IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 1, 2020, accepted September 19, 2020, date of publication September 30, 2020,
date of current version October 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028039

A Novel Decomposing Model With Evolutionary Algorithms for Feature Selection in Long Non-Coding RNAs

ROBSON P. BONIDIA[1,2], JAQUELINE SAYURI MACHIDA[1], TATIANNE C. NEGRI[3], WONDER A. L. ALVES[3], (Member, IEEE), ANDRÉ Y. KASHIWABARA[1], DOUGLAS S. DOMINGUES[1,4], ANDRÉ DE CARVALHO[2], (Member, IEEE), ALEXANDRE R. PASCHOAL[1], AND DANILO S. SANCHES[1]

[1]Department of Computer Science, Bioinformatics Graduate Program, Federal University of Technology—Paraná (UTFPR), Cornélio Procópio 86300-000, Brazil
[2]Institute of Mathematics and Computer Sciences, University of São Paulo (USP), São Carlos 13566-590, Brazil
[3]Universidade Nove de Julho (UNINOVE), São Paulo 01504-001, Brazil
[4]Department of Botany, Institute of Biosciences, São Paulo State University (UNESP), Rio Claro 13506-900, Brazil

Corresponding authors: Robson P. Bonidia (rpbonidia@gmail.com) and Danilo S. Sanches (danilosanches@utfpr.edu.br)

This work was supported in part by the master's scholarship from the Federal University of Technology—Paraná (UTFPR) under Grant April/2018, in part by the PROBAL (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)/DAAD—88887.144045/2017-00, and in part by the CAPES (PROEX-11919694/D).

ABSTRACT Machine learning algorithms have been applied to numerous transcript datasets to identify Long non-coding RNAs (lncRNAs). Nevertheless, before these algorithms are applied to RNA data, features must be extracted from the original sequences. As many of these features can be redundant or irrelevant, the predictive performance of the algorithms can be improved by performing feature selection. However, the most current approaches usually select features independently, ignoring possible relations. In this paper, we propose a new model, which identifies the best subsets, removing unnecessary, irrelevant, and redundant predictive features, taking the importance of their co-occurrence into account. The proposed model is based on decomposing solutions and is called k-rounds of decomposition features. In this model, the least relevant features are suppressed according to their contribution to a classification task. To evaluate our proposal, we extract from 5 plant species datasets, a set of features based on sequence structures, using GC content, k-mer (1-6), sequence length, and Open Reading Frame. Next, we apply 5 metaheuristics approaches (Genetic Algorithm, $(\mu + \lambda)$ Evolutionary Algorithm, Artificial Bee Colony, Ant Colony Optimization, and Particle Swarm Optimization) to select the best feature subsets. The main contribution of this work was to include in each metaheuristic a decomposition model that uses round and voting scheme. To investigate its relevance, we select the REPTree classifier to assess the predictive capacity of each subset of features selected in 8 plant species. We identified that the inclusion of the proposed decomposition model significantly reduces the dimensions of the datasets and improves predictive performance, regardless of the metaheuristic. Furthermore, the resulting pipeline has been compared with five approaches in the literature, for lncRNA, when it also showed superior predictive performance. Finally, this study generated a new pipeline to find a minimum number of features in lncRNAs and biological sequences.

INDEX TERMS Feature selection, machine learning, metaheuristic, bioinformatics, lncRNAs.

I. BACKGROUND

In recent years, the power to process and analyze biological data has advanced significantly [1]. Computational approaches have been widely used in protein structure prediction, genomics, proteomics, gene networks and protein-coding genes detection [2], [3]. Many of these approaches are based on Machine Learning (ML) and have been applied in predictive tasks. In these tasks, as data classification, ML algorithms induce predictive models able to associate predictive features from an instance to its class. Consequently, these approaches have contributed to

The associate editor coordinating the review of this manuscript and approving it for publication was Long Xu.

applications in new problems, as is the case of the Long non-coding RNAs (lncRNAs). Fundamentally, lncRNAs are a type of Non-Coding RNA (ncRNA) with a length larger than 200 nucleotides [4], and according to recent studies, play essential roles in several critical biological processes [5]–[7], including transcriptional regulation [8], epigenetics [9] and cellular differentiation [10]. They are correlated with some complex human diseases, such as cancer and neurodegenerative diseases [11], [12]. According to Wang and Chekanova [13], in plants, the lncRNAs act in flowering time control, organogenesis in roots, gene silencing, stress responses [14], [15], photomorphogenesis in seedlings, and reproduction [16]. Essentially, they are observed in almost all living beings, not only in animals and plants, but also in yeasts, prokaryotes, and even viruses [17], [18].

However, according to [19], several challenges arise when ML approaches are used to integrate biological and biomedical datasets, because these datasets have inherent complexity beyond their large sizes. Biological datasets are also high-dimensional, incomplete, biased, heterogeneous, dynamic, and noisy [19]. Another problem is that several applications in bioinformatics apply ML algorithms to sequence data, and as many ML algorithms can deal only with numerical data, sequences need to be translated into sequences of numbers. Early applications transformed each letter in the sequence to binary vector [20]. These transformations resulted in very long sequences with sparse data [19]. This difficulty grows as the size of the sequences grow. A more straightforward, and more efficient approach, adopted by most current applications of ML algorithms, extracts relevant features from the sequences. These features are based on several properties, e.g., physicochemical, ORF-based, usage frequency of adjoining nucleotide triplets, and sequence-based.

Thereby, ML algorithms have been used in several tools to predict lncRNAs, such as: CPC [21], CPC2 [22], CPAT [23], CNCI [24], PLEK [25], lncRNA-MFDL [26], LncRNA-ID [27], lncRScan-SVM [28], LncRNApred [29], DeepLNC [30], TERIUS [31], BASiNET [32], LncFinder [33], PLncPRO [34], PlantRNA_Sniffer [35] and RNAplonc [36]. In these tools, ML algorithms were applied to data from more than one species or specifically to plant, animal, and human systems. These approaches allowed access to more information about lncRNAs due to using different features and ML algorithms [28]. However, many studies have investigated the use of several features to extract significant information from lncRNAs, generating highly dimensional feature vectors.

Fundamentally, a high ratio between the number of instances and predictive features is behind the curse of dimensionality problem. In this problem, as the ratio increases, samples become so similar that it is challenging to induce models with high predictive precision. Thereby, when the number of characteristics increases substantially, the predictive performance of models induced by ML algorithms decreases. An alternative to mitigate this problem is to reduce the number of features, removing irrelevant and redundant features [37]. According to Xue *et al.* [38], the feature selection is an essential process in ML, as it reduces the feature extraction and model induction computational costs. Additionally, it can increase predictive performance. Finally, as the extraction of each feature also has an economical cost, it can reduce the financial cost of an ML-based predictive tool, making it available to a higher number of users.

The high dimensional nature of many molecular biology datasets has motivated us to look for efficient feature selection techniques [39]–[41]. In the literature, several approaches have been used in the biological data classification. In particular, feature selection techniques have been applied in the study of ncRNA, for instance, e.g., Wang *et al.* [5] (reduced from 74 to 26 - lincRNA) and Lertampaiporn *et al.* [41] (reduced from 369 to 20 - ncRNA) report experiments using metaheuristics. Pian *et al.* [29] (reduced from 89 to 30 - lncRNA) and Negri *et al.* [36] (reduced from 5,468 to 16 - lncRNA) described the use of conventional feature selection techniques, respectively: feature score criterion, WrapperSubsetEval, InfoGainAttribute, GainRatioAttributeEval. Even so, we noticed a lack of studies related to feature selection with metaheuristic techniques for lncRNAs.

Several studies have shown robust results when using metaheuristics for feature selection and other applications, such as Genetic Algorithm (GA) [5], [42]–[45], Evolutionary Algorithm ($(\mu + \lambda)$EA) [46]–[48], Bat algorithm (BA) [49]–[52], Artificial Bee Colony (ABC) [53]–[56], Ant Colony Optimization (ACO) [57], [58], and Particle Swarm Optimization (PSO) [44], [47], [59]–[61]. Furthermore, the literature presents several other representational EA-based feature selection methods, such as variable-size cooperative co-evolutionary PSO for feature selection on high-dimensional data [62], multi-objective PSO approach for cost-based feature selection in classification [63], binary differential evolution with self-learning for multi-objective feature selection [64], cost-sensitive feature selection using two-archive multi-objective ABC algorithm [65], and return-cost-based binary firefly algorithm for feature selection [66]. These metaheuristics, also known as nature-inspired algorithms [67], are suitable for feature selection, since they can easily represent features and efficiently search for a good feature subset in high dimensional datasets [68], [69]. Moreover, these metaheuristics are an efficient alternative to exact methods, which usually have high computational costs [70]. Furthermore, metaheuristics are able to avoid getting stuck in local optima, which often occurs in feature selection problems, selecting relevant features and improving predictive performance.

Therefore, based on the successful use of metaheuristics, we propose a novel way to identify the best features, removing unneeded, irrelevant, and redundant content from datasets. Our approach is based on a decomposing model, using a rounds and voting schema, which is inserted in the metaheuristics. Essentially, this study proposes an innovative procedure for feature selection (Feature Selection stage (see Figure 1)), where we consider each round as a subset of

features that are near-optimal. This permits a reduction of the search space, i.e., features that must be visited to find a global optimum, when compared to the original form of the metaheuristics. Based on this, to assess the proposed approach, we used five metaheuristics (GA, $(\mu + \lambda)$EA, ABC, ACO, and PSO). Furthermore, we investigated how the features selected by the decomposing model affected the predictive performance of three ML algorithms, i.e., J48, REPTree and Random Forest in the lncRNAs classification task. We chose these ML algorithms because they induce interpretable predictive models, allowing the understanding of the internal decision-making process. Thus, domain experts can validate the knowledge used by the models to classify new sequences. Finally, our approach focuses on the minimal optimal problem (defined by Nilsson *et al.* [71]), whose purpose is to find the smallest subset of features that contains all information. This study contributes to the area of computer science and bioinformatics, introducing a new approach for the feature selection problem in biological sequences. Thereby, we present five main contributions:

- Application of a new approach based on a decomposition model for the feature selection problem in biological sequences
- Decomposition model does not depend on the metaheuristic used;
- An in-depth analysis of 5,467 features extracted from lncRNA sequences;
- A new pipeline to find a minimum number of features in lncRNAs and biological sequences;
- A new tool able to select relevant features, with competitive classification performance.

The remainder of this article presents our methodology in Section II, proposed approach in Section III, results in Section IV, and our final considerations in Section V.

