Deep neural architectures for highly imbalanced data in bioinformatics

L. A. Bugnon, *Member, IEEE*, C. Yones, D. H. Milone *Senior Member, IEEE*, G. Stegmayer*

Manuscript submitted to the Special Issue on
Recent Advances in Theory, Methodology and Applications of Imbalanced Learning

***Abstract*—In the post-genome era, many problems in bioinformatics have arisen due to the generation of large amounts of imbalanced data. In particular, the computational classification of precursor microRNA (pre-miRNAs) involves a high imbalance in the classes. For this task, a classifier is trained to identify RNA sequences having the highest chance of being miRNA precursors. The big issue is that well-known pre-miRNAs are usually just a few in comparison to the hundreds of thousands of candidate sequences in a genome, which results in highly imbalanced data. This imbalance has a strong influence on most standard classifiers and, if not properly addressed, the classifier is not able to work properly in a real life scenario. This work provides a comparative assessment of recent deep neural architectures for dealing with the large imbalanced data issue in the classification of pre-miRNAs. We present and analyze recent architectures in a benchmark framework with genomes of animals and plants, with increasing imbalance ratios up to 1:2,000. We also propose a new graphical way for comparing classifiers performance in the context of high class imbalance. The comparative results obtained show that, at very high imbalance, deep belief neural networks can provide the best performance.**

***Index Terms*—bioinformatics, pre-miRNA classification, deep neural architectures, high class imbalance.**

I. Introduction

THE imbalanced data problem has been largely recognized as an important issue in machine learning [1]–[3]. Most machine learning algorithms work well with balanced datasets, but with imbalanced datasets supervised classifiers tend to be biased towards the majority class and have a very low performance on the minority one. Classification algorithms are designed to maximize the number of correct predictions. When the class sizes differ considerably, the classifiers better recognize the larger class obtaining a high accuracy, while the minority class has a very low recall. The classification task where one class is significantly underrepresented relative to another still remains among the leading challenges in the development of novel classification models nowadays [4]. This is of particular importance in bioinformatics in the post-genome era in studies that involve, for example, disease diagnosis based on gene expression data, protein function classification, activity prediction of drug molecules and recognition of precursor microRNAs (pre-miRNAs).

L. A. Bugnon, C. Yones, D.H. Milone and G. Stegmayer are with Research Institute for Signals, Systems and Computational Intelligence (sinc(*i*)), FICH-UNL, CONICET, Argentina. *corresponding author email: gstegmayer@sinc.unl.edu.ar

MicroRNAs (miRNAs) are a special type of non-coding RNA of 21 nucleotides in length, which can be critical regulators in the post-transcriptional regulation of gene expression [5]. Since their discovery, these molecules have revolutionized and reshaped the bases of gene regulation. They may determine the genetic expression of cells and influence the state of the tissues. MiRNAs play important regulatory roles in many fundamental biological processes such as disease development and progression. Very recent studies demonstrated that miRNAs can serve as oncogene or tumor suppressor in various cancer types [6]; thus they can assist in a better diagnosis, prognosis prediction, and therapeutic assessment of such disease [7].

However, it is very hard to identify new miRNAs experimentally [8], and this difficulty has led to the development of several computational approaches for miRNAs classification in the last ten years [9]–[11], mainly based on support vector machines (SVM) [12]–[25]. The discovery of novel miRNAs involves identifying small RNA sequences having the highest chance of being real miRNA precursors, named candidates, which can be later validated in wet experiments. In order to do that, a binary classifier is trained with the well-known pre-miRNAs of a genome. The *big issue* with this task is that well-known pre-miRNAs are, usually, just a few in comparison to the hundred of thousands sequences that can be found in a genome. This results in a highly imbalanced dataset. In a real-life scenario, the number of known pre-miRNAs is in the order of hundreds for most genomes, and in the order of thousands for the human genome (there are 1,982 human miRNAs up-to-date in the release v19 of mirBase[1]). This represents an imbalance ratio (IR) larger than 1:1,000 (1 positive class sample and 1,000 negative class samples). Furthermore, in this context, the minority class often contains very few instances with a high degree of variability, making it difficult for a classifier to generalize on unknown data [2]. A recent study has shown that most existing machine learning classifiers, in this context, cannot provide reliable performances on independent testing data because of the imbalance [26]. Very recently, deep learning and novel deep neural network architectures have been proposed to deal with this imbalanced data issue in bioinformatics. Deep learning models have shown to perform very well because they can use the multi-layer architecture to learn multiple internal levels of representation of the data

[1]The miRBase database (http://www.mirbase.org/) is the public database of published miRNA sequences and annotations.

features [27], [28], [29].

In [30] a deep belief neural network (deepBN) for identifying pre-miRNA sequences was proposed. This model has an unsupervised stage with hidden layers pre-trained as restricted Boltzmann machines, followed by a supervised tuning of the network. In [31] a deep architecture of self-organizing maps (SOMs) was proposed to overcome the problem of having very few positive class samples and a very large negative class. This model named deepSOM has several layers of hidden SOMs, where each inner SOM discards less probable candidates to pre-miRNAs. The well-known pre-miRNAs samples are used in every deep level as positive class while less likely pre-miRNA sequences are filtered level after level. The deepSOM model, however, in spite of having very good specificity and recall, has low precision because a very large number of false positive sequences remain at the last level. Although it has a heuristic rule to automatically change the map size according to data size, layer after layer, this change is static and limited. We present here two variants to the deepSOM model: the deep elastic SOM (deSOM) and the deep ensemble elastic SOM (deeSOM), which overcome the mentioned issues. In deSOM the number of deep levels not only grows automatically, but also the size of each layer changes adaptively, becoming larger or smaller as necessary according to the data at each level. The deeSOM uses an ensemble strategy to mitigate the high class imbalance, mainly at the initial levels.

In this work, we analyze and compare these very recent deep neural network approaches to deal with the imbalanced data problem in the context of pre-miRNAs classification. We provide a comprehensive study and comparative assessment, including many animal and plant genomes and increasing IR much larger than commonly published, with fair comparisons of the classifiers with the same features and datasets.

This manuscript is organized as follows. Section II presents and explains in detail the deep neural network approaches that are compared in this study. Section III presents the datasets used, the experimental setup, and performance measures. Section IV shows the results obtained and their discussion. Finally, the conclusions of this work can be found in Section V.

II. Deep neural network approaches for pre-miRNAs classification

A. The deep belief neural network

A deepBN can be built from several layers of nonlinear feedforward networks. Each layer can be pre-trained as a restricted Boltzmann machine (RBM) [32]. Each 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 [32]. Then, a final supervised stage of training is applied. It has been shown that RBMs have the universal approximation property [33].

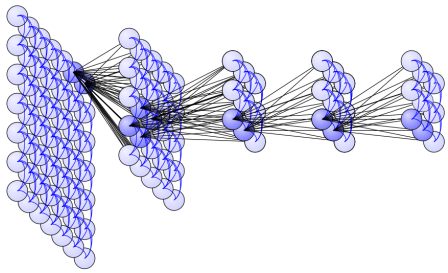

Fig. 1. The deepSOM topology. Example of an architecture with 5 levels. Dark blue neurons are pre-miRNA neurons, which provide the input to the next SOM (black lines). Levels are sequentially generated until no change is observed in the map sizes.

DeepBN models precisely based on RBMs are beginning to appear for pre-miRNA classification, with very good results [30]. These models can capture the properties of the data, learning the low-dimensional hidden features. However, the architecture of the network has to be optimized according to the task. The most relevant hyperparameters for this model are the number of deep layers, number of hidden neurons, dropout, and number of training epochs.

B. The deep self-organizing map

The first SOM-based model proposed for pre-miRNA classification has appeared very recently in [31]. It is an architecture with several levels of hidden SOMs, named deepSOM, that can be seen in Figure 1. It is based on the SOMs capability of identifying similar input patterns in the feature space and assigning them to the same neuron or a group of adjacent neurons on the map, in an unsupervised way [34]. A hierarchy of SOMs in deep nested levels refines the previous map by discarding samples that are distant to neurons with positive class samples, level after level.

