Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Complexity measures of the mature miRNA for improving pre-miRNAs prediction

Jonathan Raad*, Georgina Stegmayer and Diego H. Milone

Research Institute for Signals, Systems and Computational Intelligence sinc(i) (FICH-UNL/CONICET), Ciudad Universitaria, Santa Fe, Argentina.

*To whom correspondence should be addressed.

Associate Editor: - - - - -

Received on - - - -; revised on - - - -; accepted on - - - -

Abstract

Motivation: The discovery of microRNA (miRNA) in the last decade has certainly changed the understanding of gene regulation in the cell. Although a large number of algorithms with different features have been proposed, they still predict an impractical amount of false positives. Most of the proposed features are based on the structure of precursors of the miRNA (pre-miRNA) only, not considering the important and relevant information contained in the mature miRNA. Such new kind of features could certainly improve the performance of the predictors of new miRNAs.

Results: This paper presents three new features that are based on the sequence information contained in the mature miRNA. We will show how these new features, when used by a classical supervised machine learning approach as well as by more recent proposals based on deep learning, improve the prediction performance in a significant way. Moreover, several experimental conditions were defined and tested in order to evaluate the novel features impact in situations close to genome-wide analysis. The results show that the incorporation of new features based on the mature miRNA allow to improve the detection of new miRNAs independently of the classifier used.

Availability: https://sourceforge.net/projects/sourcesinc/files/cplxmirna/

Contact: jraad@sinc.unl.edu.ar

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In the recent decades, the discovery of new non-coding RNA molecules has changed the understanding of gene regulation in the cell. One of those molecules that caught most of the attention of the scientific community has been the microRNA (miRNA), due to its importance in the promotion or inhibition of several diseases (Bartel, 2004; Takahashi *et al.*, 2015). The miRNAs are small RNA molecules, approximately 21 bases long, which regulate gene expression in animal and plant cells through post-transcriptional control (Bartel, 2004). Given their proven role in promoting or inhibiting genes, the discovery of more miRNAs is of high interest today. Up to date, there are 38,589 miRNAs in miRBase v22[1]. Small RNA deep sequencing datasets have been used in order to support their validity. The read mapping patterns provided strong support for between 20% to 65% (depending on the species) microRNA annotations (Kozomara *et al.*, 2019). It is expected that the number of miRNAs continues growing. In fact, it has been increasing with every new release of miRBase: in v19 there were 25,141 and 30,582 in v21.

In a genome, the miRNAs are stored inside precursors that allow their recognition (Bartel, 2004). Precursors of miRNAs (pre-miRNAs) are molecules of 100 bases long approximately, which have a stem-loop structure. Experimental methods for detecting pre-miRNAs can be performed with different techniques, such as quantitative real-time PCR (qPCR), microarray and deep sequencing. These techniques present some practical difficulties when evaluating a large number of candidates. First, both qPCR and microarray suffer from low specificity and need extensive normalization (Baker, 2010; Dong *et al.*, 2013). In addition, prior knowledge is needed for the design of primers for qPCR and target sequences for microarrays, which does not allow finding novel pre-miRNAs (Pritchard *et al.*, 2012). In the case of deep sequencing, prior knowledge is not necessary but this technique is hampered by the need

[1] http://www.mirbase.org/

of extensive downstream computational analysis (Demirci *et al.*, 2017). Due to these technical and practical difficulties in detecting pre-miRNAs, computational methods have been playing an increasingly important role for their prediction (Li *et al.*, 2010; de ON Lopes *et al.*, 2014).

Among computational methods, two main prediction strategies can be considered: rule-based (RB) and machine learning (ML) based algorithms. RB algorithms evaluate measures of each sequence against reference values obtained from known pre-miRNAs. Two examples of RB tools are (Mathelier and Carbone, 2010; Friedländer *et al.*, 2011). ML based algorithms require a training step on features calculated from known pre-miRNAs and a negative set. Several RB and ML based tools were revised in (Bortolomeazzi *et al.*, 2017). The adjustment of parameters for each methods can be done automatically (by grid search or learnt from data) or manually. For example, if a given distance is calculated among sequences, a threshold must be set. If the prediction method is used with other data (for example, a newer version of miRBase), this threshold will have to be manually adjusted again. Instead, a threshold (or any other parameter) that can be automatically learnt according to data distribution, as in ML, could be used with these and with other newer data, without requiring a manual readjustment by an expert. A large number of approaches based on ML have emerged recently, for example with random forests (Vitsios *et al.*, 2017), support vector machines (Tseng *et al.*, 2017), graph based semi-supervised learning model (Yones *et al.*, 2018), and deep neural architectures (Bugnon *et al.*, 2019). Most of them propose novel ML models using a standard feature extraction. Differently, in this work we will propose novel features and will test them with standard ML classifiers. Many reviews have analysed the advantages of ML tools. For example (Chen *et al.*, 2018) reviews 20 miRNA bioinformatics tools published before 2018, where 11 out of 20 are ML-based. It concluded that classic ML methods, such as support vector machines, are still popularly used in the miRNA field, while novel and more advanced deep learning methods are beginning to appear. In (Stegmayer *et al.*, 2018), 29 pre-miRNA ML-based prediction tools published in the last 10 years are included. (Morgado and Johannes, 2017), affirmed that ML models can capture more general features that other approaches, which allows them to better detect miRNA sequences and precursors, even those with low similarity to the reference set. In (Liu, 2017) is analyzed in detail a web-server that can construct a very large variety of ML predictors for miRNAs. It is based on the fact that ML learning techniques are playing key roles in this field nowadays, but they can be cumbersome to build and use. Thus, this web server has been proposed to automatically complete the main steps for constructing a ML-predictor. A recent study (Demirci *et al.*, 2017) has shown that the computational prediction of pre-miRNAs is yet far-away from being satisfactory solved.

In order to find new candidates for pre-miRNA, structural and sequence characteristics of hairpins in a genome have to be extracted to train an ML classifier (Li *et al.*, 2010; de ON Lopes *et al.*, 2014; Shukla *et al.*, 2017). In the literature, many different features sets have been proposed, which mostly describe information of the structure of the pre-miRNA inspired by the action of Drosha (de ON Lopes *et al.*, 2014). However, although the microprocessor can takes a leading role in choosing which RNA precursors encode a miRNA, the specificity of the subsequent processes can impose additional restrictions on those hairpins that will eventually become mature miRNA (Bartel, 2018). In addition, in different studies it has been found that the selectivity of the miRNA for the target mRNA is defined by the sequence of the corresponding mature miRNA (Friedman *et al.*, 2009; Lewis *et al.*, 2005; Brennecke *et al.*, 2005; Bartel, 2009). Specifically, the mature miRNA contains two areas of union with the target sequence called seed and complementary site (Friedman *et al.*, 2009). Due to the importance that the seed has in the sequence function, the mature miRNAs can be classified on the basis of the presence of identical seed sequences into groups called miRNA families (Lewis *et al.*, 2003). In fact, some authors have proposed automatic classifiers for miRNAs families (Zou *et al.*, 2014). Therefore, given that important information is codified in the mature region, the secondary structure of the precursor by itself might not be sufficient to differentiate a true pre-miRNA from other hairpins. Our hypothesis is that the main difficulty in separating both classes is due to the omission of relevant information regarding the mature miRNA sequence in the description (feature extraction process) of the pre-miRNAs. This fact is especially notable in the prediction of novel precursors, where the features are extracted mainly from the sequences structure. A typical example of this kind of standard features (SF) is the triplets representation (Xue *et al.*, 2005), which considers the structural composition of three adjacent nucleotides and the middle base to build a vector with 32 elements. Other examples are the number of internal loops and their length (Yousef *et al.*, 2006), the z-score of the minimum free energy (Hertel and Stadler, 2006), the dinucleotide proportion (Batuwita and Palade, 2009), base pair proportion, G+C content in the terminal loop (de ON Lopes *et al.*, 2014), Shannon's entropy (zQ), base pair propensity (zP) (Ng and Mishra, 2007) and base pair distance (zD) (Ding *et al.*, 2010). Although many features have been proposed, those are mostly based on the secondary structure of pre-miRNA or the relative frequencies of dinucleotides, trinucleotides and motifs in these sequences (de ON Lopes *et al.*, 2014; Yones *et al.*, 2015). These features have been performing quite well on current classifiers (Stegmayer *et al.*, 2018). However, it can be stated that these SF do not allow to represent nor to preserve the information regarding the order in which these triads and motifs are present in the sequence, losing valuable information regarding the coding of the mature miRNA within a sequence itself.

