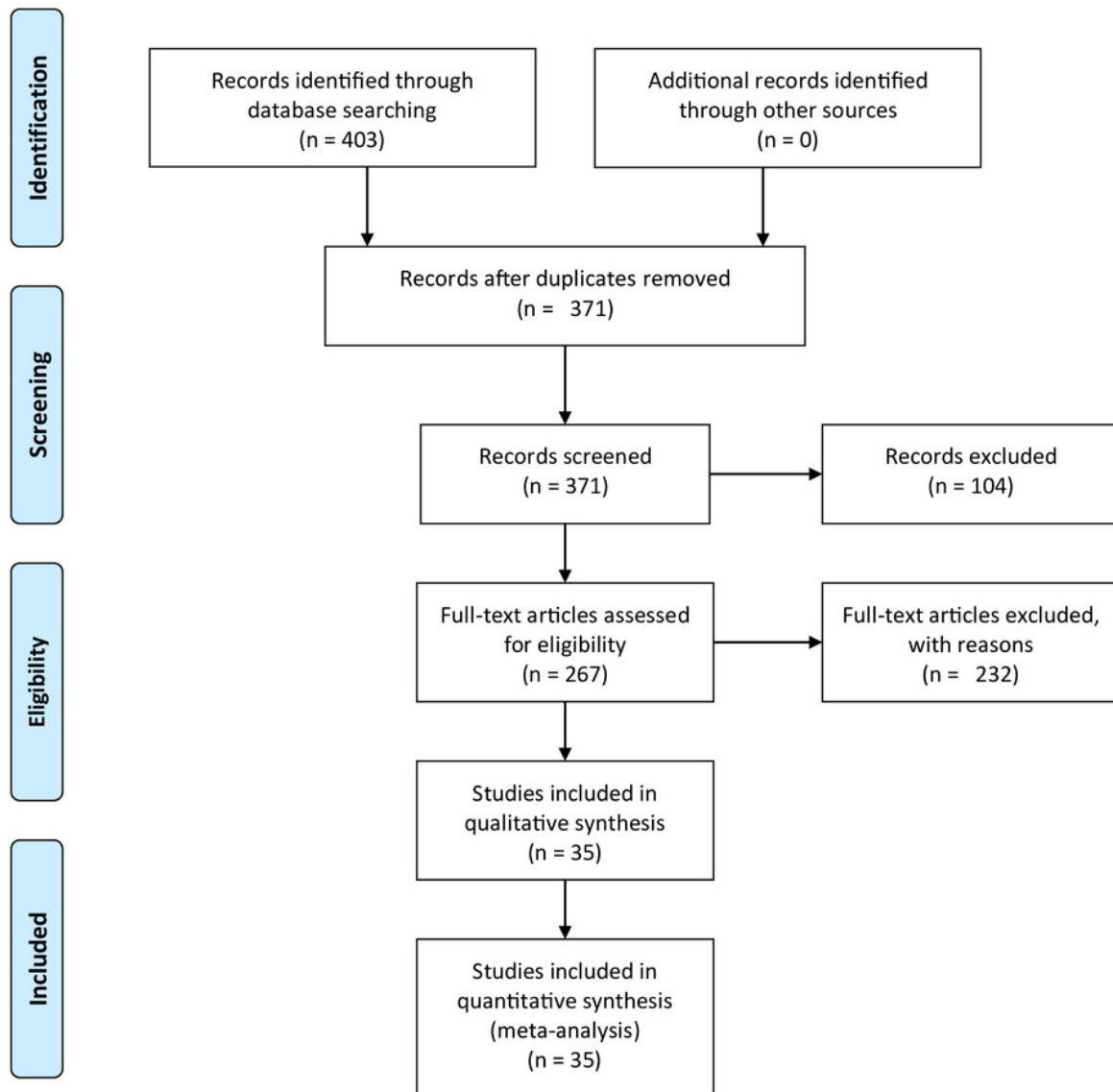
Figure 2

Stages of the systematic literature review process. Based on (Wen et al., 2012).
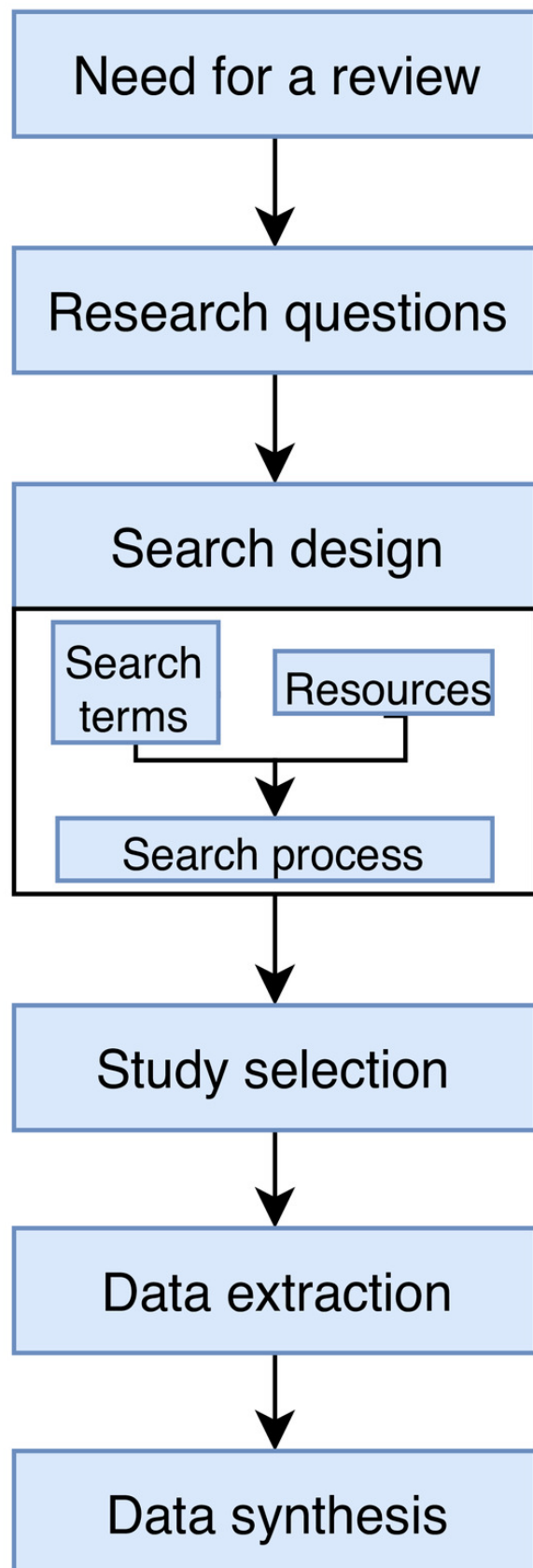
Figure 3