The training process of deepSOM starts with a root SOM on the first layer ($h = 1$). This SOM undergoes standard training with the complete set of input data, using an initial large map size. During training, each input data point is assigned to a map unit, according to the minimum Euclidean distance between the feature vector representing each sequence and each neuron centroid. When this first SOM becomes stable, that is to say, no further adaptation of the weight vectors occurs, only the data in the neurons having at least one positive class sample are chosen for training the next map. This neuron labeling occurs by taking into account the positive class data only if there is at least one positive sample in a neuron, it is labeled as a pre-miRNA neuron, no matter how many other data points are grouped there as well. In fact, there are much more candidates than true positive class samples due to the existing high class imbalance. During training, only sequences assigned to pre-miRNA neurons remain for training the next level of deepSOM. To set the size of the next deepSOM level h, the number of neurons n_h is determined according to an heuristic suggested by Kohonen [35], which states that the total number of neurons in a map is related to the number of data points to train it. This is repeated until consecutive maps do not change. After training, the best pre-miRNAs candidates

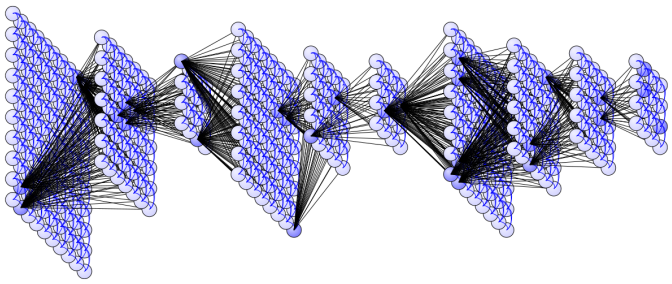

Fig. 2. The deep elastic SOM (deSOM) topology consists of several layers of SOMs, where map size in each level is adapted according to the filtering process, level after level.

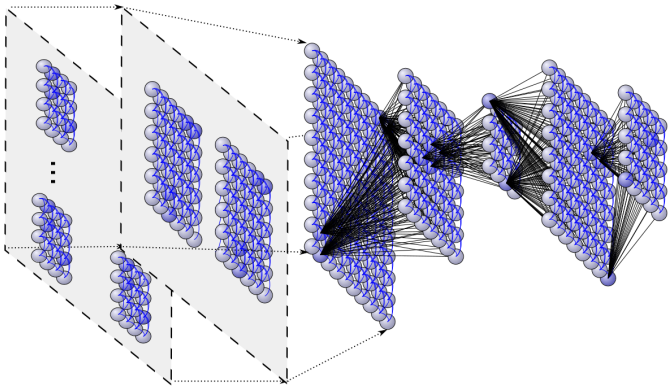

Fig. 3. The deep ensemble elastic SOM (deeSOM) architecture. It has layers of SOM-based ensembles.

are those sequences in the pre-miRNA neurons at the last level of the deepSOM. The most relevant hyperparameter for this model could be the maximum number of levels, but the last level is defined automatically when no changes are observed in the output of two consecutive SOMs.

C. The deep elastic SOM

The deep elastic SOM (deSOM) is a set of SOMs organized into a sequence of deep levels (see Figure 2), where the size of each layer is determined automatically like in deepSOM, according to the data distribution in that layer. However, in the deSOM the size of a layer can be increased if, at a certain level of depth, the map is no longer able to reduce the number of sequences. That is, the worst pre-miRNA candidates were removed and the remaining candidates are very close to the well-known pre-miRNAs. On this case, these sequences are very difficult to split with a small SOM. To do it, the adaptive algorithm of deSOM expands the map size, thus pre-miRNA neurons can be re-organized in a larger space. Therefore, several deep layers are added with this self size-adjusting method, which is triggered automatically with data reduction until only known pre-miRNA samples remain at the last layer. After training, analogously to deepSOM, the best pre-miRNAs candidates can be identified as the ones closer to the prototypes of the pre-miRNA neurons in the last levels. The high class-imbalance is being naturally tackled by this model during training, since the worst candidates (farthest to pre-miRNAs) are filtered in the initial levels and do not pass to the next ones. Another particular feature of this model is that a ranked set of candidates (for example, for further wet-lab experiments) can be obtained by checking the neurons of the next-to-last map and going back to each previous map, until a desired number of candidates is obtained. This model has not relevant hyperparameters to tune because the most important architectural configuration is self adaptive according to the training data.

D. The deep ensemble elastic SOM

As an additional way to address the high class imbalance in the initial levels, an ensemble scheme is proposed for the deSOM architecture, named deeSOM (see Figure 3). It consists of generating an ensemble of Q_ℓ SOMs at each level ℓ of the deSOM. Several parallel SOMs are used at each level and data is split among them, preserving the positive class samples and dividing the remaining ones, thus reducing the imbalance at each SOM of the ensemble. For example, let us suppose Q_1 SOMs at the initial level. These maps are trained in parallel, where all positive samples are presented to every SOM and all other samples are randomly split. This way, each SOM models just a fraction of the unlabeled space. At each map, pre-miRNA neurons are identified as those having, at least, one well-known positive sample. The sequences that are in pre-miRNA neurons are selected to pass to the next level. In the next level of deeSOM, there will be Q_2 maps. Each one receives all the positive class samples and a fraction of the unlabeled samples of the previous level.

The size and number of members in each ensemble are determined from the training data at each layer. Thus, Q_ℓ is adjusted automatically depending on data distribution. The hypothesis is that ensembles can lead to better classification performance by reducing the layer imbalance, distributing the majority-class samples among the members. This way, Q_ℓ can be automatically set to reach an appropriate imbalance in each ensemble member according to the size of the data feeding the layer. Reducing the imbalance also reduces the training data size for each ensemble member. However, there should be enough data to train each map. Thus, when the input has a high imbalance, the number Q_ℓ is set to approximate an optimal imbalance at each SOM. After a number of ensemble layers, when the IR of the output is lower than optimal, the model takes $Q_\ell = 1$ and the following layers behave as in deSOM. The most important hyperparameter of this model is the optimal imbalance for SOMs, to set each Q_ℓ.

III. Datasets, performance measures and experimental setup

For the comparisons we have created a number of datasets of varying IRs using already available public data from a benchmark dataset [35]. This provides a positive set with all well-known pre-miRNAs in miRBase v19 [36] and a negative set including random sequences from the genomes of a set of animals and plants[2], with the same sequence

[2]Source code and datasets are freely available at: https://sourceforge.net/projects/sourcesinc/files/miRimbal

TABLE I
NUMBER OF POSITIVE SAMPLES FOR DIFFERENT IR ON ANIMALS AND PLANTS DATASET.

IR	Animals	Plants
1:1	7,053	2,172
1:100	2,181	1,149
1:500	436	229
1:1000	218	114
1:1500	145	76
1:2000	109	57

length distribution than the corresponding positives. In each dataset, all the positive samples are concatenated as originally provided in [35]. Since these data points are actually mixed in the feature space (see Supp. Mat. Fig. S1), they provide homogeneous examples of positive class in a wide-variety of plants and animals genomes. In fact, since a large number of miRNAs are conserved between species in the same kingdom, a wide dispersion between well-known pre-miRNAs is not expected.

For each sequence data, commonly used features in literature have information about sequence, topology, structure [36], and motifs [37]. For fair comparisons with state-of-the-art classifiers, we have used the 28 features originally provided in [35]. These are the result of a feature selection process and have shown high discriminative power: 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.

Class imbalance has been defined as the ratio of the number of positives to the number of negative samples. A wide-range of possible IR have been taken into account, from low and moderate to very high. To this end, different artificial IRs have been produced by selecting the number of positives and negatives, ranging from 1:1 (no imbalance) up to 1:2,000 (very high imbalance). The number of positives samples for each IR is shown in Table I. The total number of samples in the animals dataset is 7,053 positives and 218,154 negatives. In the 1:100 case, there are 2,181 positives and 218,154 negatives. From this point, due to the restriction of the available data, the number of negatives is always the same and the higher imbalances are generated by reducing the number of positive cases. In the plants dataset, there are 2,172 positive and 114,929 negative samples. In 1:100, there are 1,149 positives and 114,929 negatives. Similarly to the animals datasets, for higher imbalances, the number of negative samples remains fixed. The datasets have been created starting with a random permutation of the samples and taking groups of positive cases incrementally to generate the imbalanced datasets. Thus, the smaller ones are included into the larger ones, in order to represent a real situation where there are a number of newly positives discovered each year and the total of well-known miRNAs is constantly increasing. For each model tested, a stratified 4-fold cross validation (CV) procedure has been used, giving reliable estimates of classification performance. Thus, for each imbalance in each fold, 75% of the data was used for training and the remaining 25% for testing.

In this work we focus on providing a broad spectrum of comparative results for deep neural architectures in front of very high imbalance. The aim of the comparisons is to analyze the classifiers robustness regarding how each deep neural model is able to manage the high imbalance by itself. We compared the deep neural architectures versus classical classifiers such as support vector machines (SVM) and multilayer perceptrons (MLP) [38]. Although the proposed deep models were designed to be robust to high imbalance, it is interesting to evaluate how they could work with some class balancing strategy as well. Thus, we have also included comparative results for Synthetic Minority Over-sampling (SMOTE) [2], [39]–[41]. In fact, SMOTE is the most used technique nowadays in supervised pre-miRNA classifiers [40].

