Predicting novel microRNA: a comprehensive comparison of machine learning approaches

G. Stegmayer*, L. Di Persia, M. Rubiolo, M. Gerard, M. Pividori, C. Yones, L. Bugnon, T. Rodriguez, J. Raad and D.H. Milone

Abstract

Motivation: The importance of microRNAs (miRNAs) is widely recognized in the community nowadays because these short segments of RNA can play several roles in almost all biological processes. The computational prediction of novel miRNAs involves training a classifier for identifying sequences having the highest chance of being miRNA precursors (pre-miRNAs). The big issue with this task is that well-known pre-miRNAs are usually very few in comparison to the hundreds of thousands of candidates sequences in a genome, which results in high class imbalance. This imbalance has a strong influence on most standard classifiers, and if not properly addressed in the model and the experiments, not only performance reported can be completely unrealistic, but also the classifier will not be able to work properly for pre-miRNA prediction. Besides, another important issue is that for most of the machine learning approaches already employed (supervised methods) it is necessary to have both positive and negative examples. The selection of positive examples is straightforward (well-known pre-miRNAs). However, it is very difficult to build a representative set of negative examples because they should be sequences with hairpin structure that do not actually contain a pre-miRNA.
Results: This review provides a comprehensive study and comparative assessment of methods from these two machine learning (ML) approaches for dealing with the prediction of novel pre-miRNAs: supervised and unsupervised training. We present and analyze the machine learning proposals that have appeared during the last 10 years in literature. They have been compared in several prediction tasks involving two model genomes and increasing imbalance levels. This work provides a review of existing ML approaches for pre-miRNA prediction and fair comparisons of the classifiers with same features and data sets, instead of just a revision of published software tools. The results and the discussion can help the community to select the most adequate bioinformatics approach according to the prediction task at hand. The comparative results obtained suggest that from low to mid imbalance levels between classes, supervised methods can be the best. However, at very high imbalance levels, closer to real case scenarios, models including unsupervised and deep learning can provide better performance.
Availability: http://sourceforge.net/projects/sourcesinc/files/ml4mirna/
Supplementary information: Supplementary data are available at *Briefings in Bioinformatics* online.

1 INTRODUCTION

MicroRNAs (miRNAs) are a special type of non-coding RNA of 21 nucleotides in length, which function in the post-transcriptional regulation of gene expression. Since their discovery, and without any doubt, they have reshaped the community appreciation on gene regulation. They may determine the genetic expression of cells and influence the state of the tissues [1]. Therefore, discovering new miRNAs, identifying their targets and further inferring their functions are necessary tasks for understanding miRNAs and their roles in genes regulation. Given their important role in promoting or inhibiting certain diseases and infections, the discovery of new miRNAs is of high interest today [2, 3], for example for developing biomarkers and targeted drug delivery [4, 5]. Precursors of miRNAs generated during biogenesis have a well-known RNA secondary structure, which has allowed the development of computational algorithms for their identification. They are named pre-miRNAs and are also known as hairpins. The pre-miRNAs typically exhibit a stem-loop structure with few internal loops or asymmetric bulges. However, a large amount of hairpin-like structures can be found in a genome. Due to the difficulty in systematically detecting pre-miRNAs by existing experimental techniques, which are inefficient and costly, deep sequencing [6] and computational based methods have played an increasingly important role for their prediction [7, 8]. Indeed, in the last decade many different approaches have appeared for the computational prediction of pre-miRNAs: homologous search, comparative genomics and machine learning. Although the first two kinds of methods can accurately identify miRNAs, they cannot identify those non-homologous or species-specific miRNAs, because they depend mainly on sequence conservation among multiple (possibly related) species.

Under the machine learning (ML) category, there is a large number of methods that use only RNA sequences as input data [7, 9, 10]. That is to say, the RNA is cut and to represent each sequence, features are extracted, among which sequence length can be included. Predictions are based on the inherent characteristics of the sequences and secondary structure of these types of molecules, to identify hairpin structures in non-coding and non-repetitive regions of a genome. Well-known pre-miRNAs, such as those included in miRBase [11], are used during the training process as positive samples. To date, more than 25,000 mature miRNAs have been reported in 193 species (miRBase, release 19).

Most of the published approaches deal with positive class and negative class data [12–19]. In these works, in order to train supervised classifiers and measure sensitivity and specificity in a cross-validation scheme, a reduced subset of negative examples must be artificially defined, with a pre-defined class imbalance. A set of sequences

with hairpin structures that do not contain miRNAs, for example some mRNAs, tRNAs, and rRNAs, are generally used as negative training set. These sets are used to train a predictor that should distinguish between presumed false and true pre-miRNA sequences. Main strategies used to build a negative set are: under-sampling the large set of unknown sequences; pseudo hairpins artificially created [20]; or randomly generating sequences with the same length than the positive set [21]. However, how to accurately distinguish between true de-novo pre-miRNAs and negative cases still remains an important challenge, and a careful choice of the negative dataset is crucial for supervised methods, in order to produce good and reliable predictions [10, 13]. A very small number of methods identify pre-miRNAs in an unsupervised fashion [22–26]. Basically, these methods apply clustering and then use the labels of the well-known pre-miRNAs, only after training, and just to select the clusters where look for novel pre-miRNAs.They obtain a high number of initial candidates, in the order of hundreds of thousands or tens of thousands sequences. After that, a reduced list of best candidates can be selected by applying ad-hoc rules in order to achieve a number of sequences that can be validated experimentally.

A very recent study has shown [27] that the computational prediction of pre-miRNAs is yet far-away from being satisfactory solved. The main reason is that, in spite of several existing reviews [28–31] and all the comparative works already published in the area, those are mostly centered in the revision of available prediction software or web servers. With our work, instead, we want to provide to the bioinformatics community a much more fundamental and conceptual review of existing computational ML approaches for pre-miRNA prediction. It has not been defined yet the most suitable machine learning approach to be applied for prediction. In other words, which type of learning paradigm should be applied in order to really tackle the true issue beneath the prediction in genome-wide data: the very large class imbalance.

In this review we compare and discuss the supervised and unsupervised ML paradigms, in deep, when dealing with the high class imbalance and the lack of definition for the negative set in pre-miRNA prediction. We will explain and analyze each method, providing also practical comparative results. In this comprehensive study we provide detailed analysis, in the same exactly experimental conditions and for a wide range of possible class imbalances, in order to provide a useful guide for the bioinformatics community regarding future software development for in-silico prediction of novel pre-miRNAs. The strong points of the comprehensive comparison we offer are the following: i) a methodologically rigorous and fair comparative evaluation of the most important ML approaches; ii) a deep analysis of the behavior and robustness of each ML algorithm in front of increasing class imbalance levels; iii) and to strongly support the comparative analysis from a rigorous experimental methodology, the ML methods were compared by using the same computer language and exactly the same data sets, with the same varying levels of imbalance and cross validation. All source code and datasets are available for full reproducibility.

2 MACHINE LEARNING APPROACHES

The first ML methods proposed for miRNA prediction have used simple representations to extract the main structural features of known pre-miRNAs [7]. Then, a binary classifier is trained in order to identify other sequences highly likely to be miRNA precursors. Among all possible supervised classifiers, support vector machines have been the first and most widely applied algorithm for this task [12], followed by random forest, k nearest neighbor and naive bayes (see Table 1).

A classical supervised approach needs both positive (real well-known pre-miRNA) and negative (pseudo and artificial non-pre-miRNA) sequences in order to build a binary classifier for discriminating between them. Thus, in supervised learning, the labels of the two-classes must be all known beforehand. Let be m training samples as n-dimensional vectors $\boldsymbol{x}_i = [x_{i1}, ..., x_{in}]^T$ such that $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}; \quad i = 1, \ldots, m$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$, are the response variables. Given a set of points each of which belongs to one of two possible classes, a supervised algorithm constructs a model capable of predicting whether a new point (whose class is previously unknown) belongs to one class or the other.

The k nearest neighbor (KNN) is a method that stores all the training examples as the classification model [32], without actually building a parametric model. It is the simplest classifier and an example of a lazy learner, in which all computation occurs at classification time (without training). It does not have to fit a model to the data. The probability that a point $p(\boldsymbol{x})$ falls within a volume V centered at point $\boldsymbol{x}$ is given by $\pi = \int_V p(\boldsymbol{x}) d\boldsymbol{x}$, where the integral is over the volume V. For a small volume $\pi \sim p(\boldsymbol{x})V$, the probability π may be approximated by the proportion of samples falling within V. If k is the number of samples falling within V, out of a total of m, $\pi \sim k/m$, thus $p(\boldsymbol{x}) \sim k/mV$. From a bayesian point of view we are interested in obtaining the posterior $p(y_j|\boldsymbol{x})$ for each class j. This is done by growing a region around a point $\boldsymbol{x}$ until it includes k neighbors, with a volume V, and within the k neighbors there are k_j samples from class y_j. The joint probability will be $p(\boldsymbol{x}, y_j) = k_j/mV$, with m the number of data points, and thus the posterior will be $p(y_j|\boldsymbol{x}) = p(\boldsymbol{x}, y_j)/\sum_i p(\boldsymbol{x}, y_i) = k_j/k$. For a given test example $\boldsymbol{z}$, the decision rule is to assign $\boldsymbol{z}$ to the class that has maximum a posteriori probability, $y(\boldsymbol{z}) = \arg\max_j \{p(y_j(\boldsymbol{z}))\}$, which is the same as the one that receives the largest votes k_j among the k nearest neighbors of $\boldsymbol{z}$ [33]. Standard KNN has not actually been used for pre-miRNAs prediction. In [25] the authors have seen that known precursors tend to concentrate in a particular region of the feature space. Thus, they proposed to take the coordinates of all known precursors in order to set a distance range to identify the closest candidates, instead of assigning a class label according to the k nearest neighbors. This allows different pre-miRNA structural clusters to emerge around the known precursors. The number of candidates that are included in the acceptance region is controlled by the maximum distance allowed to the closest pre-miRNA (and not by the k parameter as in standard KNN).

