Feature Extraction Approaches for Biological Sequences: A Comparative Study of Mathematical Models

Robson Parmezan Bonidia[a,b,*], Lucas Dias Hiera Sampaio[a], Fabrício Martins Lopes[a], André Carlos Ponce de Leon Ferreira de Carvalho[b], Danilo Sipoli Sanches[a]

[a]*Department of Computer Science, Bioinformatics Graduate Program (PPGBIOINFO), Federal University of Technology - Paraná, UTFPR, Campus Cornélio Procópio, Brazil.*
[b]*Institute of Mathematics and Computer Sciences, University of São Paulo - USP, São Carlos, 13566-590, Brazil*

Abstract

The number of available biological sequences has increased significantly in recent years due to various genomic sequencing projects, creating a huge volume of data. Consequently, new computational methods are needed to analyze and extract information from these sequences. Machine learning methods have shown broad applicability in computational biology and bioinformatics. The utilization of machine learning methods has helped to extract relevant information from various biological datasets. However, there are still several problems that motivate new algorithms and pipeline proposals, mainly involving feature extraction problems, in which extracting significant discriminatory information from a biological set is challenging. Considering this, our work proposes to study and analyze a feature extraction pipeline based on mathematical models (Numerical Mapping, Fourier, Entropy, and Complex Networks). As a case study, we analyze Long Non-Coding RNA sequences. Moreover, we divided this work into two studies, e.g., (I) we assessed our proposal with the most addressed problem in our review, e.g., lncRNA vs. mRNA; (II) we tested its generalization on different classification problems, e.g., circRNA vs. lncRNA. The experimental results demonstrated three main contributions: (1) An in-depth study of several mathematical models; (2) a new feature extraction pipeline and (3) its generalization and robustness

*Corresponding author
Email address: robservidor@gmail.com (Robson Parmezan Bonidia)

for distinct biological sequence classification.

Keywords: Feature Extraction; Long Non-Coding RNAs; Biological Sequences; Numerical Mapping Techniques; Fourier; Complex Networks; Shannon; Tsallis.

1. Background

In recent years, due to advances in DNA sequencing, an increasing number of biological sequences have been generated by thousands of sequencing projects [1], creating a huge volume of data [2]. During the last decade, Machine Learning (ML) methods have shown broad applicability in computational biology and bioinformatics [3]. Consequently, the ability to process and analyze biological data has advanced significantly [4]. Tools have been applied in gene networks, protein structure prediction, genomics, proteomics, protein-coding genes detection, disease diagnosis, and drug planning [5, 6]. Fundamentally, ML investigates how computers can learn (or improve their performance) based on the data. Moreover, ML is a specialization of computer science related to pattern recognition and artificial intelligence [7].

Based on this, several works have focused on investigating sequences of DNA and RNA molecules. Applying ML methods in these sequences has helped to extract important information from various datasets to explain biological phenomena [3]. The development of efficient approaches benefits the mathematical understanding of the structure of biological sequences [1], e.g., Precision cancer diagnostics [8] and the Coronavirus epidemic [9, 10]. However, according to [3, 11], there are still several challenging biological problems that motivated the emergence of proposals for new algorithms. Fundamentally, biological sequence analysis with ML presents one major problem: Feature Extraction [12].

Feature extraction seeks to generate a feature vector, optimally transforming the input data [12]. This procedure is exceptionally relevant for the success of the ML application. Another primary goal of feature extraction is to extract important information from input data compactly, as well as removing noise and redundancy to increase the accuracy of ML models [13, 12]. Furthermore, the feature extraction is an inevitable method, especially in the stage of biological sequence preprocessing [14].

Necessarily, several methods in bioinformatics apply ML algorithms for sequence classification, and as many algorithms can deal only with numerical

data, sequences need to be translated into sequences of numbers. Thereby, modern applications extract relevant features from sequences based on several biological properties, e.g., physicochemical, Open Reading Frames (ORF)-based, usage frequency of adjoining nucleotide triplets, GC content, among others. This approach is common in biological problems, but these implementations are often difficult to reuse or adapt to another specific problem, e.g., ORF features are an essential guideline for distinguishing Long non-coding RNAs (lncRNA) from protein-coding genes [15], but not useful features for classifying lncRNA classes [2]. Consequently, the feature extraction problem arises, in which extracting a set of useful features that contain significant discriminatory information becomes a fundamental step in the construction of a predictive model [16].

Therefore, these problems make the process of biological sequence classification a challenging task, creating a growing need to develop new techniques and methods to analyze sequences effectively and efficiently. Thereby, this work studies the performance of different feature extraction methods for biological sequence analysis, using mathematical models, e.g., numerical mapping, Fourier transform, entropy, and graphs. As a case study, we will use lncRNA sequences, which are fundamentally unable to produce proteins [17] and have recently casted doubt on its functionality [18].

LncRNAs present several problem classes (e.g., lncRNA vs. mRNA [19, 20] and lncRNA vs. circRNA [21]), thus enabling us to create a scenario to answer the questions raised in this work. Fundamentally, our main objective is to propose generalist techniques, demonstrating their efficiency concerning biological features. We consider biological approaches, those characteristics that present a bias to the analyzed problem or some biological explanation, e.g., ORF for lncRNA vs. mRNA [6, 15], as well as mathematical approaches and information quantity measures such as entropy. Based on this context and objectives, we assume the following hypothesis:

- **Hypothesis:** Feature extraction approaches based on mathematical models are as efficient and generalist as biological approaches.

Considering this, our work contributes to the area of computer science and bioinformatics. Specifically, it introduces new ideas and analysis for the feature extraction problem in biological sequences. Thereby, we present four new contributions: (1) A feature extraction pipeline using mathematical models; (2) Analysis of 9 different mathematical models; (3) Analysis of 6

numerical mappings with Fourier, proposing statistical characteristics; (4) The generalization and robustness of mathematical approaches for the feature extraction in biological sequences.

2. Related Works

Essentially, as emphasized, we adopt lncRNA sequences as a case study, a class of Non-Coding RNAs (ncRNAs). Fundamentally, ncRNAs are unable to produce proteins. However, these ncRNAs contain unique information that produces other functional RNA molecules [22, 17]. Moreover, they demonstrate essential roles in cellular mechanisms, playing regulatory roles in a wide variety of biological reactions and processes [22]. The ncRNAs can be classified by length into two classes: Long Non-Coding RNA (lncRNA - 200 nucleotides (nt) or more) and short ncRNA (less than 200 nt) [23, 24]. The lncRNAs are sequences with a length greater than 200 nucleotides [25], and according to recent studies, play essential roles in several critical biological processes [26, 27, 28], including transcriptional regulation [29], epigenetics [30], cellular differentiation [31], and immune response [32]. Moreover, they are correlated with some complex human diseases, such as cancer and neurodegenerative diseases [6, 33, 34].

In plants, according to [6, 35], the lncRNAs act in gene silencing, flowering time control, organogenesis in roots, photomorphogenesis in seedlings, stress responses [36, 37], and reproduction [38]. Furthermore, lncRNAs are present in large numbers in genome [39] and have similar sequence characteristics with protein-coding genes, such as 5' cap, alternative splicing, two or more exons [40], and polyA+ tails [41]. They are also observed in almost all living beings, not only in animals and plants but also yeasts, prokaryotes, and even viruses [42, 43].

According to [39], lncRNAs do not contain functional ORFs. However, recent studies have found bifunctional RNAs [44], raising the possibility that many protein-coding genes may also have non-coding functions. Furthermore, lncRNAs can be grouped into five broad categories. The classification occurs conforming to the genomic location, that is, where they are transcribed, concerning well-established markers, e.g., protein-coding genes. Among the categories are [45, 40]: sense, antisense, bidirectional, intronic, intergenic. The genomic context does not necessarily provide some information about the lncRNAs function or evolutionary origin; nevertheless, it can be used to organize these broad categories [46].

In this context, we have conducted an in-depth review of the lncRNAs classification methods, in which several approaches have been developed, such as: CPC [47], CPAT [48], CNCI [49], PLEK [50], lncRNA-MFDL [51], LncRNA-ID [52], lncRScan-SVM [53], LncRNApred [54], DeepLNC [55], PlantRNA_Sniffer [56], PLncPRO [57], RNAplonc [58], BASiNET [59], and LncFinder [20]. For better understanding, Figure 1 presents theses works divided into Mathematical, Biological, and Hybrid approaches.

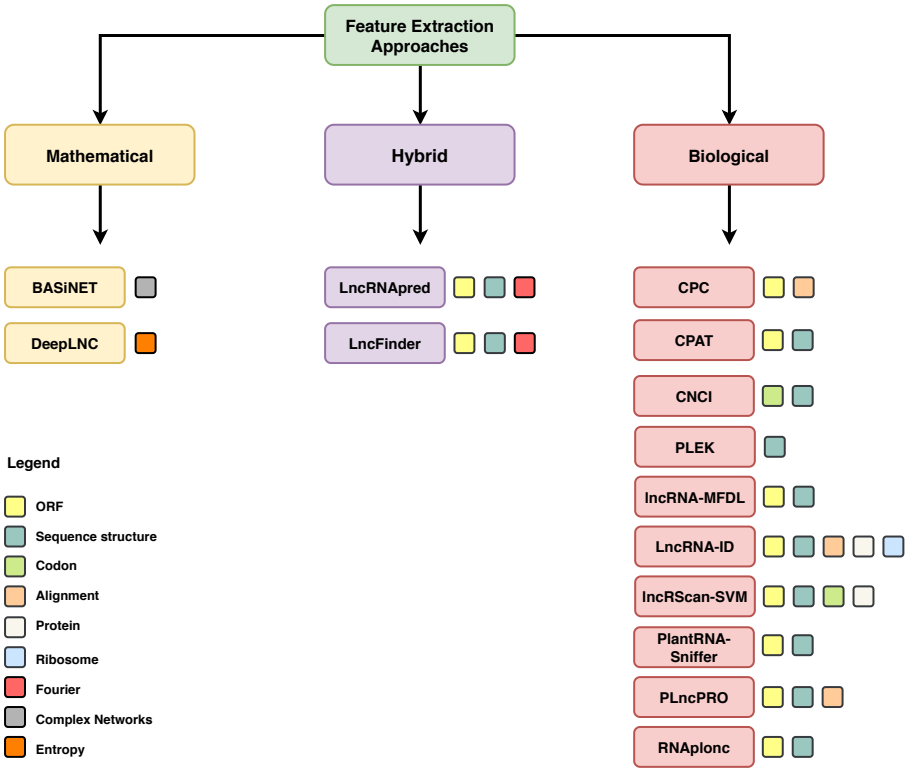

Figure 1: Feature extraction approaches in our case study divided into: Mathematical, Biological, and Hybrid.

