Robson Parmezan Bonidia

Feature Extraction and Selection Analysis in Biological Sequences: A Case Study with Metaheuristic and Mathematical Models

Cornélio Procópio, Paraná, Brazil

2020

Robson Parmezan Bonidia

Feature Extraction and Selection Analysis in Biological Sequences: A Case Study with Metaheuristic and Mathematical Models

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Bioinformática (PPGBIOINFO) da Universidade Tecnológica Federal do Paraná - Câmpus Cornélio Procópio, como parte dos requisitos necessários para a obtenção do grau de Mestre em Bioinformática.

Universidade Tecnológica Federal do Paraná – UTFPR

Câmpus Cornélio Procópio

Programa de Pós-Graduação em Bioinformática - PPGBIOINFO

Supervisor: Dr. Danilo Sipoli Sanches

Co-supervisor: Dr. Alexandre Rossi Paschoal

Cornélio Procópio, Paraná, Brazil

2020

Dados Internacionais de Catalogação na Publicação

B715 Bonidia, Robson Parmezan

Feature extraction and selection analysis in biological sequences : a case study with metaheuristic and mathematical models / Robson Parmezan Bonidia. – 2020.
104 f. : il. color. ; 31 cm.

Orientador: Danilo Sipoli Sanches.
Coorientador: Alexandre Rossi Paschoal.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. Cornélio Procópio, 2020.
Bibliografia: p. 91-104.

1. Machine Learning. 2. Mathematical models. 3. Heuristic programming. 4. Bioinformática – Dissertações. I. Sanches, Danilo Sipoli, orient. II. Paschoal, Alexandre Rossi, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. IV. Título.

CDD (22. ed.) 572.80285

Biblioteca da UTFPR - Câmpus Cornélio Procópio
Bibliotecário/Documentalista responsável:
Romeu Righetti de Araujo – CRB-9/1676

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Programa de Pós-Graduação em Bioinformática

Título da Dissertação Nº 13:

“Feature Extraction and Selection Analysis in Biological Sequences: A Case Study with Metaheuristic and Mathematical Models”.

por

Robson Parmezan Bonidia

Orientador: Prof. Dr. Danilo Sipoli Sanches
Co-orientador: Prof. Dr. Alexandre Rossi Paschoal

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM BIOINFORMÁTICA – Linha de Pesquisa: Biologia Computacional E Sistêmica, pelo Programa de Pós-Graduação em Bioinformática – PPGBIOINFO – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 14h00min do dia 12 de fevereiro de 2020. O trabalho foi __________ pela Banca Examinadora, composta pelos professores:

Prof. Dr. Danilo Sipoli Sanches
(Presidente)

Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho
(USP-SP)
Participação à distância via

Prof. Dr. Fabrício Martins Lopes
(UTFPR-CP)

Profª. Dra. Priscila Tiemi Maeda Saito
(UTFPR-CP)

Visto da coordenação:

André Yoshiaki Kashiwabara
Coordenador do Programa de Pós-Graduação em Bioinformática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

*This work is dedicated to adult children who,
when small, dreamed of becoming scientists.*

Acknowledgements

This dissertation would not have been possible without the guidance and help of several people who, in one way or another, contributed and extended their valuable assistance in the preparation and conclusion of this study.

Firstly, I would like to thank God, the most excellent teacher that anyone can know, for giving me ideas, health, and strength in difficult times to overcome and face difficulties. I also want to especially thank my supervisor, Dr. Danilo Sipoli Sanches, for his guidance during my dissertation. His perpetual energy and enthusiasm motivated me to give my best.

To professors Dr. Fabrício Martins Lopes, Dr. Alexandre Rossi Paschoal, Dr. Douglas Silva Domingues, and Dr. André Yoshiaki Kashiwabara, for the valuable ideas and advice during the execution of my work. My friends, for the encouragement, companionship, and understanding. I also thank all the professors who accompanied us and passed their knowledge and experiences.

My sincere thanks to my family, for their continued encouragement and support during my studies, especially for the four most influential people in my life, my fiancée, Amanda, my parents, Ivone, and Osmar, and my grandmother, Olga.

Last, but not least significant, I thank the Federal University of Technology - Paraná (UTFPR) (Grant: April/2018) and CAPES (Grant: April/2019) for supporting this project.

"Do not be conformed to this world,
but be transformed by the renewing of your mind.
Then you will be able to discern what is the good,
pleasing, and perfect will of God."
(Holy Bible, Romans 12, 2)

Abstract

The number of available biological sequences has increased in large amounts in past years, due to various genomic sequencing projects, creating a huge volume of data. Consequently, new computational methods are needed for the analysis and information extraction from these sequences. Machine learning methods have shown broad applicability in computational biology and bioinformatics. The application of machine learning methods has helped to extract relevant information from various biological datasets. However, there are still several challenging problems that motivate new algorithms and pipeline proposals. Therefore, this work proposes a generic machine learning pipeline for biological sequence analysis, following two main steps: (1) feature extraction and (2) feature selection. Essentially, we focus our work on the study of dimensionality reduction and feature extraction techniques, using metaheuristics and mathematical models. As a case study, we analyze Long Non-Coding RNA sequences. Moreover, we divided this dissertation into two parts, e.g., Experimental Test I (feature selection) and Experimental Test II (feature extraction). The experimental results indicated four main contributions: (1) A pipeline with five distinct metaheuristics, using a voting scheme and execution rounds, to the feature selection problem in biological sequences; (2) The metaheuristic efficiency, providing competitive classification performance; (3) A feature extraction pipeline using nine mathematical models and (4) its generalization and robustness for distinct biological sequence classification.

Keywords: Machine Learning; Feature Extraction; Feature Selection; Biological Sequences; Mathematical Models; Metaheuristics; Bioinformatics.

Resumo

O número de sequências biológicas disponíveis aumentou em grandes quantidades nos últimos anos, devido a vários projetos de sequenciamento genômico, criando um alto volume de dados. Consequentemente, novos métodos computacionais são necessários para a análise e extração de informações a partir dessas sequências. Métodos de aprendizado de máquina têm apresentado ampla aplicabilidade em biologia computacional e bioinformática. A aplicação desses métodos tem ajudado a extrair informações relevantes de vários conjuntos de dados biológicos. No entanto, ainda existem vários problemas desafiadores que motivam novas propostas de algoritmos e pipelines. Portanto, este trabalho propõe um pipeline genérico de aprendizado de máquina para análise de sequência biológica, seguindo duas etapas principais: (1) extração e (2) seleção de características. Essencialmente, concentramos nosso trabalho no estudo de técnicas de redução de dimensionalidade e extração de recursos, usando metaheurísticas e modelos matemáticos. Como estudo de caso, analisamos sequências de RNAs longos não codificantes. Além disso, dividimos esta dissertação em duas partes: Teste Experimental I (seleção de características) e Teste Experimental II (extração de características). Os resultados experimentais indicam quatro contribuições principais: (1) Um pipeline com 5 metaheurísticas diferentes, usando um esquema de votação e rodadas de execução, ao problema de seleção de características em sequências biológicas; (2) A eficiência metaheurística, proporcionando desempenho de classificação competitiva; (3) Um pipeline de extração de recursos usando 9 modelos matemáticos e (4) sua generalização e robustez para classificação de sequências biológicas distintas.

Palavras-chave: Aprendizado de Máquina; Extração de Características; Seleção de Características; Sequências Biológicas; Modelos Matemáticos; Metaheurísticas; Bioinformática.

List of Figures

List of Tables

List of abbreviations and acronyms

A	Adenine
ABC	Artificial Bee Colony
ACC	Accuracy
ACO	Ant Colony Optimization
ANN	Artificial Neural Networks
ANT	Adjoining Nucleotide Triplets
AUC	Area Under the Curve
BASiNET	BiologicAl Sequences NETwork
BLAST	Basic Local Alignment Search Tool
C	Cytosine
CFS	Correlation-Based Feature Selection
circRNA	circular RNAs
CNCI	Coding-Non-Coding Index
CPAT	Coding-Potential Assessment Tool
CPC	Coding Potential Calculator
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Network
DNR	DNA Numerical Representation
EA	Evolutionary Algorithm
ELM	Extreme Learning Machine
EN	Elastic Net
FFT	Fast Fourier Transform

FN	False Negative
FP	False Positive
FS	Feature Selection
FSBE	Forward Selection and Backward elimination
FSC	Feature Score Criterion
G	Guanine
GA	Genetic Algorithm
GC	guanine-cytosine
GFS	Greedy Forward Selection
GR	Gain Ratio
GSP	Genomic Signal Processing
HMM	Hidden Markov Mode
IG	Information Gain
lincRNA	Long Intergenic Non-Coding RNA
LR	Logistic Regression
LR	Lasso Regression
lncRNA	Long Non-Coding RNA
MCC	Matthews Correlation Coefficient
MFE	Minimum free energy
ML	Machine Learning
MLCDS	most-like CDS
mRMR	Minimal-Redundancy-Maximal-Relevance
mRNA	messenger RNA / protein-coding genes
ncRNA	Non-Coding RNA
NPV	Negative Predictive Value
ORF	Open Reading Frame

PPV	Positive Predictive Value
PSO	Particle Swarm Optimization
RF	Random Forest
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
SE	Sensitivity
SGB	Stochastic Gradient Boosting
SL	Sequence length
SNR	Signal to Noise Ratio
sncRNA	Small non-coding RNAs
SPC	Specificity
SVM	Support Vector Machine
T	Thymine
TN	True Negative
TP	True Positive
U	Uracil
WEKA	Waikato Environment for Knowledge Analysis
WT	Wilcox test

Contents

1 INTRODUCTION

Biology, in particular molecular biology, is undergoing several transformations, in which there is a growing awareness that computational and statistical models can be used to great benefit (GENTLEMAN et al., 2004). The development of high-throughput data acquisition technologies in the biological sciences has transformed biology into a data-rich science (INZA et al., 2010). In recent years, due to advances in DNA sequencing, increasing numbers of biological sequence data have been generated by thousands of sequencing projects (LOU et al., 2019), creating vast volumes of data (BONIDIA et al., 2019). Consequently, the ability to process and analyze biological data has advanced significantly (CAO et al., 2018). The development of methods to analyze this large amount of information is one of the main challenges of bioinformatics (INZA et al., 2010), in which it involves many problems that can become Machine Learning (ML) tasks.

According to Min (2010), during the last decade, ML methods have shown broad applicability in computational biology and bioinformatics. Tools have been widely applied in gene networks, protein structure prediction, genomics, proteomics, protein-coding genes detection, disease diagnosis, and drug planning (DINIZ; CANDURI, 2017; Parmezan Bonidia et al., 2019). Fundamentally, ML investigates how computers can learn (or improve their performance) based on the data. Moreover, it is a specialization of computer science related to pattern recognition and artificial intelligence (JURTZ et al., 2017).

Based on this, several works have focused on the investigating of DNA, RNA, and protein sequences. The application of ML methods in biological sequences has helped to extract important information from various datasets to explain biological phenomena. However, according to Min (2010); Xu and Jackson (2019), there are still several challenging biological problems that motivated the emergence of proposals for new algorithms. Fundamentally, biological sequence analysis with ML presents 2 major problems: (1) Feature Extraction (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015), (2) Feature Selection (SAEYS; INZA; LARRAÑAGA, 2007; WANG; WANG; CHANG, 2016).

Necessarily, as previously mentioned, several applications in bioinformatics apply ML algorithms to sequence data, and as many ML algorithms can deal only with numerical data, sequences need to be translated into sequences of numbers. Thereby, early applications transformed each letter in the sequence to binary vector (Kwan; Arniker, 2009). These transformations resulted in a very long sequence of sparse data, i.e., data is considered sparse when specific values in a dataset are missing (ZITNIK et al., 2019). This difficulty grows as the size of the sequences grows.

A more straightforward approach, adopted by most current ML applications (KONG et al., 2007; KANG et al., 2017; LI; ZHANG; ZHOU, 2014; NEGRI et al., 2018), extracts relevant features from sequences. These features are based on several properties, e.g., physicochemical, ORF-based, usage frequency of adjoining nucleotide triplets, and sequence-based. This approach is common in biological problems, but usually, 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 (BAEK et al., 2018), but not useful features for classifying lncRNA classes (BONIDIA et al., 2019). Consequently, the feature extraction problem arises, in which extracting a set of useful features that contain significant discriminatory information becomes a fundamental step in building a predictive model (MUHAMMOD et al., 2019).

Also, in the feature extraction, maintaining features with meaningful information and avoiding scarcity is extremely important (MUHAMMOD et al., 2019). Thereby, this domain has accumulated a diverse set of terminology, as feature selection (our second problem) (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015). Fundamentally, a high ratio between the number of predictive features and the number of instances is behind the curse of dimensionality problem. A term initially coined by Bellman (BELLMAN, 1961) when considering dynamic optimization problems. This term comes from the fact that the instances become so similar that it becomes challenging to induce models with high predictive accuracy. Thus, when the number of features increases substantially, the predictive performance of the models induced by ML algorithms decreases.

An alternative to deal with this problem is to reduce the number of features, removing redundant and irrelevant attributes (DY; BRODLEY, 2004), e.g., feature selection techniques. According to Xue et al. (2016), the feature selection is an essential task in ML since it reduces the feature extraction and model induction computational costs. Additionally, it can increase predictive performance. That is, as the extraction of each feature also has an economical cost, it can reduce the financial cost of an ML-based predictive tool, making it available to a more significant number of users.

Therefore, all these problems make the process of biological sequence analysis a challenging task. As a consequence, there is a growing need to develop new methods for analyzing sequences efficiently. Mainly, structures that address joint solutions. Thereby, this dissertation proposes to analyze feature selection and extraction methods for biological sequence analysis, addressing two critical phases of building a predictive model (feature extraction and selection), using metaheuristics and mathematical models. As a case study, we use lncRNA sequences, which are fundamentally unable to produce proteins (ABBAS et al., 2016) and, recently, have been remitting several doubts about its functionality (AMIN; MCGRATH; CHEN, 2019). These sequences present several classification challenges that will help in all experiments of this dissertation.

Finally, it is important to emphasize that we divided this work into two parts:

Experimental Test I (see Chapter 3) and Experimental Test II (see Chapter 4). Thus, in our first experiment, we applied metaheuristic models in the most used features for lncRNAs classification, to verify the efficiency of the proposed pipeline. Based on this, in the second experiment, we proposed feature extraction strategies with mathematical models.

1.1 Biological Sequence

Biological sequences usually refer to nucleotides or amino acid sequences, such as DNA, RNA, or protein (HAN; KAMBER; PEI, 2012). The DNA (deoxyribonucleic acid) is formed from the union of chemical compounds called nucleotides (ZAHA; FERREIRA; PASSAGLIA, 2014). Each nucleotide is composed of three substances: a nitrogenous base, a phosphate group, a five-carbon sugar (deoxyribose). The nitrogenous bases may be purines (Adenine (A) and Guanine (G)) and pyrimidines (Cytosine (C) and Thymine (T)) (ZAHA; FERREIRA; PASSAGLIA, 2014). In contrast, an RNA molecule (ribonucleic acid) is a polymer of ribonucleotide units, and its structure is identical to the single-stranded DNA, but with two central differences. The five-carbon sugar is ribose, and the nucleobase Uracil (U) is used in RNA instead of the nucleobase T (ABU-JAMOUS; FA; NANDI, 2015).

Different types of RNA are present in cells and have specific functions. Finally, proteins are long linear polymers of amino acids (in the case of DNA and RNA, the links of the chain are nucleotides) that are linked with peptide bonds and belong to the family of molecules known as polypeptides (ABU-JAMOUS; FA; NANDI, 2015). While DNA and RNA are composed of four different nucleotides, the proteins have a repository of 20 common amino acids (ALLISON, 2007).

1.2 Long Non-Coding RNAs

Fundamentally, ncRNAs are unable to produce proteins. However, these ncRNAs contain unique information that produces other functional RNA molecules (EDDY, 2001; ABBAS et al., 2016). Moreover, they demonstrate essential roles in cellular mechanisms, playing regulatory roles in a wide variety of biological reactions and processes (EDDY, 2001). The ncRNAs can be classified by length into two classes: Long Non-Coding RNA (lncRNA - 200 nt or more) and short ncRNA (less than 200 nucleotides) (KAPRANOV et al., 2007; ZHANG; TAO; LIAO, 2017). The lncRNAs are a new type of Non-Coding RNA (ncRNA) with a length greater than 200 nucleotides (LI et al., 2016), and according to recent studies, play essential roles in several critical biological processes (WANG et al., 2014; WANG et al., 2018; ZHANG et al., 2018), including transcriptional regulation (ZHOU et al., 2016), epigenetics (HASSAN et al., 2015), cellular differentiation (CIAUDO et al., 2009), and immune response (PENG et al., 2010). Also are correlated with some complex human diseases, such as cancer and neurodegenerative diseases (PASTORI; WAHLESTEDT, 2012; ZHANG et al., 2017).

In plants, according to Wang and Chekanova (2017), the lncRNAs act in gene silencing, flowering time control, organogenesis in roots, photomorphogenesis in seedlings, stress responses (DI et al., 2014; WANG et al., 2017), and reproduction (ZHANG et al., 2014). Further, lncRNAs are present in large numbers in genome (FANG; FULLWOOD, 2016) and have similar sequence characteristics with protein-coding genes, such as 5' cap, alternative splicing, two or more exons (DERRIEN et al., 2012), and polyA+ tails (CHENG et al., 2005). They are also observed in almost all living beings, not only in animals and plants but also yeasts, prokaryotes, and even viruses (MA; BAJIC; ZHANG, 2013; HU; SUN, 2016). According to Fang and Fullwood (2016), lncRNAs do not possess functional Open Reading Frames (ORFs). However, recent studies have found bifunctional RNAs (CHOONIEDASS-KOTHARI et al., 2004), 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 according to the genomic location, that is, where they are transcribed, concerning well-established markers, like protein-coding genes. Among the categories are (HE et al., 2014; DERRIEN et al., 2012):

- (a) sense: overlapping one or more exons of another transcript on the same strand;
- (b) antisense: overlapping one or more exons of another transcript on the opposite strand;
- (c) bidirectional: when lncRNA and a coding transcript are expressed together but are in opposite strands;
- (d) intronic: when derived from an intron of a second transcript;
- (e) intergenic: when it is between two genes.

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 (KUNG; COLOGNORI; LEE, 2013).

1.3 Problem Definition and Proposal

This section describes problems addressed in this dissertation in two subsections: (1) Curses of Dimensionality and The Feature Selection Problem and (2) Feature Extraction Problem. Moreover, we will briefly describe our proposals that will be presented with greater emphasis in Chapter 3 and Chapter 4.

1.3.1 Curse of Dimensionality and The Feature Selection Problem

ML algorithms and pattern recognition are subject to the so-called curse of dimensionality, which refers to analyzing and organizing data in high dimensional spaces. This problem causes performance loss in classification methods when the number of features increases substantially. Fundamentally, dimensionality presents several obstacles to the efficiency of most ML algorithms (ROKACH; MAIMON, 2015). Thereby, the high dimensionality of the input features expands the size of the search space exponentially.

Hence, the fundamental motivation for feature selection is the curse of dimensionality (DY; BRODLEY, 2004). The feature selection is a concrete way to deal with this problem. Therefore, the feature selection problem is defined as follows according to Jain and Zongker (1997): "given a set of candidate features, select a subset that performs the best under some classification system." In other words, this method is used to find an “optimal” subset of relevant features, so that accuracy increased as the attributes are reduced. Nevertheless, it is necessary to define what would be a relevant feature. In this line, the literature presents several definitions; e.g., (KOHAVI; JOHN, 1997) proposed two degrees of relevance (strong and weak) for features, being the relevance defined in absolute terms with an ideal Bayes classifier. The definitions below were described by Rudnicki, Wrzesień and Paja (2015).

- **Definition 1:** "A feature X is *strongly relevant* when removal of X alone from the data always results in deterioration of the prediction accuracy of the ideal Bayes classifier."
- **Definition 2:** "A feature X is *weakly relevant* if it is not strongly relevant and there exists a subset of features S, such that the performance of ideal Bayes classifier on S is worse than the performance on $S \cup \{X\}$."
- **Definition 3:** "A feature X is irrelevant if it is neither strongly nor weakly relevant."

Blum and Langley (1997) defined "relevant with respect to an objective." John, Kohavi and Pfleger (1994) include two notions of relevance, being: "strong relevance with respect to sample" and "strong relevance with respect to the distribution." Despite this, some authors concentrated on using relevance as a complexity measure relative to the goal. For example, relevance as a complexity measure (BLUM; LANGLEY, 1997), incremental usefulness (CARUANA; FREITAG, 1994), entropic relevance (WANG; WANG; CHANG, 2016). Nevertheless, in this work, we apply terms of Nilsson et al. (2007), which motivated by applications within bioinformatics, established the concepts of weak and strong features formally in two problems.

- **All-relevant problem:** Which consists of finding all features that bring information, regardless if the same information is contained in multiple inputs.

- **Minimal optimal problem:** Which consists of finding the smallest subset of input variables that contains all information.

Therefore, our approach focus on *minimal optimal problem*, whose purpose is to find the smallest subset of features that contains all information. Moreover, we proposed a pipeline-based approach using metaheuristics (See Chapter 3). We choose metaheuristics because they use different mechanisms to explore the search space and different strategies to deal with exploration vs. exploitation in its search for an optimal solution.

1.3.2 Feature Extraction Problem

The feature extraction seeks to generate a feature vector, optimally transforming the input data (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015). This procedure is exceptionally relevant to the success of the ML application. Another primary goal of feature extraction is to extract important features from input data compactly, as well as removing noise and redundancy to increase the accuracy of machine learning models (GUYON et al., 2008; STORCHEUS; ROSTAMIZADEH; KUMAR, 2015). Furthermore, the feature extraction is an inevitable method, especially in the stage of biological sequences preprocessing (SAIDI et al., 2012). Considering this, we propose to work with mathematical models for feature extraction (e.g., Fourier, Numerical Mappings, and Entropy - See Chapter 4), based on the excellent results presented by Machado, Costa and Quelhas (2011), Ito et al. (2018), Bonidia et al. (2019). These mathematical approaches have an advantage in terms of generalization to distinct biological sequence classification problems.

1.4 Research Questions and Hypotheses

As emphasized in the introduction, we use lncRNA sequences as a case study. Thereby, we developed an in-depth study of the lncRNAs classification methods/techniques, in which we will analyze seven questions (these questions will be answered in Chapter 2):

- **Question 1 (Q1):** What is the current research landscape?
- **Question 2 (Q2):** What are the most commonly used classification algorithms?
- **Question 3 (Q3):** What are the feature extraction methods?
- **Question 4 (Q4):** What databases are used by the works?
- **Question 5 (Q5):** Which articles use feature selection techniques?
- **Question 6 (Q6):** What are the feature selection algorithms?
- **Question 7 (Q7):** What are the evaluation metrics for predictive models?

Therefore, considering all previously discussed problems and our research questions, we assume the following hypotheses for the feature selection (PB1) and extraction (PB2) problems in biological sequences (these hypotheses will be answered in Chapter 5):

- **Hypothesis 1 (H1-PB1):** Can metaheuristics select a subset of predictive features able to improve the predictive performance of a classification model, when compared with the use of all original features, in biological sequence classification?
- **Hypothesis 2 (H2-PB1):** Are metaheuristic models more efficient than non-heuristic models for biological sequence classification?
- **Hypothesis 3 (H3-PB2):** Are mathematical models efficient for feature extraction from biological sequences?
- **Hypothesis 4 (H4-PB2):** Do mathematical models present competitive classification performance in distinct biological sequence analysis problems?
- **Hypothesis 5 (H5-PB2):** Are mathematical models more generalist than biological models in sequences classification?

1.5 Objectives

Considering our hypotheses, the general objective of this dissertation is to analyze features selection and extraction methods, to generate a generic ML pipeline for biological sequence analysis. Specifically, we concentrated our work on the study of dimensionality reduction and feature extraction techniques, using transcribed sequences of Long Non-Coding RNAs. Thus, as specific objectives, can be highlighted:

- To conduct a systematic literature review in the field of features extraction, selection, and classification in lncRNA;
- To study, analyze and apply metaheuristic optimization methods for the feature selection problem;
- To evaluate features extraction strategies, using mathematical models, for biological sequences classification;
- To apply samples selection strategies for classification;
- Validation and results analysis;
- To report and discuss the computational contribution;

- To publish results through articles (conference and journal) and writing of this dissertation.

We chose lncRNA sequences, because it is a new and relevant problem in the literature, in which, recently, it has presented several works, mainly with ML. Moreover, we will focus only on plant sequences, because it is the least addressed field by the works (see Chapter 2), consequently presenting more challenges. Finally, we chose metaheuristics and mathematical models, because our review reported several studies with biological bias features and few metaheuristic approaches, as can be seen in Chapter 2.