Classification of TEs following Rexdb and GyDB nomenclatures. Adapted from: Schietgat et al. (2018).

Levels:	Classes	Subclasses	Orders	Autonomy	Superfamilies	Lineages
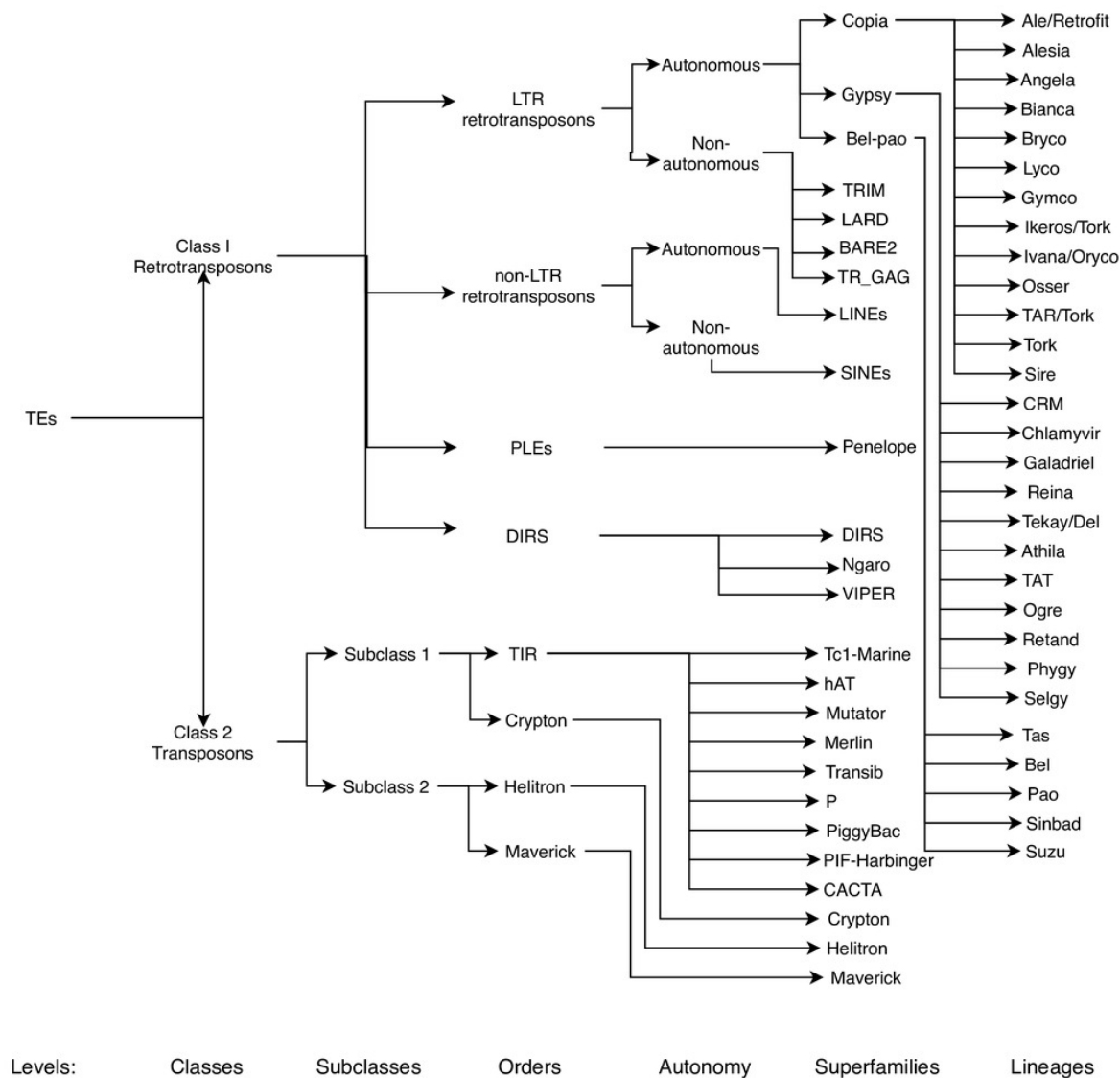
Figure 4