The CPC uses the extent and quality of the ORF, and derivation of the BLASTX [60] search to measure the protein-coding potential of a transcript. In the classification, the authors applied the LIBSVM package to train a Support Vector Machine (SVM) model, using the standard radial basis function kernel. CPAT classifies transcripts of coding and non-coding using the Logistic Regression (LR) classifier. This approach implements four features: ORF

coverage, ORF size, hexamer usage bias, and Fickett TESTCODE statistic. CNCI was induced with SVM and applies profiling Adjoining Nucleotide Triplets, and most-like CDS (MLCDS).

In contrast, PLEK (2014) is based on the k-mer scheme ($k = 1, \ldots, 5$) to predict lncRNA, also applying the SVM classifier. lncRNA-MFDL uses Deep Learning (DL) and multiple features, among them: ORF, K-mer (k = 1, 2, 3), secondary structure (minimum free energy), and MLCDS. LncRNA-ID predicts lncRNAs with Random Forest (RF) through ORF (length and coverage), sequence structure (Kozak motif), ribosome interaction, alignment (profile Hidden Markov Mode - profile HMM), and protein conservation.

lncRScan-SVM uses stop codon count, GC content, ORF (score, CDS length and CDS percentage), transcript length, exon count, exon length, and average PhastCons scores. LncRNApred classified lncRNAs with RF and features based on ORF, signal to noise ratio, k-mer (k = 1, 2, 3), sequence length, and GC content. DeepLNC uses only the k-mer scheme with entropy and Deep Neural Network (DNN). PlantRNA_Sniffer was developed in 2017 to predict Long Intergenic Non-Coding RNAs (lincRNAs). The method applied SVM and extracted features from ORF (proportion and length) and nucleotide patterns.

PLncPRO is based on machine learning and uses RF. The features selected include ORF quality (score and coverage), number of hits, significance score, total bit score, and frame entropy. RNAplonc classified sequences with the REPtree algorithm, considering 16 features (ORF, GC content, K-mer scheme ($k = 1, \ldots, 6$), sequence length). BASiNET classifies sequences based on the feature extraction from complex network measurements. Lastly, LncFinder tests five classifiers (LR, SVM, RF, Extreme Learning Machine, and Deep Learning), to apply the algorithm that obtains the highest accuracy. The authors extract features from ORF, secondary structural, and EIIP-based physicochemical properties.

In general, the aforementioned works apply supervised learning methods using binary classification (two classes - lncRNAs and protein-coding genes (mRNA)). There is a considerable amount of research on humans, followed by animals and plants. Regarding feature extraction, we observed a full domain of ORF and sequence-structure descriptors. As seen in Figure 1, there is a frequent use of biological features. On the other hand, some works have explored mathematical approaches for feature extraction, such as Genomic Signal Processing (GSP), DNA Numerical Representation (DNR) [54, 20], and Complex Networks [59]. Nevertheless, the authors used these charac-

teristics in conjunction with other biological feature extraction techniques or without testing other mathematical features. Practically no papers have focused on several mathematical approaches. Based on this, the objective of this section was to summarize the main methods of the literature and their characteristic descriptors. Therefore, we will not use the works shown for comparison, but the most applied features.

3. Materials and Methods

In this section. we describe the methodological approach used to achieve the proposed objectives, as shown in Figure 2. Essentially, we divided our study into five stages: (1) Data selection and preprocessing; (2) Feature extraction; (3) Training; (4) Testing; (5) Performance analysis. Hence, each stage of the study is described, as well as information about the adopted process.

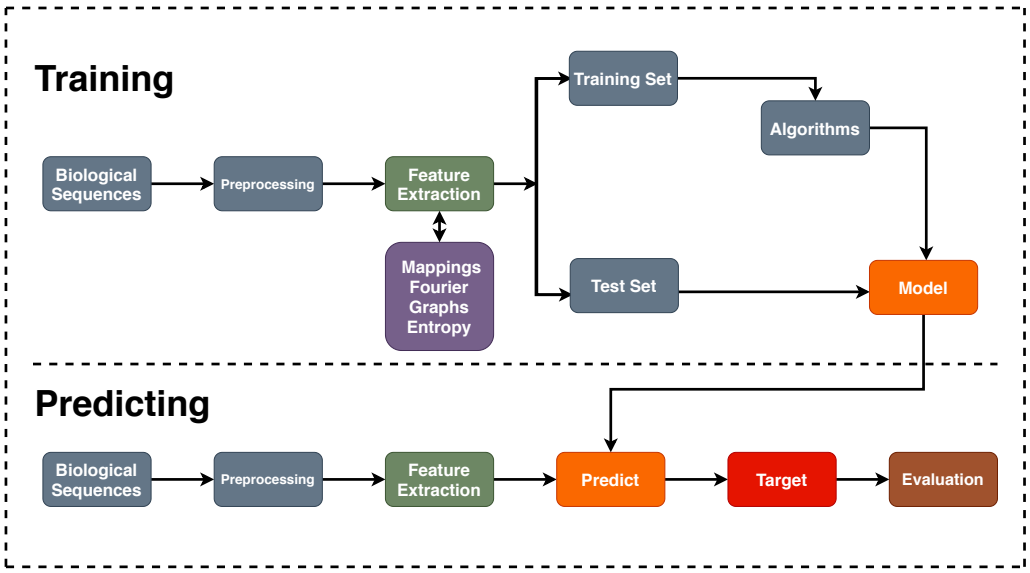

Figure 2: Proposed Pipeline. Essentially, (1) datasets are preprocessed; (2) Feature extraction techniques are applied to each dataset; (3) Machine learning algorithms are executed in the training set to induce predictive models; (4) Induced models are applied to the test set; Finally, (5) the models are evaluated.

This work was also divided into two case studies: (I) We assessed our mathematical approaches with the most addressed problem in our review,

e.g., lncRNA vs. mRNA; (II) We tested its generalization on different classification problems.

3.1. Data Selection

As previously mentioned, we chose the lncRNAs classification problem, because it is a new and relevant theme in the literature, in which, recently, it has presented several works, mainly with ML, as explored in Section 2. However, we will also adopt other datasets to assess the generalization of mathematical features. As preprocessing, we used only sequences longer than $200nt$ [50], and we also removed sequence redundancy. Moreover, the sampling method was adopted in our dataset, since we are faced with the *imbalanced data problem* [2]. Therefore, we applied random majority undersampling, which consists of removing samples from the majority class (to adjust the class distribution) [61]. Finally, we divided this paper into two case studies.

3.1.1. Case Study I

Sequences of five plant species were adopted to validate the proposed approaches. The summary of the dataset can be seen in Table 1. According to the literature approaches, this study also adopts two classes for the datasets: the positive class, with lncRNAs, and the negative class, with protein-coding genes (mRNAs).

Table 1: Adopted species to create the datasets.

Species	**Sequences**	**Samples**	**Preprocessing**	**Selected**
A. trichopoda	lncRNA	5698	4556	4556
	mRNA	26846	22326	4556
A. thaliana	lncRNA	2540	2540	2540
	mRNA	13973	13973	2540
C. sinensis	lncRNA	2562	2215	2215
	mRNA	46147	45846	2215
C. sativus	lncRNA	1929	1730	1730
	mRNA	30364	29829	1730
R. communis	lncRNA	4198	3487	3487
	mRNA	31221	29042	3487

The mRNA data of the *Arabidopsis thaliana* (obtained from CPC2 [19]) were built from the `RefSeq database` with protein sequences annotated by

Swiss-Prot [19], and lncRNA data from the `Ensembl` ($v87$) and `Ensembl Plants` ($v32$) database. The mRNA transcript data of the *Amborella trichopoda*, *Citrus sinensis*, *Cucumis sativus* and *Ricinus communis* were extracted from `Phytozome` (version 13) [62]. The lncRNAs data from these species were extracted from `GreeNC` (version 1.12) [63].

3.1.2. Case Study II

In this case study, we will apply the best mathematical models (considering accuracy) of case study I to different classification problems with lncRNAs, in order to test their generalization. Thus, divided this part into three problems:

- **Problem 1** (lncRNA vs. sncRNA): Dataset with only non-coding sequences (lncRNA and Small non-coding RNAs (sncRNAs), also obtained from [19])
  - lncRNA: 1291 sequences — sncRNA: 1291 sequences
- **Problem 2** (lncRNA vs. Antisense): Dataset with lncRNAs and long noncoding antisense transcripts (obtained from [64]).
  - lncRNA: 57 sequences — Antisense: 57 sequences
- **Problem 3** (circRNA vs. lncRNA): Dataset with lncRNA and circular RNAs (cirRNAs) sequences (circRNA obtained from PlantcircBase [65]. This problem was based on [66] and [21], in order to classify circRNA from other lncRNAs.
  - circRNA: 2540 sequences — lncRNA: 2540 sequences

It is important to emphasize that we used only sequences from *Arabidopsis thaliana* in this second case study because it is the model species in plants. Moreover, plant sequences is the least addressed field by the studies, consequently presenting more challenges.

3.2. Feature Extraction

In this section, 9 feature extraction approaches are shown: 6 numerical mapping techniques with Fourier transform, Entropy, Complex Networks. It is necessary to emphasize that we denote a biological sequence $\mathbf{s} = (s[0], s[1], \ldots, s[N-1])$ such that $\mathbf{s} \in \{A, C, G, T\}^N$ [2].

3.3. Fourier Transform and Numerical Mappings

To extract features based on a Fourier model, we applied the Discrete Fourier Transform (DFT), widely used for digital image and signal processing (here GSP), which can reveal hidden periodicities after transformation of time domain data to frequency domain space [67]. According to Yin and Yau [68], the DFT of a signal with length N, $\mathbf{x} \in \mathbb{R}^N$, at frequency k, can be defined by Equation (1):

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, 1, \ldots, N-1. \tag{1}$$

This method is has been widely studied in bioinformatics, mainly for analysis of periodicities and repetitive elements in DNA sequences [69] and protein structures [70]. This approach is shown in Figure 3 and was based on [2].

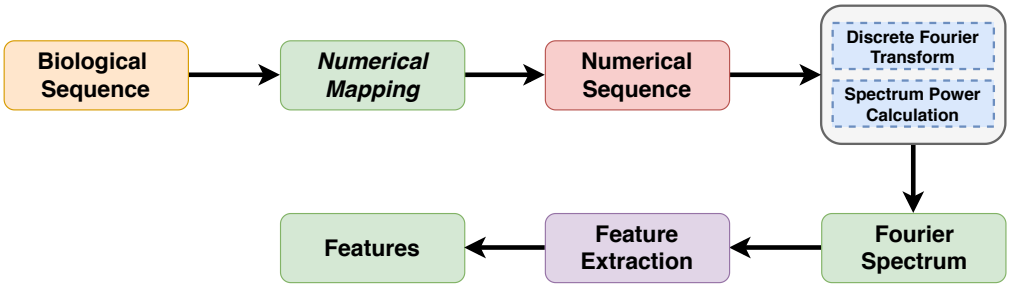

Figure 3: Fourier Transform and Numerical Mapping Pipeline. (1) Each sequence is mapped to a numerical sequence; (2) DFT is applied to the generated sequence; (3) The spectrum power is calculated; (4) The Feature Extraction is performed; Finally, (5) the features are generated.