Naive Bayes (NB) classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem [32, 34] with strong assumptions of independence between the features. It calculates the probability that a given example belongs to a certain class, making the simplifying assumption that the features constituting the instance are conditionally independent given the class. Given an example $\boldsymbol{x}$, one looks for a class y_j that maximizes the

likelihood $p(\boldsymbol{x}|y_j) = p(x_1, ..., x_n|y_j)$. The (naive) assumption of conditional independence among the features, given the class, allows to express this conditional probability $p(\boldsymbol{x}|y_j)$ as a product of simpler probabilities $p(\boldsymbol{x}|y_j) = \prod_{i=1}^{n} p(x_i|y_j)$. In this way, a NB classifier is the function that assigns to an unknown input $\boldsymbol{z}$, a class label $y(\boldsymbol{z}) = \arg\max_j \{p(y_j) \prod_{i=1}^{n} p(z_i|y_j)\}$, where the posterior is proportional to product of the prior $p(y_j)$ and the conditional probability $p(\boldsymbol{z}|y_j)$. A standard NB classifier has been used in [35] for automatically generating a model from sequence and structure information from a variety of species. This model, together with a balanced distribution of the data, has helped to reduce the false positive rate. Another work [17] has also used a classical NB classifier to identify the location and starting position of human mature miRNAs candidates based on sequence and secondary structure information of miRNA precursors.

Support vector machines (SVMs) are binary classifiers originally proposed by Vapnik [36]. SVMs can efficiently perform classification by using the so-called kernel trick, implicitly mapping the inputs into high dimensional feature spaces. SVM separates the classes in the training data by looking for the optimal separating hyper-plane with a maximal margin between the class $+1$ and the class -1 samples. For this two-classes problem, assumed as linearly separable in a mapped domain, a linear machine can be used as $y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$, where $\boldsymbol{w}$ is a weight vector and b is a bias, which defines a separation hyperplane. Suppose that we are looking for the optimal parameters $\{\boldsymbol{w}, b\}$ such as the margin is maximized. This problem can be stated as minimizing the lagrangian equation $L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{m} a_i \left\{ y_i \left(\boldsymbol{w}^T \phi(\boldsymbol{x}) + b \right) - 1 \right\}$, where $\boldsymbol{a}$ is a vector of lagrange multipliers, with $a_i \geq 0 \; \forall i$. This is a convex quadratic optimization problem which can be solved in a dual formulation with the kernel $k_f(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ by using the Karush-Kuhn-Tucker conditions [37, 38]. The vectors $\boldsymbol{x}_i$ for which the corresponding a_i are not zero are called support vectors, and they are the ones that define the separating hyperplane. For a never seen data $\boldsymbol{z}$, the output of the classifier is given by $y(\boldsymbol{z}) = \text{sgn}\left(\sum_{i \in \mathcal{S}} a_i y_i k_f(\boldsymbol{x}, \boldsymbol{x}_i) + b \right)$, where $\mathcal{S}$ is the set of support vectors. SVM has been widely used in bioinformatics [6, 39]. It has been the first computational method used for pre-miRNA prediction [9, 12]. It is still the one most widely applied in its standard form with radial basis function (gaussian) kernels and default parameters [13, 14, 40–52], with varying feature sets and genomes. In [44], a standard SVM has been applied together with a classical ML strategy for balancing imbalanced data sets [53] with the advantage in that case of improving the performance of the classical classifier. In [50], a standard SVM has been used as well, but together with a feature selection step based on genetic algorithms. In [43] three classical SVM models are applied sequentially, like filters, for increasing the specificity of predictions. In [42] an ensemble of SVM has been tried, also to improve performance. It is already well-known in ML that ensembles are superior to single classifiers. That is why in this study we have included ensemble of classifiers in the comparisons.

Ensembles of classifiers have been widely applied with success to genomics data for prediction and classification, variable selection, pathway analysis, genetic association and epistasis detection. The most popular and proven powerful classifier ensemble is random forest (RF), which is an ensemble of decision trees [54, 55]. A tree classifier consists in a number of nodes starting from a root node. At each node, the training set for that node, $\mathcal{D}$, is split into two non overlapping sets $\mathcal{D}_l$ and $\mathcal{D}_r$: a feature k is selected, and for that feature a threshold θ_k is choosen such that if $x_k \leq \theta_k$ the sample $\boldsymbol{x}$ is assigned to $\mathcal{D}_l$, or is assigned to $\mathcal{D}_r$ otherwise [56]. The tree is grown until maximum depth is reached. For the prediction of a new case $\boldsymbol{z}$, it is pushed down the tree. It is then assigned as output $h(\boldsymbol{z})$ the label of the terminal node where it ends. Since individual decision trees tend to overfit, usually bootstrap-aggregated (bagged) decision trees are used to combine the results of many trees. The final decision for an unknown input vector $\boldsymbol{z}$ using the ensemble of J trees (the RF model) is made by the rule $y(\boldsymbol{z}) = \arg\max_{w_k} \sum_j I(h_j(\boldsymbol{z}) = w_k)$, where $y_j(\boldsymbol{z})$ is the output assigned to the pattern by the j-th tree in the ensemble. For pre-miRNA prediction, in [57] authors proposed a combination of a set of standard base algorithms such as SVM and KNN, aggregating their prediction through a classical voting system. Similarly, in [58] it is proposed a bagging ensemble, that is to say, a committee of complementary base classifiers that learns from different training subsets. Actually, miPred [15] was the first truly ensemble method (RF) proposed, which achieved high discriminative power for human pre-miRNAs. More recently, HuntMi [18] performed a complete study comparing many standard supervised methods (NB, SVM and RF), where RF was confirmed to be the best one for the identification of new pre-miRNAs in animals, plants and viruses.

In unsupervised ML methods, no target variable is provided during learning. Instead, the algorithm searches for patterns and hidden structures or similarities among all the samples. The most common unsupervised method is clustering [63]. Clustering refers to grouping records, observations or cases into classes of similar objects. A cluster is a collection of data points that are similar to one another and dissimilar to data in other clusters. In order to use an unsupervised model as a classifier, the class label of the positive samples are used, after training, for labeling the clusters found. For example, for pre-miRNAs prediction, the clusters where there is at least one well-known pre-miRNA are labeled as positive. The clusters that have only unlabeled sequences are labeled as negative class. Therefore, for finding candidates to novel pre-miRNAs, only the unlabeled sequences clustered together with the labeled ones have to be looked at. The most popular and widely known unsupervised algorithms are hierarchical clustering, k-means, self-organizing maps and spectral clustering, although they have not been widely applied for pre-miRNA predictions (see Table 1).

Hierarchical clustering (HC) is one of the simplest and most popular unsupervised method in post-genomic data analyses [64]. It clusters data by forming a tree diagram or dendrogram, which shows the relationships between samples according to a distance measure. The root node of the dendrogram represents the whole data set, and each leaf is regarded as a data point. The clusters are obtained by cutting the dendrogram at different levels [65]. Agglomerative HC starts with m singleton clusters G_i, each of which includes exactly one data point, $G_i = \{\boldsymbol{x_i}\}$. The distance among two clusters G_i and G_j is defined as the minimum distance between all possible pairs of members of the clusters, $d(G_i, G_j) = \min_{\forall \boldsymbol{x}_p \in G_i, \boldsymbol{x}_q \in G_j} ||\boldsymbol{x}_p - \boldsymbol{x}_q||$. The algorithm then successively merges clusters by selecting the two with minimum distance and iterating this procedure until all samples belong to the same cluster. In [25], a standard agglomerative HC was used to perform clustering over the candidate structures along

with known pre-miRNA structures. This way, mixed clusters allow the identification of candidates that are similar to well-known precursors. HC has been applied in [26], not for the discovery of novel pre-miRNAs but for the mapping of validated miRNAs in one species to their most likely orthologues in other species. That is to say, it has been used as tool for automated miRNA mapping across and within species, through sequence similarity and secondary structure.

The k-means (KM) algorithm is one of the best known and most popular clustering algorithm [66]. It begins by assigning k centroids to data points randomly chosen from the training set, $\mathbf{c}_i = \boldsymbol{x}_{\mathrm{rnd}(1,\ldots,N)}$. At each iteration, a data points $\boldsymbol{x}_j$ is classified by assigning it to the cluster G_i whose centroid $\boldsymbol{c}_i$ is the closest one, that is $i_j^* = \arg\min_{\forall i}\{\|\boldsymbol{c}_i - \boldsymbol{x}_j\|\}$. Then, new cluster centroids are computed as the average of all the points belonging to each cluster. This process continues for each $\boldsymbol{x}_j \in \mathcal{L}$ until both, the cluster centroids and the class assignments, no longer change. To the best of our knowledge there are no published proposals for pre-miRNA prediction with KM. It has been included in this study for completeness, since it is the most used and cited clustering method over the last 50 years [66].

In recent years, spectral clustering (SC) has become one of the most popular and modern clustering algorithms [71]. SC considers each sample $\boldsymbol{x}_k$ as a vertex in a graph and weights the connections between samples with some measure of similarity [72]. This measure can be coded in a weight matrix S, where the entry s_{ij} represents the strength of the connection between samples $\boldsymbol{x}_i$ and $\boldsymbol{x_j}$. The similarity between samples is usually measured as $s_{ij} = s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/(2\sigma^2)\right)$. From the weight matrix S, a graph laplacian is built as $L = D - S$, where D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{m} s_{ij}$. Then, this unnormalized laplacian is normalized as $L_N = D^{-1/2} L D^{-1/2}$. The next step is computing the k smallest eigenvectors of L_N and store them as columns in a matrix U. This matrix results of size m by k. That is, the rows of U can be interpreted as data points $\hat{\boldsymbol{x}}_i \in \mathbb{R}^k$, which can be now clustered with any clustering algorithm. Similarly as for KM, there are no published works on SC-based pre-miRNA prediction. This algorithm has only been used in this area in [24] for clustering miRNAs with similar function.