In this work, we propose three new features that take particularly into account the order in which the nucleotides are presented in the mature miRNA, which can effectively improve the sequence representation. We will show how these novel features can improve the prediction of novel pre-miRNAs, independently of the classifier. One of the proposed features is based on the Levenshtein distance. The rationale behind it is that candidate sequences to be new miRNAs should be very similar in the region encoding the mature, and Levenshtein distance can measure it in terms of nucleotides editions. This distance has been used in other areas of bioinformatics like sequence alignment, and also to estimate the proximity between sequences (Zytnicki *et al.*, 2008; Lassmann and Sonnhammer, 2005; Billoud *et al.*, 2013). The first algorithm for global alignment was proposed as a modification of the Levenshtein distance (Needleman and Wunsch, 1970), where the problem was formulated in terms of maximizing the similarity between sequences. Subsequently, different approaches appeared such as local and semi-global alignment. The local alignment seeks to align dissimilar sequences that contain small regions of similarity in large contexts (Polyanovsky *et al.*, 2011). The semi-global alignments are used to align short sequences with large sequences, through a global alignment of the first and a local alignment of the second one (Brudno *et al.*, 2003). However, the reason why the Levenshtein distance was chosen in our work is for obtaining a numerical measure to better quantify the distance (and not maximizing the similarity) between two short sequences (mature miRNAs). Therefore, due to the conservation and the evolution of miRNAs (Wheeler *et al.*, 2009), we will show how the chains that codify the mature miRNA of possible pre-miRNA sequences are closer in this space than those that do not encode miRNAs. This way it is possible to calculate, for each candidate sequence, a distance to labeled pre-miRNAs in order to evaluate how close each candidate is to these pre-miRNA samples. Differently from (Mathelier and Carbone, 2010), where the Levenshtein distance is used as a direct calculation of the edition errors with a threshold for eliminating sequences as a first step of the processing, in our work we build a statistic that can estimate the belonging of the candidate sequence to the set of positive class examples. This way, the Levenshtein distance as a feature is more general and applicable to any

species, and can be used by any classifier. The second and third proposed features were inspired, from the point of view of the information theory, considering the randomness of a sequence that would encode a mature miRNA in the hairpin. In addition, it is known that certain mature regions have specific motifs that define their functionality and the belonging to a specific miRNA family (Bartel, 2018, 2009). In order to quantify this fact, we propose a permutation entropy (Bandt and Pompe, 2002) feature and a measure of the Lempel-Ziv complexity (Ziv and Lempel, 1978) of the sequences. We have measured the performance of these new features when used by classical supervised machine learning approaches such as Naive Bayes (NB), Random Forest (RF), k-nearest neighbor (KNN) and more recent proposals based on deep neural networks (DNN).

2 Novel features based on complexity measures

2.1 Levenshtein distance

During evolution, many miRNAs were mostly preserved among different species, sometimes suffering modifications that resulted in new miRNAs. Despite these modifications over time, the preservation of specific sequences such as the seeds of mature miRNAs has been studied, defining functionality as well as the belonging to a specific family (Bartel, 2018). This leads us to believe that the sequences that can be candidates to new pre-miRNAs should be very similar in the region encoding a mature. In other words, as a result of evolution, one would expect to have a small nucleotide edit distance in those sequences that can effectively encode miRNAs.

The Levenshtein distance, L, also known as edit distance between strings, is defined as the minimum number of operations (insertions, deletions or substitutions) required to transform one string into another one (Levenshtein, 1966). This distance between two strings $\boldsymbol{x}$ and $\boldsymbol{y}$, of lengths $|\boldsymbol{x}|$ and $|\boldsymbol{y}|$, can be calculated according to Algorithm 1. The algorithm begins verifying that both chains have a length greater than zero (line 1). If either of the two does not satisfy the condition, the algorithm returns the length of the other chain (line 2), that is, the number of insertions necessary to build it from an empty chain. If both chains satisfy the previous condition, a matrix D of $|\boldsymbol{x}|+1$ rows and $|\boldsymbol{y}|+1$ columns is created where the first row is initialized with values from 0 to $|\boldsymbol{x}|$, and the first column from 0 to $|\boldsymbol{y}|$ (lines 4 and 5). Then for each element $d_{i,j}$ in the matrix D, it is verified if x_i is equal to y_j. If this equality is satisfied, no editing operation is required. Otherwise, since one string chain can be obtained in different ways from the other one, we want to find the strings that require the fewest editing operations in relation to the other one(that is, the minimum edit distance between them). For this purpose, the minimum value of the three possible string operations is obtained in line 9, where the $d_{i-1,j}+1$, $d_{i,j-1}+1$ and $d_{i-1,j-1}+c$ corresponding to the operations of insertion, deletion and substitution, respectively. The variable c corresponds to a substitution cost. It is calculated in line 8, where $\delta(x_i, y_j)$ is the Dirac delta. The cost c is equal to 0 when both characters are equal, and 1 otherwise. It must be noted that for insertion and deletion, cost is always 1. Finally, the value found in last element of D, $d_{|\boldsymbol{x}|,|\boldsymbol{y}|}$, is assigned as the Levenshtein distance between the analyzed chains (line 10). Since this measure adds insertion steps when two chains have different lengths, it is necessary to define a way to be able to compare the distances between pairs of candidates, regardless their individual lengths are different. That is why in line 10 each distance is adjusted by subtracting the absolute difference of the lengths of the strings under analysis.

In order to be able to calculate L as a feature for each hairpin sequence, and since L is a distance between two elements, it is necessary to have a reference set for comparison. Let be $\mathcal{A}$ the set with the miRNA matures $\boldsymbol{a}_k$. Let $\boldsymbol{a}_\ell$ an element of $\mathcal{A}$ for which we wants to obtain the L feature.