Number of relevant publications found per year.
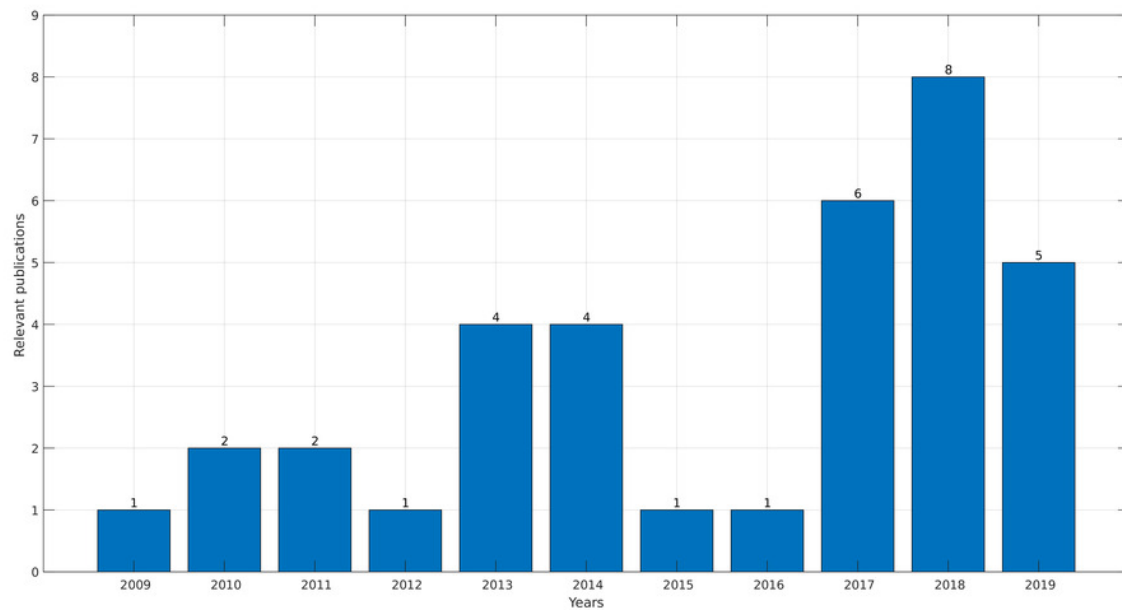
Figure 5

Source of selected publications. A) percentage of publications in each source. B) distribution of publications in journals.
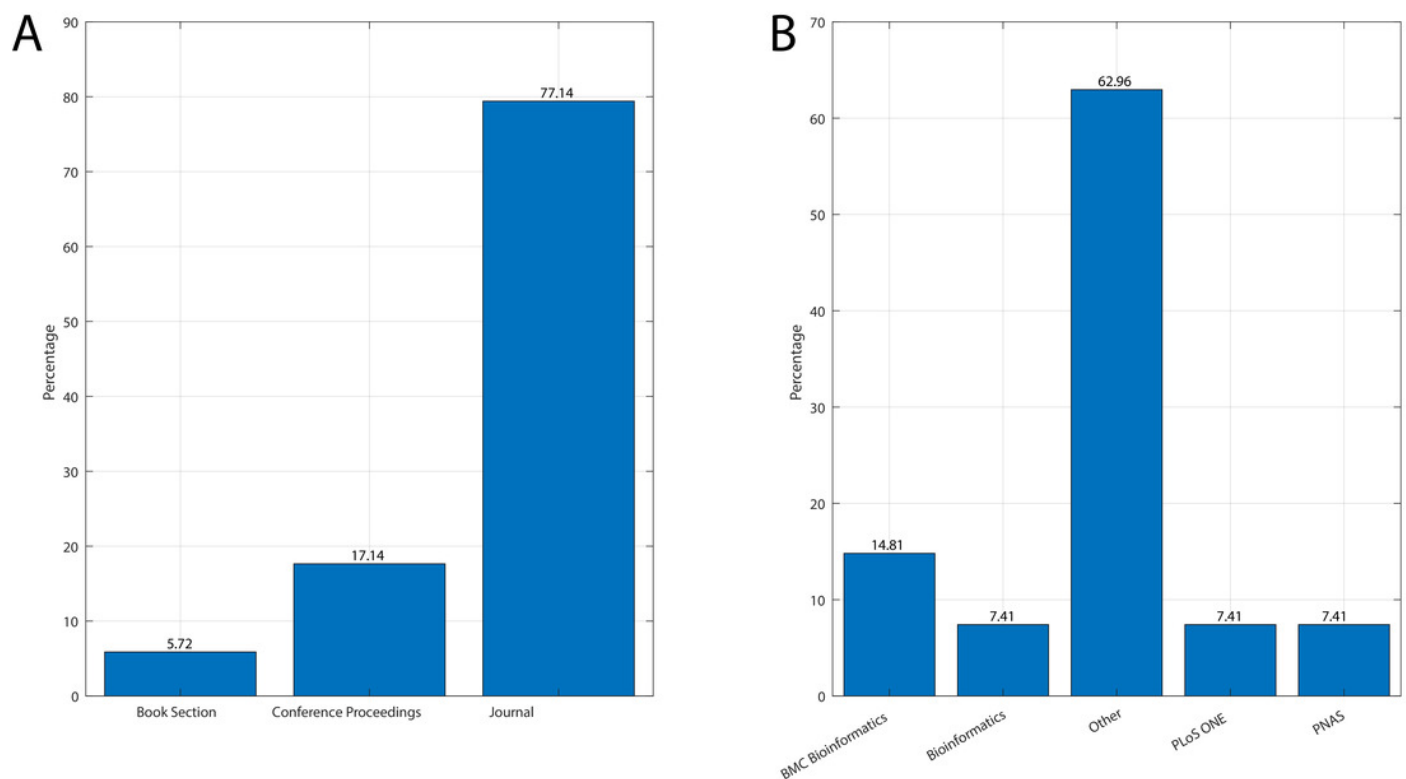
Figure 6