SOMs are a special class of neural networks that use unsupervised learning, based on the idea of neurons that compete in response to a given input [67]. The training begins by choosing random weights $w_{ij} \in [-0.5, +0.5]$ for each neuron in the map. Given an input sample, its distance to each neuron weights is computed and the winning neuron for this data sample is looked by $i_j^* = \arg\min_{\forall i}\{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}$. The weight vector of this winning neuron (and a number of its neighbors) is further moved with $\boldsymbol{w}_i = \boldsymbol{w}_i + \eta\,(\boldsymbol{x}_r - \boldsymbol{w}_i)\,, \forall i \in N_c$, where N_c is a set of neighboring neurons of the winning neuron, and η the learning rate [68]. This is repeated until no significant changes in weights are performed. SOMs have the capability of identifying similar input patterns in the feature space, by assigning them to the same neuron or a group of adjacent neurons on the map [69, 70]. The first SOM proposed for pre-miRNA prediction has appeared very recently in [22]. In [23], a deepSOM architecture with several levels of hidden SOMs was proposed, where each inner SOM discards bad candidates to pre-miRNAs. Only best candidates survive to the next SOM level. The size of the maps in the topology and the number of layers are automatically adjusted according to the data samples in each level. At the last level, the unlabeled data in the neurons having clustered (at least) one well-known labeled sample are identified as the best pre-miRNAs candidates.

A deep neural network (deepNN) can be built from several layers of nonlinear feedforward networks. Layers that are commonly used in deep learning include latent variables organized layer-wise in deep generative models such as the restricted Boltzmann machines (RBM) [59]. A single RBM consists of a layer that receives the input vectors $\boldsymbol{x}$, and has a set of connection weights w_{ij} in a hidden layer of neurons with activation outputs $\boldsymbol{h} = [h_1, ..., h_P]$. The joint distribution of hidden variables $\boldsymbol{h}$ and observation samples $\boldsymbol{x}$ can be written as $p(\boldsymbol{x}, \boldsymbol{h}) \propto e^{-E(\boldsymbol{x},\boldsymbol{h})}$, where $E(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{h}^T W \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{h}$ is the energy function, W is the weight matrix, and $\boldsymbol{b}$ and $\boldsymbol{c}$ are bias vectors for the input and the hidden layer. The parameters $\{W, \boldsymbol{b}, \boldsymbol{c}\}$ can be learnt by an unsupervised algorithm based on Gibbs sampling [59]. After this unsupervised stage, a supervised training is applied and therefore this model uses an hybrid learning approach. It has been shown that RBMs have the universal approximation property [60]. Very recently, in [61, 62] a deepNN for pre-miRNA prediction was proposed. It consists of three hidden layers, each pre-trained as a RBM. For the first hidden layer, the input patterns $\boldsymbol{x}$ are used to produce an internal representation $\boldsymbol{h}'$. Once the parameters of this network are trained, a second layer is trained using the $\boldsymbol{h}'$ as input and producing a second hidden layer $\boldsymbol{h}''$. This is repeated to produce $\boldsymbol{h}'''$. Then, a final standard feedforward output layer trained by error backpropagation is added, which predicts the class labels.

3 MATERIALS AND PERFORMANCE MEASURES

3.1 Data sets

Selecting an informative feature set is very important for the pre-miRNA prediction, and most commonly used feature sets contain information about sequence, topology and structure [73]. The earliest work in this field proposed features named triplets, computed from the sequence itself [12]. miPred [15] was the first method that proposed a representative feature set with great discriminative power, adopted by most other methods [18]. Thus, we have used them in this study: triplets, maximal length of the amino acid string, cumulative size of internal loops found in the secondary structure, and percentage of low complexity regions detected in the sequence. The features have been normalized with z-score.

For this study we have created a number of datasets of varying levels of class imbalance using already available public data [18], which provide a negative class and a positive class with all well-known pre-miRNAs in miRBase v17 [11] for *Homo sapiens* (1,406 positive and 81,228 negative samples) and *Arabidopsis thaliana* (231 positive and 28,359 negative samples). Differently from most published reviews, in this work we focus on providing a broad spectrum of comparative results for ML methods regarding the large class imbalance issue, allowing a comprehensive assessment of the supervised versus unsupervised approaches at increasing imbalance levels. Thus, different artificial imbalance

ratios (IR) (defined as the ratio of negative to positive class samples) have been produced for this comparative study by randomly varying the number of available elements in each class, ranging from 1:1 (no imbalance) up to 1:2,000 (very high imbalance).

A classical ML strategy for balancing imbalanced data sets (for the supervised models) has been evaluated as well: the synthetic minority oversampling technique (SMOTE) [53], which is an approach for oversampling the minority class. In fact, SMOTE is the most used technique nowadays in supervised pre-miRNA classifiers [74]. It is limited to the strict assumption that the local space between any two positive instances is positive. SMOTE first randomly selects several nearest neighbors of a minority class instance, and produces new instances based on linear interpolations between the original examples and the randomly selected nearest neighbors. It produces artificial samples as convex combinations of each positive sample and one of its nearest neighbors.

3.2 Measures

The prediction quality of each model was assessed by the following classical classification measures: sensitivity (s^+), specificity (s^-), precision (p), and harmonic mean of sensitivity and precision (F_1)

$$s^+ = \frac{TP}{TP+FN}, \qquad p = \frac{TP}{TP+FP},$$

$$s^- = \frac{TN}{TN+FP}, \qquad F_1 = 2\frac{s^+ p}{s^+ + p},$$

where TP, TN, FP and FN are the number of true positive, true negative, false positive and false negative classifications, respectively.

The s^+ measures how good is a classification method at recognizing (and not missing) the true positives. The s^-, instead, measures the recognized true negatives. The precision p measures the relation between true positives and false positives, which in this large imbalance context is very important because false positives, regardless of being just a fraction of the total of negatives, are a very large number of samples in comparison to true positives. This is of relevance especially when thinking in a realistic scenario. Considering the characteristics of the prediction under study and given the large class imbalances, it is important to take into account both sensitivity and the number of false positives. Therefore, F_1, being the harmonic score between precision and recall, is used as a global comparative measure among many prediction methods.

For each ML model tested, a stratified 10-fold cross validation (CV) procedure has been used, giving reliable estimates of classification performance. Each model hyperparameters were determined with an inner grid search of a range of possible values, within each training partition. In the case of supervised methods, hyperparameters are, for example, the number of neighbors in KNN or the number of layers and neurons in deepNN. In the case of unsupervised models, the corresponding hyperparameter is number of clusters. If a small number of clusters is used, the larger the clusters the more likely it is that they include a well-known positive. However, at the same time, more false positives can be obtained and precision falls. Therefore, there is a trade-off between sensibility and precision when clusters grow.

The performance in each experiment is reported as the average values on 10 folds for the test partitions only. In order to statistically evaluate the differences between classifiers, that is, to detect differences in methods across multiple imbalanced data sets, a Friedman rank test at significance level $\alpha = 0.05$ is carried out for F_1. After that, the Nemenyi test will be used as a post-hoc test in order to show which methods are significantly different from each other according to the mean rank differences of the groups [75].

4 RESULTS

4.1 Supervised models performance

Table 2 and 3 show the results after testing all the supervised ML approaches included in this review, ranging from very low to very high class imbalance; without and with SMOTE for class balancing, respectively. It can be seen that in both tables and all cases, in both datasets and for all methods, the s^- (true negative rate) is always very high, above 95.00%. In most cases and for most classifiers, it is around 99.00%. This is an expected as well as a useless result, because due to the existing large class imbalance and the abundace of negative cases, any classifier could be good to accurately detect the negative cases just by always predicting "negative class" at the output. This is not useful, however, from a practical point of view, since the true interest is in the minority (positive) class. Actually, looking at the s^+ (true positive rate) and p together, or the global measure F_1, is where one can really understand how hard this problem is, as imbalance increases.

In Table 2, it can be clearly seen how imbalance has a direct (and negative) impact on all the supervised classifiers performance, in particular for SVM that is the most widely used in literature. For example, for SVM in *H. sapiens* and very low class imbalance (1:1 to 1:50), s^+ goes from 96.14% to 66.21%. It keeps decreasing up to an extremely low value of less than 10% at the highest imbalance level (1:2,000). This means that most positive samples (well-known pre-miRNAs) will not be correctly recognized with this method at such high imbalance level. The SVM precision begins at a high 96.79% but then it decreases, reaching an extremely low p of 20.00% at the highest imbalance because there are many false positives returned by SVM at this level. It should be mentioned that this really bad performance would not be, however, correctly reported if accuracy had been calculated (as many published work do), since it is a performance measure that is biased towards the majority class and does not take into account p and s^+ together. Instead, F_1, that takes into account p and s^+ together, correctly reflects this performance decrease as imbalance growths, showing how SVMs can have, in a very large imbalance situation a

Table 1: Machine learning approaches for pre-miRNAs prediction.

ML method	**Name**	**positive class**	**negative class**	**Reference**
SVM	Triplet-SVM	*H. sapiens*	random pseudo hairpins from *H. sapiens*	[12]
	MirAbela	*H. sapiens*, *M. musculus*, *R. norvegicus*	tRNA, rRNA and mRNA from *H. sapiens*	[40]
	RNAmicro	animals (nematodes, insects, and vertebrates)	random shuffling of animal miRNA features and tRNAs	[13]
	MicroprocessorSVM	*H. sapiens*	ncRNA from *H. sapiens*	[41]
	MiRFinder	animals (human, mouse, pig, cattle, dog and sheep)	random sequences from human and mouse	[14]
	MIRenSVM	*H. sapiens*, *A. gambiae*	pseudo hairpins	[42]
	mirCos	*H. sapiens*, *M. musculus*	random sampling of training genomes	[43]
	microPred	*H. sapiens*	pseudo hairpins and other ncRNAs from *H. sapiens*	[44]
	PlantMiRNAPred	all miRNA plants in miRBase	pseudo hairpins from the protein coding sequences of *A. thaliana* and *G. max*	[45]
	miRPara	animals, plants and virus in miRBase	sequences with pri-miRNAs identical to the positive class	[46]
	SMIRP	species-specific positive sets from miRBase	ncRNA	[47]
	iMiRNA-SSF	*H. sapiens*	pseudo premiRNAs from *H. sapiens*	[48]
	ViralmiR	virus	random virus sequences, human pre-miRNAs and pseudo-hairpins from *H. sapiens*	[49]
	YamiPred	*H. sapiens*, animals	random pseudo hairpins from *H. sapiens* and ncRNAs	[50]
	iMcRNA-PseSSC	*H. sapiens*	random pseudo hairpins from *H. sapiens*	[51]
	MiRNA-dis	*H. sapiens*	random pseudo hairpins from *H. sapiens*	[52]
KNN	MinDist	*D. melanogaster*, *A. gambiae*	random sequences	[25]
NB	BayesMirnaFind	*H. sapiens*, *M. musculus*	potential negative stem-loops	[35]
	MatureBayes	*H. sapiens*, *M. musculus*	random sequences from *H. sapiens*, *M. musculus*	[17]
Ensemble	miPred	*H. sapiens*	pseudo-premiRNAs from *H. sapiens*	[15]
	HuntMi	*H. sapiens*, *A. thaliana*, animals, plants	*H. sapiens*, *A. thaliana*, animals, plants	[18]
	pMIRNA	*H. sapiens*, *O. sativa* and *A. thaliana*	pseudo hairpins and ncRNAs	[57]
	miR-BAG	animals (human, mouse, rat, dog, nematode and fruit fly)	pseudo-hairpins of tRNA, rRNA, sRNA, mRNA	[58]
Deep NN	DP-miRNA	*H. sapiens*, animals	random pseudo hairpins from *H. sapiens* and ncRNAs	[61, 62]
	deepSOM	*H. sapiens*, *A. thaliana*, animals, plants, *C. elegans*		[23]
HC	MinDist	*D. melanogaster*, *A. gambiae*		[25]
	MapMi	animals		[26]
SOM	miRNA-SOM	*C. elegans*, *E. multi*		[22]
SC	CWLan	*H. sapiens*		[24]