Algorithm 1: Levenshtein distance

Input : $\boldsymbol{x}$, $\boldsymbol{y}$ RNA sequence strings
Output: L Levenshtein distance

1 **if** $|\boldsymbol{x}||\boldsymbol{y}| = 0$ **then**
2 $\quad L \leftarrow \max\{|\boldsymbol{x}|, |\boldsymbol{y}|\}$
3 **else**
4 $\quad d_{i,0} \leftarrow i \ \forall i$
5 $\quad d_{0,j} \leftarrow j \ \forall j$
6 $\quad$ **for** $i \leftarrow 1$ **to** $|\boldsymbol{x}|$ **do**
7 $\quad\quad$ **for** $j \leftarrow 1$ **to** $|\boldsymbol{y}|$ **do**
8 $\quad\quad\quad c \leftarrow 1 - \delta(x_i, y_j)$
9 $\quad\quad\quad d_{i,j} \leftarrow \min\{d_{i-1,j}+1,\ d_{i,j-1}+1,\ d_{i-1,j-1}+c\}$
10 $\quad L \longleftarrow d_{|\boldsymbol{x}|,|\boldsymbol{y}|} - ||\boldsymbol{x}| - |\boldsymbol{y}||$
11 **return** L

Then, the median of the distance of $\boldsymbol{a}_\ell$ to all the other elements of the set can be as feature of $\boldsymbol{a}_\ell$, that is

$$L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \text{med}_{\forall k\neq\ell}\{\boldsymbol{a}_k, \boldsymbol{a}_\ell\}, \tag{1}$$

where $\mathcal{A}\setminus\boldsymbol{a}_\ell$ is the set $\mathcal{A}$ without the element $\boldsymbol{a}_\ell$. Then, each candidate can have its mature coding in different regions (5p or 3p), it is necessary to extract two chains $\boldsymbol{a}_\ell^{5p}$ and $\boldsymbol{a}_\ell^{3p}$. Thus, two L measures for each $\boldsymbol{a}_\ell$ are obtained and the maximum edit value between both $L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p})$ and $L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})$ is selected as the final $L(\boldsymbol{a}_\ell)$. That is, the L feature is not based on the distance to the primary mature strand alone, but also to its corresponding complementary star strand as well. When the distance with respect to both strands is calculated, selecting afterwards the maximum, both strands must comply with a certain minimum distance to the known miRNAs so that the L feature evidences a miRNA. That is to say, this way, none of the two strands has an excessive distance to the known pre-miRNAs.

2.2 Permutation entropy

The section in the hairpin that encodes the mature miRNA contains specific patterns of the nucleotides order in its seed and in its complementary region (Friedman *et al.*, 2009; Lewis *et al.*, 2005; Bartel, 2009). Thus, it can be expected that pre-miRNAs have less randomness in that section than any other sequences. Therefore, a measure capable of quantifying such randomness in sequence patterns could be useful to detect the true pre-miRNAs.

The Shannon entropy is widely used to measure the randomness of a sequence: the more random, the larger the entropy (Shannon, 2001). The drawback of this approach when analyzing miRNA sequences is that the information of the internal order of the nucleotides is lost when calculating the relative frequencies. To solve this, Bandt and Pompe in (Bandt and Pompe, 2002) proposed a new coding based on permutation patterns in the sequence, where the entropy is estimated from the relative frequencies of these patterns. The measure was called permutation entropy (PE). In this case, the probability distribution of $\boldsymbol{x}$ was replaced by the relative frequencies $p_{\boldsymbol{\pi}}$ of all possible patterns $\boldsymbol{\pi}$ that can be found within $\boldsymbol{x}$.

When working with PE, it is necessary to previously choose the length of the patterns to be permuted. This parameter is called order N. Thus, defined the order, $N!$ patterns $\boldsymbol{\pi}$ of length N are obtained. For example, selecting $N = 3$, then 6 possible patterns are possible: (1,2,3) (1,3,2) (2,1,3) (3,2,1) (3,1,2) (2,3,1). If the frequencies of these patterns are calculated in $\boldsymbol{x}$, then the corresponding PE can be estimated as

$$PE_N(\mathbf{x}) = -\sum_{i=1}^{N!} p_{\boldsymbol{\pi}_i} \cdot \log_2(p_{\boldsymbol{\pi}_i}), \tag{2}$$

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Raad, G. Stegmayer & D. H. Milone; "Complexity measures of the mature miRNA for improving pre-miRNAs prediction"
Bioinformatics, 2020.

When N is too small, relevant information from the system dynamics cannot be captured. On the other hand, if N is very large, the sequence will require a longer length in order to obtain a good estimation of the probability of each pattern. Therefore, as a practical rule (Bandt and Pompe, 2002), N must be selected in such a way that $N! \ll |\boldsymbol{x}|$. In the case of RNA sequences, they are encoded in an alphabet of 4 nucleotides that can form different combinations. In order to analyze as many combinations as possible, and due to the fact that the mature sequences have an approximate length of 25 nt, N should be just 2 or 3.

2.3 Lempel-ziv complexity

When observing the specificity of the mature sequence with respect to its corresponding target mRNA, from an information theory point of view, there must be syntactic rules that avoid any random mutation to modify their function. In other words, the coding of a mature sequence should be contained in a 'dictionary', so that more complex combinations of nucleotides are constructed from simpler combinations. Since the sequence of a mature must be encoded only by specific 'words', it is expected for those candidates that encode miRNA to have a smaller dictionary than those candidates that do not. Therefore, it could be very useful to have a measure to quantify this complexity in a sequence of nucleotides.

The Lempel-Ziv (LZ) algorithm allows the calculation of such complexity in a finite sequence based on the analysis of its "production process" (Lempel and Ziv, 1976). Let $\boldsymbol{a}$ be a RNA sequence, which is composed of the 4 nucleotides. We define $\boldsymbol{a}(i,j)$ as a subsequence of $\boldsymbol{a}$ that is composed of the elements that are between the indices i and j. We say that $\boldsymbol{a}$ is reproducible from $\boldsymbol{a}(1,j)$, if $\boldsymbol{a}(j+1,|\boldsymbol{a}|)$ is a sub-word of $\boldsymbol{a}$ that is contained in $\boldsymbol{a}(1,j)$. Then, we say that $\boldsymbol{a}$ is producible from $\boldsymbol{a}(1,j)$, if we add a new element at the end of the sequence $\boldsymbol{a}$ that cannot be obtained by reproducing $\boldsymbol{a}(1,j)$. In other words, a chain $\boldsymbol{a}$ can be obtained from the extension of smaller chains by two processes: reproduction (when the extension is done by copying a substring of the smallest chain) or production (when the extension is done by a new substring that is not contained in the initial chain). For example, given the sequence ACACCA, we can obtain the dictionary A | C | AC | CA. Then, the sequence ACACCACAA is obtained by production when adding a new substring CAA that is not contained in the dictionary. However, the chain ACACCAAC is obtained from the original sequence ACACCA by reproduction of AC element.