1.6 Justification

As previously mentioned, the ability to process and analyze biological data has advanced significantly in bioinformatics (CAO et al., 2018). However, data continues to grow not only in terms of the abundance of patterns but also of the dimensionality of attributes (features) (WANG; WANG; CHANG, 2016). Considering this, developing or finding an appropriate approach to effectively represent a biological sequence becomes one of the most challenging tasks (LIU et al., 2014). For this reason, recently, some ML pipelines have been developed to help in the biological sequence analysis (LIU, 2017), such as: PseKNC (CHEN et al., 2014a), PseKNC-General (CHEN et al., 2014b), repDNA (LIU et al., 2014), Pse-in-One (LIU et al., 2015), BioSeq-Analysis (LIU, 2017), pysster (BUDACH; MARSICO, 2018), FeatureSelect (MASOUDI-SOBHANZADEH; MOTIEGHADER; MASOUDI-NEJAD, 2019), Seq2Feature (NIKAM; GROMIHA, 2019) and PyFeat (MUHAMMOD et al., 2019). Based on these tools, we can generate five considerations (C).

- **C1:** *PseKNC, PseKNC-General, Pse-in-One, repDNA* and *PyFeat* are specific toolkit to generate features of DNA/RNA sequences.
- **C2:** *Pysster* generates predictive models with convolutional neural network, in which sequences are classified by learning sequence and motifs.
- **C3:** *FeatureSelect* is an application for feature selection based on wrapper approaches.
- **C4:** *Seq2Feature* is a comprehensive web-based feature extraction tool for protein and DNA sequences.
- **C5:** *BioSeq-Analysis* is a platform for biological sequences analysis which can do feature extraction, predictor construction and performance evaluation.

Therefore, we can say that our proposal is innovative when compared to available ML pipelines in the literature for the following reasons (R):

- **R1:** Some tools have been proposed to build predictors of biological sequence analysis (*BioSeq-Analysis* and *Pysster*), but most focus only on individual steps, while our approach addresses two key steps (Feature Extraction and Feature Selection).
- **R2:** The feature extraction tools have a bias in biological features, while our approach proposes to work with mathematical models (e.g., Fourier, Entropy, and Complex Network).
- **R3:** The only pipeline that uses metaheuristics is *FeatureSelect*. However, it focuses only on this individual step using wrapper approaches. In contrast, we develop filter approaches.
- **R4:** Although *BioSeq-Analysis* contemplates all steps proposed in this dissertation; it does not use metaheuristics and mathematical models.

1.7 Dissertation Organization/Outline

- Chapter 2, which follows this introduction, covers a literature review of related works to lncRNA classification. This chapter is designed to answer questions raised in Section 1.4.
- Chapter 3 describes methodological procedures used to achieve the proposed objectives in our first problem (feature selection).
- Chapter 4 analyzes mathematical models for feature extraction, to propose efficient and generalist techniques for biological sequence analysis problems.
- Chapter 5 discusses our findings in terms of whether they support our hypotheses.
- Finally, Chapter 6 presents the conclusions of this dissertation and discusses the final considerations and suggestions for future studies.

2 RELATED WORK

This chapter covers a literature review of related works to lncRNA classification, as well as algorithms and methods applied. Moreover, this chapter is designed to answer questions raised in Section 1.4. Thereby, our search string was run on five electronic databases (IEEE Xplore Digital Library, ACM Digital Library, Science Direct, Semantic Scholar, and PubMed). We considered published papers in journals in the English language. Nevertheless, beyond our review, we used as reference Han et al. (2016), Han et al. (2018), and Antonov et al. (2018), who published surveys on the subject in question.

2.1 The Landscape of lncRNA Classification (Q1, Q2)

For a better understanding, we divide this section into six parts: General Application Methods (Section 2.1.1), Specific Methods for Plants (Section 2.1.2), Feature Extraction (Section 2.1.3), lncRNA Databases (Section 2.1.4), Feature Selection of lncRNA (Section 2.1.5), and Evaluation Metrics (Section 2.1.6).

2.1.1 General Application Methods

In this section, our review selected methods that were applied in more than one species or specific to animal and human systems, as shown in Table 1. Furthermore, we report each algorithm used in Table 2. Thus, the methods reviewed were: CPC (KONG et al., 2007), CPAT (WANG et al., 2013), CNCI (SUN et al., 2013), PLEK (LI; ZHANG; ZHOU, 2014), lncRNA-MFDL (FAN; ZHANG, 2015), LncRNA-ID (ACHAWANANTAKUN et al., 2015), lncRScan-SVM (SUN et al., 2015), LncRNApred (PIAN et al., 2016), DeepLNC (TRIPATHI et al., 2016), BASiNET (ITO et al., 2018), and LncFinder (HAN et al., 2018).

The method CPC (Coding Potential Calculator - 2007) evaluates the protein-coding potential of a transcript based on two features categories. The extent and quality of the Open Reading Frame (ORF), and derivation of BLASTX search. As a prediction method, the authors used the LIBSVM package to train a Support Vector Machine (SVM) model, using the standard radial basis function kernel (RBF kernel) (KONG et al., 2007). CPAT (Coding-Potential Assessment Tool - 2013) classifies transcripts of coding and non-coding using Logistic Regression (LR). This model uses four sequence features: ORF coverage, ORF size, hexamer usage bias, and Fickett TESTCODE statistic (WANG et al., 2013). CNCI (Coding-Non-Coding Index - 2013) was modeled with SVM and used profiling Adjoining Nucleotide Triplets (ANT - 64*64) and most-like CDS (MLCDS) (SUN et al., 2013).

Table 1 – Methods that were applied in more than one species or specific to animal and human systems.

Method	Published year	Species	Prediction
CPC	2007	Multi-Species	ncRNA
CPAT	2013	Human; mouse; fly; zebrafish	lncRNA
CNCI	2013	Animals; plants	lncRNA
PLEK	2014	Multi-Species	lncRNA
lncRNA-MFDL	2015	Human	lncRNA
LncRNA-ID	2015	Human; mouse	lncRNA
lncRScan-SVM	2015	Human; mouse	lncRNA
LncRNApred	2016	Human	lncRNA
DeepLNC	2016	Human	lncRNA
BASiNET	2018	Multi-Species	lncRNA; sncRNA
LncFinder	2018	Multi-Species	lncRNA

Source – Elaborated by the author.

Table 2 – Query file format and prediction algorithm.

Method	Query file format	Prediction Algorithm
CPC	FASTA	SVM
CPAT	BED; FASTA	LR
CNCI	FASTA; GTF	SVM
PLEK	FASTA	SVM
lncRNA-MFDL	-	DL
LncRNA-ID	BED; FSATA	RF
lncRScan-SVM	GTF	SVM
LncRNApred	FASTA	RF
DeepLNC	FASTA	DNN
BASiNET	FASTA	J48; RF
LncFinder	FASTA	LR; SVM; RF; ELM; DL

Source – Elaborated by the author.

In contrast, PLEK (2014) is based on the k-mer scheme ($k = 1, 2, 3, 4, 5$) to predict lncRNA, also applying the SVM classifier (LI; ZHANG; ZHOU, 2014). lncRNA-MFDL (2015) uses Deep Learning (DL) and multiple features, among them: ORF, K-mer ($k = 1, 2, 3$), secondary structure (minimum free energy), and MLCDS (FAN; ZHANG, 2015). LncRNA-ID (2015) predicts lncRNAs with Random Forest (RF) through of ORF (length and coverage), sequence structure (Kozak motif), ribosome interaction, alignment (profile Hidden Markov Mode - profile HMM), and protein conservation (ACHAWANANTAKUN et al., 2015).

lncRScan-SVM (2015) uses stop codon count, GC content, ORF (*txCdsPredict* - score, CDS length and CDS percentage), transcript length, exon count, exon length, and alignment (PhastCons scores) (SUN et al., 2015). LncRNApred (2016) classified lncRNAs with RF and features based on ORF, signal to noise ratio, k-mer ($k = 1, 2, 3$), sequence length, and GC content (PIAN et al., 2016). DeepLNC (2016) uses only the k-mer scheme ($k = 2, 3, 4, 5$) and Deep Neural Network (DNN) (TRIPATHI et al., 2016).

BASiNET (BiologicAl Sequences NETwork - 2018) classify sequences based on the feature extraction from complex network measurements (ITO et al., 2018). Finally, LncFinder (2018) uses five classifiers (LR, SVM, RF, Extreme Learning Machine (ELM) and DL) to apply the algorithm that obtains the highest accuracy. Moreover, the authors use features of ORF, secondary structural, and electron-ion interaction (HAN et al., 2018).

2.1.2 Specific Methods for Plants

This section presents specific works for plants, according to Table 3. The methods reviewed were: PlantRNA_Sniffer (VIEIRA et al., 2017), PLncPRO (SINGH et al., 2017), RNAplonc (NEGRI et al., 2018), and Ensemble (SIMOPOULOS; WERETILNYK; GOLDING, 2018). Essentially, our review revealed there is a lack of specific approaches to predict lncRNAs in plants when compared to the previous section. For example, 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.

Table 3 – Specific methods for plants.

Method	Year	File format	Algorithm
PlantRNA_Sniffer	2017	FASTA	SVM
PLncPRO	2017	FASTA	RF
RNAplonc	2018	FASTA	REPtree
Ensemble	2018	FASTA	SGB; RF

Source – Elaborated by the author.

PLncPRO (2017) is based on machine learning and uses RF. The features include ORF quality (score and coverage), number of hits, significance score, total bit score, and frame entropy. RNAplonc (2018) considered 5468 features (ORF, GC content, K-mer ($k = 1, 2, 3, 4, 5, 6$), sequence length, and minimum free energy), besides classifying sequences with the REPtree algorithm.

Finally, Simopoulos, Weretilnyk and Golding (2018) applied Stochastic Gradient Boosting (SGB) and RF models. Features include mRNA length, ORF (length), GC content, Fickett score, hexamer score, alignment identity in SwissProt database, length of alignment in SwissProt database, the proportion of alignment length and mRNA length, proportion of alignment length and ORF length, transposable element, and sequence percent divergence from a transposable element.

2.1.3 Feature Extraction (Q3)

The feature extraction is one of the most critical steps in the elaboration of a predictor (FAN; ZHANG, 2015). Therefore, we map all extracted features by the methods previously exposed (see Section 2.1.1 and 2.1.2), as shown in Table 4 and Table 5. The features were divided into five groups: ORF, codon, sequence structure, alignment, Ribosome, and Protein. Essentially, the most commonly used group is sequence structure, followed by ORF, Codon, and Alignment. Moreover, for better understanding, we elaborate definitions of the main features, among them:

- **k-mer:** According to Sievers et al. (2017), a k-mer analysis is defined as extraction and counting of each DNA word with length k (k bases along one strand), using a "sliding window" approach to eliminate the arbitrary point influence. Li, Zhang and Zhou (2014) define a k-mer analysis on Equation (2.1).

$$\begin{aligned} &f_i = \frac{c_i}{s_k} w_k, \\ &k = 1,2,3,4,5,6; i = 1,2,...,5460, \\ &s_k = l - k + 1, \\ &w_k = \frac{1}{4^{6-k}}. \end{aligned} \tag{2.1}$$

  Where c_i denotes the number of the segments; i is the number of the patterns; s_k total of the segments along sequence with size of k. Moreover, f_i is the frequency of use multiplied by a factor of w_k, used to facilitate discrimination (LI; ZHANG; ZHOU, 2014; HAN et al., 2016).

- **GC content:** The GC content (guanine-cytosine content) is the percentage of nitrogenous bases in a DNA or RNA molecule that is guanine or cytosine (AMR; FUNKE, 2015), represented by Equation (2.2).

$$\frac{C+G}{A+C+G+T} \tag{2.2}$$

- **Sequence length (SL):** It is the number of amino acids in the sequence. For example, $SL = A + C + G + T$.

- **Minimum free energy (MFE):** MFE is a feature that evaluates the stability of the secondary structure in transcripts (FAN; ZHANG, 2015). An example of software to calculate MFE is the RNAfold (LORENZ et al., 2011).

- **ORF:** The term ORF is of central importance to gene discovery. According to Sieber, Platzer and Schuster (2018), "an ORF is a sequence that has a length divisible by

Table 4 – Summary of the extracted features in each method (ORF, Codon and Sequence structure).

Method	**ORF**	**Codon**	**Sequence structure**
CPC	Quality; Coverage; Integrit	No	No
CPAT	Length; Coverage	No	Fickett score; Hexamer Score
CNCI	No	ANT; Codon-bias	MLCDS
PLEK	No	No	k-mer ($k = 5$)
lncRNA-MFDL	Length; Coverage	No	k-mer ($k = 3$); Minimum free energy; MLCDS
LncRNA-ID	Length; Coverage	No	Kozak motif
lncRScan-SVM	Score, CDS length; CDS percentage	Distribution of stop codon	GC content; Transcript length; Exon count; Exon length
LncRNApred	Length; Coverage	No	Signal to noise ratio; k-mer ($k = 3$); GC content; Sequence length
DeepLNC	No	No	k-mer ($k = 2, 3, 4, 5$)
BASiNET	No	No	Nucleotide pattern (3)
LncFinder	ORF	-	DFT + EIIP, Minimum free energy, secondary structure
PlantRNA-Sniffer	Proportion; Length	No	Nucleotide pattern (10)
PLncPRO	Score; Coverage	No	Number of hits;
RNAplonc	Score, CDS Sizes; CDS starts, CDS stop, CDS percent	No	k-mer ($k = 6$); GC content; Sequence length; Minimum free energy
Ensemble	Length	No	mRNA length; GC content; Hexamer Score; Fickett score

Source – Elaborated by the author.

Table 5 – Summary of the extracted features in each method (Alignment, Ribosome and Protein).

Method	Alignment	Ribosome	Protein
CPC	BLASTX	No	No
CPAT	No	No	No
CNCI	No	No	No
PLEK	No	No	No
lncRNA-MFDL	No	No	No
LncRNA-ID	Profile HMM	Ribosome interaction	Protein conservation
lncRScan-SVM	PhastCons scores	No	No
LncRNApred	No	No	No
DeepLNC	No	No	No
BASiNET	No	No	No
LncFinder	-	-	-
PlantRNA-Sniffer	No	No	No
PLncPRO	BLASTX	No	No
RNAplonc	No	No	No
Ensemble	BLASTX	No	No

Source – Elaborated by the author.

three and begins with a translation start codon (ATG) and ends at a stop codon." The literature presents several tools for prediction of ORF, among them: txCdsPredict, ORFfinder, and OrfPredictor.

- **MLCDS:** The authors Sun et al. (2013) and Fan and Zhang (2015) used this feature to discover the MLCDS region of a transcript. They applied the sliding window method to analyze each transcript. The technique was applied six times, generating six reading frames. The purpose was to find sub-sequences with greater coding capacity.
- **Codon-bias:** This feature evaluates the coding-non-coding bias for each of the 61 codon types (discarding three stop codons) (SUN et al., 2013).

- **Hexamer Score:** This metric determines the hexamer usage bias in a specific sequence. Essentially, positive values report a coding sequence and negative values a non-coding sequence (WANG et al., 2013). Equation (2.3) shows the calculation used by Wang et al. (2013):

$$HexamerScore = \frac{1}{m}\sum_{i=1}^{m} log\left(\frac{F(H_i)}{F'(H_i)}\right) \tag{2.3}$$

Where the probability of a sequence is calculated, and then the logarithm of the ratio of these probabilities is used as the potential coding score. The authors used $F(h_i)$ $(i = 0, 1, ..., 4095)$ and $F'(h_i)$ $(i = 0, 1, ..., 4095)$ to represent the frame hexamer frequency for a given hexamer sequence $S = H_1, H_2, ...H_m$ (WANG et al., 2013).

- **Fickett score:** It is a feature that distinguishes protein-coding RNA and ncRNA, which uses a combinatorial effect of nucleotide composition and codon usage bias (WANG et al., 2013), as explained in Equation (2.4):

$$FickettScore = \sum_{i=1}^{8} p_i w_i \tag{2.4}$$

Where the Fickett score is obtained by computing four-position values and composition (nucleotide content - eight values in total) of a given sequence, these values are converted into probabilities (p) of coding. Each probability is multiplied by a weight (w). This value reflects the probability of each parameter alone, detecting coding or non-coding sequences (WANG et al., 2013).

- **Kozak motif:** The Kozak motif has an impact on the protein translation efficiency and is found in the region around of the start codon and has an optimal sequence of GCCRCC**AUGG** (R represents purine) (XU et al., 2010).

- **BLAST:** Basic Local Alignment Search Tool (BLAST) is a widely applied algorithm in the search to proteins and DNA databases for sequence similarity (ALTSCHUL et al., 1997).

- **Profile HMM:** Achawanantakun et al. (2015) used HMM to measure the conservation of transcripts. According to Eddy (1996), "the key idea is that an HMM is a finite model that describes a probability distribution over an infinite number of possible sequences."

2.1.4 lncRNA Databases (Q4)

The advancement of new experimental techniques and sequencing technology has increased the development of multiple databases to exploit potential functions of lncRNA. Therefore, we mapped the main databases, as shown in Table 6.

Table 6 – Summary of lncRNA Databases.

Database	Year	Species	Reference
LncRNAdb	2010	Eukaryote	Amaral et al. (2010)
NONCODE	2011	17 species	Bu et al. (2011)
DIANA-LncBase	2012	Human, mouse	Paraskevopoulou et al. (2012)
LncRNADisease	2012	lncRNA-disease association	Chen et al. (2012)
LNCipedia	2012	Human	Volders et al. (2012)
LncRNome	2013	Human	Bhartiya et al. (2013)
Linc2GO	2013	Human	Liu et al. (2013)
PLncDB	2013	Plant species	Jin et al. (2013)
LncRBase	2014	Human, mouse	(CHAKRABORTY et al., 2014)
lncRNAWiki	2014	Human	Ma et al. (2014)
lncRNAMap	2014	Human	Chan, Huang and Chang (2014)
lncRNAtor	2014	5 species	Park et al. (2014)
LncRNA2Function	2015	Human	Jiang et al. (2015)
GreeNC	2015	37 plant species	Gallart et al. (2015)
PLNlncRbase	2015	Plant species	Xuan et al. (2015)
CANTATAdb	2015	Plant species	Szcześniak, Rosikiewicz and Makałowska (2015)
lncRInter	2017	15 Species	Liu et al. (2017)
DLREFD	2017	Disease	Sun et al. (2017)

Source – Adapted from Fritah, Niclou and Azuaje (2014) and Chen et al. (2018).

After analyzing the main existing databases for lncRNA, we decided to revise datasets used by each method covered in this review, according to Table 7. This table aims to map which sequences the methods use as negative data to predict lncRNAs.

Table 7 – Summary of the sequences and databases used by each method.

Method	Sequences and Database
CPC	ncRNA (Rfam and RNADB); protein-coding (EMBL)
CPAT	lncRNA (GENCODE); protein-coding (RefSeq)
CNCI	lncRNA (GENCODE and Ensembl); protein-coding (RefSeq and Ensembl)
PLEK	lncRNA (GENCODE); protein-coding (RefSeq)
lncRNA-MFDL	lncRNA (GENCODE); protein-coding (RefSeq)
LncRNA-ID	lncRNA (GENCODE); protein-coding (GENCODE)
lncRScan-SVM	lncRNA (GENCODE); protein-coding (GENCODE)

LncRNApred	lncRNA (NONCODE); protein-coding (UCSC database)
DeepLNC	lncRNA (LNCipedia); protein-coding (RefSeq)
BASiNET	ncRNA and lncRNA (GENCODE and Ensembl); protein-coding (RefSeq)
LncFinder	lncRNA (GENCODE and Ensembl); protein-coding (GENCODE and Ensembl)
PlantRNA-Sniffer	lncRNA (CANTATAdb); protein-coding (Ensembl)
PLncPRO	lncRNA (CANTATAdb); protein-coding (Phytozome, GENCODE)
RNAplonc	lncRNA (PLNlncRbase and GreeNC); protein-coding (Phytozome)
Ensemble	lncRNA (lncRNAdb, lncRNAdisease, Ensembl, PNRD, and RNAcentral); protein-coding (Phytozome)

Source – Elaborated by the author.

Thereby, we observed that all methods reviewed used protein-coding genes as a negative weight to train classifiers in the detection of lncRNAs, which justified the high amount of features to find coding sequences (e.g., ORF and MLCDS). Regarding databases, the most used by the methods were GENCODE, RefSeq, Ensembl, and Phytozome.

2.1.5 Feature Selection of lncRNAs (Q5, Q6)

In Bioinformatics, data are becoming bigger not only in terms of the abundance of patterns but also in the dimensionality of features. Thus, lncRNA detection is no different, since the reviewed works present a high number of input attributes. According to Wang, Wang and Chang (2016), this fact can significantly degrade the accuracy of learning algorithms, especially when there is the presence of irrelevant or redundant features. Based on this problem, some lncRNAs researchers have applied feature selection methods, as presented in Table 8. We found eight papers, among them: Wang et al. (2014) applied Genetic Algorithm (GA); Lertampaiporn et al. (2014) Correlation-Based Feature Selection (CFS) and GA; Pian et al. (2016) Feature Score Criterion (FSC).

Tripathi et al. (2016) combined Forward Selection and Backward Elimination (FSBE); Ventola et al. (2017) used several feature selection approaches and algorithms, such as Filter (Wilcox test (WT), Information Gain (IG), Gain Ratio (GR), and Recursive Feature Elimination (RFE)); Wrapper (RFE and Greedy Forward Selection (GFS)); Embedded (Lasso regression (LR), Elastic Net (EN), and Random Forest (RF)). Yang et al. (2018) applied Minimal-Redundancy-Maximal-Relevance (mRMR); Han et al. (2018) used RFE; and finally, Negri et al. (2018) analyzed methods present in Weka, such as IG, and GR.

Table 8 – Feature Selection in lncRNAs.

Reference	**Method**	**Application**
Wang et al. (2014)	Heuristic	Human
Lertampaiporn et al. (2014)	Heuristic	Multi-Species
Pian et al. (2016)	Statistical	Human
Tripathi et al. (2016)	Heuristic	Human
Ventola et al. (2017)	Heuristic and Statistical	Human; Mouse; Zebrafish
Yang et al. (2018)	Heuristic	Human; Mouse
Han et al. (2018)	Heuristic	Multi-Species
Negri et al. (2018)	Statistical	Plant

Source – Elaborated by the author.

Essentially, our review showed that, again, most of the works that used feature selection methods are applied in animal and human systems (87.50%) and only a specific article for plants (12.50%). We also note that many papers did not conduct an in-depth study of features, using this method only as a means to reduce dimensionality, but without understanding the efficiency of its attributes. Lastly, although the heuristic algorithms are more applied, only two works used meta-heuristic.

2.1.6 Evaluation Metrics (Q7)

After feature extraction, selection, and classification, the next step is to analyze the efficiency of the developed model. Therefore, we report the primary metrics used by the reviewed articles, as exposed in Table 9. Different performance metrics were applied, among them: Sensitivity (SE), Specificity (SPC), Accuracy (ACC), F1-score, Positive Predictive Value (PPV), Negative Predictive Value (NPV), Matthews Correlation Coefficient (MCC), Area Under the Curve (AUC), and Receiver Operating Characteristic (ROC) Curve.

It is important to highlight that all metrics used are associated with the confusion matrix, which is one of the most intuitive and easy methods applied to find the Accuracy and Precision of a model. Thereby, True Positive (TP) estimator measures the correctly predicted positive class; True Negative (TN) estimator represents the negative class

Table 9 – Performance metrics used by each method.