Sources: https://omictools.com, [9, 27–31]

very poor global performance. In *A. thaliana*, similar conclusions can be achieved for SVM, s^- is always high; s^+ is high (around 80.00-90.00%) at low imbalance but at medium and high imbalance levels (from 1:50 on) it goes down to less than 50.00%. At the extremes, from 1:500 and on, s^+ has diminished to unacceptable levels. Regarding precision of SVM, it has been maintained higher than 90.00% values up to imbalance 1:200; being however extremely poor after 1:500, meaning that it has identified a very large number of false positives. This is correctly reflected by F_1 with values quite below 50.00%.

In Table 3, the use of SMOTE for artificial class balancing has not significantly improved SVM performance. It can be stated that it has had a little impact at the highest imbalance levels, depending on the training data set, or it has even made it worse. In any case, the global performance F_1 is unacceptable, below 50.00%. This can be explained by the fact that SMOTE is interpolating positive samples as intermediate points between the known ones, where many of them are located very close to the other class samples. This artificial balance is not helping setting the support vectors in a better position in the decision frontier between classes. Due to the existing very high class imbalance, the hypothesis that the local space between two near positive samples corresponds to a positive class sample does not hold, especially if the boundary between classes is complex. When this is the case, the support vectors located in a region full of samples from the opposite class produce very bad test results. In fact, this can be very clearly seen in the Figure 1 of the Supplementary Material.

For the KNN classifier a similar analysis can be done. In Table 2, without SMOTE, at low imbalance levels (up to 1:10), all measures are very high in both datasets; between 86.00% and 96.00%. But at mid-level imbalance (1:100), for example in *H. sapiens*, it has already decreased performance to 66.00% in s^+ and around 40.00% at the highest class imbalance. The same with p. The global F_1 decreases very much as well with increasing imbalance, from 96.00% down to around 45.00%. In the *A. thaliana* data set, the performance values are slightly better up to imbalance 1:200, being all of them of around 56.00% at the highest imbalance. When SMOTE is applied to a KNN classifier (in Table 3), it has only a noticeable influence on s^+ at the very high imbalance levels, raising just a 10.00-20.00% more the performance values. Global performance F_1 is around 45.00-50.00% for the *H. sapiens* data set, and around 60.00-65.00% in *A. thaliana*.

Regarding RF in human data set without SMOTE (in Table 2), it has very high s^+, p and F_1 (74.00% and higher, up to 97.00%) at the very low imbalance levels (1:1 to 1:50). Then, when imbalance increases, it is more difficult to distinguish true positive from false positives. These rates diminish significatively in the middle-imbalance cases to 53.75% for s^+ and 86.23% for p at 1:500. This means that here RF can still have an acceptable precision (low number of false positives), but at the cost of missing (mis-classifying) half of the true positives. However, at a very large imbalance level (1:1,000 and 1:2,000) RF does not work properly anymore; s^+ is as low as 25.00% and p is 56.67%, meaning that this classifier is not capable of recognizing true samples. It suffers from overfitting of the negative class due to the large imbalance in data. In fact, this is perfectly reflected by F_1, which has very high values at the low imbalance

situations (90.00-96.00%), and then falls below 50.00% as imbalance increases much more. In the other dataset, the trend and global behavior is the same for RF. The use of SMOTE in RF (Table 3) is a little bit helpful at the highest imbalance, rising s^+ and p in *H. sapiens*; however for *A. thaliana* SMOTE does not help at all. A similar explanation to the use of SMOTE with SVM can be applied here as well, with SMOTE inducing changes in the decision boundary that are undesirable because there are artificial positive samples in the negative region.

NB has obtained the best results among the supervised methods evaluated regarding s^+, from mid to high imbalance ratio. For example, in *H. sapiens*, the s^+ of NB is higher than 80.00-90.00% even at the highest imbalance level, with and without SMOTE, apparently showing robustness to large class imbalance. It can be stated that it almost seems to be unaffected by the IR regarding true positives recognition. However, when looking at the precision p, extremely low values can be seen for high imbalances (1:500 and 1:1,000) in both data sets. Low values, less than 10.00% even with SMOTE, are observed. That seems to be the price to pay for high sensitivity: a very large number of false positives. This fact is correctly reflected by the global measure F_1, with values below 10.00-20.00% at the extreme imbalance, without real practical use.

Finally, the hybrid deepNN predictor behaved similarly to the other supervised models. Very high s^- is achieved no matter the imbalance ratio in both genomes. High s^+ is provided in low and mid range imbalance, with poor performance in the highest imbalance levels, which means not recognizing true positives. For example, in *H. sapiens*, at the highest imbalance level 80% of the true positives are lost; for *A. thaliana*, true positives are not recognized at all. These facts lead to very poor values in F_1 at the highest imbalance here evaluated. When SMOTE is used for balancing, the results improve in s^+, p and F_1, in particular at the highest imbalance levels. While these very recent neural models are being applied with success in many areas, and for pre-miRNA prediction have shown to be among the best models, deep neural networks have to be applied with caution in front of high class imbalance and more research is required to reach acceptable levels of performance.

4.2 Unsupervised models performance

Regarding the unsupervised ML methods in Table 4, for s^- (true negative rate), the same conclusions as before can be achieved in all cases in both datasets and for all methods: it is very high even at high imbalances, with values between 80.00% and 99.00%. This is however very misleading when true positive rate and precision are analyzed more deeply. Regarding the recognition of true positives (s^+) all the methods are equally very good at low and mid class imbalance (1:1 to 1:50), in both datasets, with high values, between 83% and 99%. Only very large imbalance, larger than 1:200, impacts on s^+ for SOM, HC and SC. In the *A. thaliana* data set, both KM and SOM maintain a very high sensitivity of 90.00% at the worst imbalance level.

In the case of KM, the s^+ is the highest of all unsupervised methods at the highest imbalance level, being 92.50% for *H. sapiens* and 90.00% for *A. thaliana*. Furthermore, it could be stated that increasing imbalance has no effect on this method when used as classifier because high performance is maintained for positive class recognition. However, precision is really bad, lower than 65.00% in low to mid imbalance and less than 5.00% at the highest imbalance, meaning that in the KM labeled clusters there is a very large amount of false positives. In fact, F_1 is very low for the mid-to-high imbalance levels, being less than 5.00% for KM in both data sets. For this method, it can be said that it will almost never miss a true candidate but many false candidates will be predicted as well.

SC has a very similar behavior to KM, thus an analogous analysis can be done. It is very good for recognizing true miRNAs at low or no imbalance, though having a very bad performance at the highest imbalance levels, with modest sensitivity and very low precision. For example, in human it has a true positive rate s^+ between 86.00% and 99.00% up to 1:200, falling to 60.00% when data imbalance is increased ten times. In the other data set, the drop is up to 60.00% as well. The same happens with SC precision: it is extremely low (less than 5.00%) at mid to high imbalances because it has a large number of false positives. F_1 reflects this fact with very poor results, indeed. The same happens in both data set.

Instead, SOM and HC have more balanced results in both s^+ and p, being equally good in both datasets at the low to middle imbalance levels. For example, SOM in *A. thaliana* maintains an always high s^+, from no imbalance to the extreme 1:1,500 imbalance level. It can be stated that this model is capable of maintaining very high true positives recognition, higher than 80.00%, no matter the imbalance present in data. The precision falls from around 94.00% to 21.00% of course, as imbalance increases, being however more than 10 times better than KM and SC. The F_1 of SOM in this data set at 1:2,000 is 26.86%, the second best global value, after HC, in comparison to all other unsupervised approaches evaluated. The F_1 of SOM in the human data set is the best one of all at the extreme 1:2,000.

HC, in the imbalance range from 1:1 to 1:100 in both data sets has high values in true positive rates and precision, and in the global F_1 measure as well. It is the best method also for the arabidopsis data set regarding precision and F_1 from mid to high imbalance level. For the mid-range it presents very good global F_1 performance around 75.00% in *A. thaliana*. At high class imbalance, however, it is affected but it preserves good precision, around 36.00-20.00%. This is reflected with a F_1 of approximately 30.00-44.00% in both data sets at the extreme 1:2,000 imbalance evaluated.

Finally, the deepSOM model is the best of all unsupervised models. This deep topology of SOMs appears as the most robust model in front of the diverse imbalance levels evaluated and for performance indexes measured. As happens with all other models, s^- is close to 100.00% in almost all cases and both genomes. At the highest imbalance levels, sensitivity remains higher than 60.00% in *H. sapiens* up to 1:500. After that levels, it is affected by the imbalance but still recognizing with acceptable precision. In the case of the *A. thaliana* genome, this balance between not loosing true positives and not having too much false positives is the best possible, beating all other unsupervised models, and even most of the supervised ones. The F_1 for deepSOM at the extreme imbalance case is the best result achieved in the Table 4.

Table 2: Supervised ML methods for pre-miRNA prediction.