Source of selected publications. A) Proportion of publications using supervised and unsupervised learning. B) Supervised learning algorithms found in publications.

Abbreviations: Random Forest (RF), Decision Trees (DT), Bayesian Networks (BN), Neural networks (NN), Hidden Markov Model (HMM), Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP).
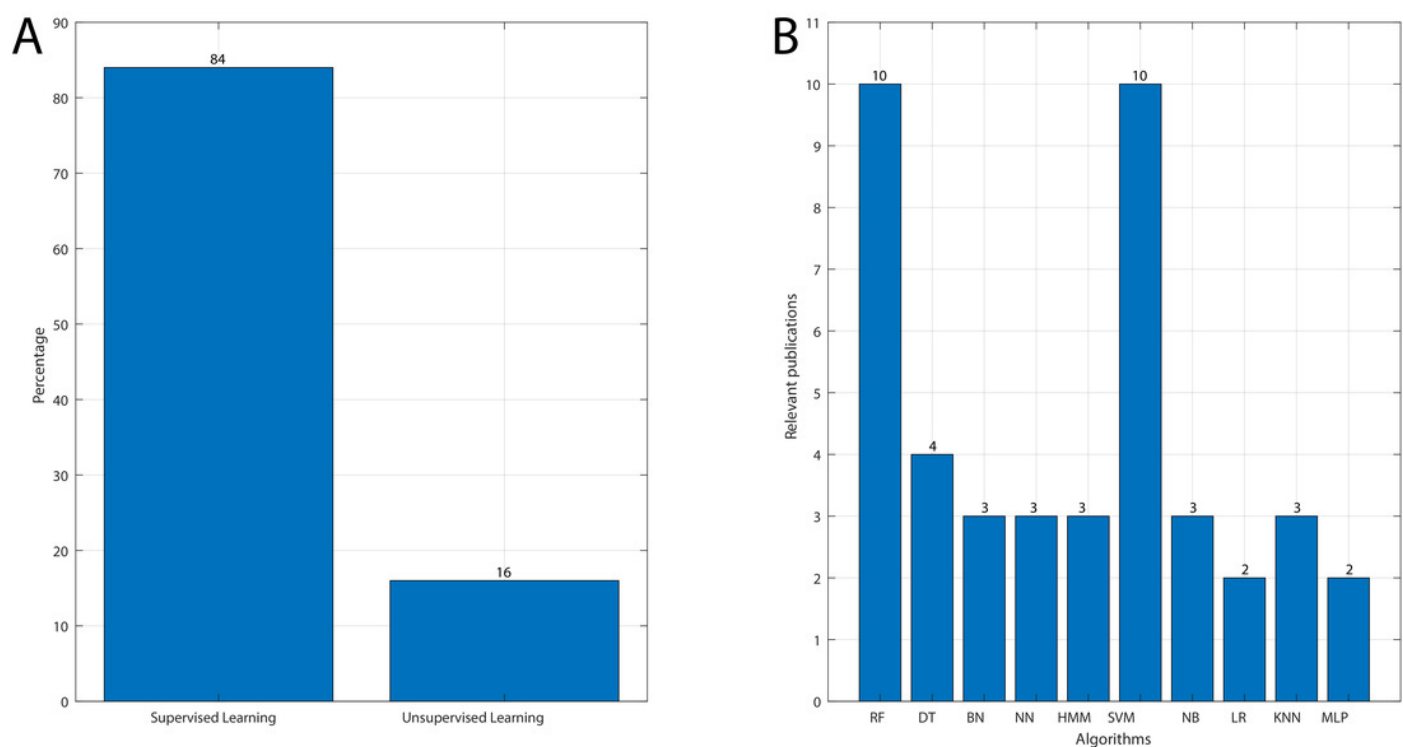
Figure 7