Method	SE	SPC	ACC	F1-score	PPV	NPV	MCC	AUC	ROC
CPC	-	-	*V*	-	-	-	-	-	-
CPAT	*V*	*V*	*V*	-	*V*	-	-	*V*	*V*
CNCI	*V*	*V*	*V*	-	-	-	-	-	*V*
PLEK	*V*	*V*	*V*	-	*V*	*V*	-	-	-
lncRNA-MFDL	*V*	*V*	*V*	-	-	-	*V*	-	-
LncRNA-ID	*V*	*V*	*V*	*V*	*V*	-	-	*V*	*V*
lncRScan-SVM	*V*	*V*	*V*	-	-	-	*V*	*V*	*V*
LncRNApred	*V*	*V*	*V*	-	-	-	*V*	-	*V*
DeepLNC	*V*	*V*	*V*	-	-	*V*	*V*	*V*	*V*
BASiNET	-	-	*V*	-	-	-	-	-	-
LncFinder	*V*	*V*	*V*	*V*	-	-	-	*V*	*V*
PlantRNA-Sniffer	-	-	*V*	-	-	-	-	-	-
PLncPRO	*V*	*V*	*V*	-	-	-	*V*	-	*V*
RNAplonc	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
Ensemble	*V*	*V*	*V*	-	-	-	-	*V*	-

Source – Elaborated by the author.

correctly classified; False Positive (FP) estimator describes all those negative entities that are incorrectly classified as positive, and False Negative (FN) estimator represents all positive that are incorrectly classified as negative. Finally, we elaborate a brief definition of each measure (WITTEN; FRANK; HALL, 2011):

- **SE:** Also known as Recall and true positive rate, measures the proportion of actual positive cases that are correctly identified (see Equation 2.5).

$$SE = \frac{TP}{TP + FN} \tag{2.5}$$

- **SPC:** Also called the true negative rate, measures the proportion of actual negative cases which are correctly identified (see Equation 2.6).

$$SPC = \frac{TN}{TN + FP} \tag{2.6}$$

- **ACC:** Accuracy is the proportion of the total number of predictions that were correct (see Equation 2.7).

$$ACC = \frac{TP + TN}{TN + FP + TP + FN} \tag{2.7}$$

- **F1-Score:** Also called F-measure, can be interpreted as a weighted average of the precision and recall (see Equation (2.8).

$$F1 - Score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2.8}$$

- **PPV:** Also called Precision, measures the proportion of positive cases that were correctly identified (see Equation 2.9).

$$PPV = \frac{TP}{TP + FP} \tag{2.9}$$

- **NPV:** Measures the proportion of negative cases that were correctly identified (see Equation 2.10).

$$NPV = \frac{TN}{TN + FN} \tag{2.10}$$

- **MCC:** It is a quality measure of two binary classifications. Basically, it returns a value between -1 and +1, where a coefficient +1 represents a perfect prediction, 0 a mean random prediction, and -1 a reverse prediction (see Equation 2.11) (BALDI et al., 2000).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{2.11}$$

- **ROC:** ROC curves depict the performance of a classifier without regard to class distribution or error costs (WITTEN; FRANK; HALL, 2011).

- **AUC:** This metric provides an aggregate measure of performance across all possible classification limits.

Finally, it is essential to emphasize that, according to Valverde-Albacete and Peláez-Moreno (2014), "predictive models with a given level of accuracy may have greater predictive power than models with higher accuracy." Therefore, we decided to apply all metrics used in the literature (see Table 9), as high accuracy is not entirely an indicator of high classifier performance (VALVERDE-ALBACETE; PELÁEZ-MORENO, 2014).

2.2 Considerations

In this chapter, we review computational methods for lncRNAs classification. In which, we observe a considerable amount of research in humans, followed by animals and plants. Moreover, all authors apply supervised learning methods using binary classification (two classes), and protein-coding genes as a negative weight for the datasets. Among the most commonly used classification algorithms are support vector machines, followed by random forest and deep learning. Regarding feature extraction, we observed a full domain of ORF features and sequence structure (features with biological bias). Due to a high number of attributes extracted by the authors, we reviewed works that performed feature selection methods for the lncRNAs problem. However, we note a lack of computational methods related to the systematic study of robust attributes, as well as the application of metaheuristic techniques and mathematical models for feature selection and extraction, respectively.

3 EXPERIMENTAL TEST I: FEATURE SELECTION PROBLEM

This chapter describes the methodological procedures used to achieve the proposed objectives in our first problem (feature selection). Fundamentally, each stage of study is described, as well as information about the process adopted for the development and application of the research. Thus, based on our systematic review presented in Chapter 2, we select the most used feature extraction methods for lncRNAs classification (ORF, sequence length, GC content, and k-mer), to apply metaheuristic models for feature selection. We design experiments to answer the hypotheses shown in Section 1.4 (H1-PB1 and H2-PB1). Therefore, this chapter is divided into two parts: Experimental Methodology and Results. Discussions will be presented in Section 5.

3.1 Experimental Methodology

We divided this first experiment into seven stages: (1) preprocessing of FASTA files, removing redundant sequences and less than 200 nucleotides; (2) Split sequences into training, and test; (3) Feature extraction; (4) Feature Selection; (5) Training; (6) Test; (7) Performance analysis. For a better understanding, Figure 1 summarizes the adopted methodological approach. Furthermore, it is necessary to emphasize that we denote a biological sequence $S = (S[0], S[1], \ldots, S[N-1])$ such that $S \in \{A, C, G, T\}^N$. Finally, we denote all sequences of our dataset (mRNA/lncRNA) by $SeqRNA$.

3.1.1 Training Set Construction

We built a training set with sequences from 5 plant species (*Arabidopsis thaliana*, *Cucumis sativus*, *Glycine max*, *Oryza sativa* and *Populus trichocarpa* - see Table 10), adopting the datasets used in Negri et al. (2018). Two classes were defined for the datasets: positive class, with lncRNAs, and negative class, with protein-coding genes (mRNAs). The lncRNA data were extracted from two public databases, `PLNlncRbase` (defined by B_{PLN}) (XUAN et al., 2015) and `GreeNC` (version 1.12 - defined by B_{Gree}) (GALLART et al., 2015).

The mRNA transcript data were collected from `Phytozome` (defined by B_{Phy}) (GOODSTEIN et al., 2011) database version 11. The choice of the mentioned databases was based on their impact and in the number of species available. In both lncRNA and mRNA datasets, we used only sequences longer than 200nt, and we removed sequence redundancy at 80% of identity (LERTAMPAIPORN et al., 2014; SU et al., 2018) using

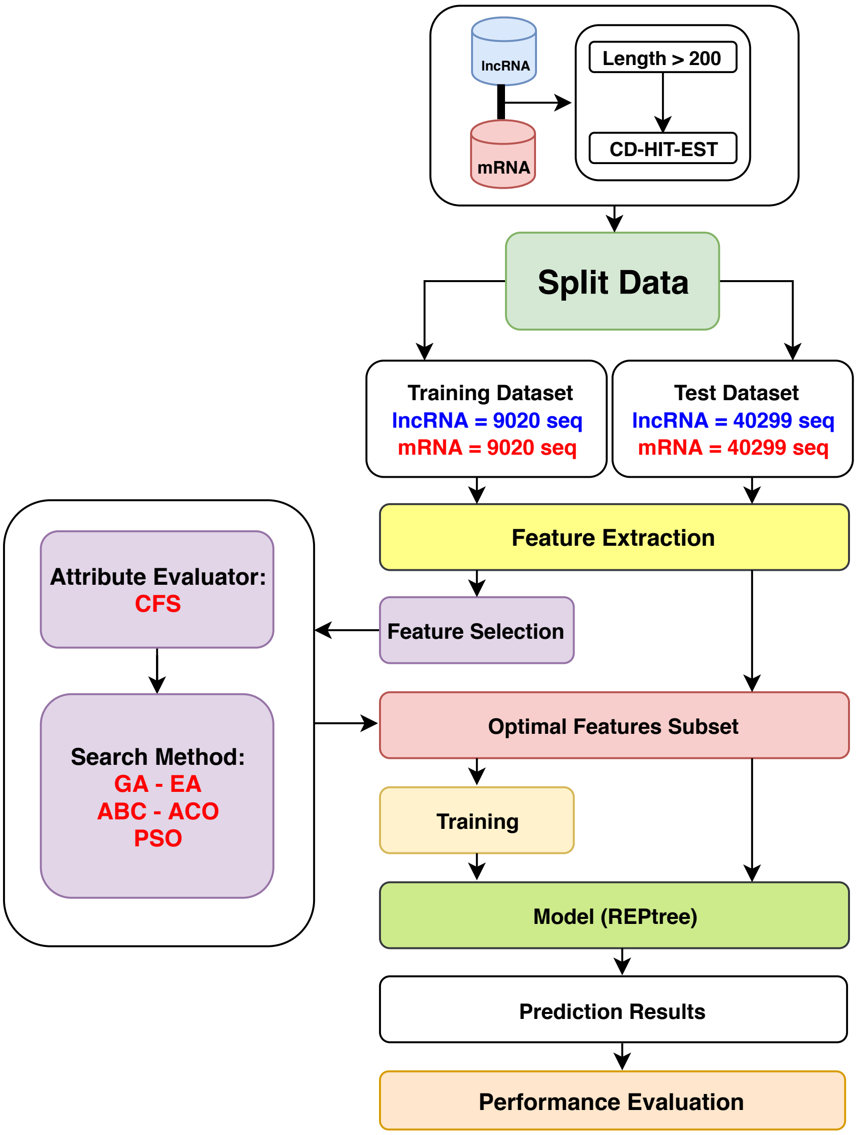


Figure 1 – Proposed Workflow for the Feature Selection Problem in lncRNAs. [1]

Source – Elaborated by the author.

CD-HIT-EST tool (v4.6.1) (LI; GODZIK, 2006). We selected randomly 1,804 sequences for each species, due to the amount of data, to balance the number of samples in each species. Therefore, a total of 9,020 lncRNA sequences and 9,020 mRNA sequences were obtained after filtering steps.

[1] Note: The FASTA format files (lncRNA - positive dataset and mRNA - negative dataset) were filtered to find sequences larger than 200 nucleotides (size > 200), and we removed redundant sequences with 80% of identity (CD-HIT-EST). This dataset was divided into training (9020 lncRNA and 9020 mRNA) and testing (40299 lncRNA and 40299 mRNA). Features were extracted from each sequence. Filters were applied to the training set to select a subset of features (see Table 12). Next, for each selected feature set, ML algorithms were applied to the data to induce predictive models. The models induced for each filter were applied to the test set, using the same selected features in the training set. Finally, the predictions of the model induced for each filter was evaluated.

Table 10 – Species used to create the training set.

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

3.1.2 Test Set Construction

To assess the performance of the algorithms used in this study, we use eight datasets of plant species (*Amborella trichopoda, Brachypodium distachyon, Citrus sinensis, Manihot esculenta, Ricinus communis, Solanum tuberosum, Sorghum bicolor* and *Zea mays*), summarized in Table 11.

Table 11 – Species used to create the test set.

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

Test sets followed the same steps as the training set (sequences longer than 200nt, and removed sequence redundancy at 80% of identity with CD-HIT-EST). The lncRNA sequences were extracted from GreeNC, and mRNA sequences from Phytozome.

3.1.3 Feature Extraction

The extraction of relevant features plays an essential role in this issue. Thus, feature extraction is one of the most critical steps in the induction of a robust predictor/classifier (FAN; ZHANG, 2015). It were extracted features considering four feature groups (ORF, sequence length, GC content, and k-mer) to distinguish lncRNA from mRNA. Basically, four sets of values were extracted from the sequences, creating four vectors, described next.

GC content descriptor

According to the literature, when we compare lncRNAs with mRNAs, the lncRNAs have low GC content (NIAZI; VALADKHAN, 2012). The GC content (guanine-cytosine content - denoted by f_{GC}), represented by Equation (2.2), is the percentage of nitrogenous bases in a DNA or RNA molecule that is guanine or cytosine (AMR; FUNKE, 2015).

k-mer descriptor

As lncRNAs normally have a low potential for protein-coding (FAN; ZHANG, 2015), the frequency of neighboring bases k (k-mer) may contain statistical information to distinguish lncRNAs from mRNAs. The k-mer is denoted in this work by f_{kmer}, corresponding to Equation (3.1).

$$
\begin{aligned}
f_{kmer}(S) = \frac{c_i^k}{N-k+1} = \Bigg(& \frac{c_1^1}{N-1+1}, \dots, \frac{c_4^1}{N-1+1}, \\
& \frac{c_{4+1}^2}{N-2+1}, \dots, \frac{c_{5460}^6}{N-6+1} \Bigg) \qquad k = 1, 2, \dots, 6.
\end{aligned}
\tag{3.1}
$$

This equation is applied to each sequence with frequencies of $k = 1, 2, 3, 4, 5, 6$. Where, c_i^k is the number of substring occurrences with length k in a sequence S with length N, in which the index $i \in \{1, 2, \dots, 4^1 + \dots + 4^k\}$ represents the analyzed substring.

Sequence length descriptor

We also used as feature the sequence length (denoted by f_{SL}), since the lncRNAs were shown to be considerably shorter than mRNAs (NIAZI; VALADKHAN, 2012; WUCHER et al., 2017).

Open Reading Frame (ORF) descriptor

Identifying candidate ORFs in the transcripts is an important guideline for distinguishing lncRNAs from mRNA (NIAZI; VALADKHAN, 2012; FRITH et al., 2006; BAEK et al., 2018). For such, we analyze the three frames in the forward strand of our sequences using the `txCdsPredict` program from the UCSC genome browser (KENT et al., 2002) [2]. We used this program, which predicts potential ORFs from a given sequence w, to extract the following features:

- txCdsPredict Score: This attribute measures the probability that a sequence is a protein. In which, a score above 1000 is likely to be a protein. The scores above 800 have 90% chance;

[2] (https://genome.ucsc.edu/ (KENT et al., 2002))

- cdsStarts: nt position for the CDS start within the transcript, zero-based;
- cdsStop: nt position for the CDS end, noninclusive;
- cdsSizes: $cdsStop\ nt\ position - cdsStart\ nt\ position$;
- cdsPercent: $\frac{cdsStop\ nt\ position + cdsStart\ nt\ position}{total\ nt\ sequence\ size}$.

The features were passed as a vector for the function that we denote by f_{ORF}, corresponding Equation (3.2).

$$f_{ORF}(S) = (Score, cdsStarts, cdsStop, cdsSizes, cdsPercent). \tag{3.2}$$

The `txCdsPredict` has been used in several studies ((SUN et al., 2013; SUN et al., 2015; LI et al., 2018)), to determine if a transcript is protein-coding and, if so, the locations of the start and stop codons. The algorithm uses ORF length, the presence of a Kozak consensus sequence at the start codon, the presence of upstream ORFs, homology in other species, and nonsense-mediated decay (KENT et al., 2002). Furthermore, several tools have used ORF features, among them: CPC, CPAT, lncRNA-MFDL, LncRNA-ID, lncRScan-SVM, LncRNApred, PlantRNA-Sniffer, PLncPRO, RNAplonc, and LncFinder.

Concatenate Feature Vectors

According to Fan and Zhang (2015), a concatenated feature vector can keep the most discriminatory information from original multi-feature sets and eliminate the redundant information from the correlation between distinct feature sets, resulting in models with robust predictive performance. To represent each transcribed sequences in the datasets, we concatenate the previously mentioned features in a new feature vector, defined as follows (Equation (3.3)):

$$\begin{aligned} V_f &= \{(X_i, Y_i) \,|\, \forall\, S_i \in \text{SeqRNA}, \\ X_i &= (f_{GC}(S_i), f_{kmer}(S_i), f_{SL}(S_i), f_{ORF}(S_i)), \\ Y_i &= Label(S_i)\}. \end{aligned} \tag{3.3}$$

Where the feature vector V_f contains the elements X_i and Y_i for every sequence S_i belonging to $SeqRNA$, such that X_i is formed by the functions ($f_{GC}(S_i)$, $f_{kmer}(S_i)$, $f_{SL}(S_i)$, $f_{ORF}(S_i)$) and Y_i by the labels 0 (mRNA); 1 (lncRNA). Therefore, we collected 5,467 genomic characteristics for each sequence relative to the training set (see Table 10): GC content (1 feature), k-mer (1-6 k-mer length = $5,460$ features), Sequence length (1 feature), and ORF metrics (5 features).

3.1.4 Data Preprocessing

Data normalization is a preprocessing technique frequently applied to a dataset before feature selection and modeling. Essentially, features can have different numeric ranges. Thus, features with a larger range which can have a stronger effect in the induction of a predictive model, mainly for distance-based ML algorithms. The application of a normalization procedure makes the ranges similar, reducing this problem (SINGH; VERMA; THOKE, 2015). We used in this work the min-max method, which reduces the data range to 0 and 1 (or -1 to 1, if there are negative values). The general formula is given as (Equation (3.4)) (SOUTO et al., 2008):

$$x'_{ij} = \frac{x_{ij} - Min(j)}{Max(j) - Min(j)}. \tag{3.4}$$

Where x is the original value and x'_{ij} is its re-scaled version. Further, $Min(j)$ and $Max(j)$ are, respectively, the smallest and the largest values of a feature j (SOUTO et al., 2008).

3.1.5 Feature Selection Techniques

Feature selection techniques are typically categorized as filters, wrappers, or embedded approaches (STAŃCZYK, 2015). Filters are applied independent of the ML algorithm used (DASH; LIU, 2003), considered as a preprocessing stage for a subsequent learning (KRIZEK, 2008). They exploit the information present in the predictive features of a dataset, assessing their relevance using measures like information gain, entropy, and consistency (DASH; LIU, 2003; STAŃCZYK, 2015). Wrappers evaluate the relevance of subsets of predictive features using an ML algorithm as an oracle (KOHAVI; JOHN, 1997), i.e., they use the accuracy of predictive models to guide the selection of an optimal subset of features (KRIZEK, 2008). The embedded approach is implemented as part of an ML algorithm that has an internal feature selection mechanism (GUYON; ELISSEEFF, 2003; LAL et al., 2006).

Table 12 – Applied Algorithms and Methods.

ID	Attribute Evaluator	Filter (Metaheuristic)
M1-GA	CFS	GA
M2-EA	CFS	$(\mu + \lambda)$EA
M3-ABC	CFS	ABC
M4-ACO	CFS	ACO
M5-PSO	CFS	PSO

In this dissertation, we applied *Filters*, to select subsets of features in a preprocessing step, independently of the ML algorithm used later. According to Guyon et al. (GUYON;

ELISSEEFF, 2003), there are several justifications for the use of filters, among them: (1) filters were successfully reported in several previous works. (2) Compared to wrappers, filters are faster. (3) Filters provide a generic selection of variables, i.e., the choice of features is not adjusted to an ML algorithm. In this study, we have used metaheuristics as filters. Also, we have worked with WEKA (Waikato Environment for Knowledge Analysis - Version 3.8 - (HALL et al., 2009)) to execute the algorithms in our study (see Table 12). Also, we have used Perl (Version 5.24.1) and Python (Version 3.5) in all experiments.

3.1.6 Objective Function

The *CFS* algorithm (HALL, 1999) was used to evaluate the feature subsets selected by the metaheuristics. This algorithm analyzes the predictive capacity and degree of redundancy of the subset. It looks for a subset that is highly correlated with the target class and has a low correlation with other features (SELVAKUBERAN; INDRADEVI; RAJARAM, 2008; FONG; BIUK-AGHAI; MILLHAM, 2018). For such, it uses a correlation-based heuristic evaluation function (see Equation 3.5). The chances of a feature to be selected depending on how well the feature can predict the correct class when other features cannot (HALL, 1999).

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3.5}$$

Where M_S is the merit of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f \in S$), and $\overline{r_{ff}}$ is the average feature-feature inter-correlation (HALL, 1999). Therefore, we investigate the performance of five metaheuristics for feature selection (see Table 12).

3.1.7 Metaheuristics

We have chosen the traditional metaheuristics considering the excellent performance for the feature selection reported in different areas of the literature (DOERING et al., 2019; NAYAR; AHUJA; JAIN, 2019; GUPTA; SHENG, 2019). Moreover, we have also previously validated the EA and ACO algorithms against five other state-of-the-art approaches (RNAplonc, CPC, CPC2, CNCI, PLEK) (Parmezan Bonidia et al., 2019). The results obtained in (Parmezan Bonidia et al., 2019) are very interesting, and support the quality of metaheuristic in the search process of dimensionality reduction and feature selection. For that reason, we extended this discussion and studied performance among five efficient metaheuristics (GA, $(\mu + \lambda)$EA, ABC, ACO, PSO). Also, we propose here to investigate not only the selected feature subsets but also the quality of each feature.

M1-GA: Genetic Algorithm

GA was developed by John Holland, his colleagues, and his students at the University of Michigan. It is a general stochastic search algorithm that effectively exploits large search spaces, which is generally required in case of feature selection. Moreover, GAs conduct global research and are based on the mechanics of natural selection. Essentially, they simulate the processes in natural systems for evolutions based on the principle of "survival of the fittest" (Charles Darwin) (GOLDBERG, 1989). Also, they work with the coding of the parameter set and use reward information (objective function). Therefore, GA applied in this dissertation is defined by Goldberg (1989), composed of three basic operators: reproduction, crossing, and mutation. The chromosome consists of binary bits, 1 to represent the attribute selection, and 0 to eliminate it.

M2-EA: Evolutionary Algorithm

The EA used in this work applies the fitness ranking selection procedure. In other words, at the end of each evolution cycle, the whole population is renewed according to generational substitution scheme. Furthermore, elitism and tournament are applied, in which the fittest individual of the population is kept in the new generation. The chromosomes (Binary encoding) are manipulated using standard genetic operators of mutation and crossover (single-point crossover, bit flip mutation). However, the EA has an extra component (a) that represents the interval width of the mutation even, where the modification has a uniform probability $[-a, a]$. Thus, for each individual, the parameter is adjusted adaptively through random mutation events (FOGEL, 1995; PHAM; CASTELLANI, 2009).

M3-ABC: Artificial Bee Colony

The ABC is bio-inspired in the food foraging behavior of bees to seek the best solution to an optimization problem. Each point in the search space is considered as a food source. The "Scout Bees" randomly sampled the space and through the fitness function, they report the quality of the visited places. The solutions are then ranked, and other "bees" are recruited to search the fitness landscape in the neighborhood of the highest ranking locations. The neighborhood of a solution is called a "flower patch". Therefore, the algorithm searches the most promising solutions and selectively explores its neighborhoods looking for the global minimum of the objective function (PHAM; CASTELLANI, 2009).

M4-ACO: Ant Colony Optimization

The ACO is a bio-inspired algorithm by the foraging behavior of some species of ants, developed by Dorigo, Maniezzo and Colorni (1996). This technique applies the pheromone method that ants deposit to demarcate a more favorable path and that must be followed by other members of the colony (DORIGO; MANIEZZO; COLORNI, 1996; DORIGO; BIRATTARI, 2011; BONIDIA et al., 2018). Fundamentally, each agent (ants) initially follows a random way, and after some time they tend to follow a single way, considered significant. They use indirect communication to indicate the best route for the other members of the colony. For this, they spread a substance called pheromone. That is, computationally, the algorithm presents a graph with n vertices and places an artificial ant in each of these. Thereby, each ant traces a path following a probabilistic equation in function of the "deposited" pheromone at each edge of the graph. Finally, after the construction of all routes, the pheromone intensity in each edge is increased according to the quality of the generated solution.

M5-PSO: Particle Swarm Optimization

It is a bio-inspired computational algorithm in the social behavior metaphor about the interaction between individuals (particles) of a group (swarm), developed in 1995 by Kennedy and Eberhart. This algorithm was implemented based on the observation of flocks of birds and shoals of fish in search of food in a certain region (MORAGLIO et al., 2008; KENNEDY, 2011; BONIDIA et al., 2018). The PSO is a population-based stochastic global optimization algorithm (KENNEDY, 2006). The version applied in this research uses the geometric framework, where it presents an intimate relation between a simplified form of PSO (without the inertia term) and evolutionary algorithms. This framework enables to generalize, in a natural, rigorous, and automatic way, PSO for any search space for which a geometric crossover is known (MORAGLIO et al., 2008). This algorithm was developed using theoretical tools of evolutionary algorithms, that is, geometric crossing and geometric mutation. Basically, there is no velocity, the equation of position update is the convex combination, there is mutation and the parameters w_1, w_2, and w_3 are non-negative and add up to one (KENNEDY, 2011).

3.1.8 Execution Rounds

At this stage, we proposed a new metaheuristic execution pipeline, as shown in Figure 2. In which, each metaheuristic was submitted to five different trails (called here as round). Also, for each round, the algorithm was run 10 times (empirically chosen value with different seeds), where the best individual found in each run is selected, generating a ranking of best candidates. Then, considering the ranking of candidates, we have selected most voted features (vote = 100%, i.e., features incident to all candidates). So, we start a new round considering only the selected features from previous round. This process was repeated five times (five rounds), totaling 50 runs.