H. sapiens

Imbalance	**SVM**				**KNN**				**RF**				**NB**				**deepNN**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	96.14	96.79	96.79	96.45	96.64	95.70	95.64	96.17	96.64	97.01	97.00	96.82	92.29	97.15	97.29	94.64	95.43	96.78	96.79	96.07
5	89.96	91.17	98.25	90.47	88.86	93.31	98.72	91.00	91.14	94.82	99.00	92.93	91.36	89.31	97.81	90.29	92.43	93.21	98.63	92.77
10	84.86	89.14	98.97	86.93	86.00	90.12	99.05	87.99	86.50	93.20	99.37	89.70	90.86	81.23	97.97	86.04	87.86	91.41	99.16	89.49
50	66.21	80.43	99.68	72.58	36.00	81.82	99.67	76.94	73.50	90.10	99.84	80.91	87.29	47.66	98.09	61.65	75.07	90.02	99.83	81.79
100	52.10	79.08	99.86	62.64	66.79	73.49	99.75	69.82	66.67	88.60	99.92	75.92	87.78	32.89	98.20	47.82	75.93	84.39	99.85	79.60
200	32.75	74.28	99.94	45.15	59.00	69.23	99.86	63.06	59.25	87.29	99.96	70.35	86.75	19.15	98.19	31.36	63.75	83.09	99.93	71.05
500	15.63	77.00	99.99	25.11	51.25	60.96	99.92	54.15	53.75	86.23	99.98	65.21	89.38	11.36	98.62	20.16	60.62	82.87	99.97	68.05
1000	7.50	45.00	99.99	12.67	41.25	45.94	99.95	42.48	33.75	77.17	99.99	45.93	86.25	5.80	98.61	10.86	48.75	65.14	99.97	55.02
1500	10.00	50.00	100.00	16.67	46.00	50.96	99.96	44.12	34.00	81.33	100.00	45.14	94.00	4.99	98.86	9.46	44.00	74.33	99.99	52.08
2000	5.00	20.00	100.00	8.00	42.50	55.83	99.98	45.71	25.00	56.67	99.99	33.71	95.00	3.54	98.70	6.82	20.00	51.67	99.99	27.71

A. thaliana

Imbalance	**SVM**				**KNN**				**RF**				**NB**				**deepNN**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	89.57	98.30	98.26	93.48	95.22	97.10	96.96	96.03	95.65	98.28	98.26	96.91	92.17	98.20	98.26	94.98	93.91	96.99	96.96	95.34
5	77.83	98.21	99.74	86.37	93.91	96.54	99.30	95.05	93.91	97.93	99.57	95.73	90.87	93.19	98.61	91.83	91.30	87.33	97.30	89.06
10	69.13	98.76	99.91	80.96	92.61	96.50	99.65	94.45	92.17	98.26	99.83	95.04	90.87	88.51	98.79	89.58	89.57	95.79	99.57	92.27
50	47.83	98.42	99.98	63.37	90.43	92.99	99.86	91.64	86.52	96.94	99.94	91.10	89.57	63.84	98.98	74.42	86.96	92.16	99.84	89.27
100	38.70	97.55	99.99	54.62	83.04	89.68	99.90	85.90	83.48	94.41	99.95	88.19	88.26	47.59	99.00	61.60	91.74	90.87	99.90	91.09
200	27.86	93.33	99.99	41.29	85.00	91.10	99.96	87.72	78.57	94.66	99.98	85.48	87.14	31.72	99.05	46.34	85.00	93.22	99.96	88.35
500	8.00	30.00	100.00	12.38	76.00	81.31	99.96	76.62	68.00	91.83	99.99	75.96	78.00	12.80	99.07	21.90	56.00	58.83	99.98	54.36
1000	0.00	0.00	100.00	0.00	65.00	60.00	99.97	59.00	40.00	55.00	99.99	43.33	65.00	9.13	99.57	15.95	70.00	72.33	99.98	66.72
1500	0.00	0.00	100.00	0.00	60.00	55.00	99.99	56.67	50.00	45.00	99.99	46.67	60.00	8.92	99.70	15.34	0.00	0.00	100.00	0.00
2000	0.00	0.00	100.00	0.00	60.00	55.00	99.99	56.67	40.00	40.00	99.99	40.00	70.00	8.64	99.67	15.12	0.00	0.00	100.00	0.00

Table 3: Supervised ML methods (with SMOTE) for pre-miRNA prediction.

H. sapiens

Imbalance	**SVM**				**KNN**				**RF**				**NB**				**deepNN**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	96.14	96.79	96.79	96.45	96.64	95.70	95.64	96.17	96.64	97.01	97.00	96.82	92.29	97.15	97.29	94.64	95.36	96.86	96.86	96.07
5	72.57	95.11	99.25	82.28	91.14	85.35	96.87	88.13	94.00	91.82	98.32	92.88	92.14	89.34	97.80	90.69	96.07	86.05	96.84	90.71
10	68.43	92.85	99.47	78.73	88.93	79.44	97.70	83.88	92.86	87.16	98.63	89.91	92.07	81.49	97.92	86.45	96.07	76.16	96.94	84.85
50	52.64	87.68	99.85	65.67	79.36	61.47	99.00	69.22	84.71	73.42	99.39	78.65	90.43	45.70	97.86	60.71	90.29	54.87	98.49	68.08
100	39.63	85.25	99.93	53.66	72.84	53.10	99.35	61.32	80.62	69.31	99.64	74.41	91.11	30.96	97.97	46.19	84.57	48.30	99.10	61.45
200	25.25	72.57	99.95	37.08	65.25	43.34	99.58	52.01	73.75	64.33	99.79	68.52	91.00	17.89	97.94	29.90	78.75	38.77	99.36	51.49
500	15.00	69.17	99.99	23.55	62.50	40.39	99.81	48.54	65.00	60.75	99.92	61.84	91.88	9.85	98.33	17.78	72.50	43.27	99.81	53.75
1000	15.00	69.83	99.99	23.13	53.75	28.26	99.87	36.59	48.75	47.58	99.94	47.32	93.75	4.76	98.14	9.05	61.25	29.83	99.86	39.98
1500	16.00	36.67	99.99	22.14	60.00	33.92	99.92	42.28	60.00	55.57	99.97	57.00	94.00	4.21	98.65	8.06	62.00	37.64	99.93	45.57
2000	12.50	45.00	100.0	19.33	62.50	43.28	99.95	49.65	52.50	39.45	99.97	43.61	92.50	2.95	98.48	5.71	67.50	50.92	99.96	54.69

A. thaliana

Imbalance	**SVM**				**KNN**				**RF**				**NB**				**deepNN**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	89.57	98.30	98.26	93.48	95.22	97.10	96.96	96.03	95.65	98.28	98.26	96.91	92.17	98.20	98.26	94.98	93.48	97.46	97.39	95.31
5	83.91	98.01	99.65	90.13	94.78	90.52	97.91	92.42	95.22	97.08	99.39	96.01	91.74	93.66	98.70	92.51	98.70	75.09	93.04	84.94
10	78.26	98.00	99.83	86.68	94.35	89.08	98.79	91.48	94.35	97.76	99.78	95.95	91.74	89.05	98.83	90.24	97.83	66.72	95.06	79.21
50	37.83	96.11	99.95	51.01	90.43	80.40	99.55	84.93	91.30	92.10	99.83	91.31	90.43	62.51	98.91	73.77	94.78	68.90	99.04	78.95
100	23.48	96.41	99.99	36.69	87.83	73.22	99.67	79.37	90.43	89.69	99.90	89.95	89.57	45.80	98.92	60.47	91.74	72.87	99.65	80.91
200	17.86	88.00	99.99	28.74	83.57	77.59	99.88	80.27	87.14	88.24	99.94	87.23	89.29	28.63	98.89	43.26	91.43	74.14	99.82	80.98
500	10.00	26.67	100.00	14.05	84.00	66.55	99.92	72.74	74.00	85.67	99.98	77.74	84.00	10.53	98.73	18.66	82.00	71.71	99.94	74.23
1000	5.00	10.00	100.00	6.67	70.00	45.00	99.94	53.33	45.00	45.00	99.98	42.33	80.00	6.37	99.22	11.78	80.00	49.52	99.94	59.11
1500	10.00	10.00	100.00	10.00	70.00	47.50	99.96	54.00	50.00	45.00	99.98	46.67	80.00	11.00	99.74	19.20	90.00	47.83	99.95	59.00
2000	0.00	0.00	100.00	0.00	70.00	63.33	99.98	65.00	30.00	25.00	99.98	26.67	80.00	9.15	99.69	16.29	90.00	60.00	99.96	67.86

Table 4: Unsupervised ML methods for pre-miRNA prediction.

H. sapiens

Imbalance	**KM**				**SOM**				**HC**				**SC**				**deepSOM**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	99.64	65.52	46.86	78.98	96.14	93.30	93.07	94.69	98.14	87.40	85.79	92.40	99.36	82.10	78.21	89.88	95.07	95.16	95.14	95.10
5	99.43	36.52	65.18	53.34	92.14	82.88	96.19	87.23	94.14	77.56	94.54	85.01	98.57	48.93	79.45	65.37	90.14	87.62	97.44	88.82
10	99.00	23.43	67.48	37.84	90.57	76.10	97.16	82.68	91.36	72.14	96.47	80.59	98.71	32.09	79.13	48.41	89.64	81.15	97.92	85.16
50	98.57	8.28	78.17	15.27	85.29	43.55	97.80	57.64	83.21	48.24	98.21	61.04	98.43	7.98	77.31	14.75	83.79	46.33	98.07	59.65
100	97.65	5.44	83.00	10.30	80.37	36.92	98.62	50.56	79.63	41.43	98.87	54.44	96.17	4.84	81.09	9.21	77.53	40.62	98.87	53.27
200	95.25	3.34	86.36	6.45	75.50	27.69	99.02	40.44	70.00	33.18	99.30	44.94	86.75	2.90	85.69	5.62	69.00	36.91	99.40	47.74
500	92.50	2.94	93.88	5.69	75.00	25.00	99.54	37.20	69.38	29.32	99.65	40.61	73.75	1.98	92.79	3.86	64.38	37.70	99.78	47.02
1000	91.25	1.96	95.41	3.84	61.25	15.68	99.67	24.91	58.75	18.53	99.74	27.88	70.00	1.22	94.40	2.39	47.50	31.16	99.90	37.23
1500	92.00	2.48	97.67	4.82	70.00	22.29	99.82	32.90	62.00	18.21	99.83	27.82	54.00	0.98	96.74	1.93	40.00	28.57	99.93	31.66
2000	92.50	2.69	98.19	5.22	65.00	21.98	99.89	32.48	67.50	20.34	99.87	30.60	60.00	1.16	97.25	2.26	42.50	30.78	99.94	34.32