The classification 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),

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

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

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 for recognizing (and not missing) the true positives of the problem. 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 and a practical application. Considering the characteristics of the classification problem 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 sensitivity, is used as a global comparative measure among classifiers.

The precision versus recall curve (PRC) is also a well-known performance indicator. Recent studies [2], [42] have shown that this representation is preferred over the receiver operating characteristics (ROC) plot to assess binary classifiers with highly imbalanced data. In these problems, a classifier could reach a good performance in terms of the ROC curve, but the number of false positives could be very high because of the size of the majority class, providing a bad quality list of pre-miRNA candidates in this context. PRC plots, instead, can provide a more clear assessment of performance due to the fact that they evaluate the fraction of true positives among the total number of candidates. Thus, an objective comparison between models has been performed with the area under the curve of precision-recall ($\hat{AUC}_{PR}$). As datasets have different imbalances and precision changes exponentially, a logarithmic ratio is defined as

$$A\hat{U}C_{PR} = 1 - \frac{\log\left(AUC_{PR}\right)}{\log\left(AUC_{PRG}\right)},$$

where AUC_{PRG} is the area under the curve for a random guess classifier, which is smaller for higher imbalances. This way, $A\hat{U}C_{PR}$ takes values in the interval [0,1] and results can be compared more easily between datasets with different imbalances.

Each model hyperparameters were determined with an inner grid search within each training partition, with internal validation subsets. In the case of the deeSOM, the optimal imbalance for the ensemble members (for defining Q_ℓ) was determined in preliminary experiments using a single SOM with different imbalances. The best trade-off was found around 1:1,000. Thus, Q_ℓ is automatically adjusted to approximate this imbalance in each SOM. For smaller imbalances $Q_\ell = 1$, exactly like a deSOM. For the deepBN hyperparameters, it has been already shown in diverse benchmarks that best results can be achieved with up to 3 layers, with a variable number of neurons each [43]. Thus, several architectures were evaluated, up to 3 layers and the number of hidden units in $\{2^3, 2^4, ..., 2^9\}$. Furthermore, different number of training epochs were used, from 16 to 512. In addition to the grid search, a random search strategy was applied in a wider range of hyperparameters [44]. However, optimal hyperparameters were very similar and yielded results equivalent to the grid search ones, but with a time cost substantially higher. Thus, grid search was used in the final experiments. An hyperparameter sensitivity evaluation was performed (Supp. Mat. Fig S2), where an optimal region can be seen with the learning rate in the range [0.05,0.1] and the batch size in the range [16,32]. Dropout was found to be in detriment of performance in all cases. The optimal number of training epochs was found to be between 64 and 256. In those ranges, performance variances were very small, that is, the approach was stable. Therefore, it can be considered that in those ranges the performance is insensitive to the hyperparameters chosen.

The performance in each experiment is reported as the average measures for the test partitions. In order to statistically evaluate the differences between classifiers, that is, to detect differences in methods across multiple imbalanced datasets, a Friedman rank test at significance level $\alpha = 0.05$ is carried out for F_1. After that, the Nemenyi test are 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 [45].

IV. Results and discussion

Table II to V show the results after testing all the approaches included in this study, ranging from low to very high imbalance, without and with SMOTE for balancing the data. In Table II, for animals dataset, it can be clearly seen how imbalance has a direct (and very negative) impact on MLP and SVM, which is the approach most widely used in this application domain. For the no imbalance case (1:1), all measures for SVM are equally very high, above 90%. But for 1:100, precision holds being higher than 90% and s^+ has dramatically drop to less than 30%, impacting on F_1 that has reduced to half its value in comparison to the situation of no class imbalance. From 1:500 and on, s^+ keeps decreasing (and so does F_1) up to an extremely low value of less than 5% at 1:1,000 IR, being zero at the highest imbalances. This means that most positive samples (well-known pre-miRNAs) will not be correctly recognized with this method at such IR. The SVM precision begins at a high 95.96% but then it decreases, reaching an extremely low p at the highest IR because of the high false positives count. It should be mentioned that this bad performance would not has been correctly reported if accuracy had been used as performance measure, because it is biased towards the majority class and does not take into account p and s^+ together. Instead, F_1 reflects this performance decrease as IR growths, showing how SVMs can have, in a very high IR situation, an extremely poor performance. Regarding s^-, it can be seen that in both datasets and for most methods, the specificity is always very high, between 90.00% and 100.00%. This is an expected as well as a useless result, because, from a practical point of view, the true interest is in the minority (positive) class. Actually, looking at the s^+ and p together (or the global measure F_1), one can really understand how hard this problem is, as imbalance increases. Table III reports models performance when SMOTE is used for balancing the classes. Of course, it has no effect when there is no imbalance. SMOTE begins to impact the performance of SVM in 1:500 and from this point forward. SVM with SMOTE can reach some $F_1 > 0$ at the highest imbalance levels. However, it has no competitive results with any of the deep neural models. For example, at 1:2,000, F_1 is now 8.51%, being anyway half the value of the F_1 of the worst deep neural model for animals (deepSOM) without SMOTE. In Table IV and V, for the case of pre-miRNA classification in plants without and with SMOTE, a similar analysis and the same trends for the SVM classifier can be found. The exactly same global analysis can be applied for the MLP classifier, in both datasets.

The deep architectures evaluated in this work have all shown that deep layers seems to be the most appropriate processing model for this type of highly imbalanced data. In Table II and Table IV, all deep models without SMOTE, at 1:1 have similar high performances, with balanced results. For 1:100, all deep models loose around a 10-20% in s^+ and around a 10% in precision, reflected by the F_1 score, that drops to almost half its previous value, in some cases. However, in relation to SVM, deSOM and deeSOM still have high values for F_1, higher than 60%. Moreover, in Table IV, all deep architectures have 20% more F_1 than SVM, and the deepBN has double F_1 than SVM, a remarkably high 85.44%. From 1:500 and on, imbalance has effect on F_1 in all deep SOMs architectures and variants, being however always much better than SVM that drops to less than 10%. Notably, deepBN holds a global performance around 50% at the higher IRs. In particular, for the deepSOM, s^- is close to 100.00% in almost all imbalances and both types of genomes. In the imbalances shown at the middle of the tables, sensitivity is between 40% and 60%. At the highest IR, deepSOM is affected by the imbalance and classifies with low precision, less than 20%. At the largest imbalance, deepSOM has F_1 of 15% for animals and 20% for plants. It can be

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
L. A. Bugnon, C. Yones, D. H. Milone & G. Stegmayer; "Deep neural architectures for highly imbalanced data in bioinformatics"
IEEE Transactions on Neural Networks and Learning Systems, 2019.

TABLE II
PERFORMANCE MEASURES FOR THE ANIMALS DATASET WHEN INCREASING IMBALANCE.

	MLP				**SVM**				**deepSOM**				**deSOM**				**deeSOM**				**deepBN**			
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	s^+	p	s^-	F_1
1:1	50.13	50.00	49.87	50.06	92.60	95.96	96.10	94.25	95.73	92.27	91.97	93.97	99.62	67.78	52.54	80.65	99.62	67.78	52.54	80.65	95.96	95.98	95.97	95.96
1:100	0.84	0.99	99.16	0.91	29.22	91.96	99.97	44.22	79.04	31.72	98.30	45.24	74.72	53.40	99.35	62.28	74.72	53.40	99.35	62.28	68.53	90.26	99.92	77.67
1:500	0.15	0.20	99.85	0.17	4.36	83.33	100.00	8.26	47.48	20.15	99.62	28.19	50.92	32.56	99.78	39.23	50.92	32.56	99.78	39.23	53.90	76.51	99.96	61.92
1:1000	0.06	0.10	99.94	0.08	1.85	100.00	100.00	3.64	37.50	15.13	99.79	21.47	34.72	30.12	99.89	30.14	34.72	30.12	99.89	30.14	47.69	65.13	99.97	50.24
1:1500	0.04	0.07	99.96	0.05	0.00	0.00	100.00	0.00	30.56	15.07	99.89	19.99	34.72	18.87	99.89	22.39	43.06	17.12	99.93	20.71	37.50	76.39	99.99	46.49
1:2000	0.03	0.05	99.97	0.04	0.00	0.00	100.00	0.00	24.07	11.06	99.90	15.09	32.41	18.01	99.90	21.63	28.70	16.79	99.93	20.71	36.11	68.24	99.98	40.09

TABLE III
PERFORMANCE MEASURES FOR THE ANIMALS DATASET WHEN INCREASING IMBALANCE, WITH SMOTE.