Overall workflow in supervised learning ML tasks applied to TEs.
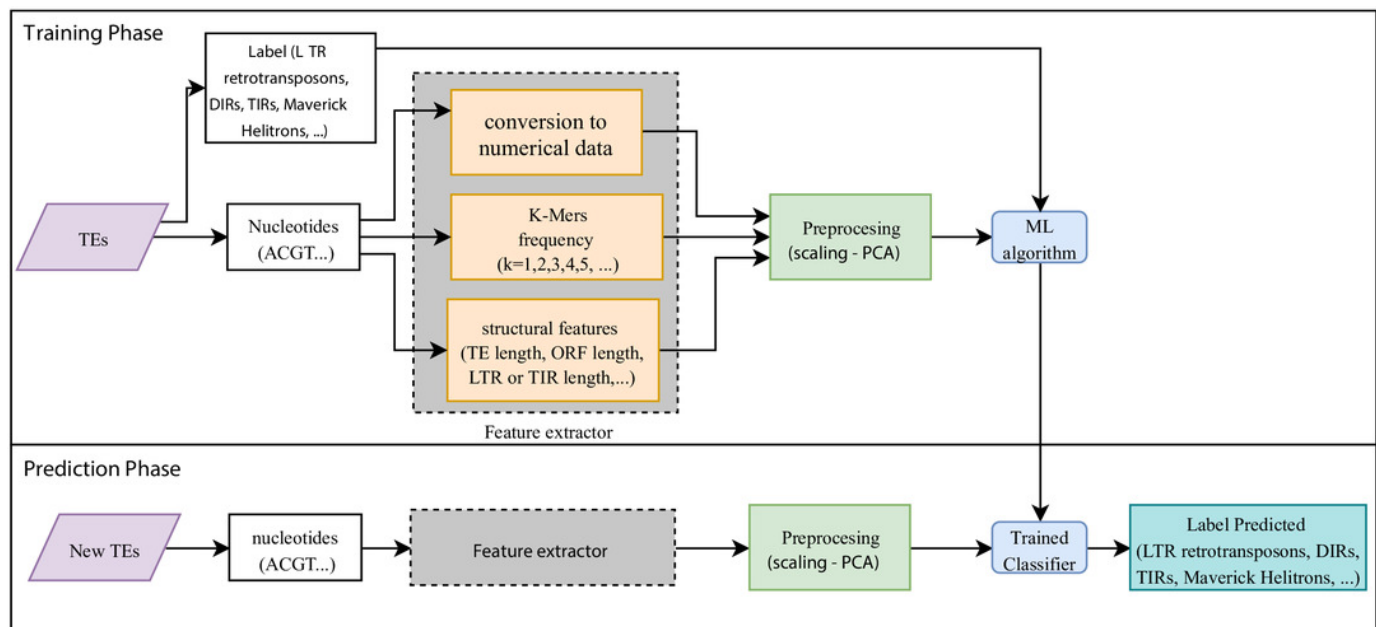
Figure 8

Overall FNN architecture used by Nakano *et al*. to classify TEs. Based on (Nakano et al., 2018b).
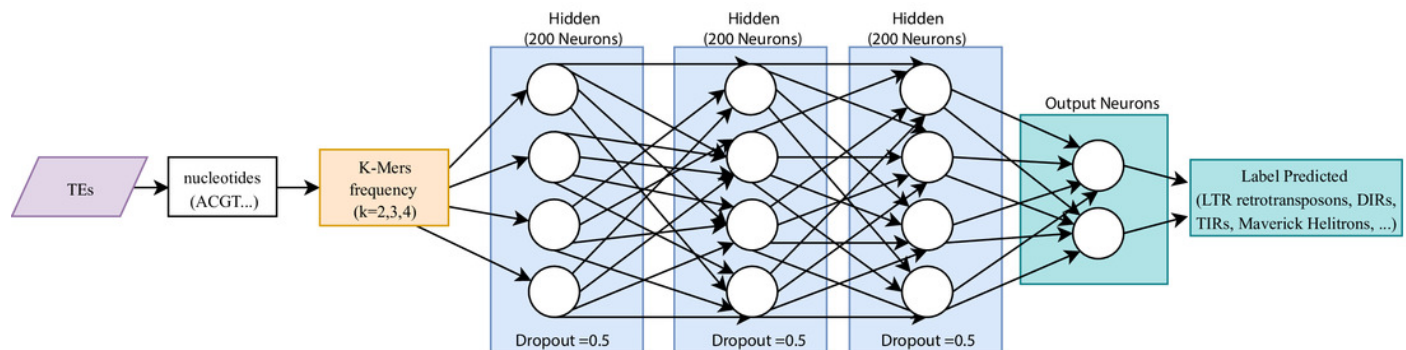
Figure 9

Overall CNN architecture used by da Cruz *et al*. to classify TEs. Based on (da Cruz et al., 2019).
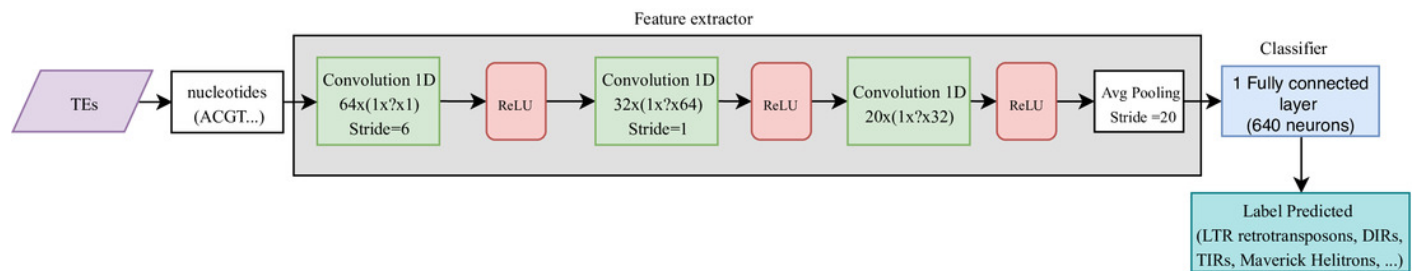
Figure 10

Equations for hierarchical metrics. Z_i and C_i correspond, respectively, to the set of true and predicted classes for an instance i.