A. thaliana

Imbalance	**KM**				**SOM**				**HC**				**SC**				**deepSOM**			
(1:IR)	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1	99.13	59.27	30.87	74.05	96.52	93.88	93.48	95.09	96.96	90.72	90.00	93.70	97.39	94.03	93.48	95.57	95.22	96.70	96.52	95.85
5	96.96	34.43	62.78	50.74	94.35	92.85	98.43	93.31	95.22	85.22	96.61	89.83	95.65	77.17	94.26	85.33	93.48	94.19	98.78	93.70
10	96.96	26.03	71.39	40.85	93.04	93.46	99.35	93.17	93.48	83.18	98.05	87.82	93.91	58.10	93.07	71.59	93.04	96.46	99.65	94.67
50	96.09	10.22	82.81	18.45	89.13	79.00	99.49	83.25	91.30	77.42	99.45	83.42	94.78	19.13	91.92	31.78	91.30	86.35	99.69	88.42
100	96.09	7.43	87.91	13.78	89.13	68.96	99.60	77.69	89.57	65.71	99.52	75.35	93.91	9.36	90.85	17.01	88.70	76.86	99.72	81.96
200	93.57	4.79	90.58	9.09	87.14	63.24	99.74	72.95	88.57	66.88	99.76	75.50	94.29	8.16	94.62	14.99	84.29	69.97	99.81	75.56
500	88.00	3.17	95.05	6.12	84.00	48.61	99.83	60.62	86.00	62.15	99.88	68.18	80.00	4.36	96.76	8.24	82.00	51.31	99.85	62.16
1000	90.00	1.38	95.23	2.72	85.00	38.36	99.89	52.04	75.00	36.19	99.91	47.78	50.00	1.26	97.30	2.46	55.00	63.33	99.97	54.67
1500	80.00	1.22	97.46	2.41	90.00	43.67	99.93	54.52	70.00	36.5	99.92	44.00	70.00	1.38	97.88	2.69	40.00	30.00	99.97	33.34
2000	90.00	1.03	96.70	2.04	50.00	20.83	99.92	26.86	60.00	24.10	99.89	31.10	60.00	1.15	98.21	2.27	60.00	45.00	99.98	50.00

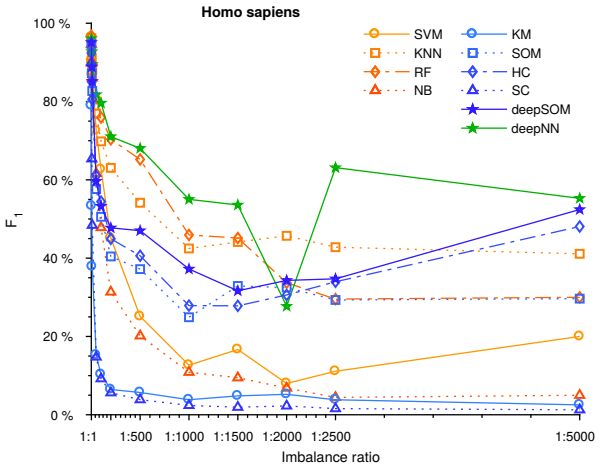

Figure 1: Machine learning approaches for pre-miRNA prediction with increasing imbalance levels in *H. sapiens* dataset. Supervised (shades of red lines), unsupervised (shades of blue lines) and hybrid learner (green line).

4.3 Global comparative results

In order to summarize the results and to more easily evaluate the global behavior of all of the ML approaches, Figure 1 and 2 show the F_1 score obtained by all methods in each data set without SMOTE, and for each imbalance level. In these figures in particular a further extreme imbalance case of twice the highest imbalance reported in the tables (1:5,000) has been included in order to show the behavior closer to more real cases. Supervised models are indicated with shades of red, unsupervised learners with shades of blue and deepNN is in green line because it is a hybrid model having an unsupervised stage before the last supervised tuning of the neural network. From the figures it can be easily seen how all methods noticeable decrease performance as imbalance increases.

Figure 1 shows the F_1 scores for all methods in the *H. sapiens* data set. Up to 1:200, most of supervised models maintain acceptable F_1 scores, being the best ones RF and KNN, together with the hybrid deepNN. At the 1:500 imbalance, two quite separated groups are easily identified: RF, KNN, deepNN, HC, SOM and deepSOM, versus very poor performance of SVM, NB, KM and SC. After this point up to 1:2,000, all methods continue falling up to less than 50.00%. Between 1:2,500 and 1:5000 all supervised methods maintain poor values, except only for KNN that falls to 41.11%. It is noteworthy that in this imbalance range, deepSOM and HC increase performance up to around 50%. The best classifiers at the highest imbalance level are the hybrid deepNN and the unsupervised learners deepSOM and HC; while all the other methods have very low performance. Moreover, it can be seen that globally, up to 1:1,500 supervised methods are the best ones, but in the highest imbalance methods including unsupervised learning stages have the best performance. That is, supervised methods are those more significantly degraded.

Figure 2 shows the F_1 scores for all methods in the *A. thaliana* data set. Again here, all methods are compromised by the increasing imbalance level. Up to 1:1,500, two groups of classifiers are easily identified. The best group includes KNN and RF together with the hybrid deepNN and the unsupervised HC, SOM and deepSOM. The worst group includes NB, SVM, KM and SC, with extremely low F_1 scores, which continue decreasing performance from this point forward. At 1:2,500 the hybrid deepNN and two unsupervised methods are the best ones. Between 1:2,500 and the highest class imbalance, deepSOM increases F_1 being the best one, followed by RF and SOM. All other models inevitably collapse because of the high class imbalance. It should be noted in particular that deepSOM, based on unsupervised models, performs much better compared with the most widely used model in this field (Triplet-SVM) in both data sets.

In order to statistically evaluate differences between all the classifiers in high class imbalance (1:1000–1:5,000) in both data sets and all folds, a Friedman rank test for F_1 was applied and resulted in P¡1.73E-47 at $\alpha = 0.05$, indicating that the differences among the scores are statistically significant. The corresponding critical difference (CD) diagram for post-hoc Nemenyi test [75], which obtained a CD=1.55, is shown in Figure 3. This statistical analysis clearly indicates that, for the high imbalance present in pre-miRNAs, the best models are deepSOM, KNN, deepNN, HC and SOM. In this group, SOM, deepSOM and HC are the best unsupervised methods, being not statistically different from KNN and the hybrid deepNN, which need positive and negative labeled data sets. The CD shows that there are no differences between NB, KM, SC and SVM, being SVM the worst one. Thus, deepSOM, KNN, deepNN, HC, SOM and RF versus NB, KM, SC and SVM are confirmed to be the best and worst classifiers for pre-miRNA prediction, respectively. The difference between these two groups of classifiers is statistically significant.

This final comparative result is very interesting. First of all, it is a strong evidence that the most used and published SVM models (such as triplet-SVM) is not the adequate classifier for pre-miRNAs predictions in close to real-life scenarios. It also clearly shows that the more recent models including unsupervised stages can be superior to standard supervised approaches. In more real situations where the imbalance can be even higher, models with unsupervised learning could be preferably explored, since they can offer better performance and they do not need a negative class definition.

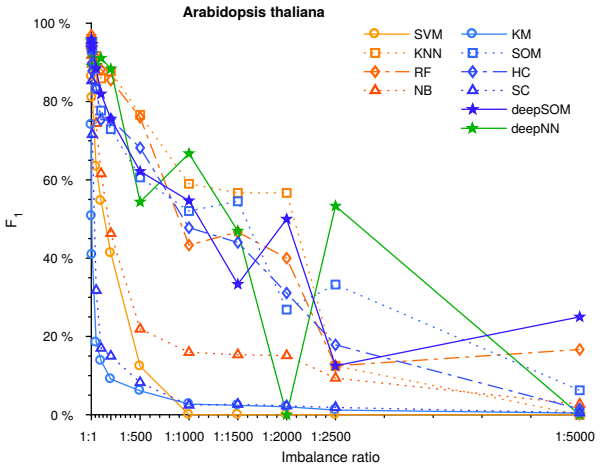

Figure 2: Machine learning approaches for pre-miRNA prediction with increasing imbalance levels in *A. thaliana* dataset. Supervised (shades of red lines), unsupervised (shades of blue lines) and hybrid learner (green line).

5 DISCUSSION

It has to be remembered at this point that the differences between unsupervised and supervised ML approaches are basically that supervised models need both class labels for training, while unsupervised methods do not need class labels during training. For classification, the class labels of the positive samples are required only after training. They first model the complete feature space regardless of class labels. Therefore, unsupervised models look at which sequences are the closest ones to well-known pre-miRNAs only after learning.

For supervised methods, a negative class definition is required. In practice, it is very difficult to build an appropiate set of negative examples for training them adequately, capable of effectively describing the non-pre-miRNA class. Even though tRNAs and rRNAs have been generally used as negative training sets, it is not known for sure whether hairpins from those RNA segments could not actually generate miRNAs [76]. Thus, the main drawback of the supervised ML approaches is the lack of an appropriate negative set, which actually represent negative examples that the classifier could find when analyzing genomics data. Since the real number of miRNAs in any given genome is still an open issue, in most works it is assumed that the probability of finding a pre-miRNA in any randomly chosen stem-loop extracted from a genome is very low. However, it cannot be generally guaranteed that hairpins that would normally be processed by the miRNA-maturation pathway are not being included in the negative training set.

This is particularly relevant in a high class imbalance context. In fact, a recent study on the impact of the negative sets when predicting human pre-miRNAs has stated that most existing supervised classifiers cannot actually provide reliable predictive performance on independent testing data sets because their negative training sets are not sufficiently representative when there is a high class imbalance [77]. In such case, the obviously negative class is well-learned and a large number of unknown sequences are predicted as positive (actually, false positives) that are very difficult to validate with wet-lab experiments. Furthermore, when positive predictivity is analyzed in detail, it may fall below 50%. In particular, it has already been shown that supervised models are less affected by low class imbalance, but they are the models most affected in presence of high class imbalance [76].