	MLP				**SVM**				**deepSOM**				**deSOM**				**deeSOM**				**deepBN**			
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	s^+	p	s^-	
1:1	50.13	50.00	49.87	50.06	92.60	95.96	96.10	94.25	95.73	92.27	91.97	93.87	99.62	67.78	52.54	80.65	99.62	67.78	52.54	80.65	95.96	95.98	95.97	95.96
1:100	3.41	0.99	96.59	1.53	32.57	86.42	99.95	47.24	95.37	8.43	89.64	15.49	98.90	2.18	55.36	4.26	98.90	2.18	55.36	4.26	88.12	38.06	98.55	53.10
1:500	1.53	0.20	98.47	0.35	7.34	69.93	99.99	13.28	87.84	3.97	95.72	7.59	92.89	0.71	73.92	1.42	92.89	0.71	73.92	1.42	75.92	37.49	99.74	50.07
1:1000	0.76	0.10	99.24	0.17	6.02	85.42	100.00	11.13	84.72	3.35	97.54	6.45	89.81	0.48	81.62	0.96	89.81	0.48	81.62	0.96	69.91	31.70	99.85	43.44
1:1500	0.56	0.07	99.44	0.12	2.78	37.50	100.00	5.16	78.47	3.27	98.44	6.27	81.25	0.48	88.82	0.96	81.25	0.48	88.82	0.96	59.72	33.37	99.92	42.53
1:2000	0.50	0.05	99.5	0.09	4.63	54.17	100.00	8.51	69.44	2.03	98.32	3.94	76.85	0.50	91.24	0.99	76.85	0.50	91.24	0.99	55.56	34.96	99.94	41.71

TABLE IV
PERFORMANCE MEASURES FOR THE PLANTS DATASET WHEN INCREASING IMBALANCE.

	MLP				**SVM**				**deepSOM**				**deSOM**				**deeSOM**				**deepBN**			
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	s^+	p	s^-	F_1
1:1	48.46	50.00	51.54	49.22	96.32	93.52	93.32	94.90	95.17	95.61	95.63	95.38	98.99	66.69	50.41	79.67	98.99	66.69	50.41	79.67	94.11	97.31	97.38	95.67
1:100	0.88	0.99	99.12	0.93	29.09	97.96	99.99	44.85	83.19	47.72	99.09	60.64	79.18	60.31	99.48	68.46	79.18	60.31	99.48	68.46	84.23	86.73	99.87	85.44
1:500	0.15	0.20	99.85	0.17	5.70	87.50	100.00	10.65	61.40	36.72	99.79	45.85	49.12	36.53	99.83	41.58	49.12	36.53	99.83	41.58	66.23	76.93	99.96	71.05
1:1000	0.06	0.10	99.94	0.07	0.00	0.00	100.00	0.00	51.79	30.29	99.88	37.99	56.25	34.94	99.89	41.48	56.25	34.94	99.89	41.48	66.07	71.56	99.97	67.47
1:1500	0.05	0.07	99.95	0.06	0.00	0.00	100.00	0.00	36.84	26.14	99.92	30.02	44.74	24.72	99.91	30.80	42.11	28.90	99.92	32.31	35.53	44.09	99.98	36.09
1:2000	0.03	0.05	99.97	0.03	0.00	0.00	100.00	0.00	28.57	16.20	99.93	20.47	44.64	27.36	99.92	31.20	48.21	27.02	99.93	33.50	30.36	58.57	99.99	33.99

TABLE V
PERFORMANCE MEASURES FOR THE PLANTS DATASET WHEN INCREASING IMBALANCE, WITH SMOTE.

	MLP				**SVM**				**deepSOM**				**deSOM**				**deeSOM**				**deepBN**			
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	s^+	p	s^-	F_1
1:1	48.46	50.00	51.54	49.22	96.32	93.52	93.32	94.90	95.17	95.61	95.63	95.38	98.99	66.69	50.41	79.67	98.99	66.69	50.41	79.67	94.11	97.31	97.38	95.67
1:100	2.58	0.99	97.42	1.43	34.23	93.34	99.98	50.09	92.86	13.78	94.15	23.97	96.43	2.14	55.48	4.19	96.43	2.14	55.48	4.19	88.68	50.11	99.09	63.77
1:500	1.58	0.20	98.42	0.35	8.33	77.12	100.00	14.74	85.96	7.00	97.64	12.90	86.84	0.54	64.49	1.06	86.84	0.54	64.49	1.06	72.81	41.63	99.79	52.80
1:1000	0.72	0.10	99.28	0.17	6.25	93.75	100.00	11.47	82.14	5.72	98.63	10.66	81.25	0.33	74.58	0.65	81.25	0.33	74.58	0.65	66.96	42.41	99.91	51.50
1:1500	0.66	0.07	99.34	0.12	5.26	66.67	100.00	9.55	75.00	4.06	98.72	7.65	65.79	0.21	77.72	0.42	65.79	0.21	77.72	0.42	67.11	41.47	99.93	50.33
1:2000	0.62	0.05	99.38	0.09	5.36	50.00	100.00	9.58	75.00	3.94	99.06	7.46	69.64	0.16	75.28	0.33	69.64	0.16	75.28	0.33	64.29	44.55	99.96	51.13

noticed that these values are improved by the other deep SOM architectures. The deSOM and deeSOM models boost all the scores of the original deepSOM model, in Table II and Table IV as well. In animals, the s^+ remains the same for these approaches, except in the case of highest imbalance, where deeSOM is better. The exactly same behaviour can be seen with the plants dataset. As a consequence, F_1 presents the same trend, in both tables: deeSOM is better than deSOM, which is better than the deepSOM. In Tables III and V, where all models have been applied after SMOTE, it has not improved the results of the deep neural architectures. In both tables the precision drops significantly, with a much more larger rate than s^+ increases, thus negatively affecting the F_1 scores. It has to be remembered that the SOM-based deep models have been specifically designed to manage large imbalance without needing any artificial balancing schema. The artificial positive samples inserted by SMOTE produce too many positive class neurons during training. This increases the amount of false positives, as many samples get associated with positive neurons, and thus the precision drops. It can be clearly seen that, globally, even the SMOTE improvements to SVM and MLP do not reach the high F_1 scores of the deep neural models without SMOTE. This is a remarkable result for the deep neural architectures. From the point of view of the application for pre-miRNA prediction, these deep models do not require additional balancing of data.

The deepBN is the best of all models and outperforms them in almost all configurations. In Table I for animals, for IR 1:100, deepBN has a very high precision (90.26%). Such high value is not reached by none of the other deep SOM-based models. Precision is high only for SVM, but at the price of a very poor s^+. DeepBN, instead, has an acceptable s^+ of around 70%, with a very good F_1 (77%). In 1:500 and 1:1,000, deepBN has yet F_1 values higher than 50%. The same happens in Table IV, for plants. Finally, at the highest imbalance of 1:2,000, it is clearly seen how the deepBN model outperforms all other architectures in animals, with s^+ 36%, a very high precision for this hard problem (68%), and F_1 approximately 40%. In the case of animals dataset, F_1 is twice as good as deeSOM (the best SOM-based architecture). In the plants datasets, deepBN is always better than the other models, in all configurations and IRs, except only in 1:2,000 where it is similar to deeSOM, being both the best ones at this extremely high IR. In Tables III and V, where deepBN has been used after SMOTE, some slightly better results are achieved for both datasets in terms of F_1. However, as SMOTE leads to a higher s^+ in exchange for a lower p, it may be considered an inferior result in terms of practical applications, where the reduction of false positives is very important. The better performance of deepBN for highly imbalanced data, in comparison to the other deep approaches, can be explained on the fact that this model can be considered as an hybrid learner. It includes an unsupervised learning stage at the beginning of training, combined with a supervised backpropagation afterwards. The first unsupervised step does not need nor uses class labels. Therefore, it can model the complete feature space, regardless of class labels, reducing the bias induced by the imbalance in the dataset under analysis.