A) Hierarchical precision, B) hierarchical recall and C) hierarchical F1-score.

$$
\text{A} \quad hP = \frac{\sum_i |Z_i \cap C_i|}{\sum_i |Z_i|}
$$

$$
\text{B} \quad hR = \frac{\sum_i |Z_i \cap C_i|}{\sum_i |C_i|}
$$

$$
\text{C} \quad hF = \frac{2 * hP * hR}{hP + hR}
$$

Literature resources used in this review.

Database	Link
Scopus	https://www.scopus.com
Science Direct	https://www.sciencedirect.com/
Web of Science	https://clarivate.com/products/web-of-science/
SpringerLink	https://link.springer.com/
PubMed	https://www.ncbi.nlm.nih.gov/pubmed/
Nature	https://www.nature.com/

Selected publications and their contribution to each research question.

Publication identifier	Year	Q1	Q2	Q3	Q4	Reference	Publication identifier	Year	Q1	Q2	Q3	Q4	Reference
P1	2017	X	X	X	X	(Yu, Yu & Pan, 2017)	P19	2013	X	X	X		(Loureiro et al., 2013b)
P2	2018	X	X	X		(Schietgat et al., 2018)	P20	2014		X	X		(Ma, Zhang & Wang, 2014)
P3	2017	X	X	X		(Arango-López et al., 2017)	P21	2010	X	X			(Dashti & Masoudi-Nejad, 2010)
P4	2013	X	X	X		(Loureiro et al., 2013a)	P22	2010		X	X		(Ding, Zhou & Guan, 2010)
P5	2011	X	X	X		(Tsafnat et al., 2011)	P23	2019				X	(Jaiswal & Krishnamachari, 2019)
P6	2018	X	X	X		(Zhang et al., 2018)	P24	2015	X	X	X	X	(Girgis, 2015)
P7	2019	X	X		X	(Eraslan et al., 2019)	P25	2018		X	X	X	(Nakano et al., 2018a)
P8	2018	X	X	X		(Douville et al., 2018)	P26	2018		X	X	X	(Zamith Santos et al., 2018)
P9	2018		X	X		(Chen et al., 2018)	P27	2009		X			(Abrusan et al., 2009)
P10	2012	X	X	X	X	(Ashlock & Datta, 2012)	P28	2019	X	X			(Su, Gu & Peterson, 2019)
P11	2017	X	X	X		(Smith et al., 2017)	P29	2017	X	X	X	X	(Nakano et al., 2017)
P12	2014	X	X	X	X	(Kamath, De Jong & Shehu, 2014)	P30	2014		X	X	X	(Brayet et al., 2014)
P13	2016	X	X	X		(Kim et al., 2016)	P31	2013		X			(Zamani et al., 2013)
P14	2018	X	X	X		(Segal et al., 2018)	P32	2019		X			(Hubbard et al., 2019)
P15	2011	X	X	X		(Rawal & Ramaswamy, 2011)	P33	2014		X	X		(Ryvkin et al., 2014)
P16	2017	X	X	X		(Tang et al., 2017)	P34	2013	X	X	X	X	(Zhang et al., 2013)
P17	2017	X	X	X		(Ventola et al., 2017)	P35	2019		X	X		(da Cruz et al., 2019)
P18	2018		X	X	X	(Nakano et al., 2018b)							

Machine learning algorithms used in publications selected in this study.

Abbreviations: Random Forest (RF), Decision Trees (DT), Bayesian Networks (BN), Neural networks (NN), Hidden Markov Model (HMM), Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), K-Nearest Neighbors (KNN), Self-Organizing Map (SOM), Principal Component Analysis (PCA), Multi-Layer Perceptron (MLP), first-order random forests (FORF). The full version of this table can be consulted in Table S1.