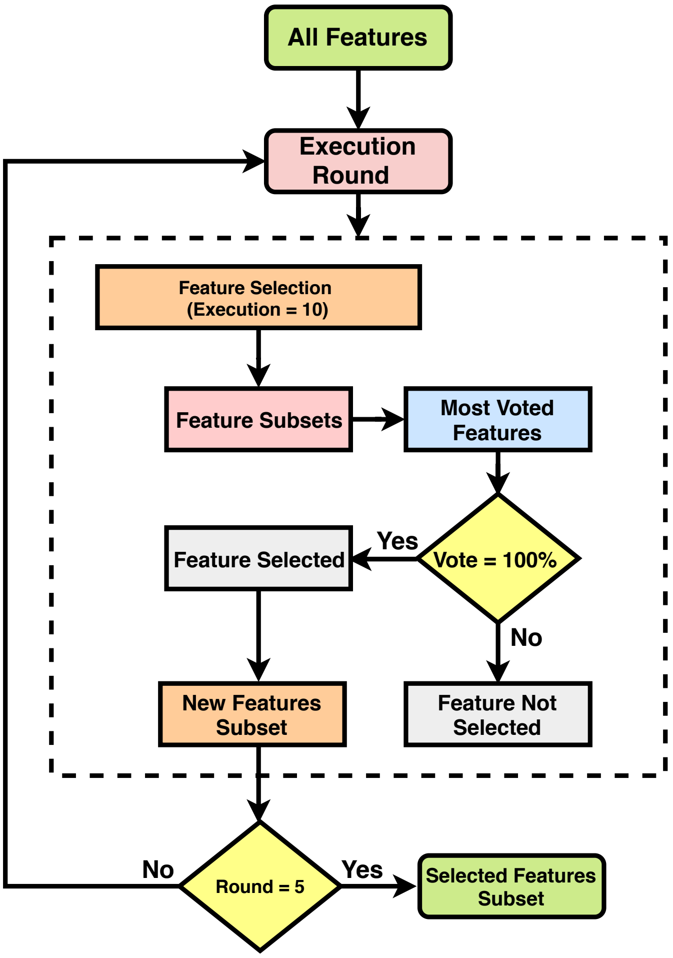


Figure 2 – Metaheuristic Execution Pipeline.

Source – Elaborated by the author.

3.1.9 Evaluation Metrics

The methods were evaluated with seven measures (FAN; ZHANG, 2015; DUDA; HART; STORK, 2012): Sensitivity (SE), Specificity (SPC), Accuracy (ACC), F1-score, Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Matthews Correlation Coefficient (MCC). These measures were used to evaluate the models' predictive performance. These measures use True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values, where: TP measures the correctly predicted lncRNAs; TN represents the correctly classified mRNAs; FP describes all those negative entities that are incorrectly classified as lncRNAs and; FN represents the true lncRNAs that are incorrectly classified as mRNAs. Nevertheless, these metrics were applied only in the test sets, which are essential for the analysis of the proposed models. Thus, the metrics used to evaluate the training set were Accuracy (ACC) and Error Rate (ER).

3.2 Results

This section shows experimental results from experiments conducted with 5,467 genomic characteristics for each sequence relative to the training and test sets, as well as a systematic study on the optimal feature subsets chosen by the metaheuristic methods in the feature selection process.

3.2.1 Hyperparameters of the algorithms

To apply the dimensionality reduction (feature selection) with algorithms exposed in Table 12, the following hyperparameter values were defined:

- **M1-GA:** Crossover Operator (single-point), Crossover Probability (0.6), Generations (20), Mutation Operator (bit-flip), Mutation Probability (0.033), Population Size (20), Selection Operator (roulette wheel);
- **M2-EA:** Generations (20), Crossover Operator (single-point), Crossover Probability (0.6), Mutation Operator (bit-flip), Mutation Probability (0.1), Population Size (20), Selection Operator (tournament);
- **M3-ABC:** Iterations (20), Population Size (30), Number of Selected Sites (15), Number of Elite Sites (8), Number of Selected Site Bee (15), Number of Elite Site Bee (30);
- **M4-ACO:** Evaporation ($rho = 0.9$), Pheromone ($\alpha = 2.0$), Heuristic ($\beta = 0.7$), Q (30), $tau0$ (0.1), Iterations (20), Population Size (20);
- **M5-PSO:** Iterations (20), Social Weight (0.33), Population Size (20), Mutation Operator (bit-flip), Mutation Probability (0.01), Individual Weight (0.34), Inertia Weight (0.30).

3.2.2 Feature Selection

At this stage, methods presented in Table 12 were applied in training set with the purpose of reducing the dimensional space of the extracted features, as shown in Table 13. For each one of the five runs, each metaheuristic selected a feature subset of decreasing size, as illustrated by Table 13. At each new run, the metaheuristic was applied to the previously selected subset, in order to further reduce the number of features. When the number of features was not reduced, we put the symbol "−". Among the selected features subsets by methods (see Table 14), two found the least amount of features (M2-EA and M3-ABC), returning a subset with 5 attributes, followed by M4-ACO (6 features), M5-PSO (7 features) and M1-GA (10 features).

Table 13 – Execution Rounds (e.g. R1 = First round).

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

Source – Elaborated by the author.

Table 14 – Optimal features subsets selected by each method.

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

Source – Elaborated by the author.

Another fact that can be observed at the methods intersection is that everyone selected two equal features (*txCdsPredict score* and *cdsSizes*). The rest of the methods demonstrated by the one or two intersections, such as M1-GA ∩ M2-EA (CCGGCA), and M1-GA ∩ M5-PSO (GGGGGG, TGACGG). We will explore more about the features obtained soon.

3.2.3 Models Training

To induce predictive models that can be interpreted, we applied three decision tree induction algorithms (Random Forest, J48, REPTree) to the training set. We performed experiments to evaluate the performance of selected features in each round by each metaheuristic. This experiment investigated if the predictive performance was maintained, as the feature sets were reduced (see Table 15). In which, J48 and REPtree algorithms presented similar performance, an average of 92.77% and 92.76% (ACC), respectively. In contrast, Random Forest had the worst performance (ACC $\approx$ 91,19%). Therefore, it was decided to use the same algorithm applied in (NEGRI et al., 2018), REPtree. Furthermore, it was observed that the methods preserved performance as the features were reduced. Thus, the optimal features subsets selected by the metaheuristics (see Table 14) were applied to construct the prediction models.

Table 15 – Classification accuracy in each execution round of the Table 13.

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

The "−" means that the algorithm obtained the same result.

Source – Elaborated by the author.

3.2.4 Performance Testing

The predictive models induced by REPTree were applied to the test sets (see Test Set Construction Method), producing the results shown in Table 16. As can be seen, the results obtained using the selected feature sets were similar. The best predictive performance regarding SE and ACC were obtained by feature set selected by M1-GA (SE: 100% and ACC: 91.29%), followed by M3-ABC (SE: 99.95% and ACC: 91.27%), and M4-ACO (SE: 99.94% and ACC: 91.27%). Regarding specificity, the best methods were M2-EA (82.61%) and M4-ACO (82.61%).

Evaluating the individual performance of the metaheuristics for different species, we noticed that the models obtained high accuracy in six datasets, among them: *C. sinensis* (ACC: 94.09% - M2-EA), *M. esculenta* (ACC: 93.30% - M2-EA), *B. distachyon* (ACC: 92.60% - M1-GA), *S. bicolor* (ACC: 92.47% - M1-GA), *R. communis* (ACC: 90.74% - M1-GA, M3-ABC and M4-ACO), and *Z. mays* (ACC: 90.73% - M1-GA). Regarding the individual sensitivity (to detect lncRNA), we achieved the best results with all species and methods, on the other hand, we reduced the specificity (to detect mRNA), especially in two sets (*A. trichopoda* and *S. tuberosum*). Therefore, to better understand the results and features, we decided to analyze the contribution of the selected features subsets, as reported in the next sections.

3.2.5 Our Approach Against All Features

In this section, we compare the best and worst model in the *performance test* (see Table 16), respectively, M1-GA and M5-PSO, against a model without feature selection (*Full features (5,467)*). In the overall average, our approach represented a gain of 4.68% (M1-GA) and 4.62% (M2-PSO) in the ACC. However, in some species, we reached an increase in the ACC of 6.40% (*S. bicolor*), 5.92% (*B. distachyon*), and 4.85% (*C. sinensis*). Fundamentally, these results expose the high efficiency of metaheuristics for feature selection in lncRNAs. These results will be discussed with greater emphasis in Chapter 5.

3.2.6 Influence and Contribution of the Selected Features

According to Zhu et al. (2010), when confronted with high-dimensional bioinformatics problems, it is important to examine and compare the contributions of different features. To better understand the influence and contribution of the features and models, we first consider the analysis of all features selected in the last run of each metaheuristic (see Table 14). For such, we plot the correlation matrix with attributes that presented at least one intersection, according to Figure 3. The chart shows the variables paired with all others.

Table 16 – Performance of all models. Each predictive model was induced using the feature subsets selected by the metaheuristics.

Species	Method - ID	TP	FP	TN	FN	SE	SPC	ACC	F1-score	PPV	NPV	MCC
A. trichopoda	M1-GA (10)	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M2-EA (5)	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M3-ABC (5)	3823	879	2944	0	**100**	77.01	88.50	89.69	81.31	**100**	79.13
	M4-ACO (6)	3822	877	2946	1	99.97	**77.06**	**88.52**	**89.70**	**81.34**	99.97	**79.14**
	M5-PSO (7)	3822	879	2944	1	99.97	77.01	88.49	89.68	81.30	99.97	79.10
B. distachyon	M1-GA (10)	4868	720	4148	0	**100**	85.21	**92.60**	**93.11**	87.12	**100**	**86.16**
	M2-EA (5)	4852	718	4150	16	99.67	85.25	92.46	92.97	87.11	99.62	85.82
	M3-ABC (5)	4863	719	4149	5	99.90	85.23	92.56	93.07	87.12	99.88	86.06
	M4-ACO (6)	4863	720	4148	5	99.90	85.21	92.55	93.06	87.10	99.88	86.04
	M5-PSO (7)	4851	716	4152	17	99.65	**85.29**	92.47	92.98	**87.14**	99.59	85.83
C. sinensis	M1-GA (10)	2292	272	2020	0	**100**	88.13	94.07	94.40	89.39	**100**	88.76
	M2-EA (5)	2292	271	2021	0	**100**	**88.18**	**94.09**	**94.42**	**89.43**	**100**	**88.80**
	M3-ABC (5)	2290	272	2020	2	99.91	88.13	94.02	94.36	89.38	99.90	88.66
	M4-ACO (6)	2291	272	2020	1	99.96	88.13	94.04	94.38	89.39	99.95	88.71
	M5-PSO (7)	2291	272	2020	1	99.96	88.13	94.04	94.38	89.39	99.95	88.71
M. esculenta	M1-GA (10)	3017	405	2612	0	**100**	86.58	93.29	93.71	88.16	**100**	87.37
	M2-EA (5)	3017	404	2613	0	**100**	**86.61**	**93.30**	**93.72**	**88.19**	**100**	**87.40**
	M3-ABC (5)	3017	405	2612	0	**100**	86.58	93.29	93.71	88.16	**100**	87.37
	M4-ACO (6)	3015	405	2612	2	99.93	86.58	93.25	93.68	88.16	99.92	87.29
	M5-PSO (7)	3016	405	2612	1	99.97	86.58	93.27	93.69	88.16	99.96	87.33
R. communis	M1-GA (10)	4080	756	3324	0	**100**	81.47	**90.74**	**91.52**	84.37	**100**	**82.91**
	M2-EA (5)	4078	756	3324	2	99.95	81.47	90.71	91.50	84.36	99.94	82.85

	M3-ABC (5)	4080	756	3324	0	**100**	81.47	**90.74**	**91.52**	84.37	**100**	**82.91**
	M4-ACO (6)	4078	754	3326	2	99.95	**81.52**	**90.74**	**91.52**	**84.40**	99.94	82.89
	M5-PSO (7)	4077	756	3324	3	99.93	81.47	90.70	91.48	84.36	99.91	82.82
S. tuberosum	M1-GA (10)	5607	1352	4255	0	**100**	75.89	**87.94**	**89.24**	80.57	**100**	**78.19**
	M2-EA (5)	5604	1352	4255	3	99.95	75.89	87.92	89.21	80.56	99.93	78.13
	M3-ABC (5)	5607	1352	4255	0	**100**	75.89	**87.94**	**89.24**	80.57	**100**	**78.19**
	M4-ACO (6)	5605	1351	4256	2	99.96	**75.91**	87.93	89.23	**80.58**	99.95	78.17
	M5-PSO (7)	5599	1351	4256	8	99.86	**75.91**	87.88	89.18	80.56	99.81	78.03
S. bicolor	M1-GA (10)	4541	684	3857	0	**100**	84.94	**92.47**	**93.00**	**86.91**	**100**	**85.92**
	M2-EA (5)	4530	683	3858	11	99.76	**84.96**	92.36	92.88	86.90	99.72	85.66
	M3-ABC (5)	4534	684	3857	7	99.85	84.94	92.39	92.92	86.89	99.82	85.74
	M4-ACO (6)	4537	684	3857	4	99.91	84.94	92.42	92.95	86.90	99.90	85.82
	M5-PSO (7)	4529	684	3857	12	99.74	84.94	92.34	92.86	86.88	99.69	85.62
Z. mays	M1-GA (10)	12071	2239	9832	0	**100**	81.45	**90.73**	**91.51**	84.35	**100**	**82.89**
	M2-EA (5)	12060	2234	9837	11	99.91	81.49	90.70	91.48	84.37	99.89	82.82
	M3-ABC (5)	12065	2234	9837	6	99.95	81.49	90.72	**91.51**	**84.38**	99.94	82.87
	M4-ACO (6)	12061	2233	9838	10	99.92	**81.50**	90.71	91.49	**84.38**	99.90	82.84
	M5-PSO (7)	12046	2233	9838	25	99.79	**81.50**	90.65	91.43	84.36	99.75	82.69
Overall Average	**M1-GA (10)**	-	-	-	-	**100**	82.58	**91.29**	**92.02**	85.27	**100**	**83.91**
	M2-EA (5)	-	-	-	-	99.90	**82.61**	91.26	91.99	**85.28**	99.89	83.82
	M3-ABC (5)	-	-	-	-	99.95	82.59	91.27	92.00	85.27	99.94	83.87
	M4-ACO (6)	-	-	-	-	99.94	**82.61**	91.27	92.00	**85.28**	99.93	83.86
	M5-PSO (7)	-	-	-	-	99.86	82.60	91.23	91.96	85.27	99.83	83.77

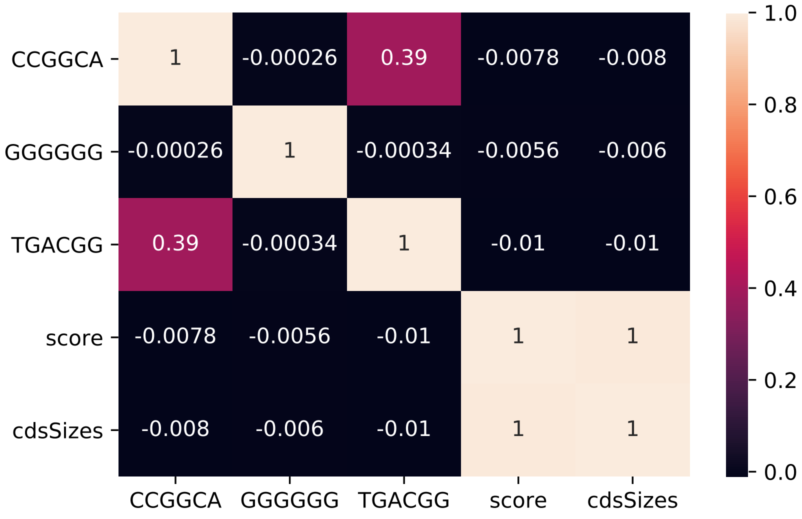


Figure 3 – Correlation matrix. The chart reports the correlation coefficients of the training data to features that presented at least one intersection: CCGGCA, GGGGGG, TGACGG, txCdsPredict score, and cdsSizes.

Source – Elaborated by the author.

The graph presents that ORF features (*txCdsPredict score* and *cdsSizes*) have a positive correlation, whereas the k-mers do not present any correlation. Therefore, we now looked only for the best-performing methods (M1-GA, M2-EA, M3-ABC, and M4-ACO) with all features from each one of them (see Figure 4 and Table 14 - 19 attributes). We trained a new model with REPTree algorithm, in which it obtained an ACC: 92.76% and ER: 7.24%. After, we applied tests, in which it obtained SE: 99.87%, SPC: 82.63% and ACC: 91.25% (see Table 17).

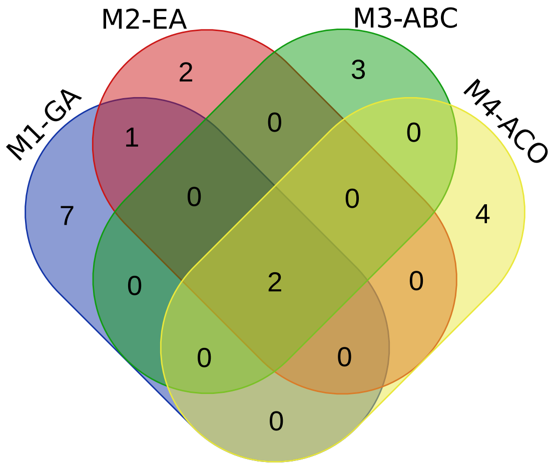


Figure 4 – Union of the best models. M1-GA: 8 features of k-mer and 2 of ORF; M2-EA: 3 features of k-mer and 2 of ORF; M4-ABC: 3 features of k-mer and 2 of ORF; M6-ACO: 4 features of k-mer and 2 of ORF.

Source – Elaborated by the author.

Table 17 – The performance with union of the best models - See Figure 4.

Species	TP	FP	TN	FN	SE	SPC	ACC
Amborella trichopoda	3822	878	2945	1	99.97	77.03	88.50
Brachypodium distachyon	4859	719	4149	9	99.82	85.23	92.52
Citrus sinensis	2290	272	2020	2	99.91	88.13	94.02
Manihot esculenta	3012	404	2613	5	99.83	86.61	93.22
Ricinus communis	4079	755	3325	1	99.98	81.50	90.74
Solanum tuberosum	5596	1346	4261	11	99.80	75.99	87.90
Sorghum bicolor	4527	681	3860	14	99.69	85.00	92.35
Zea mays	12061	2229	9842	10	99.92	81.53	90.73
Overall Average	**-**	**-**	**-**	**-**	**99.87**	**82.63**	**91.25**

Source – Elaborated by the author.

Then, we apply the same idea (training and testing) only using these two common features (*txCdsPredict score* and *cdsSizes* - see Table 18). Thereby, It is possible to notice that with only two features, the model obtained a similar result to the best method (M1-GA), with SE: 99.96%, SPC: 82.59%, and ACC: 91.27%. These two analyses demonstrated that ORF features are strongest in common for all methods, besides presenting a great classification in all sets. To confirm this hypothesis, we decided to apply new experiments looking for other features, as explained in the next section.

Table 18 – Feature Intersecting Performance - See Figure 4.

Species	TP	FP	TN	FN	SE	SPC	ACC
Amborella trichopoda	3821	879	2944	2	99.95	77.01	88.48
Brachypodium distachyon	4867	720	4148	1	99.98	85.21	92.59
Citrus sinensis	2292	272	2020	0	100	88.13	94.07
Manihot esculenta	3017	405	2612	0	100	86.58	93.29
Ricinus communis	4075	755	3325	5	99.88	81.50	90.69
Solanum tuberosum	5606	1352	4255	1	99.98	75.89	87.93
Sorghum bicolor	4536	684	3857	5	99.89	84.94	92.41
Zea mays	12068	2235	9836	3	99.98	81.48	90.73
Overall Average	**-**	**-**	**-**	**-**	**99.96**	**82.59**	**91.27**

Source – Elaborated by the author.

3.2.7 Analysis of ORF Features

Our next investigation was to remove the two strong ORF attributes (*txCdsPredict score* and *cdsSizes*). The hypothesis was to confirm if these attributes are robust enough for proper classification, and maybe to find for new efficient features. The new dataset contained 5,465 features (GC content (1 feature), k-mer (1-6 k-mer length = 5,460 features), Sequence length (1 feature), and ORF metrics (3 features)). Therefore, we re-apply five rounds of optimal feature subsets selection with the best-performing methods (top three in this experiment), including M1-GA, M3-ABC, and M4-ACO. The process followed the same methodology, as illustrated in Figure 1. The results can be seen in Table 19.

Table 19 – Optimal features subsets selected by the experiments with methods M1-GA, M3-ABC, and M4-ACO.

M1-GA	**M3-ABC**	**M4-ACO**
CCAGG, TCTGC CATCAA, CTCATG CTGCAG, GAAGGA CCGGGG, GGAATT GGACCC, cdsStop cdsPercent	GGGTCG cdsStop cdsPercent	CCGGA, CCTGG GTTGC, TGCGG AAGGCC, ACCTCC ACGGAG, ACTGGG AGAGCT, AGCTGG ATCTGG, CAAGGA CAGAGT, CTTGAC GACAGC, GAGGGG GGGTGC, GGTTAT TGCTGC, TGGGCT GCTGTT, GCTCTG GCCTTC, GATGAG TTCTGG, cdsStop cdsPercent

Source – Elaborated by the author.

Essentially, M1-GA method selected 11 features (9 k-mer, 2 ORF), M3-ABC 3 (1 k-mer, 2 ORF), and M4-ACO 27 (25 k-mer, 2 ORF). Again, two ORF features ($cdsStop, cdsPercent$) presented intersection between all the models, as shown in Figure 5. For a better analysis, Table 21 demonstrates performance data with the new features. Essentially, the new models also presented an excellent result when compared to the first tests (see Table 16), in which M1-GA again obtained the best performance in overall average (ACC: 89.03%, SE: 97.74% and SPC: 80.31%), followed by M3-ABC (ACC: 88.76%, SE: 97.43% and SPC: 80.10%) and M4-ACO (ACC: 88.76%, SE: 97.50% and SPC: 80.03%).

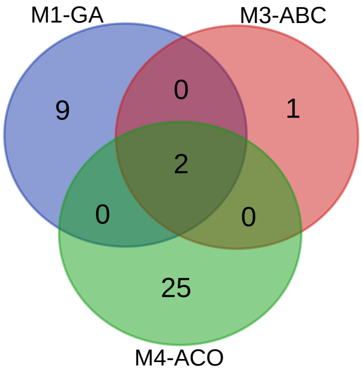


Figure 5 – Union of cardinalities. This figure shows the union of the selected features by the new experiments (M1-GA, M3-ABC and M4-ACO).

Source – Elaborated by the author.

It should be noted that if we compare the best features in the two tests, the loss in accuracy is approximately 2.26%. Therefore, there are other features with efficacy in the classification. Nevertheless, again, beyond to k-mers, the methods choose other ORF features, which indicates the high efficiency of these attributes. Thus, for better analysis, we plotted again a scatter chart with the features that presented intersection between all methods in the two experiments (Table 14 and 19), according to Figure 6.

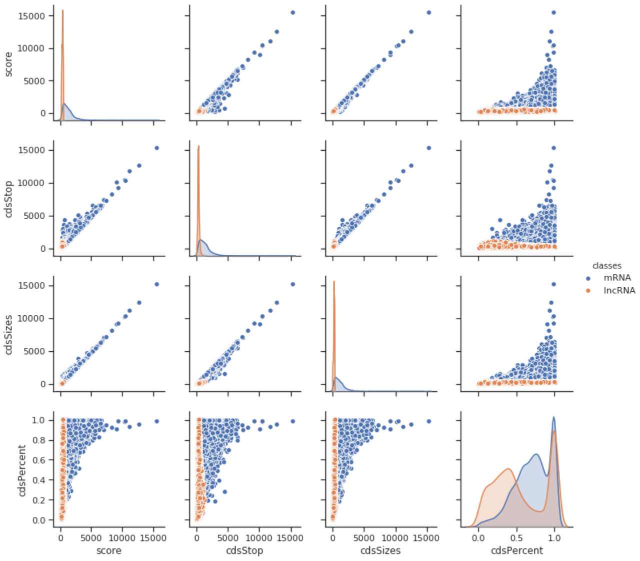


Figure 6 – The scatter chart reports the propagation of training data to features that presented intersection between all methods in the two experiments: (1) txCdsPredict score and cdsSizes; (2) cdsStop and cdsPercent.

Source – Elaborated by the author.

We can observe that most features show a positive correlation, which explains its performance in all tests. Finally, to verify if only the previously mentioned attributes have high efficiency, we performed the last experiments with the k-mer descriptor.

3.2.8 k-mer Descriptor Analysis

In our preliminary analysis, we have noticed the high frequency of the ORF descriptor in the experiments. For that reason, we have proposed a different analysis exploring the effect of k-mers on predictive performance. In that case, we have induced the REPTree algorithm into three different feature sets (1: k-mer (1-6 (5,460 features)), 2: k-mer + ORF (all features), 3: Only ORF (*txCdsPredict score* and *cdsSizes*)). Thus, the induced models were applied to the test sets (see Table 20), where we assessed the ACC and SE (to classify lncRNAs).