If we concatenate all the processes by which the chain $\boldsymbol{a}$ can be formed, the history of its construction $H(\boldsymbol{a})$, is obtained. With this history, we can measure the complexity of such construction as the number of steps necessary to generate it. In addition, since it is possible to obtain a chain from another one in different ways, we are interested in finding the history that has the minimum necessary number of steps. If we consider each step of the process as reproduction or production, then $\boldsymbol{a}$ can be analyzed as a process of z steps $H(\boldsymbol{a}) = H_1(\boldsymbol{a})H_2(\boldsymbol{a})...H_z(\boldsymbol{a})$ with $h_0 \equiv 0$.

Then, let $|H(\boldsymbol{a})|$ be the number of steps in $H(\boldsymbol{a})$. The Lempel-Ziv complexity of a sequence $\boldsymbol{a}$ is thus defined as $\mathrm{lz}(\boldsymbol{a}) = \min\{|H(\boldsymbol{a})|\}$, regarding all the histories of $\boldsymbol{a}$. Then, to obtain a measure that is independent of the length of $\boldsymbol{a}$,

$$LZ(\boldsymbol{a}) = \frac{\mathrm{lz}(\boldsymbol{a}) \log_4 |\boldsymbol{a}|}{|\boldsymbol{a}|}, \tag{3}$$

where 4 in the base of the logarithm represents the number of nucleotides.

3 Materials, measures and experimental setup

3.1 Datasets

For this study we have created a number of datasets of varying ratios of class imbalance, testing pre-miRNA predictors with and without the proposed new features. We have used an already available public dataset (Gudyś *et al.*, 2013), which provides negative and positive samples of all known pre-miRNAs in miRBase (Kozomara and Griffiths-Jones, 2010) for *Homo sapiens* (1,406 positives and 81,228 negatives). The standard features are those used in the mostly cited works (see details in the Supplementary Material) (Stegmayer *et al.*, 2018; Jiang *et al.*, 2007; Gudyś *et al.*, 2013; Batuwita and Palade, 2009). The varying ratios of class imbalance allows to evaluate the robustness of the new features in situations closer to those found in a real genome, where the number of positive miRNAs is very low with respect to the number of hairpins without miRNA in the rest of a complete genome. For this purpose, datasets were generated by random sampling from 1:500 (1 positive in 500 negatives) to a very high imbalance 1:10,000 (1 positive in 10,000 negatives).

3.2 Performance measures

For performance evaluation, the following standard measures have been used

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

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

Matthew correlation coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

Kappa coefficient

$$\kappa = \frac{a - a_c}{1 - a_c},$$

where TP, TN, FP and FN are true positives, true negatives, false positives and false negatives, respectively; N is the total number of observations; $a = (TP + TN)/N$ is the standard accuracy and a_c is the accuracy by chance, that is, the one provided by a classifier assigning randomly a positive or negative label to each sample.

The true positives rate is measured with s^+, while the true negatives rate is measured with s^-. The precision p is key to evaluate the performance of a classifier in the context of large imbalances due to the impact of false positives. Although only a small fraction of the negatives are misclassified, it becomes a large number in comparison to the number of positives. This detail is fundamental when a realistic scenario is considered, where biologists need only a small set of candidates. Thus, F_1 becomes the best measure to compare classification methods in large class imbalances, combining s^+ and p through the harmonic mean. Furthermore, we used two more combined measures, MCC and κ, which are also used for imbalanced datasets.

3.3 Experimental setup

To calculate the features, the secondary structure of all sequences (positives and negatives) was predicted with RNAfold (Lorenz *et al.*, 2011), with 37°C and the remaining parameters by default. After that, the 5p and 3p chains were extracted with 40 nt length from the terminal loop. In this way, the specific position of the mature miRNA within the chain is not required. Thus, it is possible to calculate the feature without any additional information for unknown hairpins. This is important because different isomiRs of the same chain can be generated depending on the position of the cut (Bartel, 2018).

The performance in each experiment is reported as the average value of 8 folds for the imbalances from 1:500 to 1:1,000, and 4 folds for the imbalances from 1:1,500 to 1:10,000, using the test partition only.

This difference in the number of folds selected for each case is due to the decrease in the number of positives when the imbalance increases. To assess whether there is a statistically significant difference in the performance of the proposed sets of features, the Friedman test was performed for the F_1 measure with a significance level of $\alpha = 0.01$. Finally, to evaluate which features have statistically different performances, the Nemenyi post-hoc test was used (Demšar, 2006).

The LD feature must be calculated taking into account that the reference set (the positive pre-miRNAs) changes with each training partition. Therefore, only the mature miRNAs found in each training set $\mathcal{A}$ of each corresponding fold are used, thus avoiding introducing *a-priori* information from the corresponding test set. For the training sequences, the distance of each training sample $\boldsymbol{a}_\ell \in \mathcal{A}$ is calculated as $L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \max\{L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p}), L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})\}$. In the case of the test samples $\boldsymbol{t}_\ell$, all the sequences in the train set can be used and the feature is calculated as $L_{\mathcal{A}}(\boldsymbol{t}_\ell) = \max\{L_{\mathcal{A}}(\boldsymbol{t}_\ell^{5p}), L_{\mathcal{A}}(\boldsymbol{t}_\ell^{3p})\}$.

For the PE calculation, we selected $N = 2$ because this value showed the best performance in preliminary tests. We codified each nucleotide A, C, G, U with an integer from 1 to 4 according to its relative frequencies in the sequences. To combine the information from both chains 3p and 5p, we calculated PE for each one and selected the smallest one. That is, the PE of order 2 of each test candidate $\boldsymbol{t}$ is calculated as $PE_2(\boldsymbol{t}) = \min\{PE_2(\boldsymbol{t}^{5p}), PE_2(\boldsymbol{t}^{3p})\}$. In the same way the LZ of each test candidate $\boldsymbol{t}$ was calculated as $LZ(\boldsymbol{t}) = \min\{LZ(\boldsymbol{t}^{5p}), LZ(\boldsymbol{t}^{3p})\}$.

These new features were tested with Naive Bayes (NB), Random Forest (RF), k-nearest neighbor (KNN) and Deep Neural Network (DNN) classifiers. These classifiers have been chosen because they have provided the best performances in a very recent review study on pre-miRNA prediction approaches (Stegmayer *et al.*, 2018).

NB classifiers are a family of probabilistic classifiers based on applying Bayes' theorem (Webb, 2002) with strong assumptions of independence between the features. It calculates the probability that a given example belongs to a certain class, under the assumption that the features are conditionally independent given the class. A NB classifier can be seen as a probability function that assigns, to an unknown input $\boldsymbol{z}$, a class label $y(\boldsymbol{z})$, which is proportional to the product of the prior $p(y_j)$ and the conditional probability $p(\boldsymbol{z}_j|y_j)$. Gaussian distributions were used to train this model in our experiments. RF is an ensemble of decision trees (Breiman, 2001). A decision tree classifier is composed by a number of nodes starting from a root node. At each node, the training set is split into two non overlapping sets: for a selected feature, a threshold is chosen such that the sample is assigned to some set (Breiman, 2001). The tree is grown until a maximum depth. For the prediction of a new case, it is pushed down the tree and assigned the label of a terminal node. To avoid overfitting, bootstrap-aggregated (bagged) is used by combining the results of many trees. The final decision for an unknown input vector is made by taking the majority vote of the trees in the ensemble. We used 100 trees for all cases.