To calculate DFT, we will use the Fast Fourier Transform (FFT), that is a highly efficient procedure for computing the DFT of a time series [71]. However, to use GSP techniques, a numeric representation should be used for the transformation or mapping of genomic data. In the literature, distinct DNR techniques have been developed [72]. According to Mendizabal-Ruiz et al. [73], these representations can be divided into three categories: single-value mapping, multidimensional sequence mapping, and cumulative sequence mapping. Thereby, we study 6 numerical mapping techniques (or representations), which will be presented below: Voss [74], Integer [73, 75], Real [76], Z-curve [77], EIIP [78] and Complex Numbers [72, 79, 80].

3.3.1. Voss Representation

This representation can use single or multidimensional vectors. Fundamentally, this approach transforms a sequence $\mathbf{s} \in \{A, C, G, T\}^N$ into a matrix $\mathbf{V} \in \{0,1\}^{4\times N}$ such that $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]^T$, where T is the transpose operator and each $\mathbf{v}_i$ array is constructed according to the following relation:

$$v_i[n] = \begin{cases} 1, & s[n] = \alpha[i] \\ 0, & s[n] \neq \alpha[i] \end{cases}, \text{ where } \alpha = (A, C, G, T), \qquad n = 0, 1, \ldots, N-1. \tag{2}$$

As a result, each row of matrix $\mathbf{V}$ may be seen as an array that marks each base position such that the first row denotes the presence of base A, row two for base C, row three base G and the last row for base T. For example, let $\mathbf{s} = (G, A, G, A, G, T, G, A, C, C, A)$ be a sequence that needs to be represented using Voss representation, therefore, $\mathbf{v}_1 = (0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1)$, which represents the locations of bases A, $\mathbf{v}_2 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0)$ for bases C, $\mathbf{v}_3 = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0)$ for the G bases, $\mathbf{v}_4 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$ for T bases. Then, using the DFT in the indicator sequences shown above, we obtain (see Equation 3):

$$V_i[k] = \sum_{n=0}^{N-1} v_i[n] e^{-j\frac{2\pi}{N}kn}, \ \forall\ i \in [1, 4], \qquad k = 0, 1, \ldots, N-1. \tag{3}$$

The power spectrum of a biological sequence can be obtained by Equation (4):

$$P_V[k] = \sum_{i=1}^{4} |V_i[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{4}$$

3.3.2. Integer Representation

This representation is one-dimensional [75, 73]. This mapping can be obtained by substituting the four nucleotides (T, C, A, G) of a biological sequence for integers (0, 1, 2, 3), respectively, e.g., let $\mathbf{s}$ = (G, A, G, A, G, T, G, A, C, C, A), thus, $\mathbf{d}$ = (3, 2, 3, 2, 3, 0, 3, 2, 1, 1, 2), as exposed in Equation (5). The DFT and power spectrum are presented in Equation (6).

$$d[n] = \begin{cases} 3, & s[n] = G \\ 2, & s[n] = A \\ 1, & s[n] = C, \\ 0, & s[n] = T \end{cases} \qquad n = 0, 1, \ldots, N-1. \tag{5}$$

$$D[k] = \sum_{n=0}^{N-1} d[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_D[k] = |D[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{6}$$

3.3.3. Real Representation

In this representation, Chakravarthy et al. [76] use real mapping based on the complement property of the complex mapping of [69]. This mapping applies negative decimal values for the purines (A, G), and positive decimal values for the pyrimidines (C, T), e.g., let $\mathbf{s} = (G, A, G, A, G, T, G, A, C, C, A)$, thus, $\mathbf{r}$ = (-0.5, -1.5, -0.5, -1.5, -0.5, 1.5, -0.5, -1.5, 0.5, 0.5, -1.5), as Equation (7) and Equation (8).

$$r[n] = \begin{cases} -0.5, & s[n] = G \\ -1.5, & s[n] = A \\ 0.5, & s[n] = C, \\ 1.5, & s[n] = T \end{cases} \qquad n = 0, 1, \ldots, N-1. \tag{7}$$

$$R[k] = \sum_{n=0}^{N-1} r[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_R[k] = |R[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{8}$$

3.3.4. Z-curve Representation

The Z-curve scheme is a three-dimensional curve presented by [77], to encode DNA sequences with more biological semantics. Essentially, we can inspect a given sequence $s[n]$ of length N, taking into account the n-th element of the sequence $(n = 1, 2, \ldots, N)$. Then, we denote the cumulative occurrence numbers A_n, C_n, G_n and T_n for each base A, C, G and T, as the number of times that a base occurred from $s[1]$ up until $s[n]$. Fundamentally, this method reduces the number of indicator sequences from four (Voss) to three (Z-curve) in a symmetrical way for all four components [81]. Therefore:

$$A_n + C_n + G_n + T_n = n \tag{9}$$

Where the Z-curve consists of a series of nodes $P_1, P_2, \ldots, P_N$, whose coordinates $x[n]$, $y[n]$, and $z[n]$ $(n = 1, 2, \ldots, N)$ are uniquely determined by the Z-transform, shown in Equation (10):

$$P[n] = \begin{cases} x[n] = (A_n + G_n) - (C_n + T_n) \\ y[n] = (A_n + C_n) - (G_n + T_n), \\ z[n] = (A_n + T_n) - (C_n + G_n) \end{cases} \tag{10}$$
$$x[n], y[n], z[n] \in [-n, n], \qquad n = 1, 2, \ldots, N.$$

The coordinates $x[n]$, $y[n]$, and $z[n]$ represent three independent distributions that fully describe a sequence [72]. Therefore, we will have three distributions with definite biological significance: (1) $x[n]$ = purine/pyrimidine, (2) $y[n]$ = amino/keto, (3) $z[n]$ = weak hydrogen bonds/strong hydrogen bonds [77], e.g., let $\mathbf{s}$ = (G, A, G, A, G, T, G, A, C, C, A), thus, $\mathbf{x}$ = $(1, 2, 3, 4, 5, 4, 5, 6, 5, 4, 5)$; $\mathbf{y}$ = (-1, 0, -1, 0, -1, -2, -3, -2, -1, 0, 1); $\mathbf{z}$ = $(-1, 0, -1, 0, -1, 0, -1, 0, -1, -2, -1)$. Essentially, the difference between each dimension at the n-th position and the previous $(n-1)$ position can be either 1 or -1 [77]. Therefore, we may define the following set of equations in order to update the values of each dimension array considering that $x[-1] = y[-1] = z[-1] = 0$:

$$x[n] = \begin{cases} x[n-1] + 1, & s[n] = A \text{ or } G \\ x[n-1] - 1, & s[n] = C \text{ or } T \end{cases} \tag{11}$$

$$y[n] = \begin{cases} y[n-1] + 1, & s[n] = A \text{ or } C \\ y[n-1] - 1, & s[n] = G \text{ or } T \end{cases}, \qquad n = 1, 2, \ldots, N. \tag{12}$$

$$z[n] = \begin{cases} z[n-1] + 1, & s[n] = A \text{ or } T \\ z[n-1] - 1, & s[n] = G \text{ or } C \end{cases} \tag{13}$$

Finally, the DFT and power spectrum of the Z-Curve representation may be defined as [82]:

$$X[k] = \sum_{n=1}^{N} x[n] e^{-j\frac{2\pi}{N}kn}, \quad Y[k] = \sum_{n=1}^{N} y[n] e^{-j\frac{2\pi}{N}kn}, \quad Z[k] = \sum_{n=1}^{N} z[n] e^{-j\frac{2\pi}{N}kn}. \tag{14}$$

$$P_C[k] = |X[k]|^2 + |Y[k]|^2 + |Z[k]|^2, \qquad k = 1, 2, \ldots, N. \tag{15}$$

3.3.5. EIIP Representation

Nair and Sreenadhan [78] proposed EIIP values of nucleotides to represent biological sequences and to locate exons. According to the authors, a numerical sequence representing the distribution of free electron energies can be called *"EIIP indicator sequence"*, e.g., let $\mathbf{s}$ = (G, A, G, A, G, T, G, A, C, C, A), thus, $\mathbf{b}$ = (0.0806, 0.1260, 0.0806, 0.1260, 0.0806, 0.1335, 0.0806, 0.1260, 0.1340, 0.1340, 0.1260), as shown in Equation (16). The DFT and power spectrum of this representation are presented in Equation (17).

$$b[n] = \begin{cases} 0.0806, & s[n] = G \\ 0.1260, & s[n] = A \\ 0.1340, & s[n] = C, \\ 0.1335, & s[n] = T \end{cases} \qquad n = 0, 1, \ldots, N-1. \tag{16}$$

$$B[k] = \sum_{n=0}^{N-1} b[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_B[k] = |B[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{17}$$

3.3.6. Complex Numbers Representation

This numerical mapping has the advantage of better translating some of the nucleotides features into mathematical properties [80] and represents the complementary nature of AT and CG pairs [72]; e.g., let $\mathbf{s}$ = (G, A, G, A, G, T, G, A, C, C, A), thus, $\bar{\mathbf{r}} = (-1-j,\ 1+j,\ -1-j,\ 1+j,\ -1-j,\ 1-j,\ -1-j,\ 1+j,\ -1+j,\ -1+j,\ 1+j)$, as shown in Equation (18). The DFT and power spectrum of this representation are presented in Equation (19).

$$\bar{r}[n] = \begin{cases} -1-j, & s[n] = G \\ 1+j, & s[n] = A \\ -1+j, & s[n] = C, \\ 1-j, & s[n] = T \end{cases} \qquad n = 0, 1, \ldots, N-1. \tag{18}$$

$$\bar{R}[k] = \sum_{n=0}^{N-1} \bar{r}[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_{\bar{R}}[k] = |\bar{R}[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{19}$$

3.3.7. Features

The feature extraction is applied in each representation with Fourier transform, adopting Peak to Average Power Ratio (PAPR), mistakenly confused with the Signal to Noise Ratio (SNR), average power spectrum, median, maximum, minimum, sample standard deviation, population standard deviation, percentile (15/25/50/75), amplitude, variance, interquartile

range, semi-interquartile range, coefficient of variation, skewness, and kurtosis. Since according to [83] the RNA has a statistical phenomenon known as period-3 behavior or 3-base periodicity, where the peak power will always be at the sample $N/3$. Nevertheless, the PAPR is defined as [84]:

$$PAPR = \frac{\max_{0 \leq k \leq N-1}(P[k])}{\frac{1}{N}\sum_{k=0}^{N-1} P[k]} \tag{20}$$

3.4. Entropy

Information theory has been widely used in bioinformatics [85, 86]. Based on this, we consider the study of [87], which applied an algorithmic and mathematical approach to DNA code analysis using entropy and phase plane. Fundamentally, according to [86], entropy is a measure of the uncertainty associated with a probabilistic experiment. To generate a probabilistic experiment, we use a known method in bioinformatics, the k-mer (our pipeline is shown in Figure 4).