In order to summarize the previous results, the statistical analysis and the global behavior of the approaches are shown in Figure 4. This figure includes the F_1 score obtained by all methods in each dataset, and for each IR. From the figure it can be easily seen how all methods decrease performance as imbalance increases. However, three kind of behavior are detected: the very poor performance of SVM and MLP, the deep SOMs topologies in the middle, and the best performance of deepBN. In 1:1, all methods are equally good. In 1:100 an abrupt drop in performance begins, and the three groups of behavior appear. At the highest imbalance, deepBN is easily identified as the best; deepSOM, deSOM, and deeSOM are, together, the second best models; and SVM and MLP have an extremely low performance. It should be noticed that, for 1:1,500 and 1:2,000, the deep SOM architectures are closer to the deepBN in performance, being deepBN and deeSOM the best approaches at the extreme IR in plants. In order to statistically evaluate differences among all the classifiers in high class imbalance, for both datasets and with all folds, a Friedman rank test for F_1 was applied and resulted in $p = 4.43 \times 10^{-30}$ at $\alpha = 0.05$, indicating that there is a statistically significant difference among the scores. The corresponding critical difference (CD) diagram for post-hoc Nemenyi test [45], which obtained a CD $= 1.088$, is also shown in Figure 4. The difference between the groups of classifiers is statistically significant. This statistical analysis clearly indicates that, for the imbalance present in the pre-miRNAs classification problem in bioinformatics, the best model is the deepBN. SVM and MLP are the worst models for this problem. Therefore, they will not be included in the following analyzes. While the three SOM-based deep architectures are equivalent according to F_1, deeSOM is capable of providing a much better precision. At maximum IR, deeSOM has a precision that is twice as good as the deepSOM precision. This final comparative result shows that the more recent deep neural network models can better handle the large imbalance present in this hard task. The deepSOM architecture and its variants can intrinsically handle the imbalance by filtering data from level to level, reducing the number of negative class samples. As explained above, the deepBN has an initial training stage in which the class labels are not used because its objective is to better model the complete feature space. This first unsupervised stage is very important to obtain a good internal representation while reducing the impact of the high imbalance present in the data.

The mean times taken to train each deep neural model in each imbalance level are reported in Figure 5, for the animals and plants datasets. It can be seen that all SOM models drop execution time as the imbalance ratio increases. Instead, deepBN significantly increases execution time. This can be explained by the fact that all layers of deepBN are trained with the complete training data, in each imbalanced situation. Instead, the deep SOM models discard a large amount of data during training, especially in the first levels. In the deeper levels, only the data that gets into the positive class neurons remains. The important time difference for 1:100 in the animals dataset can be explained by the deSOM training process. Since this IR has many positive samples in comparison to the

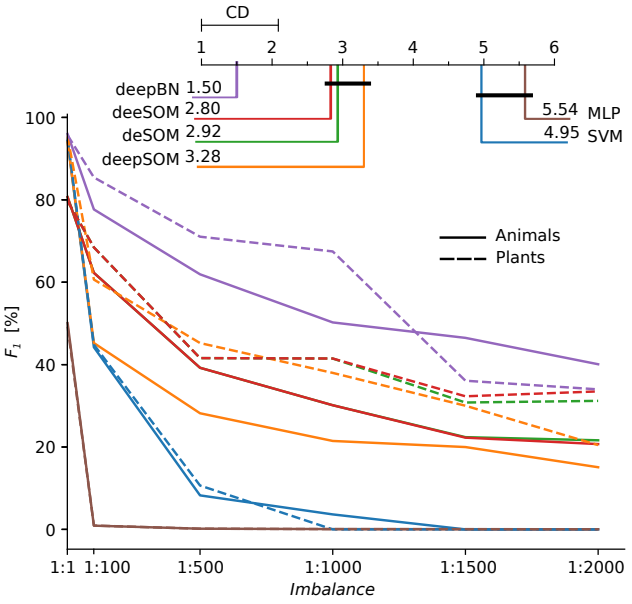

Fig. 4. F_1 score evolution for the deep versus classical approaches to the problem of pre-miRNA classification, with increasing IR in both datasets. Critical difference (CD) diagram for Nemenyi tests is shown above the curves.

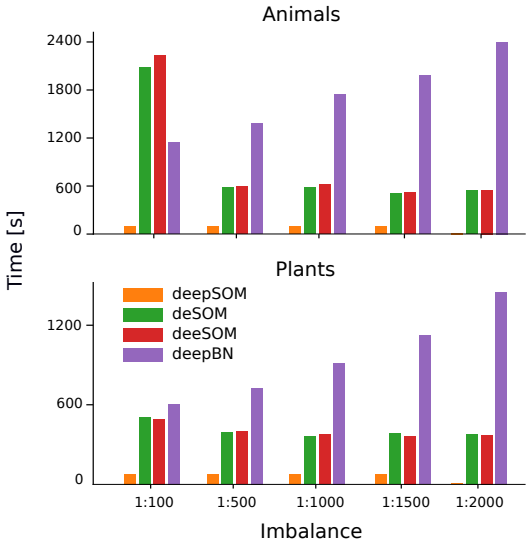

Fig. 5. Performance times in both datasets for each imbalance ratio.

other imbalances (see Table I), and positive samples are more spread in the feature space, there will be many more positive neurons in the first layers. As many samples get associated with these positive neurons, a low number of samples is discarded at the initial layers. Thus, the increased number of deep layers and their sizes require more training time. However, in the highest imbalances, for which the SOM-based models have been designed, the advantage in computing time is very important because the number of positive neurons is low. Therefore, the number of candidates is quickly reduced level after level.

As stated in Section III, the area under the precision recall curve (AUC_{PR}) is the best measure to analyze the performance in front of large class imbalance. The corresponding $\hat{AUC}_{PR}$ for the methods are presented in Tables VI and VII for the animals and plants datasets, respectively. These tables show that results are consistent with previous analysis, with deSOM and deeSOM improving deepSOM, and deepBN always reaching the highest scores. For deepBN, it can be seen that the score is very high in all cases. Moreover, from a very practical point of view, biologists often want to know how many wet experiments should be done to discover novel miRNAs in a given genome. In order to answer this, a detailed example of the pre-miRNA candidates returned by each method is presented in Figure 6. In this figure we propose a new graphical way for comparing classifiers performance in front of high class imbalance. Given the classification score of each method, for all test samples, the curves in the figure were generated from the sensitivity and number of candidates ($C = TP + FP$) found at each threshold level of the scores. Each figure shows the number of sequences considered as candidates at each level of sensitivity measured in the testing partitions, with IR ranging from 1:500 to 1:2,000. At the upper left corner is the total number of sequences for each dataset. Each point in the figure represents the output of each model in terms of number of candidates for a given sensitivity. It is important to note that the lower the value in the plot, the better is the model performance. In these plots, the curve where $FP = 0$ (that is, where all candidates are true positives), is the lowest bond for all methods. For example, let us take the case of imbalance 1:1,000 for animals dataset. In order to get a 0.90 of sensitivity, the deepBN model provides around 1,000 candidates, while for the same s^+, the deSOM architecture provides more than 10,000 candidates. Even worst, deepSOM would provide more than 100,000 candidates in order to discover the same number of new miRNAs.

For all imbalances in Figure 6, it can be seen that candidates are rapidly decreasing by several orders of magnitude. The initial slope in the curves shows how large quantities of low-quality candidates are being discarded. As it can be seen, for maximum recall all methods have the minimum precision, that is, a high number of false positives. As C falls, low-quality candidates are discarded (thus, precision is improved), at the cost of losing recall. In this experiment, the deepSOM was

TABLE VI
$\hat{AUC}_{PR}$ FOR ANIMALS DATASET

IR	deepSOM	deSOM	deeSOM	deepBN
1:100	0.72	0.83	0.83	0.96
1:500	0.68	0.78	0.78	0.94
1:1000	0.66	0.75	0.75	0.91
1:1500	0.67	0.73	0.75	0.90
1:2000	0.63	0.71	0.71	0.87

TABLE VII
$\hat{AUC}_{PR}$ FOR PLANTS DATASET

IR	deepSOM	deSOM	deeSOM	deepBN
1:100	0.82	0.86	0.86	0.97
1:500	0.79	0.79	0.79	0.95
1:1000	0.78	0.83	0.83	0.94
1:1500	0.75	0.79	0.78	0.87
1:2000	0.67	0.76	0.80	0.91