KNN is a method that stores all the training examples as the classification model, without building a parametric model. All computation occurs at testing time (without training). It does not fit a model to the data. KNN just looks for the k nearest neighbors in all the training dataset at testing time, and classifies according to the majority class of the neighbors (Webb, 2002). Therefore, the only parameter that needs to be set is the number of neighbors k. Euclidean distance was used with $k = 1$ for imbalances ratio less than 1:1,500 and $k = 3$ for the other ones.

A DNN can be built from several feedforward layers of nonlinear neurons. 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) (Fischer and Igel, 2012). After the unsupervised stage to train each RBM layer, a supervised training is applied to the full network. Therefore, this model uses a hybrid learning approach. In this work, we used a network with 3 hidden layers and an output layer of 2 neurons. For imbalance of 1:500: 256, 128 and 16 neurons were used in each layer. For the second imbalance, 1:1,000: 256, 128, and 128 neurons were used in each layer. For the other cases: 256, 256, and 64 neurons were used for each layer. In all cases, the network was trained with cross entropy function and a batch size of 16. The optimization of these hyperparameter was done following (Stegmayer *et al.*, 2018).

4 Results and discussion

4.1 Classifiers and measures

Tables 1 to 4 present the results for each proposed new feature and the standard features (SF), for NB, RF, KNN and DNN classifiers, respectively. In each row, the performance of each classifier on a given imbalance, for all features, is reported according to MCC, κ and F_1. The best performance for each imbalance ratio and each measure is shown in bold.

Table 1 shows that, for NB with LD versus SF, the performance measures reflect consistently improvements for all imbalances. In particular, when LD are used, this classifier obtained the best rates in all imbalance cases. For the case where PE is used, improvements with respect to SF are found for all measures except for the imbalances of 1:2,000 and 1:4,000, where the performance remains the same. In the case of LZ, the same behavior is observed as in PE. In Table 2, when analyzing RF performance with the new features, all three performance measures show consistent results, that is, they improve the classifier performance in relation to SF alone. From 1:8,000 and on, all measures show that this classifier is highly affected by the imbalance. From the analysis of this table in a general way, it can be observed that the best results for each imbalance are distributed among the three features, but always exceeding SF in all cases and measures.

Table 3 shows KNN with LD versus SF. It can be seen here, again, that there is an improvement in performance when incorporating LD for imbalances less than 1:8,000. The only exception is for the imbalance of 1:4,000, where only F_1 shows an improvement in the classifier performance, while the other measures show the same result than SF alone. The other two features improve SF but only slightly and in some cases. At the highest imbalance point, KNN has an extremely poor performance, which is reflected by all measures. In Table 4, when analyzing the performance of DNN with LD versus SF, a significant improvement is observed in all the three measures and for all imbalances when the new LD feature is added to SF. For the case PE versus SF, it is observed that MCC and κ show improvements for the imbalances larger than 1:6,000. With F_1 the same improvement is found for all cases.

Finally, after a comprehensive analysis of all four tables in this section, it can be stated that, overall, improvements can be observed by all performance measures, consistently, and independently of the classifier used. It can be seen that RF and KNN show values equal to zero (or MCC of -1.0) for the largest imbalances. This is due to the bias generated by the *a-priori* probabilities of the classes, which causes the classifier to label the positive cases as part of the majority class (negative class). It is also observed that DNN achieved the highest performances for all imbalances and all features proposed, furthermore showing that these improvements are equally reflected by the three performance measures reported . For this reason, in the rest of this study, only this classifier will be used for the detailed analysis of the behavior of the proposed features. In addition, due to the fact that the three measures report a similar behavior, F_1 will be used from now on.

4.2 Detailed performance of novel features

Figure 1 shows a detailed analysis of the classification results for each of the new proposed features and SF, with DNN as classifier. The horizontal

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Raad, G. Stegmayer & D. H. Milone; "Complexity measures of the mature miRNA for improving pre-miRNAs prediction"
Bioinformatics, 2020.

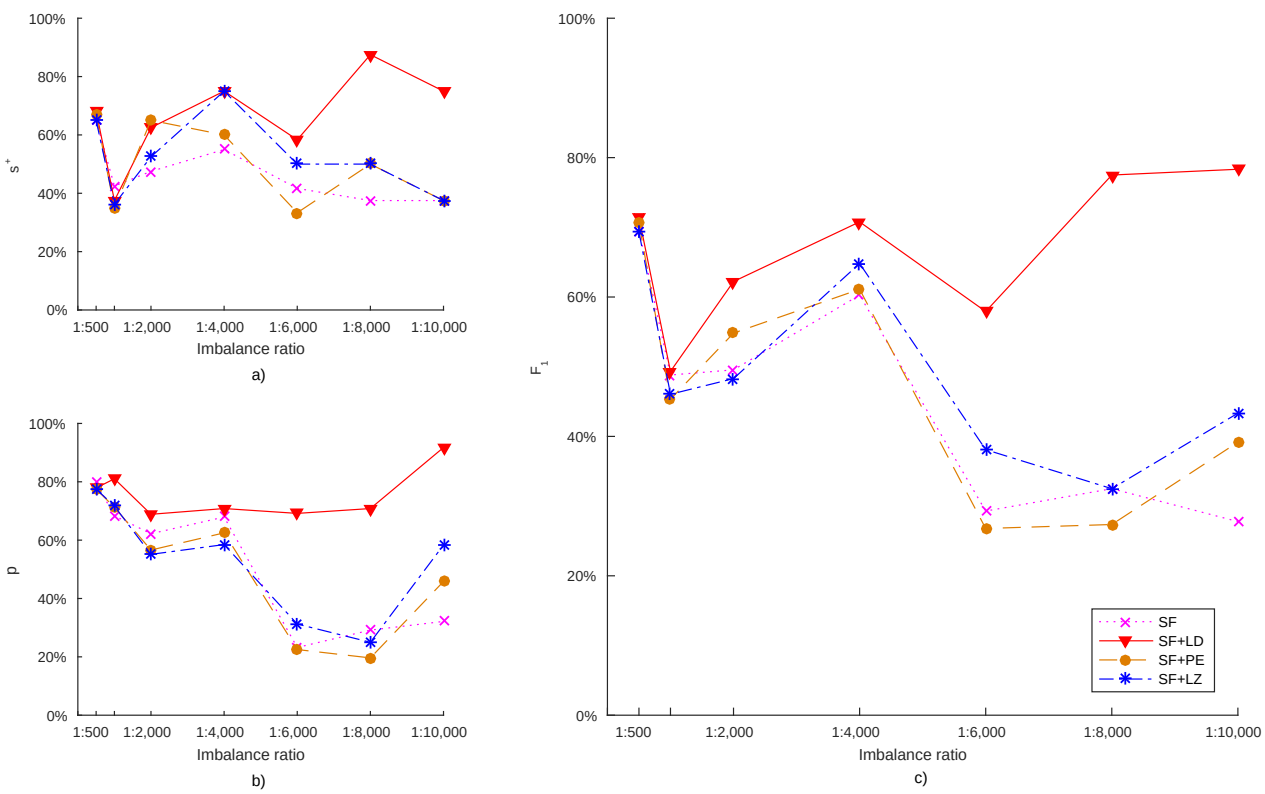

Fig. 1. Results of deep neural networks (DNN) with standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). a) Sensibility, s^+; b) Precision, p; c) F_1 score.

axis shows the imbalance ratio, while the vertical axis shows s^+, p and F_1, in Figures 1a, 1b and 1c, respectively. For more detailed information regarding the scores see Tables S1 to S4 (Supplementary Material). Since s^- has shown to be very close to 100% in all imbalances and for all features, it has not been included in the figure. This has happened because due to the high class imbalance, the negative class is the majority one and the easiest to detect, independently of the features employed.