Since the quantity of true pre-miRNA is increasing as time passes, if imbalance decreases, supervised models could perform better after SMOTE. It would strongly depend on the real quantity of pre-miRNAs present in a given organism, and the rate of novel pre-miRNA discovery for such organism. In a real scenario, the number of known pre-miRNAs is in the order of hundreds for most genomes, and in the order of thousands in the case of the human genome (there are 1881 well-known human miRNAs up-to-date in MirBase). The rest of the unlabeled sequences can be in the order of millions. Thus, it is unlikely that such high class imbalance could decrease enough to be adjusted with SMOTE.

Unsupervised models, instead, build a model from the data, reflecting the data closeness into the clusters. They only look for the closer centroid to each sample, and classify sequences as pre-miRNA candidates if this centroid belongs to a pre-miRNA cluster. An unsupervised approach does not use class labels to shape the acceptance region in the learning stage and, since it does not try to identify the optimal margin between positive and negative examples, it is also less likely to suffer from overtraining. Thus, it can be stated that the negative class labels have a less important role in unsupervised methods than in supervised ones. For supervised learners, both types of labels must be defined. In unsupervised ones, the negative class labels can be missing. After training, the class assignment depends only on the presence of (at least) one positive case in a cluster, no matter how many other unknown negatives samples there are also, together, in the same cluster. All the feature space is learned with the same detail, that is, both positive and negative classes are modeled although in an unsupervised fashion, without class labels. After training, the original labels of the positive class are used to identify the clusters that have the best pre-miRNAs candidates.

From the results obtained in the comparisons, it can be stated that regarding the question on which ML approach is the best (supervised or unsupervised), the answer strongly depends on the level of class imbalance present. The comparative results show that when there is low to mid imbalance, supervised models have the best performance. However, in the case of high class imbalance, closer to a real genome-wide prediction task where there can be one

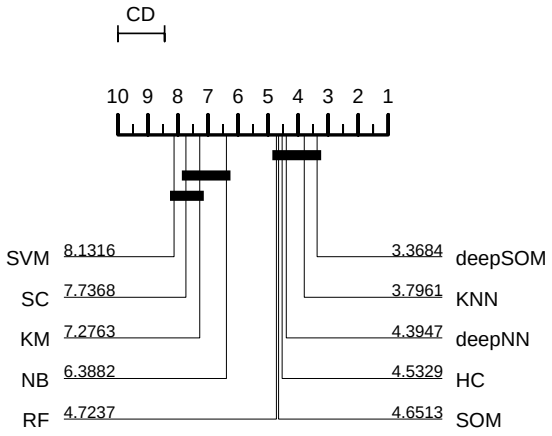

Figure 3: Statistical significance diagram. Critical difference (CD) diagram for Nemenyi tests performed on human and arabidopsis datasets for F_1 of supervised and unsupervised ML approaches for pre-miRNA prediction. Bold lines indicate groups of classifiers which are not significantly different (their average ranks differ by less than CD value).

hundred of well-known pre-miRNA in millions of unlabebeled sequences to classify, models including unsupervised learning could perform better, being also more simple to understand, generate and use.

It can be stated that unsupervised learning has many advantages for this application and that there is plenty of room for future research in computational algorithms based on this approach for novel miRNA precursors discovery. For example, a negative class must not be defined nor artificially created. This makes these models much simpler to build for a non-expert user, since all available sequence data can be provided as input without the labeling process. Independently from the method and the number of clusters, after training the results are more easy to interpret. Once the clusters that have well-known pre-miRNAs are located, the better candidates to new pre-miRNAs can be identified very fast, simply as those other sequences within such clusters. Furthermore, in a high class-imbalance context such as the pre-miRNA prediction, this is of particular relevance because in a more real scenario, there are always very few positive examples in proportion to the unknown ones. Thus, when working with genome-wide data, the unsupervised approach could be more naturally suited to learn the specific characteristics of the well-known examples, even if only a few, not being biased by the class imbalance and the sequences that are distant to pre-miRNA clusters. This way, it can be stated that these kind of models can be used in more realistic situations where genome-wide data is under analysis. In summary, with the unsupervised approach, all genome-wide data of the same species could be simply used, and the best highly-likely candidates to pre-miRNAs can be easily identified, after training, as those sequences clustered together with well-known pre-miRNAs.

CONCLUSIONS

In pre-miRNA prediction there is a very high class imbalance between well-known pre-miRNAs and unlabeled sequences that the supervised classification models cannot properly handle. This work provides to the bioinformatics community a conceptual and updated review of existing ML approaches for pre-miRNA prediction. The results obtained in the comparisons indicate that unsupervised machine learning and deep neural architectures can be more suited for future research in computational methods for pre-miRNA prediction than classical supervised approaches, such as SVM. Comparative results have clearly shown that unsupervised approaches and deep neural networks including unsupervised learning are capable of maintaining good performance rates, while classical supervised models quickly deteriorate when class imbalance increases. Additionally, the unsupervised approaches are more naturally suited to an end user that has good knowledge on the pre-miRNAs of the genome under study, but has no knowledge regarding the definition of a negative class for training a supervised classifier.

KEY POINTS

- The computational prediction of novel microRNAs involves identifying good candidate sequences in high class imbalanced data in the context of machine learning.
- Supervised models need the definition of positive and negative class samples. But negative samples must be defined artificially by a manual process.
- Unsupervised machine learning methods do not need class labels for training. The class labels, only of the positive samples, are used later for the classification task. These approaches model the complete feature space with the same detail, regardless of class labels. Thus, learning is not biased by the majority class.
- The answer to the question on which machine learning approach is the best (supervised or unsupervised) for this task strongly depends on the level of class imbalance present.
- This study has shown that supervised methods are those more significantly degraded as imbalance ratio increases. At high class imbalance and as long as the imbalance remains high, like in a real genome-wide prediction task with about 1,000 well-known pre-miRNAs and millions of unlabeled sequences, unsupervised approaches can be a better choice.

- Future research in computational methods for pre-miRNA prediction should be oriented towards the design of deep neural networks predictors and models that include unsupervised learning stages, in order to properly handle the inherent high class imbalance.

FUNDING

This work was supported by National Scientific and Technical Research Council (CONICET) [PIP 2013 117], Universidad Nacional del Litoral (UNL)[CAI+D 548, 082, 076, 042], Universidad Tecnologica Nacional (UTN) [PID 4442] and Agencia Nacional de Promocion Cientifica y Tecnologica (ANPCyT) [PICT 2014 2627, 2015 2472].

BIOGRAPHIES

G. Stegmayer is Assistant Professor in the Department of Informatics at Universidad Nacional del Litoral (UNL), and Independent Researcher at the sinc(i) Institute, National Scientific and Technical Research Council (CONICET), Argentina. Her current research interest involves machine learning, data mining and pattern recognition in bioinformatics.

L. Di Persia is Assistant Professor in the Department of Informatics at UNL and Adjunct Researcher at CONICET. He is vice-director of the sinc(i) Institute. His research interests include signal processing, pattern recognition and machine learning, applied to audio, speech, biomedical and bioinformatics data.

M. Rubiolo is Assistant Researcher at sinc(i), Teaching Assistant in the Department of Informatics at UNL, and Adjunct Professor at the Department of Information Systems Engineering in UTN-FRSF. His research involves neural networks, pattern recognition and bioinformatics.

M. Gerard received a degree in Biotechnology in 2007 and a PhD in Engineering in 2013 from UNL. He is Assistant Researcher at sinc(i) and Teaching Assistant in the Department of Informatics at UNL. His research interests include bioinformatics, machine learning and swarm intelligence.

M. Pividori is a postdoctoral fellow at sinc(i). He is currently in a research stay at the University of Chicago, US, working in the Section of Genetics Medicine. His research interests involve data mining and bioinformatics, with particular focus on consensus clustering.

C. Yones received the Computer Engineering degree in 2014 from UNL. Since 2014 he is a PhD student at sinc(i). His research interests include machine learning, data-mining, semi-supervised learning, with applications in bioinformatics.

L. Bugnon is a PhD student at sinc(i) since 2013. His research interests include automatic learning, pattern recognition, signal and image processing, with applications to affective computing, biomedical signals and bioinformatics.

T. Rodriguez is a PhD student at sinc(i). His research interests include pattern recognition in big data and bioinformatics.

J. Raad received the Bioengineering degree in 2012. He is a Ph.D. student at sinc(i). His research interests include data mining with applications in bioinformatics.

D.H. Milone is Full Professor in the Department of Informatics at UNL and Principal Research Scientist at CONICET. He is Director of sinc(i). His research interests include statistical learning, signal processing, neural and evolutionary computing, with applications to biomedical signals and bioinformatics.

sinc(i) - Research Institute for Signals, Systems and Computational Intelligence. Research at sinc(i) aims to develop new algorithms for machine learning, data mining, signal processing and complex systems, providing innovative technologies for advancing healthcare, bioinformatics, precision agriculture, autonomous systems and human-computer interfaces. The sinc(i) was created and is supported by the two major institutions of highest education and research in Argentina: the National University of Litoral (UNL) and the National Scientific and Technical Research Council (CONICET).

References

[1] D Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116:281–297, 2004.

[2] T Huan, J Rong, C Liu et al. Genome-wide identification of microrna expression quantitative trait loci. *Nature Communications*, 6(1):6601, 2015.

[3] R Takahashi, H Miyazaki, and F Takeshita. Loss of microRNA-27b contributes to breast cancer stem cell generation by activating ENPP1. *Nature Communications*, 6(1):7318, 2015.

[4] C Cheng, R Bahal, I Babar et al. MicroRNA silencing for cancer therapy targeted to the tumour microenvironment. *Nature*, 518(1):107–110, 2015.

[5] C-Y Lai, Y Sung-Liang, MH Hsieh et al. MicroRNA expression aberration as potential peripheral blood biomarkers for schizophrenia. *PLoS One*, 6(6):e21635, 2011.

[6] V Williamson, A Kim, B Xie et al. Detecting miRNAs in deep-sequencing data: a software performance comparison and evaluation. *Briefings in Bioinformatics*, 14(1):36–45, 2013.