II. MATERIALS AND METHODS

In this section, we describe our methodological approach designed to achieve the proposed objectives. Figure 1 summarizes the pipeline developed in this study. Basically, we divided the process into seven stages: (1) preprocessing of FASTA files; (2) Split sequences into training (see Table 1) and test (see Table 2); (3) Features extraction; (4) Feature Selection; (5) Training; (6) Test; (7) Performance analysis. Nevertheless, it is necessary to emphasize that we denote a sequence w over the alphabet $\beta = \{A, C, G, T\}$ by $w = w_1, w_2, \ldots, w_{|w|}$, where $|w|$ is the sequence length $w_l \in \beta$. Furthermore, we denoted by $|w|_\sigma$ the number of symbols $\sigma \in \beta$ in w. Therefore, a substring of length $k > 0$ starting at position l in w is given by $w_{l,k} = w_l, w_{l+1}, \ldots, w_{l+k-1}$. Finally, we denote all sequences of our dataset (mRNA/lncRNA) by *SeqRNA*.

A. TRAINING SET CONSTRUCTION

We built a training set with 5 plant species (*Arabidopsis thaliana*, *Cucumis sativus*, *Glycine max*, *Oryza sativa* and *Populus trichocarpa* - see Table 1), adopting the data representation used in Negri *et al.* [36]. Two classes were defined for the datasets: positive class, lncRNAs, and negative class, protein-coding genes (mRNAs). The lncRNA data were extracted from two public databases, `PLNlncRbase` (defined by B_{PLN}) [72] and `GreeNC` (version 1.12 - defined by B_{Gree}) [73].

TABLE 1. Species used to create the training set. The "#used" demonstrates the total number of sequences used after filtering steps.

	lncRNA			mRNA	
Species	B_{PLN}	B_{Gree}	#used	B_{Phy}	#used
A. thaliana	119	3010	1804	35386	1804
C. sativus	8	1935	1804	30364	1804
G. max	1	6693	1804	88647	1804
O. sativa	38	5238	1804	52424	1804
P. trichocarpa	15	5574	1804	73013	1804
Total	**181**	**22450**	**9020**	**279834**	**9020**

The mRNA data were downloaded from `Phytozome` (defined by B_{Phy}) [74] database version 11. The datasets were chosen considering the publicly available annotation and its phylogenetic diversity [36]. As preprocessing, we removed sequence redundancy at 80% of identity using the `CD-HIT-EST` tool (v4.6.1) [75] and we selected only sequences longer than 200nt [41], [76]. Furthermore, we are faced with the imbalanced data problem. Therefore, we applied random sampling, selecting 1,804 sequences for each species. After preprocessing, we used a total of 9,020 lncRNA sequences and 9,020 mRNA sequences.

B. TEST SET CONSTRUCTION

To assess the proposed approach, we created eight datasets of plant species (*Amborella trichopoda, Brachypodium distachyon, Citrus sinensis, Manihot esculenta, Ricinus communis, Solanum tuberosum, Sorghum bicolor* and *Zea mays*), summarized in Table 2.

TABLE 2. Species used to create the test set. The "#used" reports the total number of sequences selected in each species.

Species	lncRNA	# used	mRNA	# used
A. trichopoda	5698	3823	26846	3823
B. distachyon	5584	4868	52972	4868
C. sinensis	2562	2292	46147	2292
M. esculenta	3468	3017	41381	3017
R. communis	4198	4080	31221	4080
S. tuberosum	6680	5607	51472	5607
S. bicolor	5305	4541	47205	4541
Z. mays	18154	12071	88760	12071
Total	**51649**	**40299**	**386004**	**40299**

The test sets were treated in the same way as the training set. The lncRNA sequences were extracted from `GreeNC`, and mRNA sequences from `Phytozome`.

C. FEATURE EXTRACTION

The feature vector has a strong effect on the performance of predictive models. Thereby, the extraction of relevant features plays an essential role, being one of the most critical steps in the induction of a robust predictor/classifier [26]. This

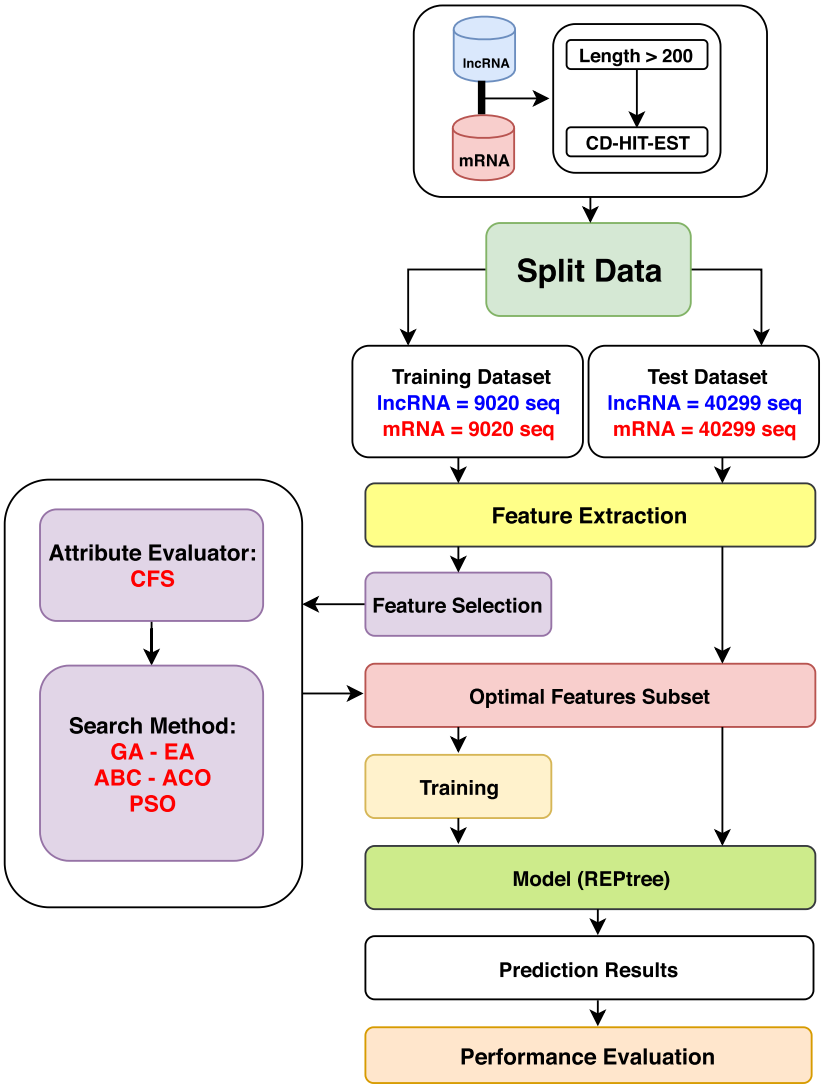


FIGURE 1. Proposed Pipeline for the Feature Selection Problem in lncRNAs data. The FASTA format files (lncRNA - positive dataset and mRNA - negative dataset) were filtered to find sequences larger than 200 nucleotides (size > 200), and redundant sequences with 80% of identity (CD-HIT-EST) were removed. This dataset was divided into training (9020 lncRNA and 9020 mRNA) and test (40299 lncRNA and 40299 mRNA). Features were extracted from each sequence. Filters were applied to the training set to select a subset of features. Next, for each selected feature set, ML algorithms were applied to the data to induce predictive models. The models induced for each filter were applied to the test set, using the same selected features in the training set. Finally, the predictions of the model induced for each filter were evaluated.

study used four descriptor groups to distinguish lncRNA from mRNA. That is, four sets of values were extracted from the sequences, creating four vectors, described next.

1) GC CONTENT DESCRIPTOR

According to the literature, when we compare lncRNAs with mRNAs, the lncRNAs have low GC content [77]. The calculus of the GC content (guanine-cytosine content - denoted by f_{GC}) is illustrated by Equation (1).

$$f_{GC}(w) = \frac{|w|_G + |w|_C}{\sum_{\sigma \in \beta} |w|_\sigma}. \quad (1)$$

where the sum of symbols $|w|_G$ and $|w|_C$ is divided by the total length of the sequence (sum of $|w|_A$, $|w|_T$, $|w|_C$, $|w|_G$).

2) K-MER DESCRIPTOR

The frequency of neighboring bases k (k-mer) may contain statistical information to distinguish lncRNAs from mRNAs. We denoted k-mer by f_{kmer}, according to Equation (2).

$$f_{kmer}(w) = \left(\frac{c_1^1}{|w|-1+1}, \ldots, \frac{c_4^1}{|w|-1+1}, \frac{c_{4+1}^2}{|w|-2+1}, \ldots, \frac{c_{5460}^6}{|w|-6+1} \right). \quad (2)$$

This Equation was applied to the transcription sequences with frequencies of $k = 1, 2, 3, 4, 5, 6$. In this equation, c_i^k is the amount occurrences of substrings with length k in w, in which the index $i \in \{1, 2, \ldots, 4^1 + \ldots + 4^k\}$ represents the analyzed substring.

3) SEQUENCE LENGTH DESCRIPTOR

Because the lncRNAs were shown to be considerably shorter than the mRNAs, the sequence length (denoted by f_{SL} in Equation (3)) was also adopted [77], [78].

$$f_{SL}(w) = \sum_{\sigma \in \beta} |w|_\sigma . \tag{3}$$

4) OPEN READING FRAME (ORF) DESCRIPTOR

Identifying candidate ORFs in the transcripts is an extremely relevant guideline for distinguishing lncRNAs from mRNA [77], [79], [80]. For such, we analyze the three frames in the forward strand of our sequences using the `txCdsPredict` program from the UCSC genome browser [81].[1] Thereby, we applied this software, which predicts potential ORFs from a given sequence w, to extract the features: txCdsPredict Score, cdsStarts, cdsStop, cdsSizes, and cdsPercent. The features were represented as a vector for the function that we denote by f_{ORF}, corresponding to Equation (4).

$$f_{ORF}(w) = (Score, cdsStarts, cdsStop, cdsSizes, cdsPercent). \tag{4}$$

The `txCdsPredict` has been used in several studies ([28], [82], [83]) to determine whether a transcript is protein-coding and, if so, the locations of the start and stop codons. The algorithm uses ORF length, the presence of a Kozak consensus sequence at the start codon, the presence of upstream ORFs, homology in other species, and nonsense-mediated decay [81]. Furthermore, several tools have used ORF features, among them: [21], [23], [26]–[29], [33], [36].