Figure 1 clearly shows how the DNN classifier is capable of maintaining performance at increasing imbalances, and even increasing both s^+ (Figure 1a) and p (Figure 1b) when the new LD feature is used. This is a remarkable result, which has a direct impact in the impressive good performance of DNN with LD in F_1. In Figure 1c, when analyzing the performance of DNN with SF versus LD, it is observed that F_1 is significantly higher for all the imbalances when the new LD feature is used. For example, it can be seen that for the imbalances between 1:500 and 1:10,000, F_1 with SF goes down from almost 70% to around 20%. In this same imbalance range, however, DNN with LD goes up to almost 80%. It can also be noticed that the precision of the classifier increases very much with the incorporation of LD up to a very high level (higher than 90%) at the highest imbalance here studied. This is a very important result in practical terms, especially for imbalances closer to real cases where genome-wide data is used, because it assures to reduce remarkably the amount of false positives. Due to the fact that, in general terms, s^+ is also improved when LD is used, the F_1 increases in all cases as the imbalance increases. This is very interesting, since the ability to avoid false positives seems to be robust to the imbalance and the size of the positive set, without thereby influencing the detection of positives cases. When analyzing all the figures in a global way, an improvement of LD with respect to SF is observed for all the measures, which presents a clear trend to increase as the imbalance increases. The other features have more variable performance. In summary, it can be affirmed that a very important improvement in performance is obtained when using LD in the feature set, even at the highest imbalance.

An interesting point to discuss here is why LD shows such a robust behavior to imbalance. Generally, the algorithms for pre-miRNA prediction use public databases for training, which generates a bias towards previously known pre-miRNAs. Given that most of them have a stem-loop structure, and most of the features are based on that structure, with these standard features it is difficult to recognize possible new miRNAs that differ from the canonical ones. However, the inclusion of a sequence feature such as LD, calculated from the mature miRNA, is disruptive in this sense because it allows to take into account different information from the candidates, not related nor biased towards the structure alone. Thus, in a different space, generated by the novel features, the distances are different and the sequences that were not close according to standard features can be near now in the new space generated with the information of the mature miRNA. A second argument is that LD is not calculated only with the information of each candidate, but it is a distance of each sequence with respect to the whole reference set. A third point of view is that it can be said that this feature could be capable of obtaining a large robustness in front of candidates sequences that may have a more recent structure. This would be due to the incorporation of mature information that is complementary to the structure of each candidate. Thus, it could be possible to find new pre-miRNAs that differ from the canonical pre-miRNAs. One last interesting

Table 1. Naive Bayes classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.314	0.197	0.200	**0.324**	**0.207**	**0.210**	0.315	0.198	0.201	0.317	0.199	0.202
1:1,000	0.223	0.107	0.111	**0.234**	**0.115**	**0.119**	0.227	0.109	0.113	0.224	0.108	0.111
1:2,000	0.180	0.066	0.067	**0.184**	**0.069**	**0.071**	0.179	0.065	0.067	0.179	0.065	0.067
1:4,000	0.166	0.056	0.058	**0.180**	**0.066**	**0.067**	0.166	0.056	0.058	0.167	0.057	0.058
1:6,000	0.142	0.040	0.044	**0.164**	**0.052**	**0.057**	0.146	0.042	0.046	0.143	0.040	0.044
1:8,000	0.143	0.040	0.041	**0.178**	**0.061**	**0.063**	0.145	0.041	0.043	0.146	0.042	0.044
1:10,000	0.130	0.038	0.041	**0.153**	**0.052**	**0.061**	0.134	0.040	0.043	0.134	0.040	0.042

Table 2. Random Forest classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.650	0.630	0.633	0.664	0.646	0.646	0.664	0.646	0.646	**0.682**	**0.666**	**0.654**
1:1,000	0.602	0.532	0.510	**0.612**	**0.545**	**0.526**	0.498	0.456	0.453	0.591	0.518	0.492
1:2,000	0.418	0.298	0.279	**0.500**	**0.400**	0.372	0.447	0.333	0.311	**0.500**	**0.400**	**0.380**
1:4,000	0.447	0.333	0.266	0.387	0.261	0.208	**0.500**	**0.400**	**0.339**	0.387	0.261	0.194
1:6,000	-1.000	0.000	0.000	**0.289**	**0.154**	**0.125**	-1.000	0.000	0.000	-1.000	0.000	0.000
1:8,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000
1:10,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000

Table 3. K-nearest neigbor classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.531	0.530	0.527	**0.568**	**0.568**	**0.574**	0.531	0.530	0.531	0.531	0.530	0.531
1:1,000	0.421	0.421	0.411	**0.441**	**0.441**	**0.447**	0.421	0.421	0.414	0.409	0.409	0.419
1:2,000	0.399	0.373	0.383	**0.494**	**0.476**	**0.478**	0.372	0.345	0.356	0.448	0.426	0.414
1:4,000	**0.592**	**0.518**	0.451	**0.592**	**0.518**	**0.476**	0.404	0.400	0.442	0.592	**0.518**	0.451
1:6,000	0.408	0.286	0.250	**0.577**	**0.500**	**0.367**	0.408	0.286	0.225	0.408	0.286	0.225
1:8,000	0.354	0.222	0.167	0.354	0.222	0.167	0.354	0.222	0.167	0.354	0.222	0.167
1:10,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000

Table 4. Deep neural networks classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.704	0.702	0.695	**0.725**	**0.724**	**0.714**	0.697	0.697	0.707	0.704	0.702	0.693
1:1,000	0.499	0.492	0.488	**0.544**	**0.508**	**0.493**	0.472	0.451	0.453	0.483	0.464	0.461
1:2,000	0.508	0.506	0.496	**0.617**	**0.617**	**0.622**	0.506	0.490	0.548	0.495	0.494	0.483
1:4,000	0.564	0.564	0.603	**0.699**	**0.698**	**0.708**	0.600	0.600	0.611	0.699	**0.698**	0.648
1:6,000	0.400	0.400	0.293	**0.764**	**0.737**	**0.579**	0.333	0.333	0.268	0.463	0.461	0.381
1:8,000	0.320	0.316	0.325	**0.935**	**0.933**	**0.775**	0.408	0.400	0.274	0.408	0.400	0.325
1:10,000	0.320	0.316	0.278	**0.866**	**0.857**	**0.783**	0.612	0.545	0.392	0.612	0.545	0.433

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Raad, G. Stegmayer & D. H. Milone; "Complexity measures of the mature miRNA for improving pre-miRNAs prediction"
Bioinformatics, 2020.

point to discuss is whether LD results can be biased towards larger miRNAs classes or families. Since in Eq. (1) LD is calculated as a statistic of the distances to each mature miRNAs of the training set, the choice of this statistic was not trivial. Firstly, the minimum has been chosen in order to avoid a possible bias towards the most numerous families. However, the results obtained showed a wide overlap of both classes, because the minimum considers only the most similar sequence. In contrast, the median is a more informative statistic because it uses the complete training set of known miRNAs. Thus, class distributions were shown to be more separated (see Figure S1 in the Supplementary Material).