The final classification did not present robust results with only k-mer features (SE: 77.73% and ACC: 75.15%) when compared with k-mer + ORF (SE: 88.38% and ACC:

Table 20 – Comparative Performance between three models, k-mer (1-6), k-mer + ORF, only ORF.

Species	Model	SE	ACC
A. trichopoda	K-mer (1-6)	96.34	76.31
	K-mer (1-6) + ORF	89.88	84.51
	Only ORF	**99.95**	**88.48**
B. distachyon	K-mer (1-6)	51.50	69.35
	K-mer (1-6) + ORF	85.62	86.68
	Only ORF	**99.98**	**92.59**
C. sinensis	K-mer (1-6)	86.82	80.78
	K-mer (1-6) + ORF	89.18	89.22
	Only ORF	**100**	**94.07**
M. esculenta	K-mer (1-6)	81.01	78.55
	K-mer (1-6) + ORF	89.56	88.96
	Only ORF	**100**	**93.29**
R. communis	K-mer (1-6)	95.00	76.96
	K-mer (1-6) + ORF	91.05	87.11
	Only ORF	**99.88**	**90.69**
S. tuberosum	K-mer (1-6)	84.48	70.93
	K-mer (1-6) + ORF	88.41	83.90
	Only ORF	**99.98**	**87.93**
S. bicolor	K-mer (1-6)	52.85	69.98
	K-mer (1-6) + ORF	85.16	86.07
	Only ORF	**99.89**	**92.41**
Z. mays	K-mer (1-6)	73.81	78.31
	K-mer (1-6) + ORF	88.21	86.40
	Only ORF	**99.98**	**90.73**
Overall Average	K-mer (1-6)	77.73	75.15
	K-mer (1-6) + ORF	88.38	86.61
	Only ORF	**99.96**	**91.27**

Source – Elaborated by the author.

86.61%) and especially when compared with ORF (SE: 99.96% and ACC: 91.27%). These additional experiments again support the efficiency of the ORF features.

3.2.9 Evaluation against other classifier tools

Lastly, the several experiments performed in this work pointed to the great efficiency of the ORF descriptor. Thus, in a final analysis, we compare the two best features (*txCdsPredict score* and *cdsSizes*) against other five state-of-the-art programs: RNAplonc (NEGRI et al., 2018) (specifically for plants - 16 features), CPC (KONG et al., 2007) (Multi-Species - 6 features), CPC2 (KANG et al., 2017) (Multi-Species - 4 features), CNCI (SUN et al., 2013) (Animals and Plants - 5 features), PLEK (LI; ZHANG; ZHOU, 2014) (Multi-Species - $1,364$ features), as shown in Table 22.

Table 21 – Performance of the selected features subsets by the new experiments M1-GA, M3-ABC, and M4-ACO (see Table 19). Recalling that the attributes *txCdsPredict score* and *cdsSizes* were removed, in order to find new relevant features.

Species	Tools - ID	TP	FP	TN	FN	SE	SPC	ACC	F1-score	PPV	NPV	MCC
A. trichopoda	M1-GA (11)	3812	875	2948	11	**99.71**	77.11	88.41	89.59	81.33	**99.63**	78.86
	M3-ABC (3)	3811	873	2950	12	99.69	**77.16**	**88.43**	**89.60**	**81.36**	99.59	78.88
	M4-ACO (27)	3812	874	2949	11	**99.71**	77.14	**88.43**	**89.60**	81.35	**99.63**	**78.89**
B. distachyon	M1-GA (11)	4694	875	3993	174	**96.43**	**82.03**	**89.23**	**89.95**	**84.29**	**95.82**	**79.28**
	M3-ABC (3)	4668	883	3985	200	95.89	81.86	88.88	89.61	84.09	95.22	78.53
	M4-ACO (27)	4669	902	3966	199	95.91	81.47	88.69	89.45	83.81	95.22	78.20
C. sinensis	M1-GA (11)	2227	348	1944	65	**97.16**	84.82	**90.99**	**91.51**	**86.49**	**96.76**	**82.61**
	M3-ABC (3)	2212	348	1944	80	96.51	84.82	90.66	91.18	86.41	96.05	81.89
	M4-ACO (27)	2220	347	1945	72	96.86	**84.86**	90.86	91.38	86.48	96.43	82.31
M. esculenta	M1-GA (11)	2932	435	2582	85	**97.18**	**85.58**	**91.38**	**91.85**	**87.08**	**96.81**	**83.33**
	M3-ABC (3)	2919	438	2579	98	96.75	85.48	91.12	91.59	86.95	96.34	82.76
	M4-ACO (27)	2924	449	2568	93	96.92	85.12	91.02	91.52	86.69	96.51	82.61
R. communis	M1-GA (11)	4060	771	3309	20	**99.51**	**81.10**	**90.31**	**91.12**	**84.04**	**99.40**	**82.01**
	M3-ABC (3)	4055	771	3309	25	99.39	**81.10**	90.25	91.06	84.02	99.25	81.87
	M4-ACO (27)	4054	771	3309	26	99.36	**81.10**	90.23	91.05	84.02	99.22	81.84
S. tuberosum	M1-GA (11)	5496	1526	4081	111	**98.02**	**72.78**	**85.40**	**87.04**	**78.27**	**97.35**	**73.17**
	M3-ABC (3)	5486	1564	4043	121	97.84	72.11	84.97	86.69	77.82	97.09	72.39
	M4-ACO (27)	5478	1564	4043	129	97.70	72.11	84.90	86.62	77.79	96.91	72.21
S. bicolor	M1-GA (11)	4378	842	3699	163	**96.41**	**81.46**	**88.93**	**89.70**	**83.87**	**95.78**	**78.75**
	M3-ABC (3)	4369	864	3677	172	96.21	80.97	88.59	89.40	83.49	95.53	78.10

	M4-ACO (27)	4366	852	3689	175	96.15	81.24	88.69	89.48	83.67	95.47	78.26
Z. mays	M1-GA (11)	11773	2702	9369	298	**97.53**	**77.62**	**87.57**	**88.70**	**81.33**	**96.92**	**76.68**
	M3-ABC (3)	11725	2739	9332	346	97.13	77.31	87.22	88.37	81.06	96.42	75.95
	M4-ACO (27)	11754	2755	9316	317	97.37	77.18	87.28	88.44	81.01	96.71	76.12
Overall Average	**M1-GA (11)**	-	-	-	-	**97.74**	**80.31**	**89.03**	**89.93**	**83.34**	**97.31**	**79.34**
	M3-ABC (3)	-	-	-	-	97.43	80.10	88.76	89.69	83.15	96.94	78.80
	M4-ACO (27)	-	-	-	-	97.50	80.03	88.76	89.69	83.10	97.01	78.81

Source – Elaborated by the author.

Table 22 – Comparative performance between our best features (*txCdsPredict score* and *cdsSizes*), RNAplonc, CPC, CPC2, CNCI, and PLEK for five plant species.

Species	Technique - ID	TP	FP	TN	FN	SE	SPC	ACC	F1-score	PPV	NPV	MCC
A. trichopoda	**Best Features**	3821	879	2944	2	99.95	77.01	88.48	89.66	81.30	99.93	79.06
	RNAplonc	3823	879	2944	0	**100**	77.01	**88.50**	**89.69**	**81.31**	**100**	**79.13**
	CPC	3823	1877	1946	0	**100**	50.90	75.45	80.29	67.07	100	58.43
	CPC2	2513	618	3205	1310	65.73	**83.83**	74.78	72.27	80.26	70.99	50.40
	CNCI	2665	1171	2651	1158	69.71	69.36	69.54	69.59	69.47	69.60	39.07
	PLEK	3823	2857	966	0	**100**	25.27	62.63	72.80	57.23	100.00	38.03
B. distachyon	**Best Features**	4867	720	4148	1	**99.98**	85.21	**92.59**	**93.10**	87.11	**99.88**	**86.13**
	RNAplonc	4753	677	4191	115	97.64	86.09	91.87	92.31	**87.53**	97.33	84.29
	CPC	4846	1685	3183	22	99.55	65.39	82.47	85.03	74.20	99.31	69.09
	CPC2	4312	666	4202	556	88.58	86.32	87.45	87.59	86.62	88.31	74.92
	CNCI	2571	426	4442	2297	52.81	**91.25**	72.03	65.38	85.79	65.91	47.73
	PLEK	4082	1449	3419	786	83.85	70.23	77.04	78.51	73.80	81.31	54.60
C. sinensis	**Best Features**	2292	272	2020	0	**100**	88.13	94.07	94.40	89.39	**100**	88.76
	RNAplonc	2290	267	2025	2	99.91	88.35	**94.13**	**94.45**	**89.56**	99.90	**88.86**
	CPC	2268	746	1546	24	98.95	67.45	83.20	85.49	75.25	98.47	69.97
	CPC2	1889	231	2061	403	82.42	**89.92**	86.17	85.63	89.10	83.64	72.54
	CNCI	1765	485	1807	527	77.01	78.84	77.92	77.72	78.44	77.42	55.86
	PLEK	2172	827	1465	120	94.76	63.92	79.34	82.10	72.42	92.43	61.69
M. esculenta	**Best Features**	3017	405	2612	0	**100**	86.58	**93.29**	**93.71**	88.16	**100**	**87.37**
	RNAplonc	3014	403	2614	3	99.90	86.64	93.27	93.69	88.21	99.89	87.31
	CPC	2980	838	2179	37	98.77	72.22	85.50	87.20	78.05	98.33	73.64

	CPC2	2645	332	2685	372	87.67	**89.00**	88.33	88.25	**88.85**	87.83	76.67
	CNCI	2580	786	2231	437	85.52	73.95	79.73	80.84	76.65	83.62	59.86
	PLEK	2849	1153	1864	168	94.43	61.78	78.11	81.18	71.19	91.73	59.47
S. bicolor	**Best Features**	4536	684	3857	5	**99.89**	84.94	**92.41**	**92.94**	86.90	**99.87**	**85.79**
	RNAplonc	4375	629	3912	166	96.34	86.15	91.25	91.67	**87.43**	95.93	82.93
	CPC	4511	1481	3060	30	99.34	67.39	83.36	85.65	75.28	99.03	70.42
	CPC2	4025	597	3944	516	88.64	86.85	87.74	87.85	87.08	88.43	75.50
	CNCI	2317	383	4158	2224	51.02	**91.57**	71.29	64.00	85.81	65.15	46.59
	PLEK	3626	1225	3316	915	79.85	73.02	76.44	77.21	74.75	78.37	53.00

Source – Elaborated by the author.

We randomly chose 5 species for evaluation. In which, our model (with two features) reported the best performance of ACC in three species, *B. distachyon* = 92.59% (M2-ACO), *M. esculenta* = 93.29% (M1-EA), and *S. bicolor* = 92.41% (M2-ACO). In contrast, the RNAplonc tool obtained the best ACC in *A. trichopoda* (88.50%) and *C. sinensis* (94.13%). However, with only a difference of 0.02% and 0.06% of our model, respectively. Moreover, we use 14 features unless RNAplonc in this experiment. In relation to SE (to predict lncRNAs), our models were the best in four species, with the exception of *A. trichopoda*, in which three tools also achieved the best result (RNAplonc, CPC, PLEK). However, in SPC (to predict mRNA), the best tool was CPC2 (*A. trichopoda*, *C. sinensis*, *M. esculenta*) and CNCI (*B. distachyon*, *S. bicolor*). In the overall average, our model had an ACC of 92.17% across all datasets, that is, 0.37%, 10.17%, 7.28%, 18.07%, and 17.46% more than RNAplonc (91.80%), CPC (82.00%), CPC2 (84.89%), CNCI (74.10%), and PLEK (74.71%), respectively. Finally, these results indicate the great efficiency of the ORF descriptor, especially the features *txCdsPredict score* and *cdsSizes*.

4 EXPERIMENTAL TEST II: FEATURE EXTRACTION PROBLEM

Regarding feature extraction in lncRNAs (see Chapter 2), we observed a full domain of ORF features and sequence structure. Despite a large number of new approaches, with excellent results for the problem in question, there is the frequent use of biological features (e.g., ORF, GC content, alignment, among others). As presented in the previous chapter, the ORF descriptor, after several experiments, proved to be the most efficient attribute for lncRNAs classification. Thus, this chapter is dedicated to analyzing mathematical models for feature extraction in order to propose efficient and generalist techniques for biological sequence analysis problems.

In our case study (lncRNAs), some works have explored mathematical models for feature extraction, such as Genomic Signal Processing (GSP) and DNA Numerical Representation (DNR) (PIAN et al., 2016; HAN et al., 2018) techniques and Complex Networks (ITO et al., 2018). Nevertheless, the authors used these approaches in conjunction with other features (e.g., ORF, GC content, alignment, among others) or without testing other mathematical models. Therefore, at this stage, as a starting point, nine mathematical models for feature extraction will be analyzed: six numerical mapping techniques with Fourier transform; Tsallis and Shannon entropy; Graphs (complex networks). Fundamentally, this chapter elaborates experiments to answer the hypotheses presented in Section 1.4 (H3-PB2, H4-PB2, and H5-PB2), divided into two parts: Experimental Methodology and Results. Discussions will be presented in Section 5.

4.1 Experimental Methodology

We divided our approach into five stages, as shown in Figure 7: (1) Data selection and preprocessing; (2) Feature extraction; (3) Training; (4) Test; (5) Performance analysis.

4.1.1 Data Selection

As previously mentioned, our central hypothesis is to demonstrate the efficiency of mathematical models in biological sequence classification problems using the same pipeline (see Figure 7). For this, we chose as a case study the lncRNAs classification problem, which is much addressed in the literature. However, we will also use other datasets to evaluate the generalization of mathematical models. Therefore, we divided this chapter into two case studies.

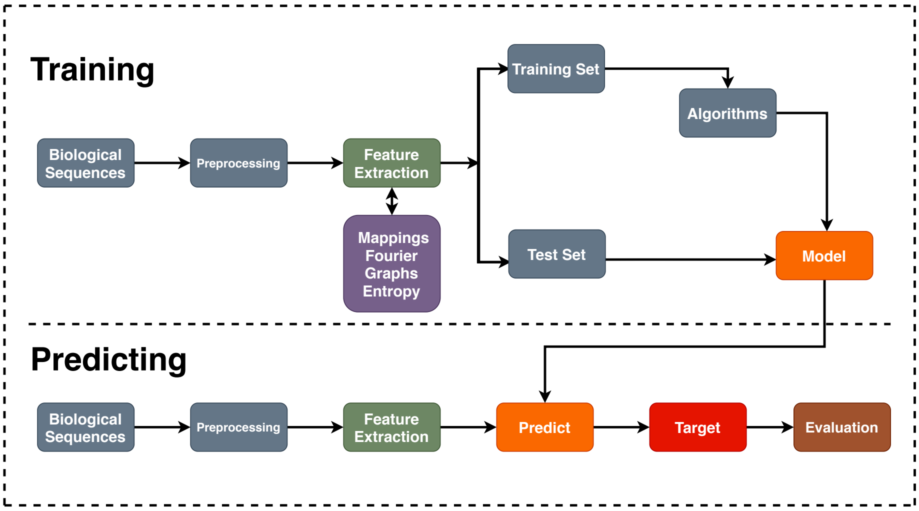


Figure 7 – Proposed Pipeline for the Feature Extraction Problem. Essentially, (1) datasets are preprocessed; (2) Feature extraction techniques are applied to each dataset; (3) Machine learning algorithms are applied to the training set to induce predictive models; (4) Induced models are applied to the test set; Finally, (5) the models are evaluated.

Source – Elaborated by the author.

Case Study I

This experiment is our main approach. Thus, sequences of five plant species were adopted in order to assess the proposed method. The summary of the dataset construction can be seen in Table 23. Following the literature methods, this work also adopts two classes for the datasets: positive class, with lncRNAs, and negative class, with protein-coding genes (mRNAs).

Table 23 – Species used to create the training set.

Species	Sequences	Amount	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

Source – Elaborated by the author.

The mRNA data of the *Arabidopsis thaliana* (obtained from CPC2 (KANG et al., 2017)) were built from the `RefSeq database` with protein sequences annotated by Swiss-Prot (KANG et al., 2017), and lncRNA data from the `Ensembl` (*v*87) 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) (GOODSTEIN et al., 2011). The lncRNAs data from these species were extracted from `GreeNC` (version 1.12) (GALLART et al., 2015). Basically, as preprocessing, we used only sequences longer than $200nt$ (LI; ZHANG; ZHOU, 2014), and we also removed sequence redundancy. Moreover, the sampling method was adopted in our dataset, since we are faced with the *imbalanced data problem*. Thus, we applied random majority under-sampling, which consists of removing samples from the majority class (to adjust the class distribution) (LIU, 2004). Finally, we follow the same prepossessing of the Experimental Test I, but we changed the sequences, in order to generate a different dataset.

Case Study II

In this second case study, we will apply the best mathematical models of the case study I to different classification problems with lncRNAs in order to test their generalization. Thus, we divided into four problems:

- **Problem 1** (lncRNA vs. sncRNA): Dataset with only non-coding sequences (lncRNA and Small non-coding RNAs (sncRNAs), also obtained from CPC2 (KANG et al., 2017))
    - lncRNA: 1291 sequences
    - sncRNA: 1291 sequences
- **Problem 2** (mRNA vs. sncRNA): Dataset with mRNA and sncRNA sequences (sncRNA obtained from CPC2 (KANG et al., 2017)). This problem was proposed based on Kang et al. (2017) and (ITO et al., 2018).
    - mRNA: 1291 sequences
    - sncRNA: 1291 sequences
- **Problem 3** (Antisense vs. lncRNA): Dataset with lncRNAs and long noncoding antisense transcripts (obtained from Chen et al. (2011)).
    - lncRNA: 57 sequences
    - Antisense: 57 sequences
- **Problem 4** (circRNA vs. lncRNA): Dataset with lncRNA and circular RNAs (cirRNAs) sequences (circRNA obtained from PlantcircBase (CHU et al., 2017). This problem was proposed based on Pan and Xiong (2015) and Chen et al. (2018), in order to classify circRNA from other lncRNAs.
    - circRNA: 2540 sequences

– lncRNA: 2540 sequences

It is important to emphasize that we used in this second case study only sequences from *Arabidopsis thaliana* because it is the best-noted model species.

4.1.2 Feature Extraction

According to (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015), the feature extraction seeks to generate a feature vector by optimally transforming the input data. This procedure is extremely relevant to the success of the machine learning application. Another major goal of feature extraction is to extract important features from the input data, as well as remove noise and redundancy (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015; GUYON et al., 2008). Considering this, the feature extraction methods are shown, in which nine mathematical models will be analyzed: six numerical mapping techniques with Fourier transform (see Section 4.1.3), Tsallis and Shannon entropy (see Section 4.1.4), Complex Networks (see Section 4.1.5). Nevertheless, it is necessary to emphasize that we denote a biological sequence $S = (S[0], S[1], \ldots, S[N-1])$ such that $S \in \{A, C, G, T\}^N$.

4.1.3 Fourier Transform and Numerical Mappings

To generate features based in a Fourier approach, we apply the Discrete Fourier Transform (DFT), widely used for digital signal processing (here GSP), that can reveal hidden periodicities after the transformation of time domain data to frequency domain (YIN; CHEN; YAU, 2014). According to Yin and Yau (YIN; YAU, 2005), the DFT of a signal with length N, $x[n]$ ($n = 0, 1, \ldots, N-1$), at frequency k, can be defined by Equation (4.1):

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, 1, \ldots, N-1. \tag{4.1}$$

This method is extensively studied in bioinformatics, mainly for analysis of periodicities and repetitive elements in DNA sequences (ANASTASSIOU, 2001) and protein structures (MARSELLA et al., 2009). This approach is shown in Figure 8 and was based on Bonidia et al. (2019).

To calculate DFT, we use the Fast Fourier Transform (FFT), which is a highly efficient procedure for computing the DFT of a time series (COCHRAN et al., 1967). However, to use GSP techniques, it is necessary to apply a numeric representation for the transformation or mapping of genomic data. In literature, distinct DNR techniques have been developed (ABO-ZAHHAD; AHMED; ABD-ELRAHMAN, 2012). According to Mendizabal-Ruiz et al. (MENDIZABAL-RUIZ et al., 2017), these representations can be divided into three categories: single-value mapping, multidimensional sequence

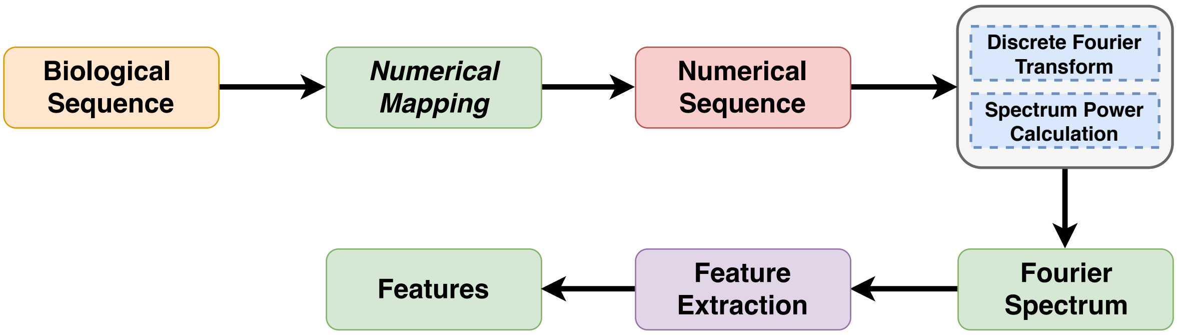


Figure 8 – Fourier Transform and Numerical Mapping Pipeline. (1) Each sequence is mapped to a numerical sequence; (2) DFT is applied to the generated sequence; (3) The spectrum power is calculated; (4) The feature extraction is performed; Finally, (5) the features are generated.

Source – Elaborated by the author.

mapping, and cumulative sequence mapping. Therefore, we study 6 numerical mapping techniques (or representations), which will be presented below: Voss (VOSS, 1992), Integer (MENDIZABAL-RUIZ et al., 2017; CRISTEA, 2002), Real (CHAKRAVARTHY et al., 2004), Z-curve (ZHANG; ZHANG, 1994), EIIP (NAIR; SREENADHAN, 2006) and Complex Numbers (ABO-ZAHHAD; AHMED; ABD-ELRAHMAN, 2012; Anastassiou, 2001; YU; LI; YU, 2018).