5) CONCATENATE FEATURE VECTORS

According to Fan & Zhan [26], a concatenated feature vector can keep the discriminatory information from original multi-feature sets and eliminate the redundant information from the correlation between distinct feature sets, resulting in models with robust predictive performance. To denote each transcribed sequence in the dataset, we concatenate the previously mentioned features in a new feature vector, defined as follows (Equation (5)):

$$\begin{aligned} V_f &= \{(X_i, Y_i) | \forall\, w_i \in \text{SeqRNA}, \\ X_i &= (f_{GC}(w_i), f_{kmer}(w_i), f_{SL}(w_i), f_{ORF}(w_i)), \\ Y_i &= Label(w_i)\}. \end{aligned} \tag{5}$$

where, feature vector V_f contains the elements X_i and Y_i for every sequence w_i belonging to *SeqRNA*, such that X_i is formed by the functions ($f_{GC}(w_i)$, $f_{kmer}(w_i)$, $f_{SL}(w_i)$, $f_{ORF}(w_i)$) and Y_i by labels 0 (mRNA); 1 (lncRNA). Therefore, we extracted 5,467 genomic characteristics for each sequence relative to the training set (see Table 1): GC content (1 feature), k-mer (1-6 k-mer length = 5, 460 features), Sequence length (1 feature), and ORF metrics (5 features).

[1] (https://genome.ucsc.edu/ [81])

D. DATA NORMALIZATION

In this work, we used the min-max normalization method, which reduces the data range to 0 and 1 (or -1 to 1, if there are negative values). The general formula is given as (Equation (6)) [84]:

$$x'_{ij} = \frac{x_{ij} - Min(j)}{Max(j) - Min(j)}. \tag{6}$$

where x is the original value and x'_{ij} is its normalized version. Further, $Min(j)$ and $Max(j)$ the smallest and the largest values of a feature j, respectively [84].

E. FEATURE SELECTION TECHNIQUES

Feature selection techniques are typically categorized as filters, wrappers, or embedded approaches [85]. Filters are applied independent of the ML algorithm used [86], considered as a preprocessing stage for a subsequent learning [87]. They exploit the information present in the predictive features of a dataset, assessing their relevance using measures such as information gain, entropy, and consistency [85], [86]. Wrappers evaluate the relevance of subsets of predictive features using an ML algorithm as an oracle [88], i.e., they use the accuracy of predictive models to guide the selection of an optimal subset of features [87]. The embedded approach is implemented as part of an ML algorithm that has an internal feature selection mechanism [39], [89].

In this paper, we applied *Filters* to select subsets of features in a preprocessing step, independently of the ML algorithm used. According to Guyon *et al.* [39], there are several justifications for the use of filters, among them: (1) filters were successfully reported in several previous works. (2) Compared to wrappers, filters are faster. (3) Filters provide a generic selection of variables, i.e., the choice of features is not adjusted to a particular ML algorithm.

F. EVALUATION METRICS

The algorithms were assessed with seven measures [26], [90]: Sensitivity (SE), Specificity (SPC), Accuracy (ACC), F1-score, Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Matthews Correlation Coefficient (MCC). These measures were used to evaluate the models' predictive performance. These measures use True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values, where: TP measures the correctly predicted lncRNAs; TN represents the correctly classified mRNAs; FP describes all those negative entities that are incorrectly classified as lncRNAs and; FN represents the true lncRNAs that are incorrectly classified as mRNAs.

G. METAHEURISTICS

For the experiments, we chose five metaheuristics, among them: GA [5], [42], $(\mu + \lambda)$EA [46]–[48], ABC [54]–[56], ACO [57], [58], PSO [44], [47], [59]–[61]. These metaheuristics were based on successful applications for feature selection and other areas of applications, such as: Engineering, Computer Science, Manufacturing, and so on. Moreover, they are considered state-of-art in Evolutionary Algorithms. Thereby, these metaheuristics are briefly described next.

1) GENETIC ALGORITHM (GA)

GA is a general stochastic search algorithm that effectively exploits large search spaces, which is generally required in selection cases. Moreover, GAs conduct global search and are based on the mechanics of natural selection. Essentially, GAs simulate the processes in natural systems for evolutions based on the principle of the ''survival of the fittest'' (Charles Darwin) [91]. Moreover, they work with the coding of the parameter set and use reward information (objective function). Therefore, GA applied in this paper comprises three basic operators: reproduction, crossing, and mutation.

2) $(\mu + \lambda)$EVOLUTIONARY ALGORITHM (EA)

The $(\mu + \lambda)$EA used in this work applies the fitness ranking selection procedure. In other words, at the end of each evolution cycle, the whole population is renewed according to generational substitution scheme. Furthermore, elitism and tournament are applied, in which the fittest individual of the population is kept in the new generation. The chromosomes (Binary encoding) are manipulated using standard genetic operators of mutation and crossover (single-point crossover, bit flip mutation). However, the $(\mu + \lambda)$EA has an extra component (a) that represents the interval width of the mutation, where the modification has a uniform probability $[-a, a]$. Thus, for each individual, the parameter is adjusted adaptively through random mutation events [47], [48].

3) ARTIFICIAL BEE COLONY (ABC)

The ABC is bio-inspired in the food foraging behavior of bees to seek the best solution to an optimization problem. Each point in the search space is considered as a food source. The ''Scout Bees'' randomly sampled the space and through the fitness function, they report the quality of the visited places. The solutions are then ranked, and other ''bees'' are recruited to search the fitness landscape in the neighborhood of the highest ranking locations. The neighborhood of a solution is called a ''flower patch''. Therefore, the algorithm searches the most promising solutions and selectively explores its neighborhoods looking for the global minimum of the objective function [92].

4) ANT COLONY OPTIMIZATION (ACO)

The ACO is a bio-inspired algorithm by the foraging behavior of some species of ants, developed by [93]. This technique applies the pheromone method that ants deposit to demarcate a more favorable path, which must be followed by other members of the colony [93], [94]. Fundamentally, each agent (ants) initially follows a random way, and after some time they tend to follow a single way, considered significant. They use indirect communication to indicate the best route for the other members of the colony. To do this, they spread a substance called pheromone. That is, computationally, the algorithm presents a graph with n vertices and places an artificial ant in each of these. Thereby, each ant traces a path following a probabilistic equation in function of the ''deposited'' pheromone at each edge of the graph. Finally, after constructing all routes, the pheromone intensity in each edge is increased according to the quality of the generated solution.

5) PARTICLE SWARM OPTIMIZATION (PSO)

It is a bio-inspired computational algorithm in the social behavior metaphor about the interaction between individuals (particles) of a group (swarm), developed in 1995 by Kennedy and Eberhart. This algorithm was implemented based on the observation of flocks of birds and shoals of fish in search of food in a certain region [95]. The PSO is a population-based stochastic global optimization algorithm [96]. The version applied in this research uses the geometric framework, where it presents a close relationship between a simplified form of PSO (without the inertia term) and evolutionary algorithms. This framework enables us to generalize, in a natural, rigorous, and automatic way, PSO for any search space for which a geometric crossover is known [95]. This algorithm was developed using theoretical tools of evolutionary algorithms, that is, geometric crossing and geometric mutation. Basically, there is no velocity, the equation of position update is the convex combination, there is mutation and the parameters w_1, w_2, and w_3 are non-negative and add up to one [97].

III. DECOMPOSING MODEL FOR FEATURE SELECTION

In this study, we propose a new approach based on a decomposing model for feature selection, which uses a specific metaheuristic to search the best feature set N times, finding N reasonable solutions (i.e., feature subsets less redundant and irrelevant than the whole set) and selects the best subset with the fitness function used in the metaheuristic search. Features from the best subset remain in the dataset, with the others being discarded (backward elimination), reducing the original dataset. This step is called the *first round*. Next, the algorithm starts the *second round*, updating the individuals in the population. In this case, the metaheuristic used will perform the search process in the reduced feature space (feature subset obtained in the first round).

The iterative process (number of rounds) finishes when: 1) the best solution is evaluated by the fitness function as being worse than the whole set, 2) the best solution has not been improved for a given number of rounds or 3) the maximum round is reached. Notice that with these conditions, the algorithm can execute some rounds, just one (as a base

metaheuristic) or none (no reduction) automatically, based on the fitness. This paper also includes an additional step for decomposing model using voting scheme, called voted solutions. Instead of selecting the best solution from the population, i.e. the best fitness value, the algorithm generates solutions with the most frequently selected features.

In this case, the procedure called voted solutions, verify all solutions in the population and highlight each feature from the dataset that appears with a specific frequency (e.g., a feature frequency is 2 if it appears twice in the population). A voted solution comprises only the features that appear more than a given frequency called minimal frequency. The frequency is a percentage that features occurs in the population. In that case, the frequency is obtained by multiplying the population size and a specific rate, called voting rate. To better understand this process, **Algorithm 1** introduces a pseudo code of round steps and **Algorithm 2** shows the generation of voted solutions.

Algorithm 1 Decomposing Model for Feature Selection

```
Input  : dataset
Output: reduced_dataset
Define max_round parameter;
Initialize counter stagnant_round as 0;
Define max_stagnant_round parameter;
Define metaheuristic for the search process;
Define population_size parameter;
for round ← 1 to max_round do
    Initialize population;
    for h ← 1 to population_size do
        individual ← get solution passing dataset and
        parameters to fitness function;
        Append individual in population;
    voted_solutions ← Generate_voted_solutions();
    if Round_stagnation() == True then
        Break round loop;
    dataset ← Reduce dataset keeping features from
    voted_solutions;
Export dataset;
```

Algorithm 2 Generate_Voted_Solutions()

```
Output: voted_solutions
Define voting_rate parameter;
Initialize vector voted_solutions;
minimal_frequency ← voting_rate * population_size;
for feature in population_size do
    feature_frequency ← how many times feature
    appears in population;
    if feature_frequency ≥ minimal_frequency then
        Append feature to voted_solutions;
Return voted_solutions;
```

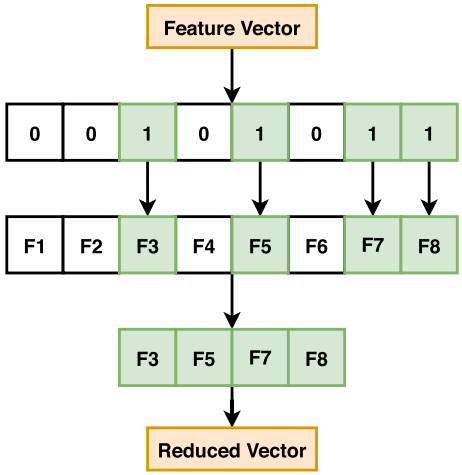


FIGURE 2. An illustration of our encoding scheme for all metaheuristics.

A. ENCODING SCHEME

For the experiments performed in this study, the candidate solutions in all five metaheuristics were represented by binary encoding (each solution as a N-bit binary string), where the individuals/particles are vectors of zeros and ones, representing a feature subset, as shown in Figure 2. For instance, the value 0 in the first position means that the feature does not belong to the subset. The opposite happens if the value is 1, i.e., the feature belongs to the subset.