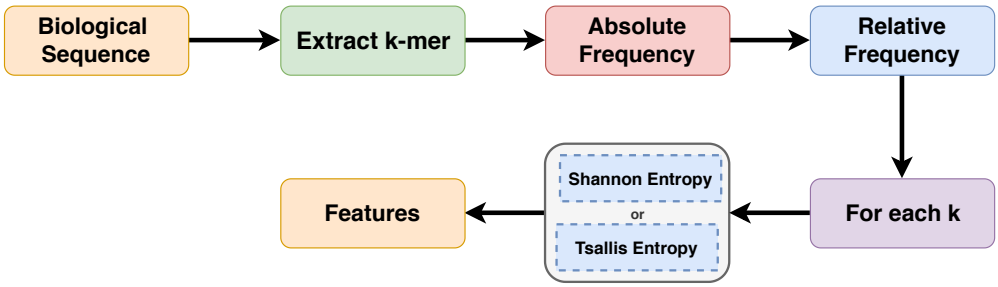

Figure 4: Entropy Pipeline. (1) Each sequence is mapped in k-mers; (2) The absolute frequency of each k is calculated; (3) Based on absolute frequency, the relative frequency is calculated; (4) The Tsallis or Shannon entropy is applied to each k; Finally, (5) features are generated.

In this method, each sequence is mapped in the frequency of neighboring bases k, generating statistical information. The k-mer is denoted in this work by P_k, corresponding to Equation (21).

$$P_k(\mathbf{s}) = \frac{c_i^k}{N-k+1} = \left(\frac{c_1^1}{N-1+1}, \ldots, \frac{c_4^1}{N-1+1}, \frac{c_{4+1}^2}{N-2+1}, \ldots, \frac{c_i^k}{N-k+1} \right) \qquad k = 1, 2, \ldots, 24. \tag{21}$$

We applied this equation to each sequence with frequencies of $k = 1, 2, \ldots, 24$. Where, c_i^k is the number of substring occurrences with length k in a sequence ($\mathbf{s}$) with length N, in which the index $i \in \{1, 2, \ldots, 4^1 + \ldots + 4^k\}$ represents the analyzed substring. For a better understanding, Figure 5 demonstrated an example with $k = 6$ and $k = 9$.

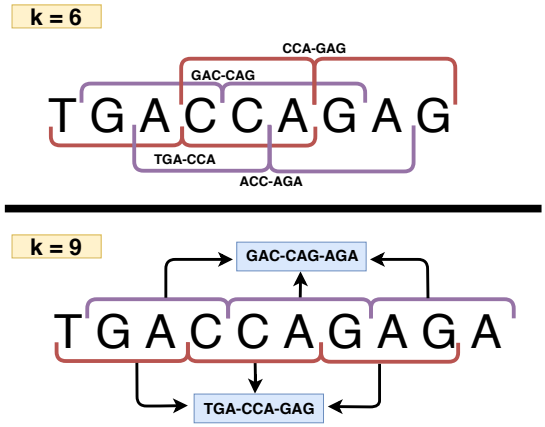

Figure 5: k-mer Workflow. Example with $k = 6$ and $k = 9$.

Basically, histograms with short bins are adopted, such as [{A}, {C}, {G}, {T}], that occur for $k = 1$, up to histograms with long sequence counting bins such as [{$GGGGGGGGGGGG$}, ..., {$AAAAAAAAAAAA$}], that result for $k = 12$. Where, after counting the absolute frequencies of each k, we generate relative frequencies (see Equation (21)), and then apply Shannon and Tsallis entropy to generate the features.

3.4.1. Shannon and Tsallis Entropy

Fundamentally, we chose Shannon entropy, because it quantifies the amount of information in a variable [88], that is, we can reach a single value that

quantifies the information contained in different observation periods (e.g., our case: k-mer). However, according to [89], it is important to explore a generalized form of the Shannon's entropy. Based on this, we have opted for a generalized entropy proposed by Tsallis, applied by several works in the literature [90, 91]. Thereby, for a discrete random variable F taking values in $\{f[0], f[1], f[2], \ldots, f[N-1]\}$ with probabilities $\{p[0], p[1], p[2], \ldots, p[N-1]\}$, represented as $P(F = f[n]) = p[n]$. The Shannon (Equation 22) and Tsallis (Equation 23) entropy associated with this variable is given by the following expressions:

$$H_S[k] = -\sum_{n=0}^{N-1} p_k[n]\ log_2\ p_k[n] \qquad k = 1, 2, \ldots 24. \tag{22}$$

$$H_T[k] = \frac{1}{q-1}\left(1 - \sum_{n=0}^{N-1} p_k[n]^q\right) \qquad k = 1, 2, \ldots 24. \tag{23}$$

Where k represents the analyzed k-mer, N the number of possible events and $p[n]$ the probability that event n occurs.

3.5. Complex Networks

Complex networks are widely used in mathematical modeling and have been an extremely active field in recent years [92], as well as becoming an ideal research area for mathematicians, computer scientists, and biologists. Based on this, we consider the study of [59], in which we propose a feature extraction model based on complex networks, as shown in Figure 6.

Each sequence is mapped to the frequency of neighboring bases k (k = 3 - see Figure 5). This mapping is converted into an undirected graph represented by an adjacency matrix, in which we applied a threshold scheme for feature extraction, thus generating our characteristic vector. Fundamentally, we represent our structure by undirected weighted graphs. According to [92], a graph $G = \{V, E\}$ is structured by a set V of vertices (or nodes) connected by a set E of edges (or links). Each edge reflects a link between two vertices, e.g., $e_p = (i, j)$ connection between the vertices i and j [92]. If there is an edge connecting the vertices i and j, the elements a_{ij} are equal to 1, and equal to 0 otherwise.

In our case, the graph is undirected, that is, the adjacency matrix A is symmetric, e.g., elements $a_{ij} = a_{ji}$ for any i and j [92]. Furthermore, we apply a threshold scheme presented by [59], in which we extract weight of

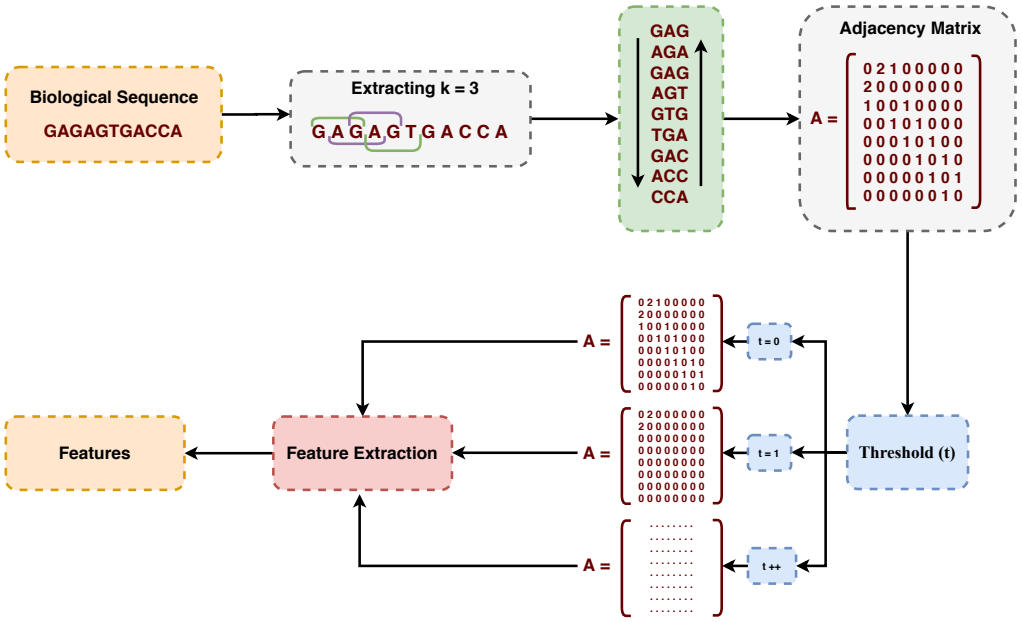

Figure 6: Complex Networks Pipeline. (1) Each sequence is mapped in the frequency of neighboring bases k (k = 3); (2) This mapping is converted to a undirected graph represented by an adjacency matrix; (3) Feature extraction is performed using a threshold scheme; Finally, (4) the features are generated.

the edges to capture adjacencies at different frequencies. Finally, as features, several network characterization measures were obtained, based on [59, 93], among them: Betweenness, assortativity, average degree, average path length, minimum degree, maximum degree, degree standard deviation, frequency of motifs (size 3 and 4), clustering coefficient.

3.6. Normalization, Training and Evaluation Metrics

Data normalization is a preprocessing technique often applied to a dataset. Essentially, features can have different dynamic ranges. This problem may have a stronger effect in the induction of a predictive model, mainly for distance-based ML algorithms. Consequently, the application of a normalization procedure makes the ranges similar, reducing this problem [94]. We used the min-max normalization, which reduces the data range to 0 and 1 (or -1 to 1, if there are negative values) [2]. The general formula is given as (Equation (24)) [95]:

$$x'_{ij} = \frac{x_{ij} - \min(j)}{\max(j) - \min(j)}. \quad (24)$$

Where x is the original value and x'_{ij} is its normalized version. Furthermore, $\min(j)$ and $\max(j)$ are, respectively, the smallest and largest values of a feature j [6, 95]. Next, we investigate three classification algorithms, such as Random Forest (RF) [96], AdaBoost [97] and CatBoost [98]. We chose these ML algorithms because they induce interpretable predictive models when humans can easily understand the internal decision-making process. Thus, domain experts can validate the knowledge used by the models for the classification of new sequences [6]. Finally, to induce our models, we used 70% of samples for *training* (with 10-fold cross-validation) and 30% for *testing*, as shown in Table 2.

Table 2: Number of sequences used for training and testing in each dataset.

Case Study	Dataset	Samples	Training	Testing
I	*A. trichopoda*	9112	6378	2734
	A. thaliana	5080	3556	1524
	C. sinensis	4430	3101	1329
	C. sativus	3460	2422	1038
	R. communis	6974	4881	2093
II	*lncRNA vs. sncRNA*	2582	1807	775
	lncRNA vs. Antisense	114	79	35
	circRNA vs. lncRNA	5080	3556	1524

The methods were evaluated with four measures: Sensitivity (SE - Equation 26), Specificity (SPC - Equation 27), Accuracy (ACC - Equation 25), and Cohen's kappa coefficient [99] (Equation 28).

$$ACC = \frac{TP + TN}{TN + FP + TP + FN} \quad (25) \qquad SPC = \frac{TN}{TN + FP} \quad (27)$$

$$SE = \frac{TP}{TP + FN} \quad (26) \qquad Kappa = \frac{p_o - p_e}{1 - p_e} \quad (28)$$

These measures use True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values, where: TP measures the correctly

predicted positive label; TN represents the correctly classified negative label; FP describes all those negative entities that are incorrectly classified as positive and; FN represents the positive label that are incorrectly classified as the negative label.