Voss Representation

This representation can use single or multidimensional vectors. Fundamentally, this approach transforms a sequence $S \in \{A,\ C,\ G,\ T\}^N$ 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:

$$v_i[n] = \begin{cases} 1, & S[n] = \alpha[i] \\ 0, & S[n] \neq \alpha[i] \end{cases}, \text{ where } \alpha = (A,\ C,\ G,\ T), \qquad n = 0, 1, \ldots, N-1. \tag{4.2}$$

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 $S = (G, A, G, A, G, T, G, A, C, C, A)$ 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 4.3):

$$V_i[k] = \sum_{n=0}^{N-1} v_i[n] e^{-j\frac{2\pi}{N}kn}, \ \forall\ i \in [1, 4], \qquad k = 0, 1, \ldots, N-1. \tag{4.3}$$

The power spectrum of a biological sequence can be obtained by Equation (4.4):

$$P_V[k] = \sum_{i=1}^{4} |V_i[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{4.4}$$

Integer Representation

This representation is one-dimensional (CRISTEA, 2002; MENDIZABAL-RUIZ et al., 2017). 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 S = (G, A, G, A, G, T, G, A, C, C, A), thus, d = (3, 2, 3, 2, 3, 0, 3, 2, 1, 1, 2), as exposed in Equation (4.5). The DFT and power spectrum are exposed in Equation (4.6).

$$d[n] = \begin{cases} 3, & S[n] = G \\ 2, & S[n] = A \\ 1, & S[n] = C \\ 0, & S[n] = T \end{cases}, \qquad n = 0, 1, \ldots, N-1. \tag{4.5}$$

$$D[k] = \sum_{n=0}^{N-1} d[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_D[k] = |D[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{4.6}$$

Real Representation

In this representation, Chakravarthy et al. (CHAKRAVARTHY et al., 2004) use real mapping based on the complement property of the complex mapping of (ANASTASSIOU, 2001). This mapping applies negative decimal values for the purines (A, G), and positive decimal values for the pyrimidines (C, T), e.g., let $S = (G, A, G, A, G, T, G, A, C, C, A)$, thus, 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 (4.7) and Equation (4.8).

$$r[n] = \begin{cases} -0.5, & S[n] = G \\ -1.5, & S[n] = A \\ 0.5, & S[n] = C \\ 1.5, & S[n] = T \end{cases}, \qquad n = 0, 1, \ldots, N-1. \tag{4.7}$$

$$R[k] = \sum_{n=0}^{N-1} r[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_R[k] = |R[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{4.8}$$

Z-curve Representation

The Z-curve scheme is a three-dimensional curve presented by (ZHANG; ZHANG, 1994), 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 ($n = 1, 2, \dots, N$). 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 (SHAO; YAN; SHAO, 2013). Therefore:

$$A_n + C_n + G_n + T_n = n \tag{4.9}$$

Where the Z-curve consists of a series of nodes $P_1, P_2, \dots, P_N$, whose coordinates $x[n]$, $y[n]$, and $z[n]$ ($n = 1, 2, \dots, N$) are uniquely determined by the Z-transform, shown in Equation (4.10):

$$P[n] = \begin{cases} x[n] = (A_n + G_n) - (C_n + T_n) \equiv R_n - Y_n \\ y[n] = (A_n + C_n) - (G_n + T_n) \equiv M_n - K_n, \\ z[n] = (A_n + T_n) - (C_n + G_n) \equiv W_n - S_n \end{cases} \\ x[n], y[n], z[n] \in [-n, n], \qquad n = 1, 2, \dots, N. \tag{4.10}$$

Where R, Y, M, K , W and S denote the bases of purine ($R = A, G$), pyrimidine ($Y = C, T$), amino ($M = A, C$), keto ($K = G, T$), weak hydrogen bonds ($W = A, T$) and strong hydrogen bonds ($S = G, C$), respectively (SHAO; YAN; SHAO, 2013; ZHANG, 1997). The coordinates $x[n]$, $y[n]$, and $z[n]$ represent three independent distributions that completely describe a sequence (ABO-ZAHHAD; AHMED; ABD-ELRAHMAN, 2012). Therefore, we will have three distributions with definite biological significance: (1) $x[n]$ = purine/pyrimidine, (2) $y[n]$ = amino/keto, (3) $z[n]$ = strong hydrogen bonds/weak hydrogen bonds (ZHANG; ZHANG, 1994), e.g., let S = (G, A, G, A, G, T, G, A, C, C, A), thus, $x = (1, 2, 3, 4, 5, 4, 5, 6, 5, 4, 5)$; y = (-1, 0, -1, 0, -1, -2, -3, -2, -1, 0, 1); $z = (-1, 0, -1, 0, -1, 0, -1, 0, -1, -2, -1)$. Essentially, the difference between each dimension at the n-th position and the previous $(n - 1)$ position can be either 1 or -1 (ZHANG; ZHANG, 1994). Therefore, we may define the following set of equations in order to update the values of each dimension array considering that $x[-1] = y[-1] = z[-1] = 0$:

$$x[n] = \begin{cases} x[n-1] + 1, & S[n] = A \text{ or } G \\ x[n-1] - 1, & S[n] = C \text{ or } T \end{cases}. \tag{4.11}$$

$$y[n] = \begin{cases} y[n-1]+1, & S[n] = A \text{ or } C \\ y[n-1]-1, & S[n] = G \text{ or } T \end{cases}, \qquad n = 1, 2, \ldots, N. \tag{4.12}$$

$$z[n] = \begin{cases} z[n-1]+1, & S[n] = A \text{ or } T \\ z[n-1]-1, & S[n] = G \text{ or } C \end{cases}. \tag{4.13}$$

Finally, the DFT and the power spectrum of the Z-Curve representation may be defined as (ZHANG, 1997):

$$X[k] = \sum_{n=1}^{N} x[n] e^{-j\frac{2\pi}{N}kn}, \quad Y[k] = \sum_{n=1}^{N} y[n] e^{-j\frac{2\pi}{N}kn}, \quad Z[k] = \sum_{n=1}^{N} z[n] e^{-j\frac{2\pi}{N}kn}. \tag{4.14}$$

$$P_C[k] = |X[k]|^2 + |Y[k]|^2 + |Z[k]|^2, \qquad k = 1, 2, \ldots, N. \tag{4.15}$$

EIIP Representation

Nair and Sreenadhan (NAIR; SREENADHAN, 2006) 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 S = (G, A, G, A, G, T, G, A, C, C, A), thus, 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 (4.16). The DFT and power spectrum of this representation are presented in Equation (4.17).

$$b[n] = \begin{cases} 0.0806, & S[n] = G \\ 0.1260, & S[n] = A \\ 0.1340, & S[n] = C \\ 0.1335, & S[n] = T \end{cases}, \qquad n = 0, 1, \ldots, N-1. \tag{4.16}$$

$$E[k] = \sum_{n=0}^{N-1} b[n] e^{-j\frac{2\pi}{N}kn}, \qquad P_E[k] = |E[k]|^2, \qquad k = 0, 1, \ldots, N-1. \tag{4.17}$$

Complex Number Representation

This numerical mapping has the advantage of better translating some of the nucleotides features into mathematical properties (YU; LI; YU, 2018) and represents the complementary nature of AT and CG pairs (ABO-ZAHHAD; AHMED; ABD-ELRAHMAN, 2012); e.g., let S = (G, A, G, A, G, T, G, A, C, C, A), thus, $cr = (-1-j,\ 1+j,\ -1-j,$

$1+j$, $-1-j$, $1-j$, $-1-j$, $1+j$, $-1+j$, $-1+j$, $1+j$), as shown in Equation (4.18). The DFT and power spectrum of this representation are presented in Equation (4.19).

$$cr[n] = \begin{cases} -1-j, & S[n]=G \\ 1+j, & S[n]=A \\ -1+j, & S[n]=C \\ 1-j, & S[n]=T \end{cases}, \qquad n = 0,1,\dots,N-1. \tag{4.18}$$

$$CR[k] = \sum_{n=0}^{N-1} b[n]e^{-j\frac{2\pi}{N}kn}, \qquad P_{CR}[k] = |CR[k]|^2, \qquad k = 0,1,\dots,N-1. \tag{4.19}$$

Features

Finally, we apply the feature extraction in each representation with Fourier, adopting Signal to Noise Ratio (SNR) (SHAO; YAN; SHAO, 2013), 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. The SNR uses the statistical phenomenon known as period-3 behavior or 3-base periodicity (YIN; YAU, 2007). Therefore, let $\bar{E}$ denote the average, then (see Equation (4.20)):

$$\bar{E} = \frac{1}{N}\sum_{k=0}^{N-1} P[k], \qquad k = 0,1,\dots,N-1. \tag{4.20}$$

$$SNR = \frac{P(\frac{N}{3})}{\bar{E}}. \tag{4.21}$$

Several studies have demonstrated (SHAO; YAN; SHAO, 2013; YIN; YAU, 2007) that there is a peak in the frequency $N/3$ of the Fourier power spectrum in coding sequences, in contrast, this 3-base periodicity does not exist in most non-coding sequences.

4.1.4 Entropy

Information theory has been widely applied in bioinformatics (VINGA, 2013; BARROS-CARVALHO; SLUYS; LOPES, 2017; PRITIŠANAC et al., 2019). Based on this, we consider the study of (MACHADO; COSTA; QUELHAS, 2011), which applied an algorithmic and mathematical approach to DNA code analysis using entropy and phase plane. Fundamentally, according to (VINGA, 2013), entropy is a measure of the uncertainty associated with a probabilistic experiment. Thus, to generate a probabilistic experiment, we use a known method in bioinformatics, the k-mer (our pipeline is shown in Figure 9).

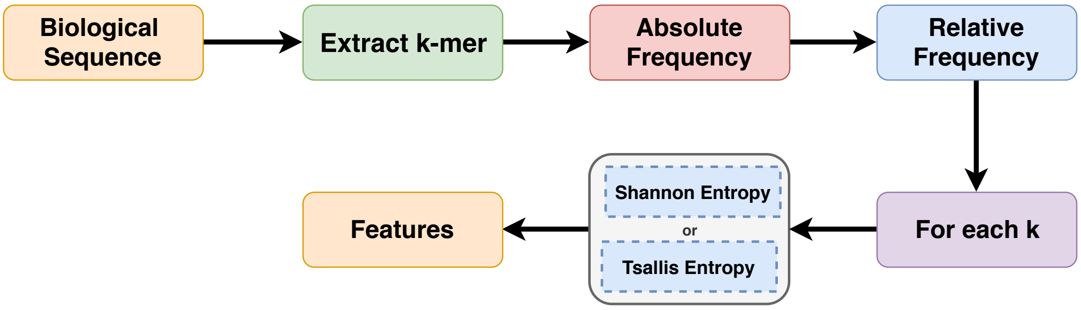


Figure 9 – Entropy Pipeline. (1) Each sequence is mapped in k-mers; (2) The absolute frequency of each k is calculated; (3) Based on absolute frequency, the relative frequency is calculated; (4) A Tsallis or Shannon entropy is applied to each k; Finally, (5) features are generated.

Source – Elaborated by the author.

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 (4.22).

$$P_k(S) = \frac{c_i^k}{N-k+1} = \left(\frac{c_1^1}{N-1+1}, \ldots, \frac{c_4^1}{N-1+1}, \right. \\ \left. \frac{c_{4+1}^2}{N-2+1}, \ldots, \frac{c_i^k}{N-k+1} \right) \qquad k = 1, 2, \ldots, 24. \tag{4.22}$$

This equation is applied 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 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 10 demonstrated an example with $k = 6$ and $k = 9$.

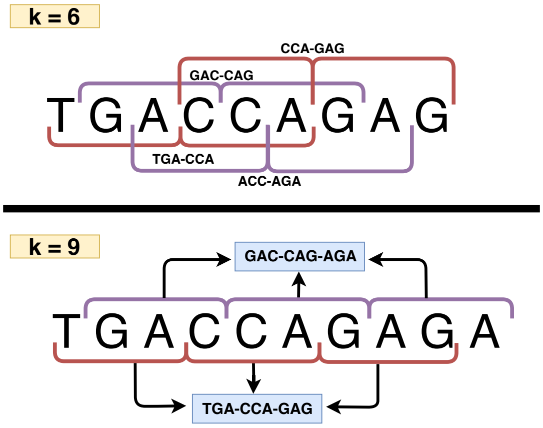


Figure 10 – k-mer Workflow. Example with $k = 6$ and $k = 9$.

Source – Elaborated by the author.

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 $[\{GGGGGGGGGGGG\}, \ldots, \{AAAAAAAAAAAA\}]$, that result for $k = 12$. Where, after counting the absolute frequencies of each k, we generate relative frequencies (see Equation (3.1)), and then apply Shannon and Tsallis entropy to generate the features.

Shannon and Tsallis Entropy

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 4.23) and Tsallis (Equation 4.24) entropy associated with this variable is given by the following expressions:

$$E_{Sh}[k] = -\sum_{n=0}^{N-1} p[n]\ log_2\ p[n] \qquad k = 1, 2, \ldots 24. \tag{4.23}$$

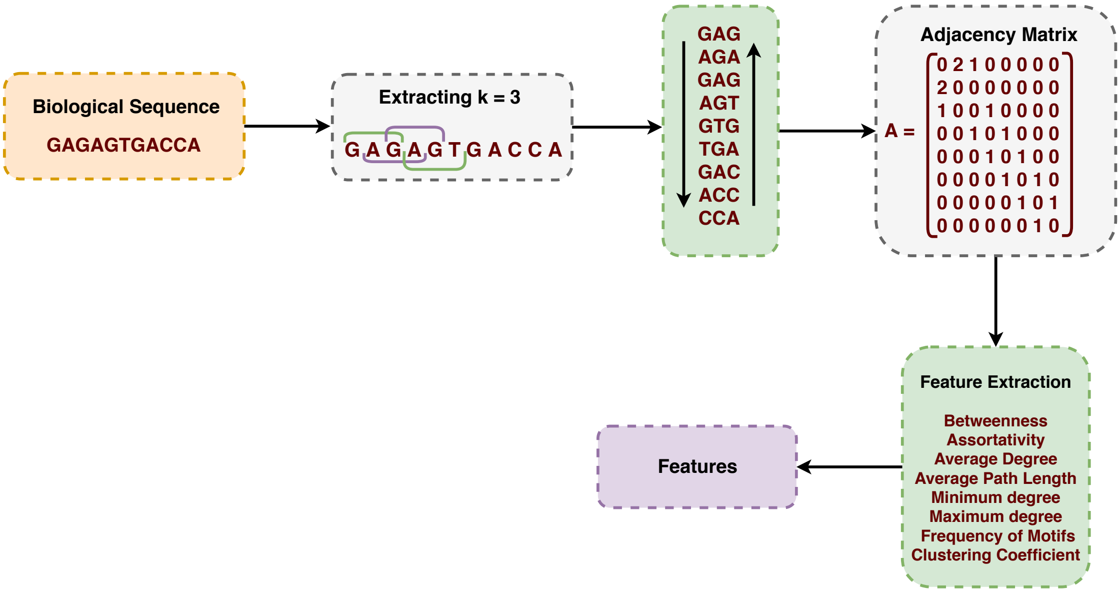
$$E_{Ts}[k] = \frac{1}{q-1}\left(1 - \sum_{n=0}^{N-1} p[n]^q\right) \quad k = 1, 2, \ldots 24. \tag{4.24}$$

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

4.1.5 Complex Networks

Complex networks are widely used in mathematical modeling and have been an extremely active field in recent years (COSTA; RODRIGUES; CRISTINO, 2008), as well as becoming an ideal research area for mathematicians, computer scientists, and biologists. Based on this, we consider the study of Ito et al. (2018), in which, we propose a feature extraction model based on complex networks, as shown in Figure 11. Here, we represent our structure of complex networks by undirected weighted graphs. According to Costa, Rodrigues and Cristino (2008), 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. The elements a_{ij} are equal to 1 whenever there is an edge connecting the vertices i and j, and equal to 0 otherwise. In our case, the graph is undirected, that is, the adjacency matrix A is symmetric, i.e., elements $a_{ij} = a_{ji}$ for any i and j. Finally, like features, several network characterization measures were obtained, based on Wang (2002), 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.



$$A = \begin{bmatrix} 0&2&1&0&0&0&0&0 \\ 2&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0 \\ 0&0&1&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix}$$

Figure 11 – 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) The feature extraction is performed; Finally, (4) the features are generated.

Source – Elaborated by the author.

4.1.6 Normalization, Training and Evaluation Metrics

Data normalization is a preprocessing technique frequently applied to a dataset. Essentially, features can have different dynamic ranges. Thus, features with a larger range which can have a stronger effect in the induction of a predictive model, mainly for distance-based ML algorithms. The application of a normalization procedure makes the ranges similar, reducing this problem (SINGH; VERMA; THOKE, 2015). We used in this work the min-max method, which reduces the data range to 0 and 1 (or -1 to 1, if there are negative values - see Equation (3.4)). Next, we investigate four classification algorithms, like Random Forest (RF) (BREIMAN, 2001), AdaBoost (HASTIE et al., 2009) and CatBoost (DOROGUSH; ERSHOV; GULIN, 2018). To induce our models, we used 70% of samples for *training* (with 10-fold cross-validation) and 30% for *testing*, as exposed in Table 24. The methods were evaluated with four measures: Sensitivity (SE - Equation 2.5), Specificity (SPC - Equation 2.6), Accuracy (ACC - Equation 2.7), and Cohen's kappa coefficient (COHEN, 1960) (Equation 4.25).

$$Kappa = \frac{p_o - p_e}{1 - p_e} \tag{4.25}$$

These measures use True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values, where: TP measures the correctly predicted lncRNAs; TN represents the correctly classified mRNAs; FP describes all those negative entities

Table 24 – Number of sequences used for training and testing in each dataset.

Case Study	Dataset	Amount	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
	mRNA vs. sncRNA	2582	1807	775
	Antisense vs. lncRNA	114	79	35
	circRNA vs. lncRNA	5080	3556	1524

Source – Elaborated by the author.

that are incorrectly classified as lncRNAs and; FN represents the true lncRNAs that are incorrectly classified as mRNAs.

4.2 Results

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

4.2.1 Case Study I

Initially, we induced our 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. Then, to estimate the real accuracy of this set, we used 10-fold cross-validation, as exposed in Table 25.

Table 25 – Real accuracy estimates 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)**

Evaluating each classifier individually, we observed that the best performance was of the CatBoost for 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 continue testing the models with CatBoost classifier. Thus, in Table 26, we present results of all mathematical models using 4 evaluation metrics.

Table 26 – 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
A. thaliana	Z-curve	0.9777	0.9383	0.9580	0.9160
	Binary	0.9619	0.9449	0.9534	0.9068
	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 we can see, 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).

4.2.2 Case Study II

After evaluating all methods in 5 different datasets (lncRNA from different species) and observing their robust 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 four new problems established in Section 4.1.1, as exposed in Table 27.

Table 27 – Performance analysis of three mathematical models, GSP (Fourier + complex numbers), entropy (Tsallis) and graphs (complex networks), for different problems.

lncRNA vs. sncRNA				**mRNA vs. sncRNA**			
Models	**SE**	**SPC**	**ACC**	**Models**	**SE**	**SPC**	**ACC**
GSP	**1.0000**	**1.0000**	**1.0000**	GSP	**1.0000**	**1.0000**	**1.0000**
Entropy	0.9974	0.9974	0.9974	Entropy	**1.0000**	**1.0000**	**1.0000**
Graphs	**1.0000**	**1.0000**	**1.0000**	Graphs	**1.0000**	**1.0000**	**1.0000**

Antisense vs. lncRNA				**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**

Again, all showed excellent results. In which, graph-based models are best in three of the four problems analyzed, followed by entropy and GSP. Our methods achieved maximum accuracy in three problems. Furthermore, in the last problem (circRNA vs. lncRNA), our approaches were excellent when compared to other works that reached ACC of 0.7780 (PAN; XIONG, 2015) and 0.7890 (CHEN et al., 2018) in their datasets against 0.8307 of our best model (graphs). However, these works use different datasets, only using these comparisons as an (indirect) reference indicator.

4.2.3 Statistical Significance Tests

We assessed the statistical significance in the two case studies (difference in accuracy), using Friedman's statistical test and the Conover post-hoc test. Thereby, our null hypothesis ($H0 = A(1) = A(2) = \ldots = A(k)$), is tested against the alternative hypothesis (H_A), at least one algorithm 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 $H0$, since that $p < 0.05$. Thus, if there are significant differences, we conclude that a post-hoc statistical analysis is necessary. Conover statistics values were obtained, as well as p-values (see Table 28), using 95% of significance ($\alpha = 0.05$).

Table 28 – 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

According to the Conover post-hoc test, entropy-based models have highly significant differences to 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, the other mathematical models in the case study I do not differ significantly, indicating the efficiency of all models in different datasets. Now, evaluating case study II, we realized that the global test with Friedman's statistical test is not significant, in which we get $\chi^2(2) = 1.64$, p-value $= 0.4412$, indicating that the three feature extraction techniques show similar performance in the problems, again, confirming the effectiveness and robustness of all mathematical models.

5 DISCUSSION

This chapter discusses our findings in terms of whether they support our hypotheses. Overall, two experimental tests were assumed in this research, and our findings fully support four hypotheses out of five raised, one being partially accepted. Thus, H1-PB1 and H2-PB1 are discussed in Section 5.1 and H3-PB2, H4-PB2, H5-PB2 in Section 5.2.

5.1 Experimental Test I

H1-PB1: Can metaheuristics select a subset of predictive features able to improve the predictive performance of a classification model ...?

Our findings fully support this hypothesis. To prove it, we compared the best and worst model in the *performance test* (see Table 16), respectively, M1-GA and M5-PSO, against a model without feature selection, as shown in Table 29. In the overall average, our approach represented a gain of 4.68% (M1-GA) and 4.62% (M2-PSO) in the ACC. However, in some species, we reached an increase in the ACC of 6.40% (*S. bicolor*), 5.92% (*B. distachyon*), and 4.85% (*C. sinensis*). Fundamentally, these results expose the high efficiency of metaheuristics for feature selection in lncRNAs.

Table 29 – Our approach against all features.

Species	Method - ID	ACC	Species	Method - ID	ACC
A. trichopoda	All Features (5,467)	84.51	*B. distachyon*	All Features (5,467)	86.68
	M1-GA (10)	**88.50**		M1-GA (10)	**92.60**
	M5-PSO (7)	88.49		M5-PSO (7)	92.47
C. sinensis	All Features (5,467)	89.22	*M. esculenta*	All Features (5,467)	88.96
	M1-GA (10)	**94.07**		M1-GA (10)	**93.29**
	M5-PSO (7)	94.04		M5-PSO (7)	93.27
R. communis	All Features (5,467)	87.11	*S. tuberosum*	All Features (5,467)	83.90
	M1-GA (10)	**90.74**		M1-GA (10)	**87.94**
	M5-PSO (7)	90.70		M5-PSO (7)	87.88
S. bicolor	All Features (5,467)	86.07	*Z. mays*	All Features (5,467)	86.40
	M1-GA (10)	**92.47**		M1-GA (10)	**90.73**
	M5-PSO (7)	92.34		M5-PSO (7)	90.65
Overall Average	**All Features (5,467)**	86.61			
	M1-GA (10)	**91.29**			
	M5-PSO (7)	91.23			

In Table 30, we also compared two random metaheuristics (M1-EA and M2-ACO) against a model without feature selection using more evaluation metrics (SE, SPC, and

Table 30 – Comparative performance between M1-EA, M2-ACO and all features for three plant species.

Species	Technique - ID	Features	SE	SPC	ACC
R. communis	M1-EA	5	**99.95**	81.47	90.71
	M2-ACO	6	**99.95**	81.52	**90.74**
	All features	5,467	91.05	**83.16**	87.11
S. tuberosum	M1-EA	5	99.95	75.89	87.92
	M2-ACO	6	**99.96**	75.91	**87.93**
	All features	5,467	88.41	**79.40**	83.90
Z. mays	M1-EA	5	99.91	81.49	90.70
	M2-ACO	6	**99.92**	81.50	**90.71**
	All features	5,467	88.21	**84.58**	86.40

ACC). In the overall average, this comparison represented a gain of 3.99% (M2-ACO) in the ACC and 10.72% (M2-ACO) in the SE (to detect lncRNAs).

H2-PB1: Are metaheuristic models more efficient than non-heuristic models for biological sequence classification?

To answer this hypothesis, we compare two best features (*txCdsPredict score* and *cdsSizes*) against other five non-heuristic state-of-the-art programs: RNAplonc (NEGRI et al., 2018) (specifically for plants - 16 features, statistics feature selection), CPC (KONG et al., 2007) (Multi-Species - 6 features), CPC2 (KANG et al., 2017) (Multi-Species - 4 features), CNCI (SUN et al., 2013) (Animals and Plants - 5 features), PLEK (LI; ZHANG; ZHOU, 2014) (Multi-Species - $1,364$ features), as shown in Table 22. In which, our model (with two features) reported the best performance of ACC in three species of the five analyzed. Thus, we assessed the statistical significance of the difference in accuracy, using Friedman's statistical test and the post-hoc test of Nemenyi, Conover, and Siegel-Castellan. According to Pohlert (2014), Friedman test is the non-parametric alternative for this type of approach with equal sample sizes.

Fundamentally, this test ranks the algorithms separately for each dataset, in which the best performance gets the rank of 1, the second best rank 2 ... (DEMŠAR, 2006). Thus, our null hypothesis ($H0 = A(1) = A(2) = \ldots = A(k)$), is tested against the alternative hypothesis (H_A), at least one metaheuristic has statistical significance ($\alpha = 0.05$, $p < \alpha$). Firstly, we apply the global test, in which the Friedman test indicates significance ($\chi^2(6) = 27.25$, p-value $= 1.301 \times 10^{-4}$), that is, we can reject $H0$, since that $p < 0.01$. Thus, if there are significant differences, we conclude that a post-hoc statistical analysis is necessary. Nemenyi, Conover and Siegel-Castellan statistic values were obtained, as well as p-values (see Table 31 - our models vs. tools), using 95% of significance ($\alpha = 0.05$).

Table 31 – Nemenyi, Conover and Siegel-Castellan statistics values - The accepted null hypothesis are in bold (p-values for $\alpha = 0.05$).