B. METAHEURISTICS AND FITNESS FUNCTION

We defined the *CFS* (Correlation-based Feature Selection) function [98] (see Equation 7) to evaluate the feature subsets selected by each metaheuristic. This function assesses the degree of redundancy and predictive capacity of a subset. It seeks a subset highly correlated with the target class and with low correlation with other features [99], [100]. The probability that a feature will be selected depends on how well it can predict the correct class, when compared to other features [98].

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{7}$$

where M_S is the "merit" of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f \in S$), and $\overline{r_{ff}}$ is the average feature-feature inter-correlation [98]. In short, we investigate the performance of five metaheuristics using *CFS* algorithm as the fitness function. Moreover, at this point of the paper, we will name GA, $(\mu + \lambda)$EA, ABC, ACO, PSO with a decomposing model and CFS algorithm as M1-GA, M2-EA, M3-ABC, M4-ACO and M5-PSO, respectively.

IV. RESULTS AND DISCUSSION

This section presents the main experimental results using a decomposing model proposed for feature selection.

A. EXPERIMENTAL SETUP

For the experiments reported in this section, we worked with Perl (Version 5.24.1) and Python (Version 3.7.4). The following hyper-parameters values were empirically defined:

- **Problem dimension:** N. of features
- **Search domain:** {0, 1}
- **Number of rounds:** $k = 5$;
- **Voting rate:** defined with 100%;
- **M1-GA:** Crossover Operator (single-point), Crossover Probability (0.6), Generations (20), Mutation Operator (bit-flip), Mutation Probability (0.033), Population Size (20), Selection Operator (roulette wheel);
- **M2-EA:** Generations (20), Crossover Operator (single-point), Crossover Probability (0.6), Mutation Operator (bit-flip), Mutation Probability (0.1), Population Size (20), Selection Operator (tournament);
- **M3-ABC:** Iterations (20), Population Size (30), Number of Selected Sites (15), Number of Elite Sites (8), Number of Selected Site Bee (15), Number of Elite Site Bee (30);
- **M4-ACO:** Evaporation (*rho* = 0.9), Pheromone (α = 2.0), Heuristic (β = 0.7), Q (30), *tau*0 (0.1), Iterations (20), Population Size (20);
- **M5-PSO:** Iterations (20), Social Weight (0.33), Population Size (20), Mutation Operator (bit-flip), Mutation Probability (0.01), Individual Weight (0.34), Inertia Weight (0.30).

B. FEATURE SELECTION PROCESS

In this step, M1-GA, M2-EA, M3-ABC, M4-ACO and M5-PSO were applied to the training set with the purpose of reducing the dimensional space of the extracted features, as shown in Table 3. For each one of the k rounds defined as a parameter, each metaheuristic selected a feature subset of decreasing size (for algorithm details, see section III), as illustrated by Table 3. When the number of features was not reduced, we put the symbol "−". Considering Table 4, we can observe that the algorithms M2-EA and M3-ABC returned a subset with 5 features, while M4-ACO, M5-PSO and M1-GA found 6,7 and 10, respectively.

TABLE 3. Execution Rounds (R) with $k = 5$.

ID	Initial features	R1	R2	R3	R4	R5
M1-GA	5,467	85	32	14	10	-
M2-EA	5,467	569	115	17	5	-
M3-ABC	5,467	12	6	5	-	-
M4-ACO	5,467	164	16	6	-	-
M5-PSO	5,467	59	11	7	-	-

It is important to highlight that two resources intersected in all algorithms (*txCdsPredict score* and *cdsSizes*). Furthermore, all metaheuristics found one or two different intersections, such as M1-GA ∩ M2-EA (CCGGCA), and M1-GA ∩ M5-PSO (GGGGGG, TGACGG). Finally, we can observe that considering only R1 (Table 3), i.e., the metaheuristic without decomposition, it can drop in a stagnation far from an ideal solution. However, when we apply our decomposing model, there is a significant improvement in the subset of features, especially from R3.

TABLE 4. Optimal feature subsets selected by each algorithm.

M1-GA	M2-EA	M3-ABC	M4-ACO	M5-PSO
ATCCCC	CCGGCA	AGCGGA	AGCACT	CGCGGA
CCGGCA	GACTAG	GGGCTA	CCGGGG	CTCGAC
CGCCTC	GAGGGC	GTCGTC	GAGCCC	GCACGC
CGGAGT	score	score	GTCGTA	GGGGGG
CGTTAG	cdsSizes	cdsSizes	score	TGACGG
CTAGGT			cdsSizes	score
GGGGGG				cdsSizes
TGACGG				
score				
cdsSizes				

C. TRAINING PHASE

In order to validate the knowledge extracted by machine learning models, it is increasingly important to use ML algorithms able to induce models that can be understood. This is one of the goals of explainable artificial intelligence [101]. To induce predictive models that can be interpreted, we selected two decision tree induction algorithms (J48, REPTree) and one algorithm that has presented very good predictive performance inducing a forest of decision trees (Random Forest). We used these algorithms to perform several experiments to assess the predictive performance obtained using selected features at each round of each metaheuristic. These experiments investigated whether the predictive performance was maintained, as the size of the feature subsets was reduced (see Table 5).

As can be seen, the REPTree and J48 algorithms presented a similar performance, 92.77% and 92.76% (ACC), respectively. On the other hand, Random Forest had the worst performance 91.19% (ACC). Given these results, we decided to apply the same algorithm used in [36], REPTree. Furthermore, it was observed that the ML algorithms kept their predictive performance as the number of features was reduced. Thus, the optimal feature subsets selected by the metaheuristics (see Table 4) were applied to induce the predictive models.

D. PERFORMANCE TEST

The predictive models induced by REPTree were applied to the test sets (see Test Set Construction Method), producing the results shown in Table 6. As can be seen, the results obtained using the selected feature subsets were similar. The best predictive performance regarding SE and ACC were obtained by using the feature subset selected by M1-GA (SE: 100% and ACC: 91.29%), followed by M3-ABC (SE: 99.95% and ACC: 91.27%), and M4-ACO (SE: 99.94% and ACC: 91.27%). Regarding specificity, the best methods were M2-EA (82.61%) and M4-ACO (82.61%).

In summary, considering this preliminary experimental analysis, we observed that our decomposing model with rounds and voting scheme led to better solutions, in terms of

TABLE 5. Training accuracy in each execution round of Table 4. The "–" means that the algorithm obtained the same result.

ID	Classifier	R1		R2		R3		R4		R5	
-	-	*ACC*	*ER*	*ACC*	*ER*	*ACC*	*ER*	*ACC*	*ER*	*ACC*	*ER*
M1-GA	*Random Forest*	90.65%	9.35%	90.64%	9.36%	90.94%	9.06%	91.08%	8.92%	-	-
	REPTree	92.75%	7.25%	92.69%	7.31%	92.72%	7.28%	92.74%	7.26%	-	-
	J48	92.46%	7.54%	92.78%	7.22%	92.78%	7.22%	92.78%	7.22%	-	-
M2-EA	*Random Forest*	89.51%	10.49%	90.65%	9.35%	90.50%	9.50%	91.28%	8.72%	-	-
	REPTree	92.77%	7.23%	92.76%	7.24%	92.67%	7.33%	92.77%	7.23%	-	-
	J48	89.50%	10.50%	92.45%	7.55%	92.78%	7.22%	92.78%	7.22%	-	-
M3-ABC	*Random Forest*	90.55%	9.45%	91.16%	8.84%	91.15%	8.85%	-	-	-	-
	REPTree	92.76%	7.24%	92.78%	7.22%	92.77%	7.23%	-	-	-	-
	J48	92.78%	7.22%	92.78%	7.22%	92.78%	7.22%	-	-	-	-
M4-ACO	*Random Forest*	90.62%	9.38%	90.69%	9.31%	91.44%	8.56%	-	-	-	-
	REPTree	92.77%	7.23%	92.72%	7.28%	92.78%	7.22%	-	-	-	-
	J48	91.81%	8.19	92.78%	7.22%	92.78%	7.22%	-	-	-	-
M5-PSO	*Random Forest*	90.65%	9.35%	90.67%	9.33%	91.27%	8.73%	-	-	-	-
	REPTree	92.76%	7.24%	92.74%	7.26%	92.74%	7.26%	-	-	-	-
	J48	92.62%	7.38%	92.78%	7.22%	92.78%	7.22%	-	-	-	-

TABLE 6. Performance of all models applied to the test sets. Each predictive model was induced using the feature subsets selected by the metaheuristics (see Table 4).