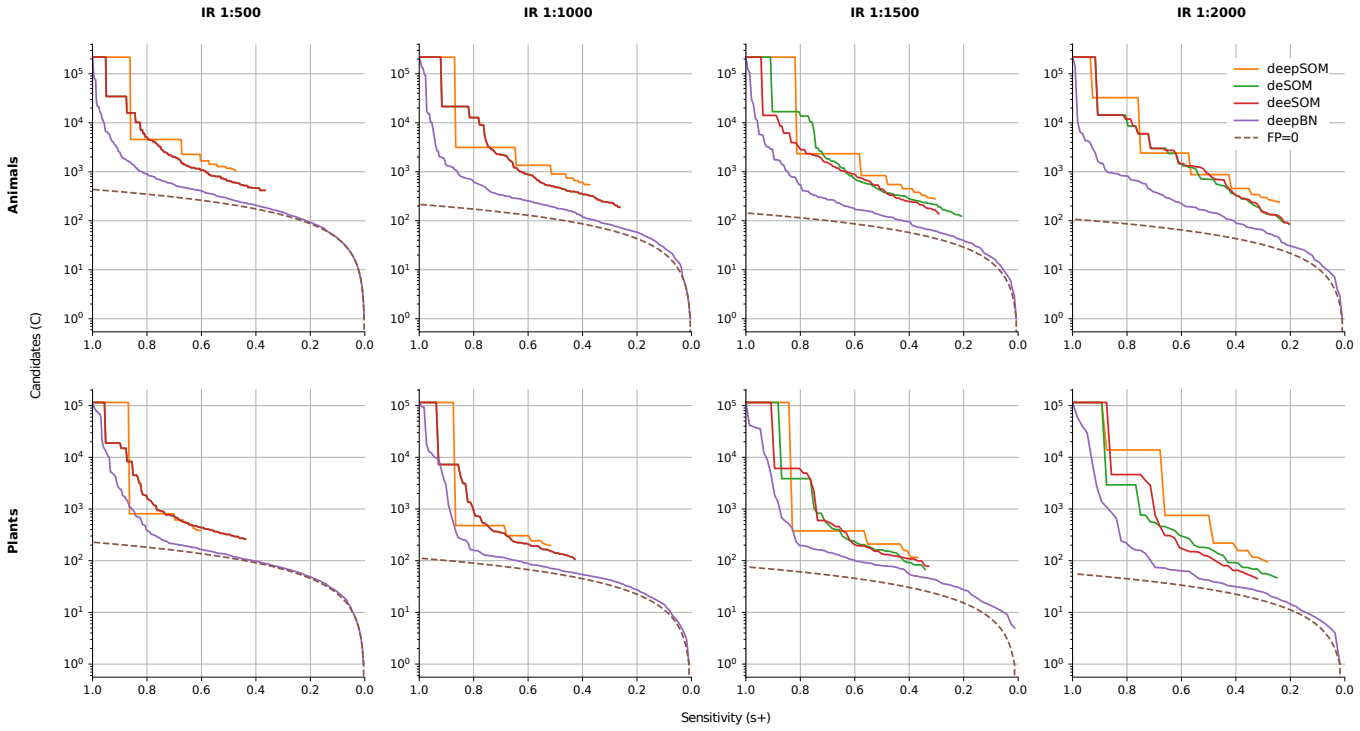

Fig. 6. Novel pre-miRNA candidates ($C = TP + FP$) versus sensitivity (s^+), on animals and plants datasets. The lower the curve, the better model performance. The dashed line is the theoretical curve in which there are no false positives ($FP = 0$).

unable to reduce the candidate number further than ∼600 sequences. This is because, differently from the elastic and ensemble topologies, it cannot adapt its size dynamically according to the data distribution in each level. Following now the evolution along the increasing imbalance, it can be seen that there is an increasing difference between deepSOM and the new SOM-based methods, which can reach better performance in the worst case scenario. The main differences can be seen in the right part of the curves, which is the most important from a practical point of view. It is interesting to note that even for very high imbalances, the methods can provide a small number of good quality candidates. For example, one could ask for the best 50 candidates of each method, which could be a reasonable number for wet experiments. In the plants dataset, for an IR of 1:2,000, deepSOM fails in this task as the smallest number of candidates that it can provide is 100, with sensitivity near 0.30. The deSOM can provide a s^+ of 0.25 (this is, 14 TP within the 50 candidates), the deeSOM provides a s^+ of 0.35 (20 TP) and deepBN reaches a sensitivity of 0.55 (31 TP). It can be seen that deepBN has an almost ideal behaviour for the top candidates, with a curve very close to $FP = 0$ for $C < 100$. These examples illustrate a very important aspect that should be measured in all the methods for pre-miRNA classification. Drastically reducing the number of candidates is an important factor to lower the costs of wet experimental confirmation of new pre-miRNAs. It can be stated that the deepSOM limitation in terms of the quality of candidates has been overcome by the novel adaptive SOM layers, which reached higher resolution. This fine grain reduction of candidates, up to only a few, allows the user to choose at which level each model should stop training, thus adjusting the precision and recall. Moreover, it can be seen that deepBN can provide a really low number of candidates, which are almost all TP, with very high precision. This can be of value from a practical point of view, where the number of high quality candidates is more important than a global accuracy measure.

V. Conclusion

In this work we have provided a comparative assessment of recent deep neural approaches for dealing with a highly imbalanced data problem in bioinformatics: the classification of pre-miRNAs. We presented and analyzed recent deep neural architectures proposals in a fair and controlled benchmark framework. Moreover, two novel deep SOM topologies capable of handling large class imbalance have been presented. The models have been compared in several classification tasks involving many genomes and increasing imbalance ratios, much larger than commonly published IRs. The comparative results obtained have shown that the model with deep learning including unsupervised generative training was the one capable of maintaining good performance rates, even at increasing IRs up to a very high imbalance.

Acknowledgements

This work was supported by National Scientific and Technical Research Council [PIP 2013 117], National University of Litoral [CAI+D 2011 548] and Agencia Nacional de Promocion Cientifica y Tecnologica (ANPCyT) [PICT 2014 2627].

REFERENCES

[1] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1119–1130, Aug 2012.

[2] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, 2009.

[3] M. Lin, K. Tang, and X. Yao, "Dynamic sampling approach to training neural networks for multiclass imbalance classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 647–660, 2013.

[4] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[5] S. Lin and R. Gregory, "Microrna biogenesis pathways in cancer," *Nature Reviews Cancer*, vol. 15, pp. 321–333, 2015.

[6] C. Croce and Y. Peng, "The role of MicroRNAs in human cancer," *Signal Transduction and Targeted Therapy*, no. 1, 2016.

[7] G. Bertoli, C. Cava, and I. Castiglioni, "MicroRNAs: New Biomarkers for Diagnosis, Prognosis, Therapy Prediction and Therapeutic Tools for Breast Cancer," *Theranostics*, vol. 5, no. 10, pp. 1122–1143, 2015.

[8] L. Li, J. Xu, D. Yang, X. Tan, and H. Wang, "Computational approaches for microRNA studies: a review," *Mammalian Genome*, vol. 21, no. 1-2, pp. 1–12, 2010.

[9] J. Allmer and M. Yousef, "Computational methods for ab initio detection of microrna," *Frontiers in Genetics*, vol. 3, no. 1, pp. 209–212, 2012.

[10] M. D. Saçar and J. Allmer, "Machine Learning Methods for MicroRNA Gene Prediction," in *Methods in molecular biology*. New York: Humana Press, Totowa, NJ, 2014, pp. 177–187.

[11] V. Shukla, V. Varghese, S. Kabekkodu, S. Mallya, and K. Satyamoorthy, "A compilation of web based research tools for mirna analysis," *Briefings in Functional Genomics*, vol. 1, no. 1, pp. 1–25, 2017.

[12] C. Xue, F. Li, T. He, G.-P. Liu, Y. Li, and X. Zhang, "Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine," *BMC Bioinformatics*, vol. 6, no. 1, p. 310, 2005.

[13] A. Sewer, N. Paul, P. Landgraf, A. Aravin, S. Pfeffer, M. J. Brownstein, T. Tuschl, E. van Nimwegen, and M. Zavolan, "Identification of clustered microRNAs using an ab initio prediction method," *BMC bioinformatics*, vol. 6, no. 1, p. 267, 2005.

[14] J. Hertel and P. F. Stadler, "Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data," *Bioinformatics*, vol. 22, no. 14, pp. e197–e202, 2006.

[15] S. A. Helvik, O. Snove, and P. Saetrom, "Reliable prediction of Drosha processing sites improves microRNA gene prediction." *Bioinformatics*, vol. 23, no. 2, 2007.

[16] T. H. Huang, B. Fan, M. Rothschild, Z. L. Hu, K. Li, and S. H. Zhao, "MiRFinder: an improved approach and software implementation for genome-wide fast microRNA precursor scans," *BMC Bioinformatics*, vol. 8, no. 1, pp. 341+, 2007.

[17] 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*, vol. 11, no. 11, p. S11, 2010.

[18] Y. Sheng, P. Engstrom, and B. Lenhard, "Mammalian MicroRNA prediction through a Support Vector Machine model of sequence and structure," *PLos ONE*, vol. 2, no. 9, p. e946, 2007.