Tools	Nemenyi	Conover	Siegel-Castellan
Our approach vs. RNAplonc	**0.9999**	**0.1890**	**0.7697**
Our approach vs. CPC	**0.3391**	7.2×10^{-10}	0.0338
Our approach vs. CPC2	**0.6272**	5.2×10^{-8}	**0.0923**
Our approach vs. CNCI	0.0104	3.1×10^{-14}	0.0006
Our approach vs. PLEK	0.0172	8.0×10^{-14}	0.0010

According to the Nemenyi post-hoc test, our approach has highly significant differences ($p < 0.02$) to CNCI and PLEK, but do not differ significantly ($p > 0.05$) to CPC and CPC2. Nevertheless, in the overall average, our model had an ACC of 92.17% across all datasets, hat is, 0.37%, 10.17%, 7.28%, 18.07%, and 17.46% more than RNAplonc (91.80%), CPC (82.00%), CPC2 (84.89%), CNCI (74.10%), and PLEK (74.71%), respectively. On the other hand, in the Conover post-hoc test, our models showed significant differences ($p < 0.01$) to CPC, CPC2, CNCI, PLEK. The Siegel-Castellan test also obtained statistical significance, $p < 0.01$, $p < 0.01$, and $p < 0.04$, for CNCI, PLEK and CPC, respectively. As expected, our metaheuristic approach and RNAplonc tool, do not differ significantly in all tests. Nevertheless, we only use 2 features, that is, a difference of 14 attributes unless RNAplonc. Therefore, our results also fully support this hypothesis, since our approach provides competitive classification performance with non-heuristic literature programs using the smallest number of features.

5.2 Experimental Test II

H3-PB2: Are mathematical models efficient for feature extraction from biological sequences?

Our findings fully support this hypothesis, since all mathematical models showed excellent results in the two case studies, as can be seen in Table 26 and Table 27. That is, all models were robust in different datasets/problems without loss of performance.

H4-PB2: Do mathematical models present competitive classification performance in distinct biological sequence analysis problems?

Our findings also fully support this hypothesis. Because, after evaluating all models in five different datasets and observing their great 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). Again, all showed excellent results. In which, graph-based models were the best in three of the four problems analyzed, followed by entropy and GSP. In the first three datasets, our methods achieved maximum accuracy. Furthermore, if we look at the last problem (circRNA vs. lncRNA), our approaches were excellent when compared to our references that reached an ACC of 0.7780 (PAN; XIONG, 2015) and 0.7890 (CHEN et al., 2018) in their datasets against 0.8307 of our best model (graph). Thus, we can say that all models maintained excellent performance in different sequence classification problems.

H5-PB2: Are mathematical models more generalist than biological models in biological sequences classification?

To answer this hypothesis, we compared the performance of the three mathematical models shown in Table 27 in relation to a model with biological bias characteristics, in four datasets ((lncRNA vs. mRNA (case study I)); (lncRNA vs. sncRNA; Antisense vs. lncRNA; circRNA vs. lncRNA (case study II)). For fair analysis, we only use datasets of *A. thaliana*. Thus, we generate our biological model using features provided by CPC2 (KANG et al., 2017), for being a widely used tool in the literature. However, we eliminated the sequence length descriptor provided by CPC2 and also any attribute that would generate this information in our approach, since that any explicit bias to this feature may facilitate the prediction. The features used in the biological model were Fickett TESTCODE score (see Equation 2.4), isoelectric point, open reading frame (ORF) length, ORF integrity. Therefore, we applied new experiments according to the same methodology (70% training and 30% test) and using CatBoost classifier, as shown in Table 32.

Table 32 – Performance analysis of three mathematical models against a biological bias 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 Model	**0.9869**	**0.9764**	**0.9816**	Biological Model	0.7855	0.8273	0.8065

Antisense vs. lncRNA				**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 Model	0.9412	0.8889	0.9143	Biological Model	0.6024	0.7612	0.6818

As can be seen, the biological model (0.9816) reported the best performance in the first dataset (lncRNA vs. mRNA), followed by our mathematical model (Entropy - 0.9587), with only a difference of 0.0229. Nevertheless, it is important to emphasize that this biological model uses the ORF descriptor, a highly employed feature for discovering coding sequences and which, according to Baek et al. (BAEK et al., 2018) 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 dissertation, this type of feature with a biological bias is often difficult to reuse or adapt to another specific problem. Thereby, our approach has an advantage in terms of generalization, since this would not be possible only with the ORF. 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. Therefore, our pipeline is robust in terms of generalization to distinguish lncRNA from mRNA, as well as other biological sequence classification problems. We also assessed the static significance of the mathematical versus biological model in the previously applied tests, in which entropy ($p < 0.0480$) and graphs ($p < 0.0200$) indicated significant results in relation to the biological model. Despite the great results, to fully support this hypothesis, we still need to develop more experiments with biological bias characteristics. Thus, we partially accept this hypothesis.

6 CONCLUSION

This dissertation proposed to analyze features selection and extraction methods for biological sequence classification, addressing two key phases of building a predictive model (feature extraction and selection). Specifically, we concentrated our work on the study of dimensionality reduction and feature extraction techniques, using metaheuristics and mathematical models. As a case study, we use lncRNA sequences, which are fundamentally unable to produce proteins, and, recently, have been remitting several doubts about its functionality. Moreover, we divided this work into two parts: Experimental Test I and Experimental Test II.

In Experimental Test I, we select the most used feature for lncRNAs classification (ORF, sequence length, GC content, and k-mer), in order to apply metaheuristic models for feature selection. Thus, we build a training set of transcript data from five plant species (*Arabidopsis thaliana, Cucumis sativus, Glycine max, Populus trichocarpa, and Oryza sativa*), divided into lncRNA samples and protein-coding genes (mRNA). Further, we assemble a feature vector using GC content, k-mer (1-6), sequence length, Open Reading Frame (score, cdsStarts, cdsStop, cdsSizes, and cdsPercent - generated by txCdsPredict). The features predictive capacity was evaluated with REPTree classifier in eight datasets of different plant species (*Amborella, Brachypodium, Citrus, Manihot, Ricinus, Solanum, Sorghum, and Zea*). Each metaheuristic algorithm underwent five execution rounds to return an optimal features subset.

From the obtained results in our experiments, two algorithms selected the least amount of features, M2-EA and M3-ABC returned a great set with 5 attributes, followed by M4-ACO (6 features), M5-PSO (7 features), and M1-GA (10 features). Regarding the performance tests, M1-GA reported the best result of SE (100%) and ACC (91.29%), followed by M3-ABC (SE: 99.95% and ACC: 91.27%), and M4-ACO (SE: 99.94% and ACC: 91.27%). Regarding specificity, the best methods were M2-EA and M4-ACO with 82.61%, respectively. Furthermore, in the overall average, our approach reached a gain of 4.68% (M1-GA) and 4.62% (M5-PSO) in the ACC when compared to a model without feature selection (all features). We also observed a high dependency of ORF-derived features (e.g., *txCdsPredict score, cdsStop, cdsSizes*, and *cdsPercent*) on methods that classify coding/lncRNAs. Based on this, we compare the two best features (*txCdsPredict score* and *cdsSizes*) against other five state-of-the-art programs: RNAplonc, CPC, CPC2, CNCI, and PLEK. In which, our model showed an average ACC of 92.17% (only with two features) across all datasets, that is, 0.37%, 10.17%, 7.28%, 18.07%, and 17.46% more than RNAplonc (91.80%), CPC (82.00%), CPC2 (84.89%), CNCI (74.10%), and PLEK (74.71%), respectively.

In Experimental Test II, we analyze mathematical models for feature extraction in order to propose efficient and generalist techniques for biological sequence analysis problems. Therefore, at this stage, as a starting point, nine mathematical models for feature extraction were analyzed: six numerical mapping techniques with Fourier transform; Tsallis and Shannon entropy; Graphs (complex networks). Thereby, several lncRNA classification problems were adopted in order to validate the proposed approach. As a result, all models presented excellent results, with the worst performance (ACC) of 89.01% and the best of 96.06% in the first case study. In the second case study, again, all showed excellent results. In which, graph-based models indicated the best performance in three of the four problems analyzed, followed by entropy and GSP. Furthermore, to assess our study, we compared the performance of three mathematical models against a biological bias model, in four different datasets. In which, our models achieved suitable results, being superior or competitive and robust in terms of generalization.

Based on this, our findings fully support four hypotheses out of five raised, one being partially accepted. Therefore, this dissertation contributes to the area of computer science and bioinformatics. Specifically, it introduces new ideas and analysis for the feature selection and extraction problem in biological sequences, using lncRNAs as a case study. Thus, we present:

- **Contributions to the feature selection problem in biological sequences:**
    1. A new pipeline with metaheuristics, using a voting scheme and execution rounds;
    2. Application of five metaheuristics to the feature selection problem in biological sequences;
    3. An in-depth analysis of 5,467 features in lncRNAs;
    4. The metaheuristic efficiency in selecting relevant features, providing competitive classification performance;
    5. A lncRNA classification tool using the best features selected by the pipeline [1].
- **Contributions to the feature extraction problem in biological sequences:**
    1. A feature extraction pipeline in biological sequences using mathematical models;
    2. Analysis of nine different mathematical models;
    3. Analysis of six numerical mappings with Fourier, proposing statistical characteristics;
    4. The generalization and robustness of mathematical models for the feature extraction in biological sequence.

[1] https://github.com/Bonidia/FeatureSelection_lncRNAs

Finally, the general contribution of this dissertation is a generic pipeline for biological sequence classification, which addresses two main phases of generating predictive models, feature extraction and selection, using metaheuristics and mathematical models.

6.1 Publications

Some works have been published during the development of the research for this dissertation. Hence, part of the results can be found in these publications:

- Parmezan Bonidia, R.; Negri, T.; Alves, W.; Domingues, D. S.; Kashiwabara, A.; Rossi Paschoal, A.; and Sipoli Sanches, D. **Feature Selection of Long Non-Coding RNAs in Plants: A Heuristic Approach with Particle Swarm Optimization**. In: X-meeting, 2018, São Pedro -SP. Proceedings X-meeting 2018, 2018. **Abstract**.

- Bonidia, R. P., Sampaio, L. D. H., Lopes, F. M., and Sanches, D. S. (2019a). **Feature Extraction of Long Non-Coding RNAs: A Fourier and Numerical Mapping Approach.** In Nyström, I., Hernández Heredia, Y., and Milián Núñez, V., editors, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pages 469–479, Cham. Springer International Publishing. **Conference, B1**.

- Parmezan Bonidia, R., Ponce de Leon Ferreira de Carvalho, A. C., Rossi Paschoal, A., and Sipoli Sanches, D. (2019). **Selecting the most relevant features for the identification of long non-coding RNAs in plants.** In 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), pages 539–544. **Conference, B2**.

- Parmezan Bonidia, R.; Negri, T.; Alves, W.; Domingues, D. S.; Kashiwabara, A.; Ponce de Leon Ferreira de Carvalho, A. C.; Rossi Paschoal, A.; and Sipoli Sanches, D. (2019). **Feature Selection of Biological Sequences: A Case Study with Metaheuristic Models in Long Non-Coding RNAs.** Expert Systems. **In writing, journal**.

- Bonidia, R. P.; Sampaio, L. D. H.; Lopes, F. M.; Ponce de Leon Ferreira de Carvalho, A. C.; and Sanches, D. S. (2019a). **Mathematical or Biological Feature Extraction Models: Which one is More Generalist in RNA Sequences Classification?** Applied Soft Computing. **In writing, journal, A1**.

6.2 Future Works

As future work, we will continue to investigate the hypotheses of this dissertation in other problems of biological sequence analysis. However, our ultimate goal is to develop a generic machine learning toolkit/pipeline for classification of biological sequences, addressing **three** key phases of building a predictive model (feature extraction, selection, and classification) and using metaheuristics, mathematical, and ensemble models. Fundamentally, we will increase a phase of our pipeline, studying new algorithms and techniques, such as:

- **Feature Extraction:** we will also study wavelet-based feature extraction techniques, complex networks, mapping with nucleotides triplets and amino acid features, chaos game representation, among others.
- **Feature Selection:** we will apply hybrid and wrapper feature selection approaches with metaheuristics.
- **Classification:** we will propose an ensemble model using metaheuristics, modeling the combination of classifiers as an optimization problem, where the weights of the confidence levels of each classifier are determined by evolutionary optimization techniques.

Bibliography

ABBAS, Q. et al. A review of computational methods for finding non-coding rna genes. *Genes*, Multidisciplinary Digital Publishing Institute, v. 7, n. 12, p. 113, 2016. Cited 2 times on page 18 and 19.

ABO-ZAHHAD, M.; AHMED, S. M.; ABD-ELRAHMAN, S. A. Genomic analysis and classification of exon and intron sequences using dna numerical mapping techniques. *International Journal of Information Technology and Computer Science*, v. 4, n. 8, p. 22–36, 2012. Cited 4 times on page 68, 69, 71, and 72.

ABU-JAMOUS, B.; FA, R.; NANDI, A. K. *Integrative cluster analysis in bioinformatics.* [S.l.]: John Wiley & Sons, 2015. Cited on page 19.

ACHAWANANTAKUN, R. et al. Lncrna-id: Long non-coding rna identification using balanced random forests. *Bioinformatics*, Oxford University Press, v. 31, n. 24, p. 3897–3905, 2015. Cited 3 times on page 26, 27, and 32.

ALLISON, L. A. *Fundamental molecular biology.* [S.l.]: Blackwell Pub., 2007. Cited on page 19.

ALTSCHUL, S. F. et al. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, Oxford University Press, v. 25, n. 17, p. 3389–3402, 1997. Cited on page 32.

AMARAL, P. P. et al. lncrnadb: a reference database for long noncoding rnas. *Nucleic acids research*, Oxford University Press, v. 39, n. suppl_1, p. D146–D151, 2010. Cited on page 33.

AMIN, N.; MCGRATH, A.; CHEN, Y.-P. P. Evaluation of deep learning in non-coding rna classification. *Nature Machine Intelligence*, Nature Publishing Group, v. 1, n. 5, p. 246, 2019. Cited on page 18.

AMR, S. S.; FUNKE, B. Targeted hybrid capture for inherited disease panels. In: *Clinical Genomics*. [S.l.]: Elsevier, 2015. p. 251–269. Cited 2 times on page 29 and 42.

ANASTASSIOU, D. Genomic signal processing. *IEEE signal processing magazine*, IEEE, v. 18, n. 4, p. 8–20, 2001. Cited 2 times on page 68 and 70.

Anastassiou, D. Genomic signal processing. *IEEE Signal Processing Magazine*, v. 18, n. 4, p. 8–20, July 2001. ISSN 1558-0792. Cited on page 69.

ANTONOV, I. V. et al. Prediction of lncrnas and their interactions with nucleic acids: benchmarking bioinformatics tools. *Briefings in bioinformatics*, 2018. Cited on page 26.

BAEK, J. et al. lncrnanet: Long non-coding rna identification using deep learning. *Bioinformatics*, Oxford University Press, v. 1, p. 9, 2018. Cited 3 times on page 18, 42, and 86.

BALDI, P. et al. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, Oxford University Press, v. 16, n. 5, p. 412–424, 2000. Cited on page 37.

BARROS-CARVALHO, G. A.; SLUYS, M.-A. V.; LOPES, F. M. An efficient approach to explore and discriminate anomalous regions in bacterial genomes based on maximum entropy. *Journal of Computational Biology*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 24, n. 11, p. 1125–1133, 2017. Cited on page 73.

BELLMAN, R. E. *Adaptive control processes: a guided tour*. [S.l.]: Princeton university press, 1961. Cited on page 18.

BHARTIYA, D. et al. lncrnome: a comprehensive knowledgebase of human long noncoding rnas. *Database*, Oxford University Press, v. 2013, 2013. Cited on page 33.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 245–271, 1997. Cited on page 21.

BONIDIA, R. P. et al. Computational intelligence in sports: A systematic literature review. *Advances in Human-Computer Interaction*, Hindawi, v. 2018, 2018. Cited on page 47.

BONIDIA, R. P. et al. Feature extraction of long non-coding rnas: A fourier and numerical mapping approach. In: NYSTRÖM, I.; HEREDIA, Y. H.; NÚÑEZ, V. M. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Cham: Springer International Publishing, 2019. p. 469–479. ISBN 978-3-030-33904-3. Cited 4 times on page 17, 18, 22, and 68.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Cited on page 76.

BU, D. et al. Noncode v3. 0: integrative annotation of long noncoding rnas. *Nucleic acids research*, Oxford University Press, v. 40, n. D1, p. D210–D215, 2011. Cited on page 33.

BUDACH, S.; MARSICO, A. pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics*, v. 34, n. 17, p. 3035–3037, 04 2018. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/bty222>. Cited on page 24.

CAO, M.-R. et al. Bioinformatic analysis and prediction of the function and regulatory network of long non-coding rnas in hepatocellular carcinoma. *Oncology letters*, Spandidos Publications, v. 15, n. 5, p. 7783–7793, 2018. Cited 2 times on page 17 and 24.

CARUANA, R.; FREITAG, D. Greedy attribute selection. In: *Machine Learning Proceedings 1994*. [S.l.]: Elsevier, 1994. p. 28–36. Cited on page 21.

CHAKRABORTY, S. et al. Lncrbase: an enriched resource for lncrna information. *PloS one*, Public Library of Science, v. 9, n. 9, p. e108010, 2014. Cited on page 33.

CHAKRAVARTHY, N. et al. Autoregressive modeling and feature analysis of dna sequences. *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corp., v. 2004, p. 13–28, 2004. Cited 2 times on page 69 and 70.

CHAN, W.-L.; HUANG, H.-D.; CHANG, J.-G. lncrnamap: a map of putative regulatory functions in the long non-coding transcriptome. *Computational biology and chemistry*, Elsevier, v. 50, p. 41–49, 2014. Cited on page 33.

CHEN, D. et al. PlantNATsDB: a comprehensive database of plant natural antisense transcripts. *Nucleic Acids Research*, v. 40, n. D1, p. D1187–D1193, 11 2011. ISSN 0305-1048. Disponível em: <https://doi.org/10.1093/nar/gkr823>. Cited on page 67.

CHEN, G. et al. Lncrnadisease: a database for long-non-coding rna-associated diseases. *Nucleic acids research*, Oxford University Press, v. 41, n. D1, p. D983–D986, 2012. Cited on page 33.

CHEN, L. et al. Discriminating cirrnas from other lncrnas using a hierarchical extreme learning machine (h-elm) algorithm with feature selection. *Molecular Genetics and Genomics*, Springer, v. 293, n. 1, p. 137–149, 2018. Cited 3 times on page 67, 80, and 85.

CHEN, W. et al. Pseknc: A flexible web server for generating pseudo k-tuple nucleotide composition. *Analytical Biochemistry*, v. 456, p. 53 – 60, 2014. ISSN 0003-2697. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0003269714001249>. Cited on page 24.

CHEN, W. et al. PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, v. 31, n. 1, p. 119–120, 09 2014. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btu602>. Cited on page 24.

CHEN, X. et al. Computational models for lncrna function prediction and functional similarity calculation. *Briefings in functional genomics*, 2018. Cited on page 33.

CHENG, J. et al. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, American Association for the Advancement of Science, v. 308, n. 5725, p. 1149–1154, 2005. Cited on page 20.

CHOONIEDASS-KOTHARI, S. et al. The steroid receptor rna activator is the first functional rna encoding a protein. *FEBS letters*, Wiley Online Library, v. 566, n. 1-3, p. 43–47, 2004. Cited on page 20.

CHU, Q. et al. Plantcircbase: a database for plant circular rnas. *Molecular plant*, Elsevier, v. 10, n. 8, p. 1126–1128, 2017. Cited on page 67.

CIAUDO, C. et al. Highly dynamic and sex-specific expression of micrornas during early es cell differentiation. *PLoS genetics*, Public Library of Science, v. 5, n. 8, p. e1000620, 2009. Cited on page 20.

COCHRAN, W. T. et al. What is the fast fourier transform? *Proceedings of the IEEE*, IEEE, v. 55, n. 10, p. 1664–1674, 1967. Cited on page 68.

COHEN, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, Sage Publications Sage CA: Thousand Oaks, CA, v. 20, n. 1, p. 37–46, 1960. Cited on page 76.

COSTA, L. d. F.; RODRIGUES, F. A.; CRISTINO, A. S. Complex networks: the key to systems biology. *Genetics and Molecular Biology*, SciELO Brasil, v. 31, n. 3, p. 591–601, 2008. Cited on page 75.

CRISTEA, P. D. Conversion of nucleotides sequences into genomic signals. *Journal of cellular and molecular medicine*, Wiley Online Library, v. 6, n. 2, p. 279–303, 2002. Cited 2 times on page 69 and 70.

DASH, M.; LIU, H. Consistency-based search in feature selection. *Artificial intelligence*, Elsevier, v. 151, n. 1-2, p. 155–176, 2003. Cited on page 44.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006. Cited on page 83.

DERRIEN, T. et al. The gencode v7 catalog of human long noncoding rnas: analysis of their gene structure, evolution, and expression. *Genome research*, Cold Spring Harbor Lab, v. 22, n. 9, p. 1775–1789, 2012. Cited on page 20.

DI, C. et al. Characterization of stress-responsive lncrnas in arabidopsis thaliana by integrating expression, epigenetic and structural features. *The Plant Journal*, Wiley Online Library, v. 80, n. 5, p. 848–861, 2014. Cited on page 20.

DINIZ, W. J. d. S.; CANDURI, F. Bioinformatics: an overview and its applications. *Genet Mol Res*, v. 16, n. 1, 2017. Cited on page 17.

DOERING, J. et al. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives*, Elsevier, p. 100121, 2019. Cited on page 45.

DORIGO, M.; BIRATTARI, M. Ant colony optimization. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 36–39. Cited on page 47.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 26, n. 1, p. 29–41, 1996. Cited 2 times on page 46 and 47.

DOROGUSH, A. V.; ERSHOV, V.; GULIN, A. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018. Cited on page 76.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012. Cited on page 48.

DY, J. G.; BRODLEY, C. E. Feature selection for unsupervised learning. *Journal of machine learning research*, v. 5, n. Aug, p. 845–889, 2004. Cited 2 times on page 18 and 21.

EDDY, S. R. Hidden markov models. *Current opinion in structural biology*, Elsevier, v. 6, n. 3, p. 361–365, 1996. Cited on page 32.

EDDY, S. R. Non-coding rna genes and the modern rna world. *Nature Reviews Genetics*, Nature Publishing Group, v. 2, n. 12, p. 919, 2001. Cited on page 19.

FAN, X.-N.; ZHANG, S.-W. lncrna-mfdl: identification of human long non-coding rnas by fusing multiple features and using deep learning. *Molecular BioSystems*, Royal Society of Chemistry, v. 11, n. 3, p. 892–897, 2015. Cited 8 times on page 26, 27, 29, 31, 41, 42, 43, and 48.

FANG, Y.; FULLWOOD, M. J. Roles, functions, and mechanisms of long non-coding rnas in cancer. *Genomics, proteomics & bioinformatics*, Elsevier, v. 14, n. 1, p. 42–54, 2016. Cited on page 20.

FOGEL, D. B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ, USA: IEEE Press, 1995. ISBN 0-7803-1038-1. Cited on page 46.

FONG, S.; BIUK-AGHAI, R. P.; MILLHAM, R. C. Swarm search methods in weka for data mining. In: ACM. *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. [S.l.], 2018. p. 122–127. Cited on page 45.

FRITAH, S.; NICLOU, S. P.; AZUAJE, F. Databases for lncrnas: a comparative evaluation of emerging tools. *Rna*, Cold Spring Harbor Lab, v. 20, n. 11, p. 1655–1665, 2014. Cited on page 33.

FRITH, M. C. et al. Discrimination of non-protein-coding transcripts from protein-coding mrna. *RNA biology*, Taylor & Francis, v. 3, n. 1, p. 40–48, 2006. Cited on page 42.

GALLART, A. P. et al. Greenc: a wiki-based database of plant lncrnas. *Nucleic acids research*, Oxford University Press, v. 44, n. D1, p. D1161–D1166, 2015. Cited 3 times on page 33, 39, and 67.

GENTLEMAN, R. C. et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, Springer, v. 5, n. 10, p. R80, 2004. Cited on page 17.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Cited on page 46.

GOODSTEIN, D. M. et al. Phytozome: a comparative platform for green plant genomics. *Nucleic acids research*, Oxford University Press, v. 40, n. D1, p. D1178–D1186, 2011. Cited 2 times on page 39 and 67.

GUPTA, B. B.; SHENG, Q. Z. *Machine Learning for Computer and Cyber Security: Principle, Algorithms, and Practices*. [S.l.]: CRC Press, 2019. Cited on page 45.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *Journal of machine learning research*, v. 3, n. Mar, p. 1157–1182, 2003. Cited 2 times on page 44 and 45.

GUYON, I. et al. *Feature extraction: foundations and applications*. [S.l.]: Springer, 2008. v. 207. Cited 2 times on page 22 and 68.

HALL, M. et al. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009. Cited on page 45.

HALL, M. A. Correlation-based feature selection for machine learning. University of Waikato Hamilton, 1999. Cited on page 45.