Species	Method - ID	TP	FP	TN	FN	SE	SPC	ACC	F1-score	PPV	NPV	MCC
A. trichopoda	M1-GA	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M2-EA	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M3-ABC	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M4-ACO	3822	877	2946	1	99.97	**77.06**	**88.52**	**89.70**	**81.34**	99.97	**79.14**
	M5-PSO	3822	879	2944	1	99.97	77.01	88.49	89.68	81.30	99.97	79.10
B. distachyon	M1-GA	4868	720	4148	0	**100**	85.21	**92.60**	**93.11**	87.12	**100**	**86.16**
	M2-EA	4852	718	4150	16	99.67	85.25	92.46	92.97	87.11	99.62	85.82
	M3-ABC	4863	719	4149	5	99.90	85.23	92.56	93.07	87.12	99.88	86.06
	M4-ACO	4863	720	4148	5	99.90	85.21	92.55	93.06	87.10	99.88	86.04
	M5-PSO	4851	716	4152	17	99.65	**85.29**	92.47	92.98	**87.14**	99.59	85.83
C. sinensis	M1-GA	2292	272	2020	0	**100**	88.13	94.07	94.40	89.39	**100**	88.76
	M2-EA	2292	271	2021	0	**100**	**88.18**	**94.09**	**94.42**	**89.43**	**100**	**88.80**
	M3-ABC	2290	272	2020	2	99.91	88.13	94.02	94.36	89.38	99.90	88.66
	M4-ACO	2291	272	2020	1	99.96	88.13	94.04	94.38	89.39	99.95	88.71
	M5-PSO	2291	272	2020	1	99.96	88.13	94.04	94.38	89.39	99.95	88.71
M. esculenta	M1-GA	3017	405	2612	0	**100**	86.58	93.29	93.71	88.16	**100**	87.37
	M2-EA	3017	404	2613	0	**100**	**86.61**	**93.30**	**93.72**	**88.19**	**100**	**87.40**
	M3-ABC	3017	405	2612	0	**100**	86.58	93.29	93.71	88.16	**100**	87.37
	M4-ACO	3015	405	2612	2	99.93	86.58	93.25	93.68	88.16	99.92	87.29
	M5-PSO	3016	405	2612	1	99.97	86.58	93.27	93.69	88.16	99.96	87.33
R. communis	M1-GA	4080	756	3324	0	**100**	81.47	**90.74**	**91.52**	84.37	**100**	**82.91**
	M2-EA	4078	756	3324	2	99.95	81.47	90.71	91.50	84.36	99.94	82.85
	M3-ABC	4080	756	3324	0	**100**	81.47	**90.74**	**91.52**	84.37	**100**	**82.91**
	M4-ACO	4078	754	3326	2	99.95	**81.52**	**90.74**	**91.52**	**84.40**	99.94	82.89
	M5-PSO	4077	756	3324	3	99.93	81.47	90.70	91.48	84.36	99.91	82.82
S. tuberosum	M1-GA	5607	1352	4255	0	**100**	75.89	**87.94**	**89.24**	80.57	**100**	**78.19**
	M2-EA	5604	1352	4255	3	99.95	75.89	87.92	89.21	80.56	99.93	78.13
	M3-ABC	5607	1352	4255	0	**100**	75.89	**87.94**	**89.24**	80.57	**100**	**78.19**
	M4-ACO	5605	1351	4256	2	99.96	**75.91**	87.93	89.23	**80.58**	99.95	78.17
	M5-PSO	5599	1351	4256	8	99.86	**75.91**	87.88	89.18	80.56	99.81	78.03
S. bicolor	M1-GA	4541	684	3857	0	**100**	84.94	**92.47**	**93.00**	**86.91**	**100**	**85.92**
	M2-EA	4530	683	3858	11	99.76	**84.96**	92.36	92.88	86.90	99.72	85.66
	M3-ABC	4534	684	3857	7	99.85	84.94	92.39	92.92	86.89	99.82	85.74
	M4-ACO	4537	684	3857	4	99.91	84.94	92.42	92.95	86.90	99.90	85.82
	M5-PSO	4529	684	3857	12	99.74	84.94	92.34	92.86	86.88	99.69	85.62
Z. mays	M1-GA	12071	2239	9832	0	**100**	81.45	**90.73**	**91.51**	84.35	**100**	**82.89**
	M2-EA	12060	2234	9837	11	99.91	81.49	90.70	91.48	84.37	99.89	82.82
	M3-ABC	12065	2234	9837	6	99.95	81.49	90.72	**91.51**	**84.38**	99.94	82.87
	M4-ACO	12061	2233	9838	10	99.92	**81.50**	90.71	91.49	**84.38**	99.90	82.84
	M5-PSO	12046	2233	9838	25	99.79	**81.50**	90.65	91.43	84.36	99.75	82.69
Overall Average	**M1-GA**	-	-	-	-	**100**	82.58	**91.29**	**92.02**	85.27	**100**	**83.91**
	M2-EA	-	-	-	-	99.90	**82.61**	91.26	91.99	**85.28**	99.89	83.82
	M3-ABC	-	-	-	-	99.95	82.59	91.27	92.00	85.27	99.94	83.87
	M4-ACO	-	-	-	-	99.94	**82.61**	91.27	92.00	**85.28**	99.93	83.86
	M5-PSO	-	-	-	-	99.86	82.60	91.23	91.96	85.27	99.83	83.77

higher accuracy and lower number of attributes, regardless of the metaheuristic used within the model. In addition, Figure 3 illustrates the ROC curve considering only the best model, i.e., M1-GA.

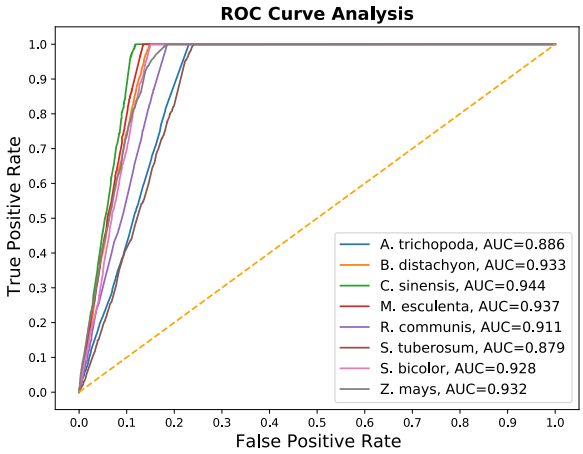


FIGURE 3. ROC curve of all species in the Performance Test, using our best model (M1-GA).

Assessing the individual performance of each metaheuristic for different species, we observed that several models obtained high ACC in six datasets, among them: *C. sinensis* (ACC: 94.09% - M2-EA), *M. esculenta* (ACC: 93.30% - M2-EA), *B. distachyon* (ACC: 92.60% - M1-GA), *S. bicolor* (ACC: 92.47% - M1-GA), *R. communis* (ACC: 90.74% - M1-GA, M3-ABC and M4-ACO), and *Z. mays* (ACC: 90.73% - M1-GA). Regarding the individual sensitivity (to detect lncRNA), we achieved the best results with all species and methods. On the other hand, we reduced the specificity (to detect mRNA), especially in two datasets (*A. trichopoda* and *S. tuberosum*). In addition, we assessed the statistical significance of the metaheuristics with the Friedman's statistical test, as follow:

- Null hypothesis: $(H0 : M(1) = M(2) = \ldots = M(k))$
- Alternative hypothesis (H_A): Metaheuristic has statistical significance $(p < 0.05)$

According to the statistical test, $\chi^2(4) = 14.577$, p-value $= 0.005664$, we can reject $H0$, since $p < 0.01$. Also, Conover statistics p-values are presented in Table 7.

TABLE 7. Conover statistics *p*-values for M1-GA, M2-EA, M3-ABC, M4-ACO, and M5-PSO.

	M1-GA	M2-EA	M3-ABC	M4-ACO
M2-EA	**0.0351**	-	-	-
M3-ABC	0.3460	0.6447	-	-
M4-ACO	0.1986	0.7639	0.7639	-
M5-PSO	**4e-05**	0.0758	**0.0045**	**0.0110**

The Conover statistics p-values show that most metaheuristics do not have a significant difference. For that reason, we can highlight that the proposed decomposition model does not depend on any of the metaheuristics used in the study.

E. DECOMPOSING MODEL COMPARED WITH ALL FEATURES IN THE DATASET

In this section, we compare M1-GA and M5-PSO (the best and the worst models shown in Table 6), with a model without feature selection (see Table 8 - *Full features (5,467)*). In the overall average, our approach represented a gain of 4.68% (M1-GA) and 4.62% (M2-PSO) in the ACC. However, in some species, we achieved an improvement (ACC) of 6.40% (*S. bicolor*), 5.92% (*B. distachyon*), and 4.85% (*C. sinensis*).

TABLE 8. Our approach against all features.

Species	Method - ID	ACC
Amborella trichopoda	All Features	84.51
	M1-GA	**88.50**
	M5-PSO	88.49
Brachypodium distachyon	All Features	86.68
	M1-GA	**92.60**
	M5-PSO	92.47
Citrus sinensis	All Features	89.22
	M1-GA	**94.07**
	M5-PSO	94.04
Manihot esculenta	All Features	88.96
	M1-GA	**93.29**
	M5-PSO	93.27
Ricinus communis	All Features	87.11
	M1-GA	**90.74**
	M5-PSO	90.70
Solanum tuberosum	All Features	83.90
	M1-GA	**87.94**
	M5-PSO	87.88
Sorghum bicolor	All Features	86.07
	M1-GA	**92.47**
	M5-PSO	92.34
Zea mays	All Features	86.40
	M1-GA	**90.73**
	M5-PSO	90.65
Overall Average	**All Features**	86.61
	M1-GA	**91.29**
	M5-PSO	91.23

Furthermore, the results demonstrated that our decomposing model, regardless of the metaheuristic, can be better than a model without feature selection, even considering the worst performance with the PSO algorithm (M5-PSO).

F. EVALUATION WITH OTHER CLASSIFIER TOOLS

Finally, we investigated the best model in the *performance test* (see Table 6), respectively, M1-GA, with five literature pipelines, i.e., the most cited: CPC [21] (6 features), CPC2 [22] (4 features), CNCI [24] (5 features), PLEK [25] (1, 364 features), and RNAplonc [36] (16 features), as shown in Table 9. We randomly chose 5 species for evaluation. In these experiments, our model (M1-GA) had the best performance for ACC (and several metrics) for four species, *A. trichopoda* = 88.50%, *B. distachyon* = 92.60%, *M. esculenta* = 93.29%, and *S. bicolor* = 92.47%.

It is important to mention that RNAplonc reached the best performance (ACC) in *A. trichopoda* (88.50%) and *C. sinensis* (94.13%) with only a difference of 0.02% from our model (in the *C. sinensis*). On the other hand, with M1-GA, we obtained a reduction of 37.5% in the number of

TABLE 9. Comparative performance between M1-GA, CPC, CPC2, CNCI, PLEK, and RNAplonc for five plant species.

Species	Pipeline	TP	FP	TN	FN	SE	SPC	ACC	F1-score	PPV	NPV	MCC
A. trichopoda	M1-GA	3823	879	2944	0	**100**	77.01	**88.50**	**89.69**	**81.31**	**100**	**79.13**
	CPC	3823	1877	1946	0	**100**	50.90	75.45	80.29	67.07	**100**	58.43
	CPC2	2513	618	3205	1310	65.73	**83.83**	74.78	72.27	80.26	70.99	50.40
	CNCI	2665	1171	2651	1158	69.71	69.36	69.54	69.59	69.47	69.60	39.07
	PLEK	3823	2857	966	0	**100**	25.27	62.63	72.80	57.23	**100**	38.03
	RNAplonc	3823	879	2944	0	**100**	77.01	**88.50**	**89.69**	**81.31**	**100**	**79.13**
B. distachyon	M1-GA	4868	720	4148	0	**100**	85.21	**92.60**	**93.11**	87.12	**100**	**86.16**
	CPC	4846	1685	3183	22	99.55	65.39	82.47	85.03	74.20	99.31	69.09
	CPC2	4312	666	4202	556	88.58	86.32	87.45	87.59	86.62	88.31	74.92
	CNCI	2571	426	4442	2297	52.81	**91.25**	72.03	65.38	85.79	65.91	47.73
	PLEK	4082	1449	3419	786	83.85	70.23	77.04	78.51	73.80	81.31	54.60
	RNAplonc	4753	677	4191	115	97.64	86.09	91.87	92.31	**87.53**	97.33	84.29
C. sinensis	M1-GA	2292	272	2020	0	**100**	88.13	94.07	94.40	89.39	**100**	88.76
	CPC	2268	746	1546	24	98.95	67.45	83.20	85.49	75.25	98.47	69.97
	CPC2	1889	231	2061	403	82.42	**89.92**	86.17	85.63	89.10	83.64	72.54
	CNCI	1765	485	1807	527	77.01	78.84	77.92	77.72	78.44	77.42	55.86
	PLEK	2172	827	1465	120	94.76	63.92	79.34	82.10	72.42	92.43	61.69
	RNAplonc	2290	267	2025	2	99.91	88.35	**94.13**	**94.45**	**89.56**	99.90	**88.86**
M. esculenta	M1-GA	3017	405	2612	0	**100**	86.58	**93.29**	**93.71**	88.16	**100**	**87.37**
	CPC	2980	838	2179	37	98.77	72.22	85.50	87.20	78.05	98.33	73.64
	CPC2	2645	332	2685	372	87.67	**89.00**	88.33	88.25	**88.85**	87.83	76.67
	CNCI	2580	786	2231	437	85.52	73.95	79.73	80.84	76.65	83.62	59.86
	PLEK	2849	1153	1864	168	94.43	61.78	78.11	81.18	71.19	91.73	59.47
	RNAplonc	3014	403	2614	3	99.90	86.64	93.27	93.69	88.21	99.89	87.31
S. bicolor	M1-GA	4541	684	3857	0	**100**	84.94	**92.47**	**93.00**	86.91	**100**	**85.92**
	CPC	4511	1481	3060	30	99.34	67.39	83.36	85.65	75.28	99.03	70.42
	CPC2	4025	597	3944	516	88.64	86.85	87.74	87.85	87.08	88.43	75.50
	CNCI	2317	383	4158	2224	51.02	**91.57**	71.29	64.00	85.81	65.15	46.59
	PLEK	3626	1225	3316	915	79.85	73.02	76.44	77.21	74.75	78.37	53.00
	RNAplonc	4375	629	3912	166	96.34	86.15	91.25	91.67	**87.43**	95.93	82.93