[19] R. Batuwita and V. Palade, "*microPred*: effective classification of pre-mirnas for human mirna gene prediction," *Bioinformatics*, vol. 25, no. 8, pp. 989–995, 2009.

[20] D. Kleftogiannis, K. Theofilatos, S. Likothanassis, and S. Mavroudi, "YamiPred: A novel evolutionary method for predicting pre-miRNAs and selecting relevant features," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, 2015.

[21] P. Xuan, M. Guo, X. Liu, Y. Huang, W. Li, and Y. Huang, "*PlantMiRNAPred*: efficient classification of real and pseudo plant pre-mirnas," *Bioinformatics*, vol. 27, no. 10, pp. 1368–1376, 2011.

[22] Y. Wu and B. Wei and H. Liu and T. Li and S. Rayner, "MiRPara: a SVM-based software tool for prediction of most probable microRNA coding regions in genome scale sequences," *BMC Bioinformatics*, vol. 12, no. 1, p. 107, 2011.

[23] R. Peace and K. Biggar and K. Storey and J.R. Green, "A framework for improving microrna prediction in non-human genomes," *Nucleic Acids Research*, vol. 43, no. 20, p. e138, 2015.

[24] K. Huang and T. Lee and Y. Teng and T.H. Chang, "ViralmiR: a support-vector-machine-based method for predicting viral microRNA precursors," *BMC Bioinformatics*, vol. 16, no. 1, p. S9, 2015.

[25] B. Liu and L. Fang and J. Chen and F. Liu and X. Wang, "miRNA-dis: microRNA precursor identification based on distance structure status pairs," *Molecular BioSystems*, vol. 11, pp. 1194–1204, 2015.

[26] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and Promising Identification of Human MicroRNAs by Incorporating a High-quality Negative Set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, Jan. 2014.

[27] S. Min and B. Lee and S. Yoon, "Deep learning in bioinformatics," *Briefings in Bioinformatics*, vol. 18, no. 5, p. 851869, 2017.

[28] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 1, no. 1, pp. 1–17, 2018.

[29] Z. Si, H. Yu, and Z. Ma, "Learning deep features for DNA methylation data analysis," *IEEE Access*, pp. 2732–2737, 2016.

[30] J. Thomas and S. Thomas and L. Sael, "DP-miRNA: An Improved Prediction of precursor microRNA using Deep Learning Model," *IEEE Int. Conf. on Big Data and Smart Computing*, vol. 1, no. 1, pp. 96–99, 2017.

[31] G. Stegmayer and C. Yones and L. Kamenetzky and D.H. Milone, "High class-imbalance in pre-miRNA prediction: a novel approach based on deepSOM," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 6, pp. 1316–1326, 2017.

[32] 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*, vol. 1, pp. 14–36, 2012.

[33] N. LeRoux and Y. Bengio, "Representational Power of Restricted Boltzmann Machines and Deep Belief Networks," *Neural Computation*, vol. 6, no. 20, pp. 1331–1649, 2008.

[34] G. Stegmayer, M. Gerard, and D. Milone, "Data mining over biological datasets: an integrated approach based on computational intelligence," *IEEE Computational Intelligence Magazine, Special Issue on Computational Intelligence in Bioinformatics*, vol. 7, no. 4, pp. 22–34, 2012.

[35] A. Gudy, M. Szczeniak, M. Sikora, and I. Makalowska, "HuntMi: an efficient and taxon-specific approach in pre-miRNA identification," *BMC Bioinformatics*, vol. 14, no. 1, pp. 83+, 2013.

[36] C. Yones, G. Stegmayer, L. Kamenetzky, and D. Milone, "miRNAfe: a comprehensive tool for feature extraction in microRNA prediction," *BioSystems*, vol. 238, pp. 1–5, 2015.

[37] T. Li, X. Zhang, F. Luo, F. Wu, and J. Wang, "Multimotifmaker: a multi-thread tool for identifying dna methylation motifs from pacbio reads," *IEEE/ACM Transactions on Computational Biology and Bioinformatics, in press*, 2018.

[38] T. Zhao, N. Zhang, Y. Zhang, J. Ren, P. Xu, Z. Liu, L. Cheng, and Y. Hu, "A novel method to identify pre-microrna in various species knowledge base on various species," *Journal of Biomedical Semantics*, vol. 8, no. 30, pp. 1679–1688, 2017.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[40] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, no. 1, p. 106, 2013.

[41] C. Huang, C. C. Loy, and X. Tang, "Discriminative sparse neighbor approximation for imbalanced learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1503–1513, 2018.

[42] T. Saito, M. Rehmsmeier, L. Hood, O. Franco, R. Pereira, and K. Wang, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, vol. 10, no. 3, 2015.

[43] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring Strategies for Training Deep Neural Networks," *Journal of Machine Learning Research*, vol. 1, pp. 1–40, 2009.

[44] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[45] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.

Leandro A. Bugnon received the Bioengineering degree (Hons.) from Universidad Nacional de Entre Ríos (UNER), Argentina, in 2013 and the Ph.D. degree on Engineering oriented to Computational Intelligence, Signals and Systems from Universidad Nacional del Litoral (UNL), Argentina, in 2018. He is with the Research Institute for Signals, Systems and Computational Intelligence - sinc(i) (FICHUNL/ CONICET). He has a postdoctoral fellowship at the National Scientific and Technical Research Council (CONICET). His research interests include automatic learning, pattern recognition, signal and image processing, with applications to affective computing, biomedical signals and bioinformatics.

Cristian Yones received the Computer Engineering degree in 2014 and the Ph.D. degree on Engineering oriented to Computational Intelligence, Signals and Systems in 2018, both from Universidad Nacional del Litoral (UNL), Argentina. He has a postdoctoral fellowship at the National Scientific and Technical Research Council (CONICET). His research interests include machine learning, data mining, semi-supervised learning, with applications in bioinformatics.

Diego H. Milone received the Bioengineering degree (Hons.) from National University of Entre Rios (UNER), Argentina, in 1998, and the Ph.D. degree in Microelectronics and Computer Architectures from Granada University, Spain, in 2003. He was with the Department of Bioengineering and the Department of Mathematics and Informatics at UNER from 1995 to 2002. Since 2003 he is Full Professor in the Department of Informatics at National University of Litoral (UNL). From 2009 to 2011 was Director of the Department of Informatics and from 2010 to 2014 was Assistant Dean for Science and Technology. Since 2006 he is a Research Scientist at the National Scientific and Technical Research Council (CONICET). Since 2015 he is Director of the Research Institute for Signals, Systems and Computational Intelligence (CONICET-UNL). His research interests include statistical learning, pattern recognition, signal processing, neural and evolutionary computing, with applications to speech recognition, affective computing, biomedical signals and bioinformatics.

Georgina Stegmayer received the Engineering degree in Information Systems from UTN-FRSF, Argentina, in 2000, and the Ph.D. degree from Politecnico di Torino, Italy, in 2006. Since 2007 she is Assistant Professor of Artificial Intelligence and Computationl Intelligence in UNL University in Argentina. She is currently Independent Researcher at the National Scientific and Technical Research Council (CONICET) of Argentina. She is author and co-author of numerous papers on journals, book chapters and conference proceedings on artificial neural networks for a wide variety of problems. Her current research interests involve machine learning, data mining and pattern recognition in bioinformatics.

Supplementary Material
Deep neural architectures for highly imbalanced data in bioinformatics