For DNN with PE it is observed that F_1 is being improved in approximately a 10%, only at the largest imbalance here analyzed, where F_1 is almost 30% with SF, and almost 40% when PE is also used. The most important and remarkable improvement is observed in p at 1:10,000, where from around 30% it goes up to more than 45%. This suggests that this feature can effectively reduce the false positives, achieving an improvement of precision in very large imbalanced problems. In summary, it can be stated that PE can only improve the performance of DNNs just for highly imbalanced cases.

In the case of LZ, when analyzing the performance of DNN with SF, versus DNN with the incorporation of LZ, it is observed that F_1 is superior for the largest imbalance. It can also be seen that the improvement of F1 is due to by a slightly improvement of p and s^+. That is, LZ can probably serve to avoid false positives, especially when a negative class is extremely large with respect to the positive class. It can be stated, in summary, that LZ can have the capacity to improve the performance of a DNN for high imbalances, mainly thanks to the improvement of p.

4.3 Global performance of novel features

Table 5 shows the results with different combinations of the proposed features for DNN. In each row F_1 can be observed for the different sets of features, for each imbalance. It can be seen that LD improves the performance of the classifier in all cases, even for very high imbalances (1:10,000). Instead, LZ and PE individually do not improve the DNN performance. F_1 in those cases remains the same or quite similar to the SF case. Observing the different combinations of features for DNN, it can be noticed that F_1 improves for all cases in LD+PE with respect to SF. In addition, for the case of 1:2,000, 1:4,000 and 1:6,000, LD+PE combined achieve a larger performance than when used separately. For LD+LZ, F_1 improves in all cases with respect to SF (except for 1:1,000, where it remains almost the same). Furthermore, for the cases of 1:4,000 and 1:8,000, LD+LZ overcome the performance of the features used separately. In the case of PE+LZ, it is observed that F_1 mostly remains the same, or improves only slightly in some cases. Finally, analyzing the behavior of the combination of all the features together, it can be stated that F_1 improved in all cases.

Table 5 shows, in a more global way, two key and complementary results. In the first place, that LD is the feature that has the best individual performance. Secondly, although the features PE and LZ individually improve the results for DNN classifier, their contributions have more impact when combined. For this reason, it can be said that the novel features presented in this work provide complementary information.

In order to evaluate the statistical significance of the results, the Friedman test for F_1 was performed, resulting in a p-value of 2.5748E-05 ($\alpha = 0.01$), which indicates that there is a statistically significant difference between the scores. Then, the Nemenyi post-hoc test for F_1 was performed. This statistical analysis clearly indicates that the results obtained for LD and the combination LD+PE+LZ are the best features, in comparison to SF, LZ and PE alone. The post-hoc test showed that there are no statistically significant difference between LD and LD+PE+LZ, as it also showed that there are no statistically significant difference between LZ, PE and SF. Thus, the difference between these two groups of features is statistically significant. Furthermore, due to the fact that there were very few positive samples in the test partitions of the highest imbalances, we have repeated the experiment 10 times with different samplings of positives in the case of LD versus SF with DNN for imbalance 1:10,000. A median F_1 of 66.67% and 30.95% were obtained, for LD and SF respectively. A Wilcoxon signed-rank test test was applied to these 40 test partitions and a p < 6.2028E-05 was obtained.

Table 5. F_1 results for different combinations of Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ) with deep neural networks. Best results in bold for each table panel, individual (left) and combined (right) features.

IR	SF	LD	PE	LZ	LD+PE	LD+LZ	PE+LZ	ALL
1:500	69.50	**71.44**	70.65	69.34	71.39	**71.68**	68.96	71.50
1:1,000	48.81	**49.33**	45.33	46.05	49.26	48.71	52.85	**53.85**
1:2,000	49.55	**62.22**	54.82	48.29	63.21	57.72	53.33	**65.34**
1:4,000	60.28	**70.78**	61.11	64.81	**78.28**	73.33	64.95	71.89
1:6,000	29.29	**57.92**	26.79	38.10	**61.67**	57.92	29.17	56.79
1:8,000	32.50	**77.50**	27.36	32.50	77.50	**85.00**	36.67	77.50
1:10,000	27.78	**78.33**	39.17	43.33	62.50	**70.00**	40.48	54.17

An interesting point to further discuss is why PE and LZ individually have not shown a robust behavior for increasing imbalances. However, when combined with LD, it has been found that those actually help improving the robustness to imbalance. This behavior suggests that these features can capture useful information from the mature, but due to its short length it is not possible to obtain values discriminative enough, by themselves, separately. However, they are more discriminative when combined with LD, because this feature does not depend on the length of the sequence itself, but on the distance to the whole reference set, as explained before. For this reason, when all the features are combined, a predominance of LD over PE and LZ is observed, although the inclusion of the latter continues to provide some discriminative information. For example, for imbalance 1:2,000, the baseline F_1 provided by SF is 49.55%, LD improves it up to 62.22% but PE and LZ are just slightly better than SF. Thus, the 65.34% of ALL is clearly dominated by LD. On the other hand, the best results of the Levenshtein distance feature can be explained based to the fact that this feature is calculated according to an external/outside set of pre-miRNAs. Instead, permutation entropy and Lempel-Ziv complexity are individual features, calculated with information within each sequence by itself. LD allows having a more accurate measure and representative sense of belonging to the positive class, since LD is a distance to a reference set of miRNAs. From another point of view, this suggests that the mature contains certain syntactic structures that guide its functioning, thus avoiding any random mutation to modify it. Therefore, by combining the information of the median distance of a candidate (LD), together with the information of its randomness (PE) and its complexity (LZ), we are restricting the number of candidate sequences just to the possible combinations of nucleotides that can allow small changes, with a defined complexity.

5 Conclusions

In the prediction of novel pre-miRNAs a large number of structural features have been proposed in order to improve the efficiency in the separation of the positive and negative classes. However, the detained performance is highly dependent on the imbalance, generating a large number of false positives. In this work, a set of new features based on the sequence information of the mature miRNA was proposed, which improve the performance independently of the classifier, decreasing the number of false positives for high imbalances. The results showed that the incorporation of the proposed measures in the mature miRNA provides a

high discriminative power. Especially, the proposed Levenshtein distance has shown to have the best performance for all the imbalances. In addition, the proposed features based in permutation entropy and Lempel-Ziv complexity showed the best performances in high imbalances when combined with Levenshtein distance. The best results of the Levenshtein distance can be explained because it is a measure to a reference set of miRNAs, which allows measuring more accurately the belonging of any sequence to the positive class. This feature has provided very high precision to the classifiers evaluated, which is one of the most important contributions of our work, because most available algorithms have a very large rate of false positives. Moreover, it has shown robustness to the imbalance, improving predictions even in large imbalance scenarios. In a future work it would be interesting to introduce the probability of mutation of each nucleotide as different penalties in the Levenshtein distance. Another important conclusion of this study is that, although for all classifiers the inclusion of the new features improved their performance, the deep neural networks was the best one to relate the structural and sequence information of each pre-miRNA.