4. Results

This section shows experimental results from 9 feature extraction approaches with mathematical models for biological sequences, divided into two parts: Case Study I and Case Study II.

4.1. Case Study I

Initially, we induced models with the RF, AdaBoost, and CatBoost classifiers in the training set of three datasets (*A. trichopoda*, *A. thaliana*, and *R. communis*). Our initial goal is to choose the best classifier to follow in the testing phases. Thereby, to estimate the real accuracy, we applied 10-fold cross-validation, as shown in Table 3.

Table 3: Accuracy for the training set (*A. trichopoda*, *A. thaliana*, and *R. communis*) using 10-fold cross-validation.

Dataset	Model	RF	AdaBoost	CatBoost
A. trichopoda	Z-curve	0.90 (± 0.03)	0.91 (± 0.02)	**0.92 (± 0.02)**
	Binary	0.92 (± 0.02)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Real	0.91 (± 0.02)	0.93 (± 0.02)	**0.94 (± 0.02)**
	Integer	0.91 (± 0.02)	0.93 (± 0.02)	**0.94 (± 0.02)**
	EIIP	0.92 (± 0.02)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Complex	0.92 (± 0.03)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Graphs	0.92 (± 0.02)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Shannon	0.92 (± 0.02)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Tsallis	0.92 (± 0.02)	**0.94 (± 0.02)**	**0.94 (± 0.02)**
A. thaliana	Z-curve	**0.95 (± 0.02)**	0.93 (± 0.03)	0.94 (± 0.02)
	Binary	**0.94 (± 0.02)**	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	Real	**0.95 (± 0.02)**	0.94 (± 0.02)	**0.95 (± 0.02)**
	Integer	**0.94 (± 0.02)**	**0.94 (± 0.02)**	**0.94 (± 0.02)**
	EIIP	**0.95 (± 0.02)**	0.94 (± 0.02)	0.95 (± 0.03)
	Complex	0.94 (± 0.02)	0.94 (± 0.02)	**0.94 (± 0.01)**
	Graphs	0.94 (± 0.02)	0.94 (± 0.02)	**0.95 (± 0.02)**

	Shannon	0.94 (± 0.02)	0.94 (± 0.02)	**0.95 (± 0.02)**
	Tsallis	**0.94 (± 0.02)**	**0.94 (± 0.02)**	**0.94 (± 0.02)**
R. communis	Z-curve	**0.93 (± 0.02)**	0.92 (± 0.02)	**0.93 (± 0.02)**
	Binary	**0.95 (± 0.01)**	0.95 (± 0.02)	0.95 (± 0.02)
	Real	**0.95 (± 0.02)**	0.94 (± 0.02)	0.94 (± 0.02)
	Integer	**0.94 (± 0.01)**	**0.94 (± 0.01)**	0.94 (± 0.02)
	EIIP	0.95 (± 0.02)	0.95 (± 0.02)	**0.95 (± 0.01)**
	Complex	0.95 (± 0.02)	**0.95 (± 0.01)**	**0.95 (± 0.01)**
	Graphs	**0.95 (± 0.01)**	**0.95 (± 0.01)**	0.95 (± 0.02)
	Shannon	0.95 (± 0.02)	0.95 (± 0.02)	**0.95 (± 0.01)**
	Tsallis	**0.95 (± 0.01)**	**0.95 (± 0.01)**	**0.95 (± 0.01)**

Assessing each classifier, we noted that the best performance was of the CatBoost with all mathematical models in *A. trichopoda*, followed by AdaBoost (6 best results) and RF (no better results). In *A. thaliana*, CatBoost kept the best performance (7 best results), followed by RF (6 best results) and AdaBoost (3 best results). In contrast, the RF classifier obtained the best results (6) in *R. communis*, followed by CatBoost (5 best results) and AdaBoost (3 best results). Based on this, we continued testing the models with the CatBoost classifier. Thus, in Table 4, we present the results of all mathematical models using 4 evaluation metrics.

Table 4: Performance analysis. This table compares the sensitivity, specificity, accuracy and kappa metrics for each model in the test sets using CatBoost classifier.

Dataset	Model	SE	SPC	ACC	Kappa
A. trichopoda	Z-curve	0.9744	0.8566	0.9155	0.8310
	Binary	0.9795	0.9005	0.9400	0.8800
	Real	**0.9802**	0.8837	0.9320	0.8639
	Integer	0.9773	0.8822	0.9298	0.8595
	EIIP	0.9781	0.8990	0.9386	0.8771
	Complex	**0.9802**	0.9012	**0.9407**	**0.8815**
	Graphs	0.9737	**0.9020**	0.9378	0.8756
	Shannon	0.9781	**0.9020**	0.9400	0.8800
	Tsallis	0.9795	0.9005	0.9400	0.8800
	Z-curve	0.9777	0.9383	0.9580	0.9160
	Binary	0.9619	0.9449	0.9534	0.9068

A. thaliana	Real	**0.9803**	0.9409	**0.9606**	**0.9213**
	Integer	0.9698	0.9436	0.9567	0.9134
	EIIP	0.9646	0.9449	0.9547	0.9094
	Complex	0.9724	0.9409	0.9567	0.9134
	Graphs	0.9685	0.9423	0.9554	0.9108
	Shannon	0.9738	**0.9462**	0.9600	0.9200
	Tsallis	0.9764	0.9409	0.9587	0.9173
C. sinensis	Z-curve	0.9021	**0.8707**	0.8864	0.7728
	Binary	0.8901	**0.8707**	0.8804	0.7607
	Real	0.9142	0.8571	0.8856	0.7713
	Integer	0.8825	0.8692	0.8758	0.7517
	EIIP	0.8840	0.8526	0.8683	0.7367
	Complex	0.9081	0.8496	0.8789	0.7577
	Graphs	0.9006	0.8632	0.8819	0.7637
	Shannon	0.9172	0.8586	0.8879	0.7758
	Tsallis	**0.9262**	0.8541	**0.8901**	**0.7803**
C. sativus	Z-curve	0.8979	0.8478	0.8728	0.7457
	Binary	0.9056	0.8459	0.8757	0.7514
	Real	0.9268	0.8439	0.8854	0.7707
	Integer	0.9056	**0.8536**	0.8796	0.7592
	EIIP	0.8979	0.8459	0.8719	0.7437
	Complex	0.9326	0.8343	0.8834	0.7669
	Graphs	0.9075	**0.8536**	0.8805	0.7611
	Shannon	0.9326	0.8382	0.8854	0.7707
	Tsallis	**0.9403**	0.8401	**0.8902**	**0.7803**
R. communis	Z-curve	0.9446	0.9140	0.9293	0.8586
	Binary	0.9417	0.9589	0.9503	0.9006
	Real	**0.9589**	0.9408	0.9498	0.8997
	Integer	0.9465	0.9456	0.9460	0.8920
	EIIP	0.9455	0.9551	0.9503	0.9006
	Complex	0.9398	0.9561	0.9479	0.8958
	Graphs	0.9455	0.9542	0.9498	0.8997
	Shannon	0.9388	0.9589	0.9489	0.8978
	Tsallis	0.9417	**0.9608**	**0.9513**	**0.9025**

As can be seen, all models presented excellent results, with the worst performance (ACC) of 0.8901 (*C. sinensis*) and the best of 0.9606 (*A. thaliana*). That is, all models were robust in different datasets without a high loss of

performance. Assessing each metric individually, we realized that in SE, the best performance was from Real representation (3 datasets), followed by Tsallis (2 datasets) and Complex numbers (1 dataset). In SPC, the best results were from Entropy (3 datasets), followed by Graphs (2 datasets). In ACC, Tsallis presented the best performance (3 datasets), followed by Real representation and Complex numbers (1 dataset). For each dataset, we can see in *A. trichopoda* the best ACC was 0.9407 (Complex); *A. thaliana* with 0.9606 (Real); *C. sinensis* with 0.8901 (Tsallis); *C. sativus* with 0.8902 (Tsallis); and *R. communis* with 0.9513 (Tsallis).

4.2. Case Study II

After evaluating all methods in 5 different datasets (lncRNA of different species) and observing their results, we applied a second case study, where we used only three mathematical models for generalization analysis, including GSP (Fourier + complex numbers), entropy (Tsallis) and graphs (complex networks). Here, our objective was to analyze how each model behaved in different biological sequence classification problems. For this, we tested 3 new datasets established in Section 3.1.2, as can be seen in Figure 7.

Again, all showed robust results, in which, graph-based models are the best in 2 of the 3 problems analyzed, followed by entropy and GSP. In the first three datasets, our methods achieved excellent accuracy. Furthermore, if we analyze at the last problem (circRNA vs. lncRNA), our approaches were effective when compared to our references that reached an ACC of 0.7780 [66] and 0.7890 [21] in their datasets against 0.8307 from our best model (graph - using these comparisons as an (indirect) reference indicator).

4.3. Statistical Significance Tests

The statistical significance was assessed in both case studies (difference in ACC), using Friedman's statistical test and the Conover post-hoc test. Thereby, our null hypothesis ($H0 = M(1) = M(2) = \ldots = M(k)$), is tested against the alternative hypothesis (H_A = at least one model has statistical significance ($\alpha = 0.05$, $p < \alpha$)). First, we apply the global test in the case study I, in which the Friedman test indicates significance ($\chi^2(8) = 17.34$, p-value $= 0.0268$), that is, we can reject $H0$, as $p < 0.05$. Thus, it is essential to execute the post-hoc statistical test. Conover statistics values were obtained, as well as p-values (see Table 5), using 95% of significance ($\alpha = 0.05$).

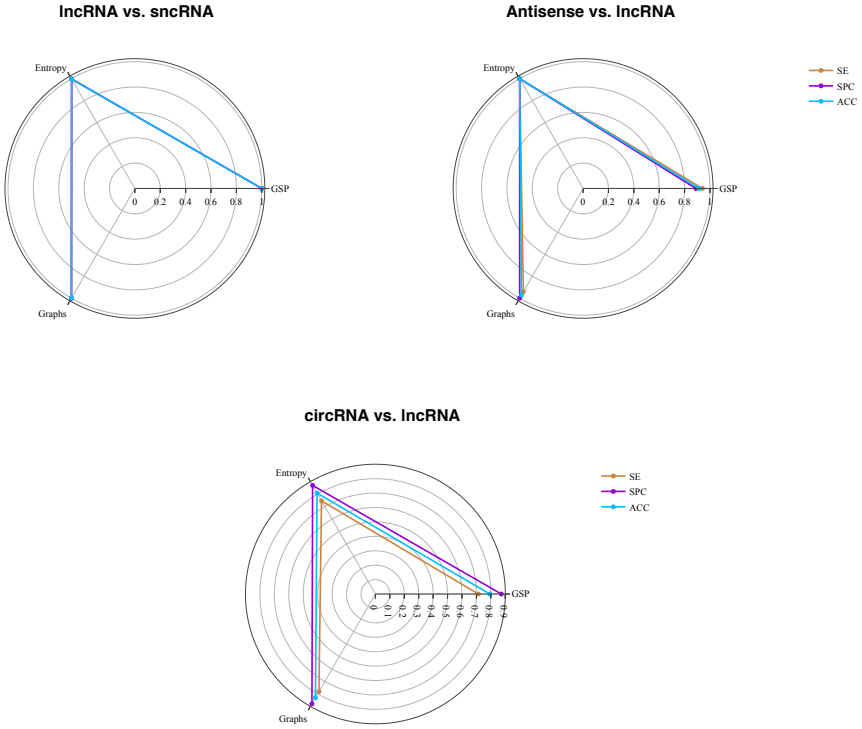

Figure 7: Performance analysis of three mathematical models, GSP (fourier + complex numbers), entropy (Tsallis) and graphs (complex networks), for different problems.