features. Concerning SE (to predict lncRNAs), M1-GA was the best alternative in four species, except for *A. trichopoda* (RNAplonc, CPC, PLEK also achieved the best result). Nevertheless, in SPC (to predict mRNA), the best pipeline was CPC2 (*A. trichopoda*, *C. sinensis*, *M. esculenta*) and CNCI (*B. distachyon*, *S. bicolor*). In the overall average, our model had an ACC of 92.19% across all datasets, that is, 0.39%, 10.19%, 7.30%, 18.09%, and 17.48% more than RNAplonc (91.80%), CPC (82.00%), CPC2 (84.89%), CNCI (74.10%), and PLEK (74.71%), respectively. Finally, we have also assessed the statistical significance using the Friedman's statistical test (following same idea in section IV-D), as follow:

- Null hypothesis: $(H0 : P(1) = P(2) = \ldots = P(k))$
- Alternative hypothesis (H_A): Some pipeline has statistical significance ($p < 0.05$).

According to the statistical test, $\chi^2(5) = 23.39$, p-value $= 0.0002842$, we can reject $H0$, since $p < 0.01$. Next, we applied a post-hoc statistical analysis. The Conover statistics p-values are presented in Table 10.

As reported in Table 10, Conover statistics show that M1-GA predictive performance was statistically better ($p < 0.01$) than the CPC, CPC2, CNCI, and PLEK tools. However, there was no statistically significant difference ($p > 0.05$) when compared with the RNAplonc tool. Although in this last comparison, there are no significant differences in the statistical test, our pipeline achieved gains in ACC. Besides, we significantly reduced the features set, using only 10 features with M1-GA, a difference of 6 features, not considering RNAplonc. These results, again, confirm the effectiveness of our approach in removing unnecessary, irrelevant, and redundant predictive features.

TABLE 10. Conover statistics *p*-values for M1-GA, CPC, CPC2, CNCI, PLEK, and RNAplonc.

	M1-GA	CPC	CPC2	CNCI	PLEK
CPC	**7.8e-10**	-	-	-	-
CPC2	**7.2e-08**	0.017	-	-	-
CNCI	**4.0e-14**	1.7e-07	1.6e-09	-	-
PLEK	**9.8e-14**	9.5e-07	6.8e-09	0.395	-
RNAplonc	0.098	1.4e-08	2.4e-06	2.5e-13	6.7e-13

G. LIMITATIONS OF THE PROPOSED APPROACH

In this paper, we have investigated the significance of features in the lncRNA classification task. For that reason, we proposed a new approach based on a decomposing model and metaheuristics. However, we investigated the influence of each feature considering only the frequency metric. On the other hand, we believe in an alternative way to select feature dependencies based on Bayesian network models and their variations. Furthermore, our study focuses on a filter approach to find the smallest subset that contains all information, preserving the interpretability of the model, i.e., selecting features from a dataset independently of any machine learning algorithm. Nevertheless, our proposed method could be applied effectively to other approaches, e.g., wrapper and hybrid.

V. CONCLUSION

Understanding the significance of features in the lncRNA identification is the next challenge in computational

biology [31]. Fundamentally, the large number of features let to a high dimensionality problem, requiring the application of feature selection techniques and the investigation of the influence and contribution of the selected features to the biological sequences classification (e.g., lncRNA). Although previous works have proposed feature selection techniques for similar problems, they select features independently, without taking into account possible dependencies. To deal with this limitation, we proposed a new approach based on a decomposing model, which uses a novel way to identify the best set of attributes by removing unneeded, irrelevant, and redundant features.

To validate the decomposing model, we applied 5 widely studied metaheuristics algorithms (Genetic Algorithm, $(\mu + \lambda)$EA, Artificial Bee Colony, Ant Colony Optimization, and Particle Swarm Optimization) for feature selection and analysis in plant lncRNAs. According to the experimental results, two algorithms selected the minimum number of features, M2-EA and M3-ABC, with 5 features each, followed by M4-ACO (6 features), M5-PSO (7 features), and M1-GA (10 features). Regarding the performance tests, M1-GA reported the best result of SE (100%) and ACC (91.29%), followed by M3-ABC (SE: 99.95% and ACC: 91.27%), and M4-ACO (SE: 99.94% and ACC: 91.27%). In addition, we have evaluated all metaheuristics with Friedman's statistical test, and it was observed that most algorithms do not have a significant difference. For that reason, we can highlight that the proposed decomposition model does not depend on any of the metaheuristics used in the paper.

Also, considering the number of rounds obtained by each algorithm, we realized that metaheuristics without decomposition model (with just one round), it can drop in a stagnation far from an ideal solution. However, when applying the decomposing model, there is a significant improvement in the subset of features, in special with three (or more) rounds. Moreover, these results suggest that our decomposing model for feature selection is an effective way to overcome some limitations of simple metaheuristics, such as premature convergence and poor ability of fine-tuning near local optimum points.

AVAILABILITY OF DATA AND MATERIALS

The tool and datasets generated are available in the Github repository: https://github.com/Bonidia/Feature Selection-FSRV.

ACKNOWLEDGEMENT

The authors would like to thank PPGBIOINFO, UTFPR-CP, USP, CAPES (Finance Code 001) for the financial support given to this research.

REFERENCES

[1] M. Cao, Z. Han, J. Liu, Y. Li, Y. Lv, J. Zhou, and J. He, "Bioinformatic analysis and prediction of the function and regulatory network of long non-coding RNAs in hepatocellular carcinoma," *Oncol. Lett.*, vol. 15, no. 5, pp. 7783–7793, 2018.

[2] M. A. Mehmood, U. Sehar, and N. Ahmad, "Use of bioinformatics tools in different spheres of life sciences," *J. Data Mining Genomics Proteomics*, vol. 5, no. 2, p. 1, 2014.

[3] W. Diniz and F. Canduri, "Bioinformatics: An overview and its applications," *Genet. Mol. Res.*, vol. 16, no. 1, pp. 1–21, 2017.

[4] A. Li, Q. Zang, D. Sun, and M. Wang, "A text feature-based approach for literature mining of lncRNA–protein interactions," *Neurocomputing*, vol. 206, pp. 73–80, Sep. 2016.

[5] Y. Wang, Y. Li, Q. Wang, Y. Lv, S. Wang, X. Chen, X. Yu, W. Jiang, and X. Li, "Computational identification of human long intergenic non-coding RNAs using a GA–SVM algorithm," *Gene*, vol. 533, no. 1, pp. 94–99, Jan. 2014.

[6] P. Ping, L. Wang, L. Kuang, S. Ye, M. F. B. Iqbal, and T. Pei, "A novel method for LncRNA-disease association prediction based on an lncRNA-disease association network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 2, pp. 688–693, Mar. 2019.

[7] W. Zhang, Q. Qu, Y. Zhang, and W. Wang, "The linear neighborhood propagation method for predicting long non-coding RNA–protein interactions," *Neurocomputing*, vol. 273, pp. 526–534, Jan. 2018.

[8] Q.-Z. Zhou, B. Zhang, Q.-Y. Yu, and Z. Zhang, "BmncRNAdb: A comprehensive database of non-coding RNAs in the silkworm, Bombyx Mori," *BMC Bioinf.*, vol. 17, no. 1, p. 370, Dec. 2016.

[9] M. Q. Hassan, C. E. Tye, G. S. Stein, and J. B. Lian, "Non-coding RNAs: Epigenetic regulators of bone development and homeostasis," *Bone*, vol. 81, pp. 746–756, Dec. 2015.

[10] C. Ciaudo, N. Servant, V. Cognat, A. Sarazin, E. Kieffer, S. Viville, V. Colot, E. Barillot, E. Heard, and O. Voinnet, "Highly dynamic and sex-specific expression of microRNAs during early ES cell differentiation," *PLoS Genet.*, vol. 5, no. 8, Aug. 2009, Art. no. e1000620.

[11] C. Pastori and C. Wahlestedt, "Involvement of long noncoding RNAs in diseases affecting the central nervous system," *RNA Biol.*, vol. 9, no. 6, pp. 860–870, Jun. 2012.

[12] Q. Zhang, Y. Wei, Z. Yan, C. Wu, Z. Chang, Y. Zhu, K. Li, and Y. Xu, "The characteristic landscape of lncRNAs classified by RBP–lncRNA interactions across 10 cancers," *Mol. BioSyst.*, vol. 13, no. 6, pp. 1142–1151, 2017.

[13] V. Hsiao-Lin Wang and A. Julia Chekanova, "Long noncoding RNAs in plants," in *Long Non Coding RNA Biol.*, M. R. S. Rao, ed. Singapore: Springer, 2017, pp. 133–154.

[14] C. Di, J. Yuan, Y. Wu, J. Li, H. Lin, L. Hu, T. Zhang, Y. Qi, M. B. Gerstein, Y. Guo, and Z. J. Lu, "Characterization of stress-responsive lncRNAs in Arabidopsis thalianaby integrating expression, epigenetic and structural features," *Plant J.*, vol. 80, no. 5, pp. 848–861, Dec. 2014.