Publication	Data Source	Task	ML algorithm	Learning method	Reference
P2	Numerical and categorical features based on coding regions	Detect LTR Retrotransposons at the super-family level	RF	Supervised	(Schietgat et al., 2018)
P3	Numerical and categorical features	Classify LTR Retrotransposons at the lineage level	DT, BN and lazy algorithms	Supervised	(Arango-López et al., 2017)
P4	Numerical and categorical features	Improve the detection and classification of TEs	NN, BN, RF, DT	Supervised	(Castellanos-Garzón & Díaz, 2013)
P5	Numerical features based on structure	Detect boundary sequences of mobile elements	HMM, SVM	Unsupervised and Supervised	(Tsafnat et al., 2011)
P6	85 numerical features in four categories (genomic, epigenetic, expression, network)	Detection of cancer-related long non-coding RNA	RF, NB, SVM, LR and KNN	Supervised	(Zhang et al., 2018)
P8	Z-score features, representing chromosome arm gains and losses.	Detection of aneuploidy	SVM	Supervised	(Douville et al., 2018)
P10	K-mer frequencies and frequencies of certain patterns	Distinguishing Endogenous Retroviral LTRs from SINEs	RF	Supervised	(Ashlock & Datta, 2012)
P11	Dinucleotide frequencies	Identification and clustering of RNA structure motifs	Density-based clustering	Unsupervised	(Smith et al., 2017)
P12	Sequences of nucleotides (DNA) and categorical features	Automatization of the process of extracting discriminatory features for determining functional properties of biological sequences	Evolutionary Feature Construction and Evolutionary Feature Selection	Unsupervised	(Kamath, De Jong & Shehu, 2014)
P14	Numerical features	Analysis of mutants	RF	Supervised	(Segal et al., 2018)
P15	Insertion sites	Identification of potential insertion sites of mobile elements	SVM	Supervised	(Rawal & Ramaswamy, 2011)
P16	Numerical features	Identification of somatic LINE-1 insertions	LR	Supervised	(Tang et al., 2017)

P17	Numerical features, ARN mononucleotides, dinucleotides and trinucleotides frequencies, Fickett score	Identification of most informative features of long non-coding transcripts	11 different feature selection approaches, SVM, RF, and NB	Supervised	(Ventola et al., 2017)
P19	Numerical and categorical features	Improve the detection and classification of TEs	NN, BN, RF, DT	Supervised	(Loureiro et al., 2013b)
P21	K-mer frequencies	Classify repetitive sequences	SVM	Supervised	(Hesam & Ali, 2010)
P22	Numerical features	Prediction of microRNA precursors	SVM	Supervised	(Ding, Zhou & Guan, 2010)
P24	Sequences of nucleotides (DNA)	Detecting repeats de novo	HMM	Supervised	(Girgis, 2015)
P26	K-mer frequencies	Classify TEs using hierarchical	DT, RF, NB, KNN, MLP, SVM	Supervised	(Zamith Santos et al., 2018)
P27	K-mer frequencies	Classify TEs	SVM	Supervised	(Abrusan et al., 2009)
P28	Numerical features based on structure	Identify sequence motifs conserved in each of the five major TIR superfamilies.	NN, KNN, RF, and Adaboost	Supervised	(Su, Gu & Peterson, 2019)
P30	Numerical features and k-mer frequencies	piRNA prediction	SVM	Supervised	(Brayet et al., 2014)
P31	Aligned genomes and binary representation (1 for mismatches and 0 for matches)	Recognition of local relationship patterns	HMM, SOM	Unsupervised	(Zamani et al., 2013)
P32	Numerical features	Compare multiple transposon insertion sequencing studies	PCA	Unsupervised	(Hubbard et al., 2019)
P33	Numerical and categorical features, nucleotide frequencies	Classify the precursors of small non-coding RNAs	RF	Supervised	(Ryvkin et al., 2014)
P34	Normalized numerical and categorical features	Prediction of transcriptional effects by intronic endogenous retroviruses	MLP NN	Supervised	(Zhang et al., 2013)

DL architectures used in genomic data reviewed in (Eraslan et al., 2019) . Architecture details used in each work can be consulted in Table S2.

Dataset Features	Task	DNN type	Framework or language	Year	Reference
Presence of binding motifs of splice factors or sequence conservation	Predict the percentage of spliced exons	Fully connected NN	TensorFlow	2017	(Jha, Gazzara & Barash, 2017)
Numerical features, k-mer frequencies (k = 1,2,3,4)	Prioritize potential disease-causing genetic variants	Fully connected NN	Matlab	2016	(Liu et al., 2016)
Chromatin marks, gene expression and evolutionary conservation	Predict cis-regulatory elements	Fully connected NN	Python	2018	(Li, Shi & Wasserman, 2018)
Microarray and sequencing data	Predict binarized in vitro and in vivo binding affinities	Convolutional NN	Python + CUDA	2015	(Alipanahi et al., 2015)
A 1,000 bp sequence	Predict the presence or absence of 919 chromatin features	Convolutional NN	LUA	2015	(Zhou & Troyanskaya, 2015)
A 600bp sequence (one-hot matrix)	Predict 164 binarized DNA accessibility features	Convolutional NN	Torch7	2016	(Kelley, Snoek & Rinn, 2016)
DNA sequence (one-hot matrix)	Classify transcription factor binding sites	Convolutional NN	Torch7	2018	(Wang et al., 2018)
DNA sequence (one-hot matrix)	Predict molecular phenotypes such as chromatin features	Convolutional NN	TensorFlow	2018	(Kelley et al., 2018)
DNA sequence (one-hot matrix) and DNAse signal	DNA contact maps	Convolutional NN	Python	2018	(Schreiber et al., 2018)
DNA sequence (one-hot matrix) and DNAse signal	DNA methylation	Convolutional NN	Theano + Keras	2017	(Angermueller et al., 2017)
DNA sequences	Transform genomic sequences to epigenomic features	Convolutional NN	PyTorch	2018	(Zhou et al., 2018)
K-mer frequencies and their positions	Predict translation efficiency	Convolutional NN	Keras	2017	(Cuperus et al., 2017)
DNA sequence (one-hot matrix) and DNAse signal	Predict RNA-binding proteins	Convolutional NN	TensorFlow	2018	(Budach & Marsico, 2018)
Numerical features	Predict microRNA (miRNA) targets	Convolutional NN	-	2016	(Cheng et al., 2015)
Numerical features	Aggregate the outputs of CNNs for predicting single-cell DNA methylation state	Recurrent NN	Theano + Keras	2017	(Angermueller et al., 2017)
RNA sequence (one-hot matrix)	Predict RNA-binding proteins	Recurrent NN	Keras	2018	(Pan et al., 2018)
DNA sequence (one-	Predict transcription	Recurrent NN	Theano +	2019	(Quang &