Table 5: Conover statistics values - The accepted alternative hypothesis is in bold (p-values for $\alpha = 0.05$).

	Z-curve	**Binary**	**Real**	**Integer**	**EIIP**	**Complex**	**Graphs**	**Shannon**
Binary	0.5580	-	-	-	-	-	-	-
Real	0.1416	0.3671	-	-	-	-	-	-
Integer	0.7896	0.3956	0.0852	-	-	-	-	-
EIIP	0.9574	0.5230	0.1284	0.8309	-	-	-	-
Complex	0.3671	0.7489	0.5580	0.2451	0.3399	-	-	-
Graphs	0.5580	1.0000	0.3671	0.3956	0.5230	0.7489	-	-
Shannon	0.0687	0.2057	0.7089	**0.0390**	0.0616	0.3399	0.2057	-
Tsallis	**0.0146**	0.0550	0.2898	**0.0075**	**0.0128**	0.1050	0.0550	0.4892

Concerning to the Conover post-hoc test, entropy-based models have highly significant differences for the Z-curve ($p < 0.0146$), Integer ($p < 0.0075$

- Tsallis and $p < 0.0390$ - Shannon), and EIIP ($p < 0.0128$). Possibly, these results indicate that entropy has a more significant performance when compared to representations with Fourier. However, other mathematical models in case study I do not differ significantly, indicating their efficiency in all datasets. Now, evaluating case study II, we realized that the global test with Friedman's statistical test is not significant, in which we obtained $\chi^2(2) = 1.64$, p-value $= 0.4412$, indicating that the three studied feature extraction techniques show a similar performance in the problems, once more confirming the effectiveness and robustness of all mathematical models.

4.4. Computational Time

In addition, we also assessed the computational time cost of each tested model. To do this, we ran three models, GSP (Fourier + complex numbers), entropy (Tsallis) and graphs (complex networks)), in 1291 random sequences, as shown in Figure 8.

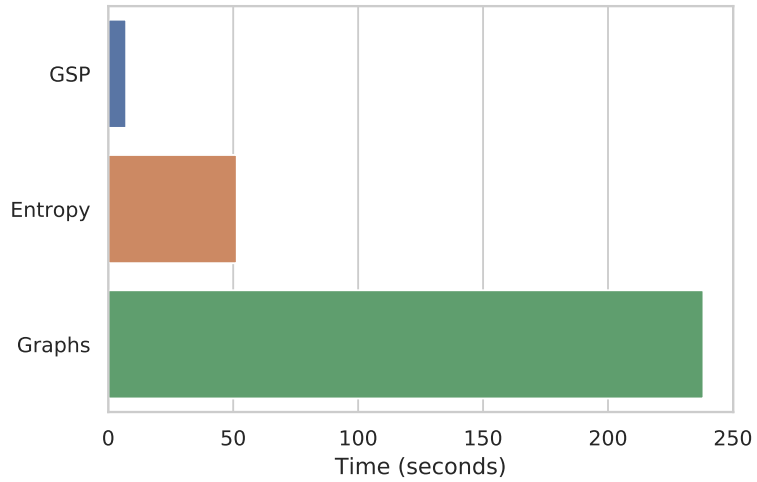

Figure 8: Execution Time.

We performed the experiments using Intel Core i3-9100F CPU (3.60GHz), 16GB memory, and running in Debian GNU/Linux 10. The lowest cost in computational time is for models based on GSP (0m7.183s) and entropy (0m51.427s), while graphs (3m58.208s) have a much higher cost. These results demonstrated that, although the models present a similar performance, the computational time efficiency is significantly different.

5. Discussion

This section discusses our findings in terms of whether they support our hypothesis (*feature extraction approaches based on mathematical models are as efficient and generalist as biological approaches*). Overall, several experimental tests were assumed in this research, in which all feature extraction approaches based on mathematical models showed excellent results, as can be seen in Table 4 and Figure 7. Regarding its performance in distinct classification problems, case study II, we used only three mathematical models for generalization analysis, including GSP (Fourier + complex numbers), entropy (Tsallis) and graphs (complex networks). In which, entropy and graph-based models reported the best performance followed by GSP. Furthermore, all models maintained robust results in different sequence classification problems.

Furthermore, to fully support our hypothesis, we also compare three mathematical models shown in Figure 7 concerning a biological and hybrid approach, in four datasets ((lncRNA vs. mRNA (case study I)); (lncRNA vs. sncRNA; lncRNA vs. Antisense; circRNA vs. lncRNA (case study II)). Thus, we generate our biological model using some of the most applied features in Figure 1. Thus, features used by the models are:

- **Biological:** The features were provided by [19]: Fickett TESTCODE score, isoelectric point, open reading frame (ORF) length, and ORF integrity.

- **Hybrid:** The features were generated by one of the most current approaches in the literature (lncFinder [20] - 2018). We classify this model as a hybrid because it uses a combination of biological and mathematical features. Among the biological characteristics is Logarithm-distance of hexamer on ORF, length and coverage of the longest ORF. Regarding mathematical features, [20] uses an EIIP-based physicochemical property with Fourier Transform (similar to our approach with GSP, but using only EIIP mapping).

For a fair comparison, the new experiments follow the same methodology (70% training, 30% test, and CatBoost classifier), as shown in Table 6.

As can be seen, the hybrid model (0.9915) reported the best performance in the first dataset (lncRNA vs. mRNA), followed by the biological (0.9816) and our mathematical model (Entropy - 0.9587), with only a difference of

Table 6: Performance analysis of three mathematical models against a biological and hybrid model for different sequence classification problems.

lncRNA vs. mRNA				**lncRNA vs. sncRNA**			
Models	**SE**	**SPC**	**ACC**	**Models**	**SE**	**SPC**	**ACC**
GSP	0.9724	0.9409	0.9567	GSP	**1.0000**	**1.0000**	**1.0000**
Entropy	**0.9764**	**0.9409**	**0.9587**	Entropy	0.9974	0.9974	0.9974
Graphs	0.9685	0.9423	0.9554	Graphs	**1.0000**	**1.0000**	**1.0000**
Biological	**0.9869**	**0.9764**	**0.9816**	Biological	0.7855	0.8273	0.8065
Hybrid	**0.9895**	**0.9934**	**0.9915**	Hybrid	0.9509	0.9485	0.9497

lncRNA vs. Antisense				**circRNA vs. lncRNA**			
Models	**SE**	**SPC**	**ACC**	**Models**	**SE**	**SPC**	**ACC**
GSP	0.9412	0.8889	0.9143	GSP	0.7139	0.8727	0.7933
Entropy	**1.0000**	**1.0000**	**1.0000**	Entropy	0.7467	0.8701	0.8084
Graphs	0.9412	1.0000	0.9714	Graphs	**0.7822**	**0.8793**	**0.8307**
Biological	0.8889	0.9412	0.9143	Biological	0.6024	0.7612	0.6818
Hybrid	0.9412	0.7778	0.8571	Hybrid	0.7283	0.8819	0.8051

0.0328 and 0.0229, respectively. However, it is relevant to highlight that the biological and hybrid models use the ORF descriptor, a highly employed feature for discovering coding sequences and which, according to [15, 6] is an essential guideline for distinguishing lncRNAs from mRNA. In other words, this explains the great result, but, as mentioned at the beginning of this manuscript, this type of feature with a biological insight is often difficult to reuse or adapt to another specific problem. Thereby, our study has an gain in terms of generalization, since this would not be possible only with the ORF. If we analyze at the hybrid model, in this first dataset, the gain was minimal compared to the biological (0.0099), which again confirms the efficiency of the previously mentioned features. This is different from our approaches, which showed an excellent result without using bias features for the analyzed problem.

Consequently, this hypothesis is proven in the other three datasets, where our mathematical models perform much better than the biological model, mainly in the fourth dataset (circRNA vs. lncRNA), in which we obtained a gain of 0.1489 in ACC. Regarding the hybrid model, it can be observed that the mixture of biological and mathematical characteristics helped to

keep the model competitive in all datasets, indicating the effectiveness of mathematical features. Even so, our models showed the best results in three of the four proposed problems. Therefore, our pipeline is efficient in terms of generalization to classify lncRNA from mRNA, as well as other biological sequence classification problems. We also assessed the statistical significance of the mathematical versus biological approach in the previously applied tests, in which entropy ($p < 0.0480$) and graphs ($p < 0.0200$) indicated significant results concerning the biological model. Lastly, considering all these findings, we fully support the suggested hypothesis.

6. Conclusion

This work proposed to analyze feature extraction approaches for biological sequence classification. Specifically, we concentrated our work on the study of feature extraction techniques using mathematical models. We analyzed mathematical models to propose efficient and generalist techniques for different problems. As a case study, we used lncRNA sequences. Moreover, we divided this paper into two case studies. In our experiments, as a starting point, 9 mathematical models for feature extraction were analyzed: 6 numerical mapping techniques with Fourier transform; Tsallis and Shannon entropy; Graphs (complex networks). Thereby, several biological sequence classification problems were adopted to validate the proposed approach.

As a result, all models presented excellent results, with performances (ACC) between 0.8901-0.9606 in case study I. In case study II, once more, all showed excellent results with models based on entropy and graphs showing the best performance, followed by GSP. Furthermore, to validate our study, we compared the performance of three mathematical models against a biological and hybrid approach, in four different datasets. In which, our models demonstrated suitable results, and was superior or competitive and robust in terms of generalization. In our experiments, we verified that mathematical approaches perform as accurately as biological approaches and have a better generalization capacity since they outperform biological features in scenarios not designed for them. Finally, among the different mathematical models tested in this work, the combination of k-mer and entropy, as well as graph-based models performs better than GSP at the cost of a significant increase in computational complexity.

Declaration of Competing interests

All authors declare that they have no conflict of interest.

Financial support

This project has been supported by a master scholarship from Federal University of Technology - Paraná (UTFPR) (Grant: April/2018) and CAPES (April/2019 and PROEX-11919694/D).

Acknowledgements

The authors would like to thank UTFPR-CP, ICMC-USP, and CAPES for the financial support given to this research.