[15] D. Wang, Z. Qu, L. Yang, Q. Zhang, Z.-H. Liu, T. Do, D. L. Adelson, Z.-Y. Wang, I. Searle, and J.-K. Zhu, "Transposable elements (TEs) contribute to stress-related long intergenic noncoding RNAs in plants," *Plant J.*, vol. 90, no. 1, pp. 133–146, Apr. 2017.

[16] Y.-C. Zhang, J.-Y. Liao, Z.-Y. Li, Y. Yu, J.-P. Zhang, Q.-F. Li, L.-H. Qu, W.-S. Shu, and Y.-Q. Chen, "Genome-wide screening and functional analysis identify a large number of long noncoding RNAs involved in the sexual reproduction of rice," *Genome Biol.*, vol. 15, no. 12, p. 512, Dec. 2014.

[17] L. Ma, V. B. Bajic, and Z. Zhang, "On the classification of long non-coding rnas," *RNA Biol.*, vol. 10, no. 6, pp. 924–933, 2013.

[18] R. Hu and X. Sun, "LncRNATargets: A platform for lncRNA target prediction based on nucleic acid thermodynamics," *J. Bioinf. Comput. Biol.*, vol. 14, no. 4, Aug. 2016, Art. no. 1650016.

[19] M. Zitnik, F. Nguyen, B. Wang, J. Leskovec, A. Goldenberg, and M. M. Hoffman, "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities," *Inf. Fusion*, vol. 50, pp. 71–91, Oct. 2019.

[20] H. K. Kwan and S. B. Arniker, "Numerical representation of DNA sequences," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, Jun. 2009, pp. 307–310.

[21] L. Kong, Y. Zhang, Z.-Q. Ye, X.-Q. Liu, S.-Q. Zhao, L. Wei, and G. Gao, "CPC: Assess the protein-coding potential of transcripts using sequence features and support vector machine," *Nucleic Acids Res.*, vol. 35, no. 2, pp. W345–W349, Jul. 2007.

[22] Y.-J. Kang, D.-C. Yang, L. Kong, M. Hou, Y.-Q. Meng, L. Wei, and G. Gao, "CPC2: A fast and accurate coding potential calculator based on sequence intrinsic features," *Nucleic acids Res.*, vol. 45, no. W1, pp. W12–W16, 2017.

[23] L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher, and W. Li, "CPAT: Coding-potential assessment tool using an alignment-free logistic regression model," *Nucleic Acids Res.*, vol. 41, no. 6, p. e74, Apr. 2013.

[24] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, and Y. Zhao, "Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts," *Nucleic Acids Res.*, vol. 41, no. 17, p. e166, Sep. 2013.

[25] A. Li, J. Zhang, and Z. Zhou, "PLEK: A tool for predicting long non-coding RNAs and messenger RNAs based on an improved K-MER scheme," *BMC Bioinf.*, vol. 15, no. 1, p. 311, 2014.

[26] X.-N. Fan and S.-W. Zhang, "LncRNA-MFDL: Identification of human long non-coding RNAs by fusing multiple features and using deep learning," *Mol. BioSyst.*, vol. 11, no. 3, pp. 892–897, 2015.

[27] R. Achawanantakun, J. Chen, Y. Sun, and Y. Zhang, "LNCRNA-ID: Long non-coding rna identification using balanced random forests," *Bioinformatics*, vol. 31, no. 24, pp. 3897–3905, 2015.

[28] L. Sun, H. Liu, L. Zhang, and J. Meng, "LncRScan-SVM: A tool for predicting long non-coding RNAs using support vector machine," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0139654.

[29] C. Pian, G. Zhang, Z. Chen, Y. Chen, J. Zhang, T. Yang, and L. Zhang, "LncRNApred: Classification of long non-coding RNAs and protein-coding transcripts by the ensemble algorithm with a new hybrid feature," *PLoS ONE*, vol. 11, no. 5, May 2016, Art. no. e0154567.

[30] R. Tripathi, S. Patel, V. Kumari, P. Chakraborty, and P. K. Varadwaj, "DeepLNC, a long non-coding RNA prediction tool using deep neural network," *Netw. Model. Anal. Health Informat. Bioinf.*, vol. 5, no. 1, p. 21, Dec. 2016.

[31] S.-W. Choi and J.-W. Nam, "TERIUS: Accurate prediction of LNCRNA via high-throughput sequencing data representing RNA-binding protein association," *BMC Bioinf.*, vol. 19, no. S1, p. 41, Feb. 2018.

[32] E. A. Ito, I. Katahira, F. F. D. R. Vicente, L. F. P. Pereira, and F. M. Lopes, "BASiNET—BiologicAl sequences NETwork: A case study on coding and non-coding RNAs identification," *Nucleic acids Res.*, vol. 46, no. 16, p. e96, 2018.

[33] S. Han, Y. Liang, Q. Ma, Y. Xu, Y. Zhang, W. Du, C. Wang, and Y. Li, "LncFinder: An integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property," *Briefings Bioinf.*, vol. 20, no. 6, pp. 2009–2027, Nov. 2019.

[34] U. Singh, N. Khemka, M. S. Rajkumar, R. Garg, and M. Jain, "PLncPRO for prediction of long non-coding RNAs (lncRNAs) in plants and its application for discovery of abiotic stress-responsive lncRNAs in rice and chickpea," *Nucleic Acids Res.*, vol. 45, no. 22, p. e183, Dec. 2017.

[35] L. Vieira, C. Grativol, F. Thiebaut, T. Carvalho, P. Hardoim, A. Hemerly, S. Lifschitz, P. Ferreira, and M. Walter, "PlantRNA_Sniffer: A SVM-based workflow to predict long intergenic non-coding RNAs in plants," *Non-Coding RNA*, vol. 3, no. 1, p. 11, Mar. 2017.

[36] T. D. C. Negri, W. A. L. Alves, P. H. Bugatti, P. T. M. Saito, D. S. Domingues, and A. R. Paschoal, "Pattern recognition analysis on long noncoding RNAs: A tool for prediction in plants," *Briefings Bioinf.*, vol. 20, no. 2, pp. 682–689, Mar. 2019.

[37] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Aug. 2004.

[38] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[39] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[40] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007.

[41] S. Lertampaiporn, C. Thammarongtham, C. Nukoolkit, B. Kaewkamnerdpong, and M. Ruengjitchatchawalya, "Identification of non-coding RNAs with a new composite feature in the hybrid random forest ensemble algorithm," *Nucleic Acids Res.*, vol. 42, no. 11, p. e93, Jun. 2014.

[42] B. Ma and Y. Xia, "A tribe competition-based genetic algorithm for feature selection in pattern classification," *Appl. Soft Comput.*, vol. 58, pp. 328–338, Sep. 2017.

[43] Z. Tao, L. Huiling, W. Wenwen, and Y. Xia, "GA-SVM based feature selection and parameter optimization in hospitalization expense modeling," *Appl. Soft Comput.*, vol. 75, pp. 323–332, Feb. 2019.

[44] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, A. K. Abasi, and S. N. Makhadmeh, "EEG signals denoising using optimal wavelet transform hybridized with efficient Metaheuristic methods," *IEEE Access*, vol. 8, pp. 10584–10605, 2020.

[45] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, A. K. Abasi, and S. N. Makhadmeh, "EEG signal denoising using hybridizing method between wavelet transform with genetic algorithm," in *Proc. 11th Nat. Tech. Seminar Unmanned Syst. Technol.* in Lecture Notes in Electrical Engineering, vol. 666, Z. Md Zain *et al.* Singapore: Springer, 2019, doi: 10.1007/978-981-15-5281-6_31.

[46] Y. Kim, W. N. Street, and F. Menczer, "Feature selection in unsupervised learning via evolutionary search," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 365–369.

[47] R. P. Bonidia, A. C. P. de Leon Ferreira de Carvalho, A. R. Paschoal, and D. S. Sanches, "Selecting the most relevant features for the identification of long non-coding RNAs in plants," in *Proc. 8th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2019, pp. 539–544.

[48] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," in *Proc. Neural Comput. Appl.*, 2020, pp. 1–17.

[49] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2012, pp. 291–297.

[50] X. Yang and X. He, "Bat algorithm: Literature review and applications," *Int. J. Bio-Inspir. Com.*, vol. 5, no. 3, pp. 141–149, 2013.

[51] D. Gupta, U. Agrawal, J. Arora, and A. Khanna, "Bat-inspired algorithm for feature selection and white blood cell classification," in *Nature-Inspired Computation and Swarm Intelligence*, X.-S. Yang, Ed. New York, NY, USA: Academic, 2020, ch. 11, pp. 179–197.

[52] F. Liu, X. Yan, and Y. Lu, "Feature selection for image steganalysis using binary bat algorithm," *IEEE Access*, vol. 8, pp. 4244–4249, 2020.

[53] M. Schiezaro and H. Pedrini, "Data feature selection based on artificial bee colony algorithm," *EURASIP J. Image Video Process.*, vol. 2013, no. 1, p. 47, Dec. 2013.

[54] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, Jun. 2014.

[55] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, and L. Gu, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Appl. Soft Comput.*, vol. 74, pp. 634–642, Jan. 2019.

[56] X. Chu, S. Li, W. Mao, W. Zhao, and L. Huang, "A binary superior tracking artificial bee colony for feature selection," in *Neural Computing for Advanced Applications—NCAA* (Communications in Computer and Information Science), vol. 1265, H. Zhang, Z. Zhang, Z. Wu, and T. Hao, Eds. Singapore: Springer, 2020, doi: 10.1007/978-981-15-7670-6_25.

[57] V.-E. Neagoe and E.-C. Neghina, "Feature selection with ant colony optimization and its applications for pattern recognition in space imagery," in *Proc. Int. Conf. Commun. (COMM)*, Jun. 2016, pp. 101–104.

[58] S. B. V. Sara and K. Kalaiselvi, "Ant colony optimization (ACO) based feature selection and extreme learning machine (ELM) for chronic kidney disease detection," *Int. J. Adv. Stud. Sci. Res.*, vol. 4, no. 1, pp. 474–481, 2019.

[59] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[60] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, Z. A. A. Alyasseri, and A. K. Abasi, "Particle swarm optimization algorithm for power scheduling problem using smart battery," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 672–677.

[61] B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, and Y. Xiao, "Bio-inspired feature selection: An improved binary particle swarm optimization approach," *IEEE Access*, vol. 8, pp. 85989–86002, 2020.

[62] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, early access, Jan. 22, 2020, doi: 10.1109/TEVC.2020.2968743.

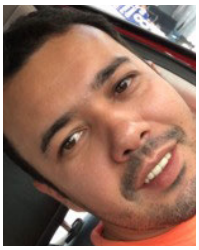
[63] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.