hot matrix)	factor binding and DNA accessibility		Keras		Xie, 2019)
RNA sequence (weight matrices)	Predict the occurrence of precursor miRNAs from the mRNA sequence	Recurrent NN	Theano + Keras	2016	(Park et al., 2016)
Gene expression level (binary, over or under-expressed)	Predict binarized gene expression given the expression of other genes	Graph-convolutional NN	Torch7	2018	(Dutil et al., 2018)
Gene expression profile and protein-protein interaction network	Classify cancer subtypes	Graph-convolutional NN	-	2017	(Rhee, Seo & Kim, 2017)

Metrics used in TEs and other similar tasks.

Adopted from (Kamath, De Jong & Shehu, 2014; Brayet et al., 2014; Ma, Zhang & Wang, 2014; Yu, Yu & Pan, 2017; Smith et al., 2017; Chen et al., 2018; Schietgat et al., 2018; Segal et al., 2018). D for detection and C for classification.

Metric	Representation	Observations	Tasks in which it was used
Accuracy	$$\frac{(TP + TN)}{(TP + FP + FN + TN)}$$	Measures the percentage of samples that are correctly classified	D, C
Precision	$$\frac{TP}{(TP + FP)}$$	Percentage of correct predictions	D
Sensitivity (recall)	$$\frac{TP}{(TP + FN)}$$	Represents the proportion of positive samples that are correctly predicted	D, C
Specificity	$$\frac{TN}{(TN + FP)}$$	Represents the proportion of negative samples that are correctly predicted	D
Matthews correlation coefficient	$$\frac{TP \times TN - FN \times FP}{\sqrt{(TP+FN) \times (TN+FP) \times (TP+FP) \times (TN+FN)}}$$	It can be a key measurement because it is a balanced measurement, even if the sizes of positive and negative samples have great differences.	D
Positive predictive value	$$\frac{TP}{(TP + FP)}$$	Percentage of correctly classified positive samples among all positive-classified ones	D, C
Performance coefficient	$$\frac{TP}{(TP + FN + FP)}$$	Ratio of correct predictions belonging to the positive class and predictions belonging to the false class	D
F1 score	$$\frac{2 \times TP}{(2\ x\ TP + FP + FN)}$$	Harmonic mean of precision and sensitivity	D
Precision-recall curves	Graphics	Plots the precision of a model as a function of its recall	D, C
Receiver Operating	Graphics	Commonly used to evaluate the	C

Characteristic curves (ROCs)		discriminative power of the classification model at different thresholds	
Area under the ROC curve (AUC)	Graphics	Summary measure that indicates whether prediction performance is close to random (0:5) or perfect (1:0). Also describes the sensitivity versus the specificity of the prediction	D, C
Area under the Precision Recall Curve (auPRC)	Graphics	Measures the fraction of negatives misclassified as positives and plots the precision vs. recall ratio	D
False positive rate	1 – Specificity	Percentage of predictions marked as belonging to the positive class, but that are part of the negative class.	D

Coding schemes for translating DNA characters in numerical representations. Adapted from (Yu, Yu & Pan, 2017).

Encoding Schemes	Codebook	Reference
DAX	{'C':0, 'T':1, 'A':2, 'G':3}	Yu *et al.* (2015)
EIIP	{'C':0.1340, 'T':0.1335, 'A':0.1260, 'G':0.0806}	Nair and Sreenadhan (2006)
Complementary	{'C':-1, 'T':-2, 'A':2, 'G':1}	Akhtar *et al.* (2008)
Enthalpy	{'CC':0.11, 'TT':0.091, 'AA':0.091, 'GG':0.11, 'CT':0.078, 'TA':0.06, 'AG':0.078, 'CA':0.058, 'TG':0.058, 'CG': 0.119, 'TC':0.056, 'AT':0.086, 'GA':0.056, 'AC':0.065, 'GT':0.065, 'GC':0.1111}	Kauer and Blöcker (2003)
Galois (4)	{'CC':0.0, 'CT':1.0, 'CA':2.0, 'CG':3.0, 'TC':4.0, 'TT':5.0, 'TA':6.0, 'TG':7.0, 'AC:8.0, 'AT: 9.0, 'AA':1.0, 'AG:11.0, 'GC':12.0, 'GT':13.0, 'GA':14.0, 'GG':15.0 }	Rosen (2006)
Orthogonal (one-hot) Encoding	{'A': [1, 0, 0, 0], 'C': [0, 1, 0, 0], 'T': [0, 0, 1, 0], 'G': [0, 0, 0, 1]}	Baldi *et al.* (2001)