[7] L Li, J Xu, D Yang et al. Computational approaches for microRNA studies: a review. *Mamm Genome*, 21(1):1–12, 2010.

[8] I Lopes, A Schliep, and A deCarvalho. The discriminant power of RNA features for pre-miRNA recognition. *BMC Bioinformatics*, 15(1):124+, 2014.

[9] V Shukla, VK Varghese, SP Kabekkodu et al. A compilation of Web-based research tools for miRNA analysis. *Briefings in Functional Genomics*, 1(1):1–25, 2017.

[10] CP Gomes, JH Cho, L Hood et al. A review of computational tools in microRNA discovery. *Frontiers in Genetics*, 4(1): 81–104, 2013.

[11] A Kozomara and S Griffiths-Jones. miRBase: integrating microRNA annotation and deep-sequencing data. *Nucleic Acids Research*, 39:152–157, 2011.

[12] C Xue, F Li, T He et al. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6(1):310, 2005.

[13] J Hertel and PF Stadler. Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, 22(14):e197–e202, 2006.

[14] T Huang, B Fan, M Rothschild et al. MiRFinder: an improved approach and software implementation for genome-wide fast microRNA precursor scans. *BMC Bioinformatics*, 8(1):341+, 2007.

[15] P Jiang, H Wu, W Wang et al. MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Research*, 35(1):W339–W344, 2007.

[16] Y Xu, X Zhou, and W Zhang. MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*, 24(1):i50–i58, 2008.

[17] K Gkirtzou, I Tsamardinos, P Tsakalides et al. MatureBayes: A probabilistic algorithm for identifying the mature miRNA within novel precursors. *PLOS one*, 5(8):e11843, 2010.

[18] A Gudy, M Szczeniak, M Sikora et al. HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC Bioinformatics*, 14(1):83+, 2013.

[19] ME Rahman, R Islam, S Islam et al. MiRANN: a reliable approach for improved classification of precursor microRNA using Artificial Neural Network model. *Genomics*, 99(4):189–194, 2012.

[20] K Ng and SK Mishra. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, 23(1):1321–1330, 2007.

[21] J Allmer and M Yousef. Computational methods for ab initio detection of micrornas. *Frontiers in Genetics*, 3(1):209–212, 2012.

[22] L Kamenetzky, G Stegmayer, L Maldonado et al. MicroRNA discovery in the human parasite *Echinococcus multilocularis* from genome-wide data. *Genomics*, 107(6):174–280, 2016.

[23] G Stegmayer, C Yones, L Kamenetzky et al. High class-imbalance in pre-miRNA prediction: a novel approach based on deepSOM. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(6):1316–1326, 2017.

[24] C Lan, Q Chen, and J Li. Grouping miRNAs of similar functions via weighted information content of gene ontology. *BMC Bioinformatics*, 17(19):507, 2016.

[25] N Mendes, S Heyne, A Freitas et al. Navigating the unexplored seascape of pre-miRNA candidates in single-genome approaches. *Bioinformatics*, 28(23):3034–3041, 2012.

[26] J Guerra-Assuncao and A Enright. MapMi: automated mapping of microRNA loci. *BMC Bioinformatics*, 11(1):133, 2010.

[27] S Demirci, J Baumbach, and J Allmer. On the performance of pre-microRNA detection algorithms. *Nature Communications*, 8(1):330–340, 2017.

[28] B Liu, J Li and M Cairns. Identifying miRNAs, targets and functions. *Briefings in Bioinformatics*, 15(1):1–19, 2014.

[29] J Hertel, D Langenberger, P Stadler. Computational prediction of microRNA genes. *Methods Mol Biol.*, 1097(1):437–456, 2014.

[30] N Mendes, A Freitas and MF Sagot Current tools for the identification of miRNA genes and their targets. *Nucleic Acids Research*, 37(8):2419–2433, 2009.

[31] A Adai, C Johnson, S Mlotshwa and et al. Computational prediction of miRNAs in Arabidopsis thaliana. *Genome Research*, 15(1):78–91, 2005.

[32] A Webb, editor. *Statistical pattern recognition.* Wiley Press, 2002.

[33] R Duda, P Hart, and D Stork. *Pattern Classification.* John Wiley and Sons, second edition edition, 2001.

[34] T Mitchell, editor. *Machine Learning.* McGraw Hill, 1997.

[35] M Yousef, M Nebozhyn, H Shatkay et al. Combining multi-species genomic data for microRNA identification using a naive bayes classifier. *Bioinformatics*, 22(1):1325–1334, 2006.

[36] V Vapnik. *The Nature of Statistical Learning Theory*. 1995.

[37] R Fan, P Chen, and C Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6(1):1889–1918, 2005.

[38] C Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[39] P Pavlidis, J Weston, J Cai et al. Gene functional classification from heterogeneous data. *Proc. 5th Annual International Conference on Computational Biology, ACM Press*, 1(1):249–255, 2001.

[40] A Sewer, N Paul, P Landgraf et al. Identification of clustered microRNAs using an ab initio prediction method. *BMC bioinformatics*, 6(1):267, 2005.

[41] SA Helvik, O Snove, and P Saetrom. Reliable prediction of Drosha processing sites improves microRNA gene prediction . *Bioinformatics*, 23(2):142–149, 2007.

[42] J Ding, S Zhou, and J Guan. MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC Bioinformatics*, 11(11):S11, 2010.

[43] Y Sheng, PG Engstrom, and B Lenhard. Mammalian MicroRNA prediction through a Support Vector Machine model of sequence and structure. *PLos ONE*, 2(9):e946, 2007.

[44] R Batuwita and V Palade. *microPred*: effective classification of pre-mirnas for human mirna gene prediction. *Bioinformatics*, 25(8):989–995, 2009.

[45] P Xuan, M Guo, X Liu et al. PlantMiRNAPred: efficient classification of real and pseudo plant pre-mirnas. *Bioinformatics*, 27(10):1368–1376, 2011.

[46] Y Wu, B Wei, H Liu et al. MiRPara: a SVM-based software tool for prediction of most probable microRNA coding regions in genome scale sequences. *BMC Bioinformatics*, 12(1):107, 2011.

[47] R Peace, K Biggar, K Storey et al. A framework for improving microrna prediction in non-human genomes. *Nucleic Acids Research*, 43(20):e138, 2015.

[48] J Chen, X Wan, and B Liu. iMiRNA-SSF: Improving the Identification of MicroRNA Precursors by Combining Negative Sets with Different Distributions. *Scientific Reports*, 6(1):19062, 2016.

[49] K Huang, T Lee, Y Teng et al. ViralmiR: a support-vector-machine-based method for predicting viral microRNA precursors. *BMC Bioinformatics*, 16(1):S9, 2015.

[50] D Kleftogiannis, K Theofilatos, S Likothanassis et al. YamiPred: A novel evolutionary method for predicting pre-miRNAs and selecting relevant features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5):1183–1192, 2015.

[51] B Liu, L Fang, F Liu et al. Identification of Real MicroRNA Precursors with a Pseudo Structure Status Composition Approach. *PLos ONE*, 10(3):e0121501, 2015.

[52] B Liu, L Fang, J Chen et al. miRNA-dis: microRNA precursor identification based on distance structure status pairs. *Molecular BioSystems*, 11:1194–1204, 2015.

[53] NV Chawla, KW Bowyer, LO Hall et al. SMOTE: Synthetic minority over-sampling. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.

[54] L Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[55] X Chen and H Ishwaran. Random forests for genomic data analysis. *Genomics*, 99(6):323–329, 2012.

[56] KP Murphy. *Machine Learning. A probabilistic approach*. The MIT Press, 2012.

[57] S Lertampaiporn, C Thammarongtham, C Nukoolkit et al. Heterogeneous ensemble approach with discriminative features and modified-smotebagging for pre-mirna classification. *Nucleic Acids Research*, 41(1):e21, 2013.

[58] A Jha, R Chauhan, M Mehra et al. miR-BAG: Bagging based identification of microrna precursors. *PLOS One*, 7(9):1–15, 09 2012.

[59] A Fischer and C Igel An Introduction to Restricted Boltzmann Machines in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. *Lecture Notes in Computer Science, Springer, Berlin, Heidelberg*, 14–36, 2012.

[60] N LeRoux and Y Bengio Representational Power of Restricted Boltzmann Machines and Deep Belief Networks. *Neural Computation*, 20(6):1631–1649, 2008.

[61] J Thomas, S Thomas and L Sael DP-miRNA: An Improved Prediction of precursor microRNA using Deep Learning Model. *IEEE Int. Conf. Big Data and Smart Computing*, 96–99, 2017.

[62] J Thomas and L Sael Deep neural network based precursor microRNA prediction on eleven species. *CoRR abs/1704.03834*, 2017.

[63] R Xu and D Wunsch. *Clustering*. Wiley and IEEE Press, 2009.

[64] J Handl, J Knowles, and D Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15): 3201, 2005.

[65] L Rokach and O Maimon, editors. *Clustering methods. Data mining and knowledge discovery handbook.* Springer, 2005.

[66] A Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010.

[67] T Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[68] T Kohonen, M Schroeder, and T Huang. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 2005.

[69] G Stegmayer, M Gerard, and DH Milone. Data mining over biological datasets: an integrated approach based on computational intelligence. *IEEE Computational Intelligence Magazine, Special Issue on Computational Intelligence in Bioinformatics*, 7(4):22–34, 2012.

[70] DH Milone, G Stegmayer, L Kamenetzky et al. *omeSOM: a software for clustering and visualization of transcriptional and metabolite data mined from interspecific crosses of crop plants. *BMC Bioinformatics*, 11:438–447, 2010.

[71] A Ng, M Jordan, and Y Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 849–856, Cambridge, MA, USA, 2001. MIT Press.

[72] U vonLuxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, December 2007.

[73] C Yones, G Stegmayer, L Kamenetzky et al. miRNAfe: a comprehensive tool for feature extraction in microRNA prediction. *BioSystems*, 238:1–5, 2015.

[74] R Blagus and L Lusa. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1):106, 2013.

[75] J Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[76] RC Prati, G Batista, and DF Silva. Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270, 2015.

[77] L Wei, M Liao, Y Gao et al. Improved and promising identification of human micrornas by incorporating a high-quality negative set. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(1):192–201, 2014.