[64] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
[65] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Syst. Appl.*, vol. 137, pp. 46–58, Dec. 2019.
[66] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Inf. Sci.*, vols. 418–419, pp. 561–574, Dec. 2017.
[67] P. Shrivastava, A. Shukla, P. Vepakomma, N. Bhansali, and K. Verma, "A survey of nature-inspired algorithms for feature selection to identify Parkinson's disease," *Comput. Methods Programs Biomed.*, vol. 139, pp. 171–179, Feb. 2017.
[68] B. de la Iglesia, "Evolutionary computation for feature selection in classification problems," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 3, no. 6, pp. 381–407, Nov. 2013.
[69] V. R. Balasaraswathi, M. Sugumaran, and Y. Hamid, "Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms," *J. Commun. Inf. Netw.*, vol. 2, no. 4, pp. 107–119, Dec. 2017.
[70] R. L. Rardin and R. Uzsoy, "Experimental evaluation of heuristic optimization algorithms: A tutorial," *J. Heuristics*, vol. 7, no. 3, pp. 261–304, 2001.
[71] R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér, "Consistent feature selection for pattern recognition in polynomial time," *J. Mach. Learn. Res.*, vol. 8, pp. 589–612, Mar. 2007.
[72] H. Xuan, L. Zhang, X. Liu, G. Han, J. Li, X. Li, A. Liu, M. Liao, and S. Zhang, "PLNlncRbase: A resource for experimentally identified lncRNAs in plants," *Gene*, vol. 573, no. 2, pp. 328–332, Dec. 2015.
[73] A. P. Gallart, A. H. Pulido, I. An. Martínez de Lagrán, W. Sanseverino, and R. A. Cigliano, "GREENC: A wiki-based database of plant lncRNAs," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1161–D1166, Jan. 2016.
[74] D. M. Goodstein, S. Shu, R. Howson, R. Neupane, R. D. Hayes, J. Fazo, T. Mitros, W. Dirks, U. Hellsten, N. Putnam, and D. S. Rokhsar, "Phytozome: A comparative platform for green plant genomics," *Nucleic Acids Res.*, vol. 40, no. D1, pp. D1178–D1186, Jan. 2012.
[75] W. Li and A. Godzik, "Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.
[76] Z.-D. Su, Y. Huang, Z.-Y. Zhang, Y.-W. Zhao, D. Wang, W. Chen, K.-C. Chou, and H. Lin, "ILoc-lncRNA: Predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC," *Bioinformatics*, vol. 34, no. 24, pp. 4196–4204, 2018.
[77] F. Niazi and S. Valadkhan, "Computational analysis of functional long noncoding RNAs reveals lack of peptide-coding capacity and parallels with 3' UTRs," *RNA*, vol. 18, no. 4, pp. 825–843, Apr. 2012.
[78] V. Wucher *et al.*, "FEELnc: A tool for long non-coding RNA annotation and its application to the dog transcriptome," *Nucleic Acids Res.*, vol. 45, no. 8, pp. e57–e57, Jan. 2017. [Online]. Available: https://doi.org/10.1093/nar/gkw1306, doi: 10.1093/nar/gkw1306.
[79] M. C. Frith, T. L. Bailey, T. Kasukawa, F. Mignone, S. K. Kummerfeld, M. Madera, S. Sunkara, M. Furuno, C. J. Bult, J. Quackenbush, C. Kai, J. Kawai, P. Carninci, Y. Hayashizaki, G. Pesole, and J. S. Mattick, "Discrimination of Non-Protein-Coding transcripts from protein-coding mRNA," *RNA Biol.*, vol. 3, no. 1, pp. 40–48, Jan. 2006.
[80] J. Baek, B. Lee, S. Kwon, and S. Yoon, "LNCRNANET: Long noncoding RNA identification using deep learning," *Bioinformatics*, vol. 34, no. 22, pp. 3889–3897, 2018.
[81] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler, "The human genome browser at UCSC," *Genome Res.*, vol. 12, no. 6, pp. 996–1006, Jun. 2002.
[82] K. Sun, X. Chen, P. Jiang, X. Song, H. Wang, and H. Sun, "ISeeRNA: Identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data," *BMC Genomics*, vol. 14, no. 2, pp. S7, 2013.
[83] X.-Q. Li, Z.-X. Ren, K. Li, J.-J. Huang, Z.-T. Huang, T.-R. Zhou, H.-Y. Cao, F.-X. Zhang, and B. Tan, "Key anti-fibrosis associated long noncoding RNAs identified in human hepatic stellate cell via transcriptome sequencing analysis," *Int. J. Mol. Sci.*, vol. 19, no. 3, p. 675, Feb. 2018.
[84] M. C. P. de Souto, D. S. A. de Araujo, I. G. Costa, R. G. F. Soares, T. B. Ludermir, and A. Schliep, "Comparative study on normalization procedures for cluster analysis of gene expression datasets," in *Proc. IEEE Int. Joint Conf. Neural Netw., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 2792–2798.
[85] U. Stańczyk, "Feature evaluation by filter, wrapper, and embedded approaches," in *Feature Selection for Data and Pattern Recognition*, U. Stańczyk and L. C. Jain, Eds. Berlin, Germany: Springer, 2015, pp. 29–44.
[86] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell.*, vol. 151, nos. 1–2, pp. 155–176, 2003.
[87] P. Krızek, "Feature selection: Stability, algorithms, and evaluation," Ph.D. dissertation, Czech Tech. Univ. Prague, Prague, Czechia, 2008.
[88] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
[89] T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff, "Embedded methods," in *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds. Berlin, Germany: Springer, 2006, pp. 137–165. Berlin Heidelberg.
[90] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NJ, USA: Wiley, 2012.
[91] E. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley, 1989.
[92] D. T. Pham and M. Castellani, "The bees algorithm: Modelling foraging behaviour to solve continuous optimization problems," *Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci.*, vol. 223, no. 12, pp. 2919–2938, Dec. 2009.
[93] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[94] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning*, vol. 1, 1st ed. Boston, MA, USA: Springer, 2011, pp. 36–39.
[95] A. Moraglio, C. D. Chio, J. Togelius, and R. Poli, "Geometric particle swarm optimization," *J. Artif. Evol. Appl.*, vol. 2008, p. 14, Jan. 2008.
[96] J. Kennedy, "Swarm intelligence," in *Handbook of Nature-Inspired and Innovative Computing*. Boston, MA, USA: Springer, 2006, pp. 187–219.
[97] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, vol. 1, 1st ed. Boston, MA, USA: Springer, 2011, pp. 760–766.
[98] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1999.
[99] K. Selvakuberan, M. Indradevi, and R. Rajaram, "Combined feature selection and classification—A novel approach for the categorization of Web pages," *J. Inf. Comput. Sci.*, vol. 3, no. 2, pp. 83–89, 2008.
[100] S. Fong, R. P. Biuk-Aghai, and R. C. Millham, "Swarm search methods in Weka for data mining," in *Proc. 10th Int. Conf. Mach. Learn. Comput.*, 2018, pp. 122–127.
[101] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

ROBSON P. BONIDIA received the M.Sc. degree in bioinformatics from the Federal University of Technology—Paraná (UTFPR), Brazil. He is currently pursuing the Ph.D. degree in computer science and computational mathematics with the University of São Paulo—USP. His main research topics are in computational biology and pattern recognition, feature extraction and selection, metaheuristics, and sports data mining.

JAQUELINE SAYURI MACHIDA studies control and automation engineering with the Federal University of Technology—Paraná (UTFPR), where she developed a scientific initiation research about machine learning, feature selection, and metaheuristics.

TATIANNE C. NEGRI received the master's degree in bioinformatics (PPGBIOINFO) from the Federal University of Technology—Paraná (UTFPR), where she developed the RNAplonc project. She is currently pursuing the Ph.D. degree in the informatics and knowledge management graduate program with UNINOVE. Her research is in deep learning model in bioinformatics problems.



WONDER A. L. ALVES (Member, IEEE) is currently a Professor in the informatics and knowledge management graduate program with UNINOVE, with the focus on machine learning and image analysis.

ANDRÉ Y. KASHIWABARA received the Ph.D. degree in computer science from the Universidade de São Paulo. He is currently an Adjunct Professor with the Department of Computer Science, Federal University of Technology—Paraná (UTFPR), Brazil. His research focuses on the development of machine learning approaches for genomics and transcriptomics.

DOUGLAS S. DOMINGUES graduated in biology from the São Paulo State University, Botucatu, Brazil, in 2003, and received the Ph.D. degree in biotechnology from the University of São Paulo, Brazil, in 2009. He is currently a Research Professor of Plant Gene Expression with the Department of Biodiversity, São Paulo State University, Rio Claro, Brazil, where he is in charge of the Genomics and Transcriptomics in Plants Group. He is the Head of the Ph.D. in plant biology with the São Paulo State University. In his research, he uses genomics and transcriptomics approaches in non-model plants to understand gene function, the evolution of gene families and genome components, as well as molecular responses to environmental constraints.

ANDRÉ DE CARVALHO (Member, IEEE) was an Associate Professor with the University of Guelph, Canada, a Visiting Professor with the University of Kent, U.K., and a Visiting Researcher with the University of Porto, Portugal, and the Alan Turing Institute, U.K. His main research areas are data mining, data science, and machine learning. He is the Founding Director of the Center of Machine Learning in Data Analysis, University of São Paulo (USP), Brazil, where he is also the Vice Dean of the Mathematics and Computer Science Institute of University of São Paulo, ICMC-USP, the Vice Director of the Center for Mathematical Sciences Applied to Industry, USP, and the Vice President of the Brazilian Computer Society, SBC. He is currently a Full Professor with the Department of Computer Science, USP. He is a member of the Technical and Scientific Council of the EMAp-FGV, the Latin America and Caribe Chapter Board of the International Network for Government Science Advice (INGSA), and the Science for Education Brazilian Network of the University Council of the USP and the Data Science and Engineering Consortium. He is also a partner in the City University's Data Science Institute and a member of the Strategy and Partnerships Board of the UKRI Centre for Doctoral Training in Accountable, Responsible, and Transparent AI (ART-AI), University of Bath, U.K.

ALEXANDRE R. PASCHOAL received the Ph.D. degree in bioinformatics from the University of São Paulo, in 2012. He is currently an Associate Professor with the Computer Science Department, Federal University of Technology - Paraná (UTFPR), Brazil. His research includes data mining, databases, bioinformatics, and pattern recognition approaches to model and understand biological data.

DANILO S. SANCHES received the Ph.D. degree in electrical engineering from the University of Sao Paulo, in 2013. He is currently an Associate Professor with the Computer Science Department, Federal University of Technology - Paraná (UTFPR), Brazil. His research includes data mining, machine learning, evolutionary algorithms, bioinformatics, and pattern recognition approaches.