References

[1] H. Lou, M. Schwartz, J. Bruck, F. Farnoud, Evolution of k-mer frequencies and entropy in duplication and substitution mutation systems, IEEE Transactions on Information Theory (2019).

[2] R. P. Bonidia, L. D. H. Sampaio, F. M. Lopes, D. S. Sanches, Feature extraction of long non-coding rnas: A fourier and numerical mapping approach, in: I. Nyström, Y. Hernández Heredia, V. Milián Núñez (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer International Publishing, Cham, 2019, pp. 469–479.

[3] R. Min, Machine Learning Approaches to Biological Sequence and Phenotype Data Analysis, University of Toronto, 2010.

[4] M.-R. Cao, Z.-P. Han, J.-M. Liu, Y.-G. Li, Y.-B. Lv, J.-B. Zhou, J.-H. He, Bioinformatic analysis and prediction of the function and regulatory network of long non-coding rnas in hepatocellular carcinoma, Oncology letters 15 (5) (2018) 7783–7793.

[5] W. J. d. S. Diniz, F. Canduri, Bioinformatics: an overview and its applications, Genet Mol Res 16 (1) (2017).

[6] R. Parmezan Bonidia, A. C. Ponce de Leon Ferreira de Carvalho, A. Rossi Paschoal, D. Sipoli Sanches, Selecting the most relevant features for the identification of long non-coding rnas in plants, in: 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), 2019, pp. 539–544. doi:10.1109/BRACIS.2019.00100.

[7] V. I. Jurtz, A. R. Johansen, M. Nielsen, J. J. Almagro Armenteros, H. Nielsen, C. K. Sønderby, O. Winther, S. K. Sønderby, An introduction to deep learning on biological sequence data: examples and solutions, Bioinformatics 33 (22) (2017) 3685–3690. doi:10.1093/bioinformatics/btx531.
URL https://doi.org/10.1093/bioinformatics/btx531

[8] M. E. Maros, D. Capper, D. T. Jones, V. Hovestadt, A. von Deimling, S. M. Pfister, A. Benner, M. Zucknick, M. Sill, Machine learning workflows to estimate class probabilities for precision cancer diagnostics on dna methylation microarray data, Nature Protocols (2020) 1–34.

[9] J. Li, W. Liu, Puzzle of highly pathogenic human coronaviruses (2019-ncov), Protein & Cell (2020) 1–4.

[10] D. Benvenuto, M. Giovanetti, A. Ciccozzi, S. Spoto, S. Angeletti, M. Ciccozzi, The 2019-new coronavirus epidemic: Evidence for virus evolution, Journal of Medical Virology 92 (4) (2020) 455–459. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmv.25688, doi:10.1002/jmv.25688.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jmv.25688

[11] C. Xu, S. A. Jackson, BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches, Genome Biology 20 (2019) 1–4. doi:https://doi.org/10.1186/s13059-019-1689-0.
URL https://doi.org/10.1186/s13059-019-1689-0

[12] D. Storcheus, A. Rostamizadeh, S. Kumar, A survey of modern questions and challenges in feature extraction, in: Feature Extraction: Modern Questions and Challenges, 2015, pp. 1–18.

[13] I. Guyon, S. Gunn, M. Nikravesh, L. A. Zadeh, Feature extraction: foundations and applications, Vol. 207, Springer, 2008.

[14] R. Saidi, S. Aridhi, E. M. Nguifo, M. Maddouri, Feature extraction in protein sequences classification: a new stability measure, in: Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, ACM, 2012, pp. 683–689.

[15] J. Baek, B. Lee, S. Kwon, S. Yoon, lncrnanet: Long non-coding rna identification using deep learning, Bioinformatics 1 (2018) 9.

[16] R. Muhammod, S. Ahmed, D. Md Farid, S. Shatabda, A. Sharma, A. Dehzangi, PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences, Bioinformatics 35 (19) (2019) 3831–3833. arXiv:http://oup.prod.sis.lan/bioinformatics/article-pdf/35/19/3831/30061688/btz165.pdf, doi:10.1093/bioinformatics/btz165. URL https://doi.org/10.1093/bioinformatics/btz165

[17] Q. Abbas, S. M. Raza, A. A. Biyabani, M. A. Jaffar, A review of computational methods for finding non-coding rna genes, Genes 7 (12) (2016) 113.

[18] N. Amin, A. McGrath, Y.-P. P. Chen, Evaluation of deep learning in non-coding rna classification, Nature Machine Intelligence 1 (5) (2019) 246.

[19] Y.-J. Kang, D.-C. Yang, L. Kong, M. Hou, Y.-Q. Meng, L. Wei, G. Gao, Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features, Nucleic acids research 45 (W1) (2017) W12–W16.

[20] S. Han, Y. Liang, Q. Ma, Y. Xu, Y. Zhang, W. Du, C. Wang, Y. Li, Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property, Briefings in Bioinformatics (2018).

[21] L. Chen, Y.-H. Zhang, G. Huang, X. Pan, S. Wang, T. Huang, Y.-D. Cai, Discriminating cirrnas from other lncrnas using a hierarchical extreme learning machine (h-elm) algorithm with feature selection, Molecular Genetics and Genomics 293 (1) (2018) 137–149.

[22] S. R. Eddy, Non-coding rna genes and the modern rna world, Nature Reviews Genetics 2 (12) (2001) 919.

[23] P. Kapranov, J. Cheng, S. Dike, D. A. Nix, R. Duttagupta, A. T. Willingham, P. F. Stadler, J. Hertel, J. Hackermüller, I. L. Hofacker, et al., Rna maps reveal new rna classes and a possible function for pervasive transcription, Science 316 (5830) (2007) 1484–1488.

[24] Y. Zhang, Y. Tao, Q. Liao, Long noncoding rna: a crosslink in biological regulatory network, Briefings in bioinformatics (2017).

[25] A. Li, Q. Zang, D. Sun, M. Wang, A text feature-based approach for literature mining of lncrna–protein interactions, Neurocomputing 206 (2016) 73–80.

[26] Y. Wang, Y. Li, Q. Wang, Y. Lv, S. Wang, X. Chen, X. Yu, W. Jiang, X. Li, Computational identification of human long intergenic non-coding rnas using a ga–svm algorithm, Gene 533 (1) (2014) 94–99.

[27] L. Wang, L. Kuang, S. Ye, M. F. B. Iqbal, T. Pei, et al., A novel method for lncrna-disease association prediction based on an lncrna-disease association network, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2018).

[28] W. Zhang, Q. Qu, Y. Zhang, W. Wang, The linear neighborhood propagation method for predicting long non-coding rna–protein interactions, Neurocomputing 273 (2018) 526–534.

[29] Q.-Z. Zhou, B. Zhang, Q.-Y. Yu, Z. Zhang, Bmncrnadb: a comprehensive database of non-coding rnas in the silkworm, bombyx mori, BMC bioinformatics 17 (1) (2016) 370.

[30] M. Q. Hassan, C. E. Tye, G. S. Stein, J. B. Lian, Non-coding rnas: Epigenetic regulators of bone development and homeostasis, Bone 81 (2015) 746–756.

[31] C. Ciaudo, N. Servant, V. Cognat, A. Sarazin, E. Kieffer, S. Viville, V. Colot, E. Barillot, E. Heard, O. Voinnet, Highly dynamic and sex-specific expression of micrornas during early es cell differentiation, PLoS genetics 5 (8) (2009) e1000620.

[32] X. Peng, L. Gralinski, C. D. Armour, M. T. Ferris, M. J. Thomas, S. Proll, B. G. Bradel-Tretheway, M. J. Korth, J. C. Castle, M. C. Biery, et al., Unique signatures of long noncoding rna expression in response to

virus infection and altered innate immune signaling, MBio 1 (5) (2010) e00206–10.

[33] C. Pastori, C. Wahlestedt, Involvement of long noncoding rnas in diseases affecting the central nervous system, RNA biology 9 (6) (2012) 860–870.

[34] Q. Zhang, Y. Wei, Z. Yan, C. Wu, Z. Chang, Y. Zhu, K. Li, Y. Xu, The characteristic landscape of lncrnas classified by rbp–lncrna interactions across 10 cancers, Molecular bioSystems 13 (6) (2017) 1142–1151.

[35] H.-L. V. Wang, J. A. Chekanova, Long noncoding rnas in plants, in: Long Non Coding RNA Biology, Springer, 2017, pp. 133–154.

[36] C. Di, J. Yuan, Y. Wu, J. Li, H. Lin, L. Hu, T. Zhang, Y. Qi, M. B. Gerstein, Y. Guo, et al., Characterization of stress-responsive lncrnas in arabidopsis thaliana by integrating expression, epigenetic and structural features, The Plant Journal 80 (5) (2014) 848–861.

[37] D. Wang, Z. Qu, L. Yang, Q. Zhang, Z.-H. Liu, T. Do, D. L. Adelson, Z.-Y. Wang, I. Searle, J.-K. Zhu, Transposable elements (te s) contribute to stress-related long intergenic noncoding rna s in plants, The Plant Journal 90 (1) (2017) 133–146.

[38] Y.-C. Zhang, J.-Y. Liao, Z.-Y. Li, Y. Yu, J.-P. Zhang, Q.-F. Li, L.-H. Qu, W.-S. Shu, Y.-Q. Chen, Genome-wide screening and functional analysis identify a large number of long noncoding rnas involved in the sexual reproduction of rice, Genome biology 15 (12) (2014) 512.

[39] Y. Fang, M. J. Fullwood, Roles, functions, and mechanisms of long noncoding rnas in cancer, Genomics, proteomics & bioinformatics 14 (1) (2016) 42–54.

[40] T. Derrien, R. Johnson, G. Bussotti, A. Tanzer, S. Djebali, H. Tilgner, G. Guernec, D. Martin, A. Merkel, D. G. Knowles, et al., The gencode v7 catalog of human long noncoding rnas: analysis of their gene structure, evolution, and expression, Genome research 22 (9) (2012) 1775–1789.

[41] J. Cheng, P. Kapranov, J. Drenkow, S. Dike, S. Brubaker, S. Patel, J. Long, D. Stern, H. Tammana, G. Helt, et al., Transcriptional maps of

10 human chromosomes at 5-nucleotide resolution, Science 308 (5725) (2005) 1149–1154.

[42] L. Ma, V. B. Bajic, Z. Zhang, On the classification of long non-coding rnas, RNA biology 10 (6) (2013) 924–933.

[43] R. Hu, X. Sun, lncrnatargets: a platform for lncrna target prediction based on nucleic acid thermodynamics, Journal of bioinformatics and computational biology 14 (04) (2016) 1650016.

[44] S. Chooniedass-Kothari, E. Emberley, M. Hamedani, S. Troup, X. Wang, A. Czosnek, F. Hube, M. Mutawe, P. Watson, E. Leygue, The steroid receptor rna activator is the first functional rna encoding a protein, FEBS letters 566 (1-3) (2004) 43–47.