Funding

This work was supported by Universidad Nacional del Litoral (CAI+D 2016 082) and Agencia Nacional de Promocion Científica y Tecnológica (PICT 2014 2627).

References

Baker, M. (2010). MicroRNA profiling: separating signal from noise. *Nature Methods*, **7**(9), 687–692.

Bandt, C. and Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, **88**(17), 174102.

Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *cell*, **116**(2), 281–297.

Bartel, D. P. (2009). MicroRNAs: target recognition and regulatory functions. *cell*, **136**(2), 215–233.

Bartel, D. P. (2018). Metazoan microRNAs. *Cell*, **173**(1), 20–51.

Batuwita, R. and Palade, V. (2009). micropred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, **25**(8), 989–995.

Billoud, B. *et al.* (2013). Computational prediction and experimental validation of microRNAs in the brown alga Ectocarpus siliculosus. *Nucleic Acids Research*, **42**(1), 417–429.

Bortolomeazzi, M. *et al.* (2017). A survey of software tools for microRNA discovery and characterization using RNA-seq. *Briefings in Bioinformatics*, **20**(3), 918–930.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Brennecke, J. *et al.* (2005). Principles of microRNA–target recognition. *PLoS biology*, **3**(3), e85.

Brudno, M. *et al.* (2003). Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, **19**(suppl_1), i54–i62.

Bugnon, L. *et al.* (2019). Deep Neural Architectures for Highly Imbalanced Data in Bioinformatics. *IEEE Transactions on Neural Networks and Learning Systems*, **6**, 1–11.

Chen, L. *et al.* (2018). Trends in the development of miRNA bioinformatics tools. *Briefings in Bioinformatics*.

de ON Lopes, I. *et al.* (2014). The discriminant power of RNA features for pre-miRNA recognition. *BMC bioinformatics*, **15**(1), 124.

Demirci, M. D. S. *et al.* (2017). On the performance of pre-microRNA detection algorithms. *Nature communications*, **8**(1), 330.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, **7**(Jan), 1–30.

Ding, J. *et al.* (2010). MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC bioinformatics*, **11**(11), S11.

Dong, H., , *et al.* (2013). MicroRNA: function, detection, and bioanalysis. *Chemical reviews*, **113**(8), 6207–6233.

Fischer, A. and Igel, C. (2012). An introduction to restricted boltzmann machines. In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 14–36, Berlin, Heidelberg. Springer Berlin Heidelberg.

Friedländer, M. R. *et al.* (2011). miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Research*, **40**(1), 37–52.

Friedman, R. C. *et al.* (2009). Most mammalian mRNAs are conserved targets of micrornas. *Genome research*, **19**(1), 92–105.

Gudyś, A. *et al.* (2013). HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC bioinformatics*, **14**(1), 83.

Hertel, J. and Stadler, P. F. (2006). Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, **22**(14), e197–e202.

Jiang, P. *et al.* (2007). MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic acids research*, **35**(suppl_2), W339–W344.

Kozomara, A. *et al.* (2019). miRBase: from microRNA sequences to function. *Nucleic acids research*, **47**(D1), D155–D162.

Kozomara, A. and Griffiths-Jones, S. (2010). miRBase: integrating microRNA annotation and deep-sequencing data. *Nucleic acids research*, **39**(suppl_1), D152–D157.

Lassmann, T. and Sonnhammer, E. L. (2005). Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC bioinformatics*, **6**(1), 298.

Lempel, A. and Ziv, J. (1976). On the complexity of finite sequences. *IEEE Transactions on information theory*, **22**(1), 75–81.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Lewis, B. P. *et al.* (2003). Prediction of mammalian microRNA targets. *Cell*, **115**(7), 787–798.

Lewis, B. P. *et al.* (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *cell*, **120**(1), 15–20.

Li, L. *et al.* (2010). Computational approaches for microRNA studies: a review. *Mammalian Genome*, **21**(1-2), 1–12.

Liu, B. (2017). BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in Bioinformatics*, **20**(4), 1280–1294.

Lorenz, R. *et al.* (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, **6**(1), 26.

Mathelier, A. and Carbone, A. (2010). MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, **26**(18), 2226–2234.

Morgado, L. and Johannes, F. (2017). Computational tools for plant small RNA detection and categorization. *Briefings in Bioinformatics*, **20**(4), 1181–1192.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, **48**(3), 443–453.

Ng, K. L. S. and Mishra, S. K. (2007). De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, **23**(11), 1321–1330.

Polyanovsky, V. O. *et al.* (2011). Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for molecular biology*, **6**(1), 25.

Pritchard, C. C. *et al.* (2012). MicroRNA profiling: approaches and considerations. *Nature Reviews Genetics*, **13**(5), 358.

Shannon, C. (2001). A mathematical theory of communication. sigmobile mob comput commun rev 5 (1): 3–55.

Shukla, V. *et al.* (2017). A compilation of web-based research tools for miRNA analysis. *Briefings in functional genomics*, **16**(5), 249–273.

Stegmayer, G. *et al.* (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*, page bby037.

Takahashi, R.-u. *et al.* (2015). Loss of microRNA-27b contributes to breast cancer stem cell generation by activating enpp1. *Nature communications*, **6**, 7318.

Tseng, K.-C. *et al.* (2017). microRPM: a microRNA prediction model based only on plant small RNA sequencing data. *Bioinformatics*, **34**(7), 1108–1115.

Vitsios, D. M. *et al.* (2017). Mirnovo: genome-free prediction of microRNAs from small RNA sequencing data and single-cells using decision forests. *Nucleic Acids Research*, **45**(21), e177–e177.

Webb, A. (2002). *Statistical pattern recognition*. Wiley Press.

Wheeler, B. M. *et al.* (2009). The deep evolution of metazoan microRNAs. *Evolution & development*, **11**(1), 50–68.

Xue, C. *et al.* (2005). Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, **6**(1), 310.

Yones, C. *et al.* (2015). miRNAfe: a comprehensive tool for feature extraction in microRNA prediction. *Biosystems*, **138**, 1–5.

Yones, C. *et al.* (2018). Genome-wide pre-miRNA discovery from few labeled examples. *Bioinformatics*, **34**(4), 541–549.

Yousef, M. *et al.* (2006). Combining multi-species genomic data for microRNA identification using a naive bayes classifier. *Bioinformatics*, **22**(11), 1325–1334.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Raad, G. Stegmayer & D. H. Milone; "Complexity measures of the mature miRNA for improving pre-miRNAs prediction"
Bioinformatics, 2020.

776 Ziv, J. and Lempel, A. (1978). Compression of individual sequences via variable-rate
777 coding. *IEEE transactions on Information Theory*, **24**(5), 530–536.
778 Zou, Q. *et al.* (2014). miRClassify: an advanced web server for miRNA family
779 classification and annotation. *Computers in biology and medicine*, **45**, 157–160.
780 Zytnicki, M. *et al.* (2008). Darn! a weighted constraint solver for RNA motif
781 localization. *Constraints*, **13**(1-2), 91–109.