L. A. Bugnon, *Member, IEEE*, C. Yones, D. H. Milone *Senior Member, IEEE*, G. Stegmayer

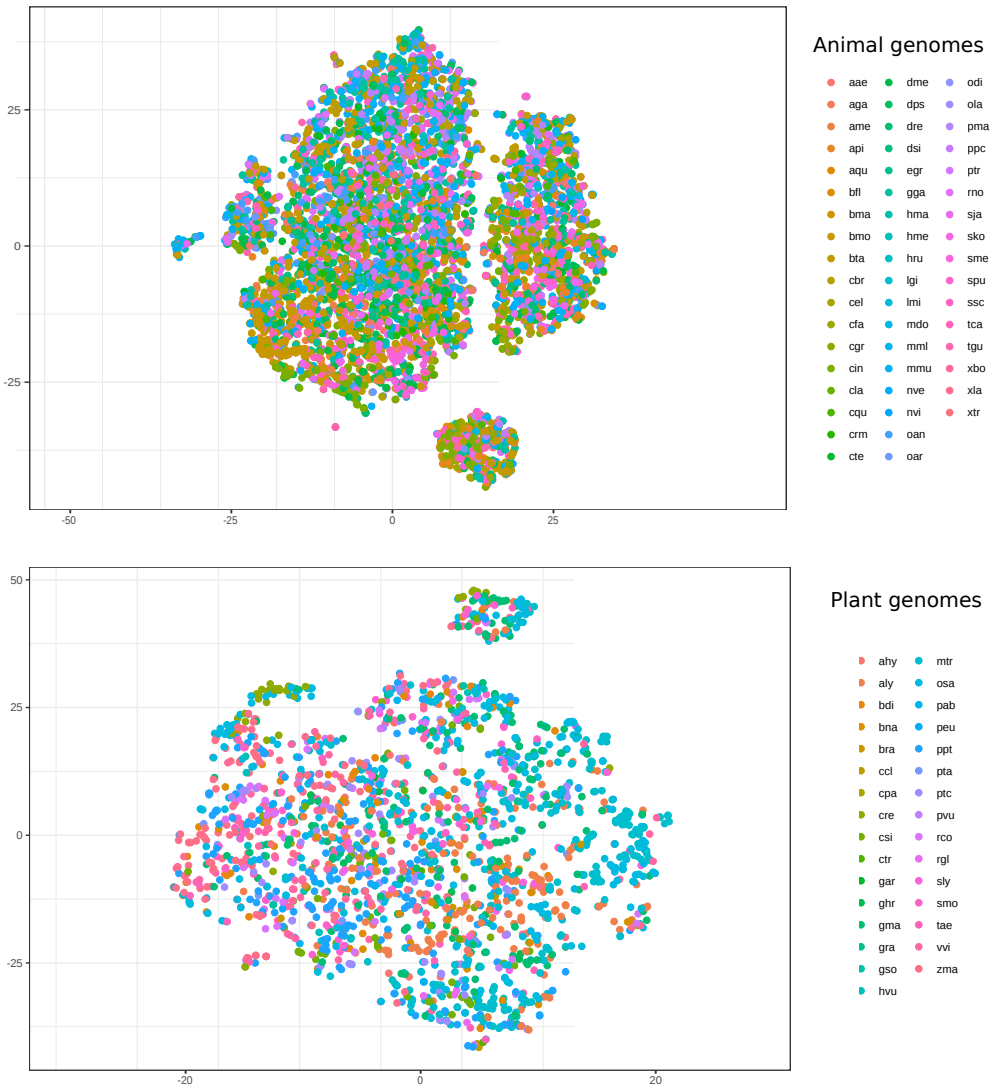

Figure S 1. Animals and plants data distribution in the feature space (t-SNE projection) of the positive class samples. A different color has been used for each species well-known pre-miRNAs. The complete list of genomes contained in the dataset is the following. In Animals: *Aedes aegypti* (aae), *Anopheles gambiae* (aga), *Apis mellifera* (ame), *Acyrthosiphon pisum* (api), *Amphimedon queenslandica* (aqu), *Branchiostoma floridae* (bfl), *Brugia malayi* (bma), *Bombyx mori* (bmo), *Bos taurus* (bta), *Caenorhabditis briggsae* (cbr), *Caenorhabditis elegans* (cel), *Canis familiaris* (cfa), *Cricetulus griseus* (cgr), *Ciona intestinalis* (cin), *Cerebratulus lacteus* (cla), *Culex quinquefasciatus* (cqu), *Caenorhabditis remanei* (crm), *Capitella teleta* (cte), *Drosophila melanogaster* (dme), *Drosophila pseudoobscura* (dps), *Danio rerio* (dre), *Drosophila simulans* (dsi) *Echinococcus granulosus* (egr), *Gallus gallus* (gga), *Hydra magnipapillata* (hma), *Heliconius melpomene* (hme), *Haliotis rufescens* (hru), *Lottia gigantea* (lgi), *Locusta migratoria* (lmi), *Monodelphis domestica* (mdo), *Macaca mulatta* (mml), *Mus musculus* (mmu), *Nematostella vectensis* (nve), *Nasonia vitripennis* (nvi), *Ornithorhynchus anatinus* (oan), *Ovis aries* (oar), *Oikopleura dioica* (odi), *Oryzias latipes* (ola), *Petromyzon marinus* (pma), *Pristionchus pacificus* (ppc), *Pan troglodytes* (ptr), *Rattus norvegicus* (rno), *Schistosoma japonicum* (sja), *Saccoglossus kowalevskii* (sko), *Schmidtea mediterranea* (sme), *Strongylocentrotus purpuratus* (spu), *Sus scrofa* (ssc), *Tribolium castaneum* (tca), *Taeniopygia guttata* (tgu), *Xenoturbella bocki* (xbo), *Xenopus laevis* (xla), and *Xenopus tropicalis* (xtr). In Plants: *Arachis hypogaea* (ahy), *Arabidopsis lyrata* (aly), *Brachypodium distachyon* (bdi), *Brassica napus* (bna), *Brassica rapa* (bra), *Citrus clementina* (ccl), *Carica papaya* (cpa), *Chlamydomonas reinhardtii* (cre), *Citrus sinensis* (csi), *Citrus trifoliata* (ctr), *Gossypium arboreum* (gar), *Gossypium hirsutum* (ghr), *Glycine max* (gma), *Gossypium raimondii* (gra), *Glycine soja* (gso), *Hordeum vulgare* (hvu), *Medicago truncatula* (mtr), *Oryza sativa* (osa), *Picea abies* (pab), *Populus euphratica* (peu), *Physcomitrella patens* (ppt), *Pinus taeda* (pta), *Populus trichocarpa* (ptc), *Phaseolus vulgaris* (pvu), *Ricinus communis* (rco), *Rehmannia glutinosa* (rgl), *Solanum lycopersicum* (sly), *Selaginella moellendorffii* (smo), *Triticum aestivum* (tae), *Vitis vinifera* (vvi), and *Zea mays* (zma).

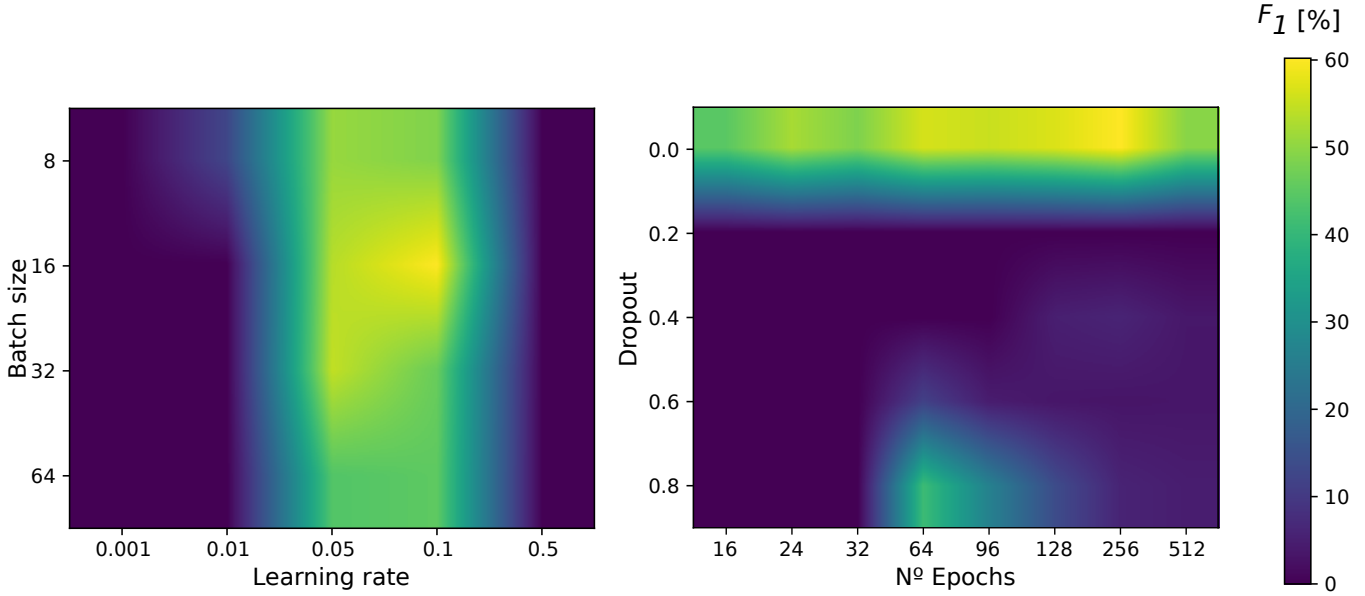

Figure S 2. F_1 score stability for different hyperparameters of deepBN on plants dataset, with imbalance 1:1000. On the left, batch size versus learning rate comparison. On the right, dropout versus the number of training epochs.