[45] Y. He, X.-M. Meng, C. Huang, B.-M. Wu, L. Zhang, X.-W. Lv, J. Li, Long noncoding rnas: Novel insights into hepatocelluar carcinoma, Cancer letters 344 (1) (2014) 20–27.

[46] J. T. Kung, D. Colognori, J. T. Lee, Long noncoding rnas: past, present, and future, Genetics 193 (3) (2013) 651–669.

[47] L. Kong, Y. Zhang, Z.-Q. Ye, X.-Q. Liu, S.-Q. Zhao, L. Wei, G. Gao, Cpc: assess the protein-coding potential of transcripts using sequence features and support vector machine, Nucleic acids research 35 (suppl_2) (2007) W345–W349.

[48] L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher, W. Li, Cpat: Coding-potential assessment tool using an alignment-free logistic regression model, Nucleic acids research 41 (6) (2013) e74–e74.

[49] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, Y. Zhao, Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts, Nucleic acids research 41 (17) (2013) e166–e166.

[50] A. Li, J. Zhang, Z. Zhou, Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme, BMC bioinformatics 15 (1) (2014) 311.

[51] X.-N. Fan, S.-W. Zhang, lncrna-mfdl: identification of human long non-coding rnas by fusing multiple features and using deep learning, Molecular BioSystems 11 (3) (2015) 892–897.

[52] R. Achawanantakun, J. Chen, Y. Sun, Y. Zhang, Lncrna-id: Long non-coding rna identification using balanced random forests, Bioinformatics 31 (24) (2015) 3897–3905.

[53] L. Sun, H. Liu, L. Zhang, J. Meng, lncrscan-svm: a tool for predicting long non-coding rnas using support vector machine, PloS one 10 (10) (2015) e0139654.

[54] C. Pian, G. Zhang, Z. Chen, Y. Chen, J. Zhang, T. Yang, L. Zhang, Lncrnapred: Classification of long non-coding rnas and protein-coding transcripts by the ensemble algorithm with a new hybrid feature, PloS one 11 (5) (2016) e0154567.

[55] R. Tripathi, S. Patel, V. Kumari, P. Chakraborty, P. K. Varadwaj, Deeplnc, a long non-coding rna prediction tool using deep neural network, Network Modeling Analysis in Health Informatics and Bioinformatics 5 (1) (2016) 21.

[56] L. M. Vieira, C. Grativol, F. Thiebaut, T. G. Carvalho, P. R. Hardoim, A. Hemerly, S. Lifschitz, P. C. G. Ferreira, M. E. M. Walter, Plantrna_sniffer: a svm-based workflow to predict long intergenic non-coding rnas in plants, Non-coding RNA 3 (1) (2017) 11.

[57] U. Singh, N. Khemka, M. S. Rajkumar, R. Garg, M. Jain, Plncpro for prediction of long non-coding rnas (lncrnas) in plants and its application for discovery of abiotic stress-responsive lncrnas in rice and chickpea, Nucleic acids research 45 (22) (2017) e183–e183.

[58] T. d. C. Negri, W. A. L. Alves, P. H. Bugatti, P. T. M. Saito, D. S. Domingues, A. R. Paschoal, Pattern recognition analysis on long non-coding rnas: a tool for prediction in plants, Briefings in bioinformatics (2018).

[59] E. A. Ito, I. Katahira, F. F. d. R. Vicente, L. F. P. Pereira, F. M. Lopes, Basinet—biological sequences network: a case study on coding and non-coding rnas identification, Nucleic acids research (2018).

[60] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, Gapped blast and psi-blast: a new generation of protein database search programs, Nucleic acids research 25 (17) (1997) 3389–3402.

[61] A. C. Liu, The effect of oversampling and undersampling on classifying imbalanced text datasets, The University of Texas at Austin (2004).

[62] D. M. Goodstein, S. Shu, R. Howson, R. Neupane, R. D. Hayes, J. Fazo, T. Mitros, W. Dirks, U. Hellsten, N. Putnam, et al., Phytozome: a comparative platform for green plant genomics, Nucleic acids research 40 (D1) (2011) D1178–D1186.

[63] A. Paytuví Gallart, A. Hermoso Pulido, I. Anzar Martínez de Lagrán, W. Sanseverino, R. Aiese Cigliano, Greenc: a wiki-based database of plant lncrnas, Nucleic acids research 44 (D1) (2015) D1161–D1166.

[64] D. Chen, C. Yuan, J. Zhang, Z. Zhang, L. Bai, Y. Meng, L.-L. Chen, M. Chen, PlantNATsDB: a comprehensive database of plant natural antisense transcripts, Nucleic Acids Research 40 (D1) (2011) D1187–D1193. arXiv:https://academic.oup.com/nar/article-pdf/40/D1/D1187/9481672/gkr823.pdf, doi:10.1093/nar/gkr823.
URL `https://doi.org/10.1093/nar/gkr823`

[65] Q. Chu, X. Zhang, X. Zhu, C. Liu, L. Mao, C. Ye, Q.-H. Zhu, L. Fan, Plantcircbase: a database for plant circular rnas, Molecular plant 10 (8) (2017) 1126–1128.

[66] X. Pan, K. Xiong, Predcircrna: computational classification of circular rna from other long non-coding rna using hybrid features, Molecular Biosystems 11 (8) (2015) 2219–2226.

[67] C. Yin, Y. Chen, S. S.-T. Yau, A measure of dna sequence similarity by fourier transform with applications on hierarchical clustering, Journal of theoretical biology 359 (2014) 18–28.

[68] C. Yin, S. S.-T. Yau, A fourier characteristic of coding sequences: origins and a non-fourier approximation, Journal of computational biology 12 (9) (2005) 1153–1165.

[69] D. Anastassiou, Genomic signal processing, IEEE signal processing magazine 18 (4) (2001) 8–20.

[70] L. Marsella, F. Sirocco, A. Trovato, F. Seno, S. C. Tosatto, Repetita: detection and discrimination of the periodicity of protein solenoid repeats by discrete fourier transform, Bioinformatics 25 (12) (2009) i289–i295.

[71] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, P. D. Welch, What is the fast fourier transform?, Proceedings of the IEEE 55 (10) (1967) 1664–1674.

[72] M. Abo-Zahhad, S. M. Ahmed, S. A. Abd-Elrahman, Genomic analysis and classification of exon and intron sequences using dna numerical mapping techniques, International Journal of Information Technology and Computer Science 4 (8) (2012) 22–36.

[73] G. Mendizabal-Ruiz, I. Román-Godínez, S. Torres-Ramos, R. A. Salido-Ruiz, J. A. Morales, On dna numerical representations for genomic similarity computation, PloS one 12 (3) (2017) e0173288.

[74] R. F. Voss, Evolution of long-range fractal correlations and 1/f noise in dna base sequences, Physical review letters 68 (25) (1992) 3805.

[75] P. D. Cristea, Conversion of nucleotides sequences into genomic signals, Journal of cellular and molecular medicine 6 (2) (2002) 279–303.

[76] N. Chakravarthy, A. Spanias, L. D. Iasemidis, K. Tsakalis, Autoregressive modeling and feature analysis of dna sequences, EURASIP Journal on Applied Signal Processing 2004 (2004) 13–28.

[77] R. Zhang, C.-T. Zhang, Z curves, an intutive tool for visualizing and analyzing the dna sequences, Journal of Biomolecular Structure and Dynamics 11 (4) (1994) 767–782.

[78] A. S. Nair, S. P. Sreenadhan, A coding measure scheme employing electron-ion interaction pseudopotential (eiip), Bioinformation 1 (6) (2006) 197.

[79] D. Anastassiou, Genomic signal processing, IEEE Signal Processing Magazine 18 (4) (2001) 8–20. doi:10.1109/79.939833.

[80] N. Yu, Z. Li, Z. Yu, Survey on encoding schemes for genomic data representation and feature learning—from signal processing to machine learning, Big Data Mining and Analytics 1 (3) (2018) 191–210.

[81] J. Shao, X. Yan, S. Shao, Snr of dna sequences mapped by general affine transformations of the indicator sequences, Journal of mathematical biology 67 (2) (2013) 433–451.

[82] C.-T. Zhang, A symmetrical theory of dna sequences and its applications, Journal of theoretical biology 187 (3) (1997) 297–306.

[83] C. Yin, S. S.-T. Yau, Prediction of protein coding regions by the 3-base periodicity analysis of a dna sequence, Journal of theoretical biology 247 (4) (2007) 687–694.

[84] H. Nikookar, Peak-to-average power ratio, in: Wavelet Radio: Adaptive and Reconfigurable Wireless Systems Based on Wavelets, Cambridge University Press, 2013, pp. 93–111. doi:10.1017/CBO9781139084697.006.

[85] I. Pritišanac, R. M. Vernon, A. M. Moses, J. D. Forman Kay, Entropy and information within intrinsically disordered protein regions, Entropy 21 (7) (2019) 662.

[86] S. Vinga, Information theory applications for biological sequence analysis, Briefings in bioinformatics 15 (3) (2013) 376–389.

[87] J. T. Machado, A. C. Costa, M. D. Quelhas, Shannon, rényie and tsallis entropy analysis of dna using phase plane, Nonlinear Analysis: Real World Applications 12 (6) (2011) 3135–3144.

[88] A. Lesne, Shannon entropy: a rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics, Mathematical Structures in Computer Science 24 (3) (2014).

[89] M. P. De Albuquerque, I. A. Esquef, A. G. Mello, Image thresholding using tsallis entropy, Pattern Recognition Letters 25 (9) (2004) 1059–1065.

[90] F. M. Lopes, E. A. de Oliveira, R. M. Cesar, Inference of gene regulatory networks from time series by tsallis entropy, BMC systems biology 5 (1) (2011) 61.

[91] A. Ramírez-Reyes, A. R. Hernández-Montoya, G. Herrera-Corral, I. Domínguez-Jiménez, Determining the entropic index q of tsallis entropy in images through redundancy, Entropy 18 (8) (2016) 299.

[92] L. d. F. Costa, F. A. Rodrigues, A. S. Cristino, Complex networks: the key to systems biology, Genetics and Molecular Biology 31 (3) (2008) 591–601.

[93] X. F. Wang, Complex networks: topology, dynamics and synchronization, International journal of bifurcation and chaos 12 (05) (2002) 885–916.

[94] B. K. Singh, K. Verma, A. Thoke, Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification, International Journal of Computer Applications 116 (19) (2015).

[95] M. C. de Souto, D. S. de Araujo, I. G. Costa, R. G. Soares, T. B. Ludermir, A. Schliep, Comparative study on normalization procedures for cluster analysis of gene expression datasets, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 2792–2798.

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

[97] T. Hastie, S. Rosset, J. Zhu, H. Zou, Multi-class adaboost, Statistics and its Interface 2 (3) (2009) 349–360.

[98] A. V. Dorogush, V. Ershov, A. Gulin, Catboost: gradient boosting with categorical features support, arXiv preprint arXiv:1810.11363 (2018).

[99] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1) (1960) 37–46.