HAN, J.; KAMBER, M.; PEI, J. 13 - data mining trends and research frontiers. In: HAN, J.; KAMBER, M.; PEI, J. (Ed.). *Data Mining (Third Edition)*. Third edition. Boston: Morgan Kaufmann, 2012, (The Morgan Kaufmann Series in Data Management Systems). p. 585 – 631. ISBN 978-0-12-381479-1. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780123814791000137>. Cited on page 19.

HAN, S. et al. Long noncoding rna identification: comparing machine learning based tools for long noncoding transcripts discrimination. *BioMed research international*, Hindawi, v. 2016, 2016. Cited 2 times on page 26 and 29.

HAN, S. et al. Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Briefings in Bioinformatics*, 2018. Cited 4 times on page 26, 28, 35, and 65.

HASSAN, M. Q. et al. Non-coding rnas: Epigenetic regulators of bone development and homeostasis. *Bone*, Elsevier, v. 81, p. 746–756, 2015. Cited on page 19.

HASTIE, T. et al. Multi-class adaboost. *Statistics and its Interface*, International Press of Boston, v. 2, n. 3, p. 349–360, 2009. Cited on page 76.

HE, Y. et al. Long noncoding rnas: Novel insights into hepatocelluar carcinoma. *Cancer letters*, Elsevier, v. 344, n. 1, p. 20–27, 2014. Cited on page 20.

HU, R.; SUN, X. lncrnatargets: a platform for lncrna target prediction based on nucleic acid thermodynamics. *Journal of bioinformatics and computational biology*, World Scientific, v. 14, n. 04, p. 1650016, 2016. Cited on page 20.

INZA, I. et al. Machine learning: an indispensable tool in bioinformatics. In: *Bioinformatics methods in clinical research*. [S.l.]: Springer, 2010. p. 25–48. Cited on page 17.

ITO, E. A. et al. Basinet—biological sequences network: a case study on coding and non-coding rnas identification. *Nucleic acids research*, 2018. Cited 6 times on page 22, 26, 28, 65, 67, and 75.

JAIN, A.; ZONGKER, D. Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 19, n. 2, p. 153–158, 1997. Cited on page 21.

JIANG, Q. et al. Lncrna2function: a comprehensive resource for functional investigation of human lncrnas based on rna-seq data. In: BIOMED CENTRAL. *BMC genomics*. [S.l.], 2015. v. 16, n. 3, p. S2. Cited on page 33.

JIN, J. et al. Plncdb: plant long non-coding rna database. *Bioinformatics*, Oxford University Press, v. 29, n. 8, p. 1068–1071, 2013. Cited on page 33.

JOHN, G. H.; KOHAVI, R.; PFLEGER, K. Irrelevant features and the subset selection problem. In: *Machine Learning Proceedings 1994*. [S.l.]: Elsevier, 1994. p. 121–129. Cited on page 21.

JURTZ, V. I. et al. An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, v. 33, n. 22, p. 3685–3690, 08 2017. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btx531>. Cited on page 17.

KANG, Y.-J. et al. Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic acids research*, Oxford University Press, v. 45, n. W1, p. W12–W16, 2017. Cited 6 times on page 18, 59, 66, 67, 83, and 85.

KAPRANOV, P. et al. Rna maps reveal new rna classes and a possible function for pervasive transcription. *Science*, American Association for the Advancement of Science, v. 316, n. 5830, p. 1484–1488, 2007. Cited on page 19.

KENNEDY, J. Swarm intelligence. In: *Handbook of nature-inspired and innovative computing*. [S.l.]: Springer, 2006. p. 187–219. Cited on page 47.

KENNEDY, J. Particle swarm optimization. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 760–766. Cited on page 47.

KENT, W. J. et al. The human genome browser at ucsc. *Genome research*, Cold Spring Harbor Lab, v. 12, n. 6, p. 996–1006, 2002. Cited 2 times on page 42 and 43.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Cited 2 times on page 21 and 44.

KONG, L. et al. Cpc: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic acids research*, Oxford University Press, v. 35, n. suppl_2, p. W345–W349, 2007. Cited 4 times on page 18, 26, 59, and 83.

KRIZEK, P. *Feature selection: stability, algorithms, and evaluation*. Tese (Doutorado) — PhD thesis, Czech Technical University in Prague., 2008. Cited on page 44.

KUNG, J. T.; COLOGNORI, D.; LEE, J. T. Long noncoding rnas: past, present, and future. *Genetics*, Genetics Soc America, v. 193, n. 3, p. 651–669, 2013. Cited on page 20.

Kwan, H. K.; Arniker, S. B. Numerical representation of dna sequences. In: *2009 IEEE International Conference on Electro/Information Technology*. [S.l.: s.n.], 2009. p. 307–310. Cited on page 17.

LAL, T. N. et al. Embedded methods. In: GUYON, I. et al. (Ed.). *Feature Extraction: Foundations and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 137–165. ISBN 978-3-540-35488-8. Disponível em: <https://doi.org/10.1007/978-3-540-35488-8_6>. Cited on page 44.

LERTAMPAIPORN, S. et al. Identification of non-coding rnas with a new composite feature in the hybrid random forest ensemble algorithm. *Nucleic acids research*, Oxford University Press, v. 42, n. 11, p. e93–e93, 2014. Cited 3 times on page 34, 35, and 39.

LI, A. et al. A text feature-based approach for literature mining of lncrna–protein interactions. *Neurocomputing*, Elsevier, v. 206, p. 73–80, 2016. Cited on page 19.

LI, A.; ZHANG, J.; ZHOU, Z. Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme. *BMC bioinformatics*, BioMed Central, v. 15, n. 1, p. 311, 2014. Cited 7 times on page 18, 26, 27, 29, 59, 67, and 83.

LI, W.; GODZIK, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, Oxford University Press, v. 22, n. 13, p. 1658–1659, 2006. Cited on page 40.

LI, X.-Q. et al. Key anti-fibrosis associated long noncoding rnas identified in human hepatic stellate cell via transcriptome sequencing analysis. *International journal of molecular sciences*, Multidisciplinary Digital Publishing Institute, v. 19, n. 3, p. 675, 2018. Cited on page 43.

LIU, A. C. The effect of oversampling and undersampling on classifying imbalanced text datasets. *The University of Texas at Austin*, Citeseer, 2004. Cited on page 67.

LIU, B. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in Bioinformatics*, v. 20, n. 4, p. 1280–1294, 12 2017. ISSN 1467-5463. Disponível em: <https://doi.org/10.1093/bib/bbx165>. Cited on page 24.

LIU, B. et al. repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, v. 31, n. 8, p. 1307–1309, 12 2014. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btu820>. Cited on page 24.

LIU, B. et al. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Research*, v. 43, n. W1, p. W65–W71, 05 2015. ISSN 0305-1048. Disponível em: <https://doi.org/10.1093/nar/gkv458>. Cited on page 24.

LIU, C.-J. et al. lncrinter: A database of experimentally validated long non-coding rna interaction. *Journal of Genetics and Genomics*, Elsevier, v. 44, n. 5, p. 265–268, 2017. Cited on page 33.

LIU, K. et al. Linc2go: a human lincrna function annotation resource based on cerna hypothesis. *Bioinformatics*, Oxford University Press, v. 29, n. 17, p. 2221–2222, 2013. Cited on page 33.

LORENZ, R. et al. Viennarna package 2.0. *Algorithms for Molecular Biology*, BioMed Central, v. 6, n. 1, p. 26, 2011. Cited on page 29.

LOU, H. et al. Evolution of k-mer frequencies and entropy in duplication and substitution mutation systems. *IEEE Transactions on Information Theory*, IEEE, 2019. Cited on page 17.

MA, L.; BAJIC, V. B.; ZHANG, Z. On the classification of long non-coding rnas. *RNA biology*, Taylor & Francis, v. 10, n. 6, p. 924–933, 2013. Cited on page 20.

MA, L. et al. Lncrnawiki: harnessing community knowledge in collaborative curation of human long non-coding rnas. *Nucleic acids research*, Oxford University Press, v. 43, n. D1, p. D187–D192, 2014. Cited on page 33.

MACHADO, J. T.; COSTA, A. C.; QUELHAS, M. D. Shannon, rényie and tsallis entropy analysis of dna using phase plane. *Nonlinear Analysis: Real World Applications*, Elsevier, v. 12, n. 6, p. 3135–3144, 2011. Cited 2 times on page 22 and 73.

MARSELLA, L. et al. Repetita: detection and discrimination of the periodicity of protein solenoid repeats by discrete fourier transform. *Bioinformatics*, Oxford University Press, v. 25, n. 12, p. i289–i295, 2009. Cited on page 68.

MASOUDI-SOBHANZADEH, Y.; MOTIEGHADER, H.; MASOUDI-NEJAD, A. Featureselect: a software for feature selection based on machine learning approaches. *BMC bioinformatics*, BioMed Central, v. 20, n. 1, p. 170, 2019. Cited on page 24.

MENDIZABAL-RUIZ, G. et al. On dna numerical representations for genomic similarity computation. *PloS one*, Public Library of Science, v. 12, n. 3, p. e0173288, 2017. Cited 3 times on page 68, 69, and 70.

MIN, R. *Machine Learning Approaches to Biological Sequence and Phenotype Data Analysis*. [S.l.]: University of Toronto, 2010. Cited on page 17.

MORAGLIO, A. et al. Geometric particle swarm optimization. *Journal of Artificial Evolution and Applications*, Hindawi, v. 2008, 2008. Cited on page 47.

MUHAMMOD, R. et al. PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences. *Bioinformatics*, v. 35, n. 19, p. 3831–3833, 03 2019. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btz165>. Cited 2 times on page 18 and 24.

NAIR, A. S.; SREENADHAN, S. P. A coding measure scheme employing electron-ion interaction pseudopotential (eiip). *Bioinformation*, Biomedical Informatics Publishing Group, v. 1, n. 6, p. 197, 2006. Cited 2 times on page 69 and 72.

NAYAR, N.; AHUJA, S.; JAIN, S. Swarm intelligence for feature selection: A review of literature and reflection on future challenges. In: KOLHE, M. L. et al. (Ed.). *Advances in Data and Information Sciences*. Singapore: Springer Singapore, 2019. p. 211–221. ISBN 978-981-13-0277-0. Cited on page 45.

NEGRI, T. d. C. et al. Pattern recognition analysis on long noncoding rnas: a tool for prediction in plants. *Briefings in bioinformatics*, 2018. Cited 7 times on page 18, 28, 35, 39, 50, 59, and 83.

NIAZI, F.; VALADKHAN, S. Computational analysis of functional long noncoding rnas reveals lack of peptide-coding capacity and parallels with 3' utrs. *Rna*, Cold Spring Harbor Lab, 2012. Cited on page 42.

NIKAM, R.; GROMIHA, M. M. Seq2Feature: a comprehensive web-based feature extraction tool. *Bioinformatics*, v. 35, n. 22, p. 4797–4799, 05 2019. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btz432>. Cited on page 24.

NILSSON, R. et al. Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, v. 8, n. Mar, p. 589–612, 2007. Cited on page 21.

PAN, X.; XIONG, K. Predcircrna: computational classification of circular rna from other long non-coding rna using hybrid features. *Molecular Biosystems*, Royal Society of Chemistry, v. 11, n. 8, p. 2219–2226, 2015. Cited 3 times on page 67, 80, and 85.

PARASKEVOPOULOU, M. D. et al. Diana-lncbase: experimentally verified and computationally predicted microrna targets on long non-coding rnas. *Nucleic acids research*, Oxford University Press, v. 41, n. D1, p. D239–D245, 2012. Cited on page 33.

PARK, C. et al. lncrnator: a comprehensive resource for functional investigation of long non-coding rnas. *Bioinformatics*, Oxford University Press, v. 30, n. 17, p. 2480–2485, 2014. Cited on page 33.

Parmezan Bonidia, R. et al. Selecting the most relevant features for the identification of long non-coding rnas in plants. In: *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2019. p. 539–544. ISSN 2643-6256. Cited 2 times on page 17 and 45.

PASTORI, C.; WAHLESTEDT, C. Involvement of long noncoding rnas in diseases affecting the central nervous system. *RNA biology*, Taylor & Francis, v. 9, n. 6, p. 860–870, 2012. Cited on page 20.

PENG, X. et al. Unique signatures of long noncoding rna expression in response to virus infection and altered innate immune signaling. *MBio*, Am Soc Microbiol, v. 1, n. 5, p. e00206–10, 2010. Cited on page 20.

PHAM, D. T.; CASTELLANI, M. The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, SAGE Publications Sage UK: London, England, v. 223, n. 12, p. 2919–2938, 2009. Cited on page 46.

PIAN, C. et al. Lncrnapred: Classification of long non-coding rnas and protein-coding transcripts by the ensemble algorithm with a new hybrid feature. *PloS one*, Public Library of Science, v. 11, n. 5, p. e0154567, 2016. Cited 5 times on page 26, 28, 34, 35, and 65.

POHLERT, T. The pairwise multiple comparison of mean ranks package (pmcmr). *R package*, v. 27, 2014. Cited on page 83.

PRITIŠANAC, I. et al. Entropy and information within intrinsically disordered protein regions. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 21, n. 7, p. 662, 2019. Cited on page 73.

ROKACH, L.; MAIMON, O. Z. *Data mining with decision trees: theory and applications. 2nd Edition*. [S.l.]: World scientific, 2015. v. 69. Cited on page 21.

RUDNICKI, W. R.; WRZESIEŃ, M.; PAJA, W. All relevant feature selection methods and applications. In: *Feature Selection for Data and Pattern Recognition*. [S.l.]: Springer, 2015. p. 11–28. Cited on page 21.

SAEYS, Y.; INZA, I.; LARRAÑAGA, P. A review of feature selection techniques in bioinformatics. *bioinformatics*, Oxford University Press, v. 23, n. 19, p. 2507–2517, 2007. Cited on page 17.

SAIDI, R. et al. Feature extraction in protein sequences classification: a new stability measure. In: ACM. *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. [S.l.], 2012. p. 683–689. Cited on page 22.

SELVAKUBERAN, K.; INDRADEVI, M.; RAJARAM, R. Combined feature selection and classification–a novel approach for the categorization of web pages. *Journal of Information and Computing Science*, Citeseer, v. 3, n. 2, p. 083–089, 2008. Cited on page 45.

SHAO, J.; YAN, X.; SHAO, S. Snr of dna sequences mapped by general affine transformations of the indicator sequences. *Journal of mathematical biology*, Springer, v. 67, n. 2, p. 433–451, 2013. Cited 2 times on page 71 and 73.

SIEBER, P.; PLATZER, M.; SCHUSTER, S. The definition of open reading frame revisited. *Trends in Genetics*, Elsevier, v. 34, n. 3, p. 167–170, 2018. Cited on page 29.

SIEVERS, A. et al. K-mer content, correlation, and position analysis of genome dna sequences for the identification of function and evolutionary features. *Genes*, Multidisciplinary Digital Publishing Institute, v. 8, n. 4, p. 122, 2017. Cited on page 29.

SIMOPOULOS, C. M.; WERETILNYK, E. A.; GOLDING, G. B. Prediction of plant lncrna by ensemble machine learning classifiers. *BMC genomics*, BioMed Central, v. 19, n. 1, p. 316, 2018. Cited on page 28.

SINGH, B. K.; VERMA, K.; THOKE, A. Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. *International Journal of Computer Applications*, Foundation of Computer Science, v. 116, n. 19, 2015. Cited 2 times on page 44 and 76.

SINGH, U. et al. 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*, Oxford University Press, v. 45, n. 22, p. e183–e183, 2017. Cited on page 28.

SOUTO, M. C. de et al. Comparative study on normalization procedures for cluster analysis of gene expression datasets. In: IEEE. *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on.* [S.l.], 2008. p. 2792–2798. Cited on page 44.

STAŃCZYK, U. Feature evaluation by filter, wrapper, and embedded approaches. In: *Feature Selection for Data and Pattern Recognition*. [S.l.]: Springer, 2015. p. 29–44. Cited on page 44.

STORCHEUS, D.; ROSTAMIZADEH, A.; KUMAR, S. A survey of modern questions and challenges in feature extraction. In: *Feature Extraction: Modern Questions and Challenges*. [S.l.: s.n.], 2015. p. 1–18. Cited 4 times on page 17, 18, 22, and 68.

SU, Z.-D. et al. iloc-lncrna: predict the subcellular location of lncrnas by incorporating octamer composition into general pseknc. *Bioinformatics*, Oxford University Press, v. 34, n. 24, p. 4196–4204, 2018. Cited on page 39.

SUN, K. et al. iseerna: identification of long intergenic non-coding rna transcripts from transcriptome sequencing data. *BMC genomics*, BioMed Central, v. 14, n. 2, p. S7, 2013. Cited on page 43.

SUN, L. et al. lncrscan-svm: a tool for predicting long non-coding rnas using support vector machine. *PloS one*, Public Library of Science, v. 10, n. 10, p. e0139654, 2015. Cited 3 times on page 26, 27, and 43.

SUN, L. et al. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic acids research*, Oxford University Press, v. 41, n. 17, p. e166–e166, 2013. Cited 4 times on page 26, 31, 59, and 83.

SUN, Y.-Z. et al. Dlrefd: a database providing associations of long non-coding rnas, environmental factors and phenotypes. *Database*, Oxford University Press, v. 2017, 2017. Cited on page 33.

SZCZEŚNIAK, M. W.; ROSIKIEWICZ, W.; MAKAŁOWSKA, I. Cantatadb: a collection of plant long non-coding rnas. *Plant and Cell Physiology*, Oxford University Press, v. 57, n. 1, p. e8–e8, 2015. Cited on page 33.

TRIPATHI, R. et al. Deeplnc, a long non-coding rna prediction tool using deep neural network. *Network Modeling Analysis in Health Informatics and Bioinformatics*, Springer, v. 5, n. 1, p. 21, 2016. Cited 4 times on page 26, 28, 34, and 35.

VALVERDE-ALBACETE, F. J.; PELÁEZ-MORENO, C. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PloS one*, Public Library of Science, v. 9, n. 1, p. e84217, 2014. Cited on page 38.

VENTOLA, G. M. et al. Identification of long non-coding transcripts with feature selection: a comparative study. *BMC bioinformatics*, BioMed Central, v. 18, n. 1, p. 187, 2017. Cited 2 times on page 34 and 35.

VIEIRA, L. M. et al. Plantrna_sniffer: a svm-based workflow to predict long intergenic non-coding rnas in plants. *Non-coding RNA*, Multidisciplinary Digital Publishing Institute, v. 3, n. 1, p. 11, 2017. Cited on page 28.

VINGA, S. Information theory applications for biological sequence analysis. *Briefings in bioinformatics*, Oxford University Press, v. 15, n. 3, p. 376–389, 2013. Cited on page 73.

VOLDERS, P.-J. et al. Lncipedia: a database for annotated human lncrna transcript sequences and structures. *Nucleic acids research*, Oxford University Press, v. 41, n. D1, p. D246–D251, 2012. Cited on page 33.

VOSS, R. F. Evolution of long-range fractal correlations and 1/f noise in dna base sequences. *Physical review letters*, APS, v. 68, n. 25, p. 3805, 1992. Cited on page 69.

WANG, D. et al. Transposable elements (te s) contribute to stress-related long intergenic noncoding rna s in plants. *The Plant Journal*, Wiley Online Library, v. 90, n. 1, p. 133–146, 2017. Cited on page 20.

WANG, H.-L. V.; CHEKANOVA, J. A. Long noncoding rnas in plants. In: *Long Non Coding RNA Biology*. [S.l.]: Springer, 2017. p. 133–154. Cited on page 20.

WANG, L. 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*, IEEE, 2018. Cited on page 19.

WANG, L. et al. Cpat: Coding-potential assessment tool using an alignment-free logistic regression model. *Nucleic acids research*, Oxford University Press, v. 41, n. 6, p. e74–e74, 2013. Cited 2 times on page 26 and 32.

WANG, L.; WANG, Y.; CHANG, Q. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, Elsevier, v. 111, p. 21–31, 2016. Cited 4 times on page 17, 21, 24, and 34.

WANG, X. F. Complex networks: topology, dynamics and synchronization. *International journal of bifurcation and chaos*, World Scientific, v. 12, n. 05, p. 885–916, 2002. Cited on page 75.

WANG, Y. et al. Computational identification of human long intergenic non-coding rnas using a ga–svm algorithm. *Gene*, Elsevier, v. 533, n. 1, p. 94–99, 2014. Cited 3 times on page 19, 34, and 35.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical machine learning tools and techniques - Third Edition.* [S.l.]: Morgan Kaufmann, 2011. Cited 2 times on page 36 and 37.

WUCHER, V. et al. Feelnc: a tool for long non-coding rna annotation and its application to the dog transcriptome. *Nucleic acids research*, Oxford University Press, v. 45, n. 8, p. e57–e57, 2017. Cited on page 42.

XU, C.; JACKSON, S. A. *Machine learning and complex biological data.* [S.l.]: BioMed Central, 2019. Cited on page 17.

XU, H. et al. Length of the orf, position of the first aug and the kozak motif are important factors in potential dual-coding transcripts. *Cell research*, Nature Publishing Group, v. 20, n. 4, p. 445, 2010. Cited on page 32.

XUAN, H. et al. Plnlncrbase: a resource for experimentally identified lncrnas in plants. *Gene*, Elsevier, v. 573, n. 2, p. 328–332, 2015. Cited 2 times on page 33 and 39.

XUE, B. et al. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 20, n. 4, p. 606–626, 2016. Cited on page 18.

YANG, C. et al. Lncadeep: An ab initio lncrna identification and functional annotation tool based on deep learning. *Bioinformatics*, 2018. Cited on page 35.

YIN, C.; CHEN, Y.; YAU, S. S.-T. A measure of dna sequence similarity by fourier transform with applications on hierarchical clustering. *Journal of theoretical biology*, Elsevier, v. 359, p. 18–28, 2014. Cited on page 68.

YIN, C.; YAU, S. S.-T. A fourier characteristic of coding sequences: origins and a non-fourier approximation. *Journal of computational biology*, Mary Ann Liebert, Inc. 2 Madison Avenue Larchmont, NY 10538 USA, v. 12, n. 9, p. 1153–1165, 2005. Cited on page 68.

YIN, C.; YAU, S. S.-T. Prediction of protein coding regions by the 3-base periodicity analysis of a dna sequence. *Journal of theoretical biology*, Elsevier, v. 247, n. 4, p. 687–694, 2007. Cited on page 73.

YU, N.; LI, Z.; YU, Z. Survey on encoding schemes for genomic data representation and feature learning—from signal processing to machine learning. *Big Data Mining and Analytics*, TUP, v. 1, n. 3, p. 191–210, 2018. Cited 2 times on page 69 and 72.

ZAHA, A.; FERREIRA, H. B.; PASSAGLIA, L. M. *Biologia Molecular Básica-5.* [S.l.]: Artmed Editora, 2014. Cited on page 19.

ZHANG, C.-T. A symmetrical theory of dna sequences and its applications. *Journal of theoretical biology*, Elsevier, v. 187, n. 3, p. 297–306, 1997. Cited 2 times on page 71 and 72.

ZHANG, Q. et al. The characteristic landscape of lncrnas classified by rbp–lncrna interactions across 10 cancers. *Molecular bioSystems*, Royal Society of Chemistry, v. 13, n. 6, p. 1142–1151, 2017. Cited on page 20.

ZHANG, R.; ZHANG, C.-T. Z curves, an intutive tool for visualizing and analyzing the dna sequences. *Journal of Biomolecular Structure and Dynamics*, Taylor & Francis, v. 11, n. 4, p. 767–782, 1994. Cited 2 times on page 69 and 71.

ZHANG, W. et al. The linear neighborhood propagation method for predicting long non-coding rna–protein interactions. *Neurocomputing*, Elsevier, v. 273, p. 526–534, 2018. Cited on page 19.

ZHANG, Y.; TAO, Y.; LIAO, Q. Long noncoding rna: a crosslink in biological regulatory network. *Briefings in bioinformatics*, 2017. Cited on page 19.

ZHANG, Y.-C. et al. Genome-wide screening and functional analysis identify a large number of long noncoding rnas involved in the sexual reproduction of rice. *Genome biology*, BioMed Central, v. 15, n. 12, p. 512, 2014. Cited on page 20.

ZHOU, Q.-Z. et al. Bmncrnadb: a comprehensive database of non-coding rnas in the silkworm, bombyx mori. *BMC bioinformatics*, BioMed Central, v. 17, n. 1, p. 370, 2016. Cited on page 19.

ZHU, L. et al. Improving the accuracy of predicting disulfide connectivity by feature selection. *Journal of computational chemistry*, Wiley Online Library, v. 31, n. 7, p. 1478–1485, 2010. Cited on page 52.

ZITNIK, M. et al. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, v. 50, p. 71 – 91, 2019. ISSN 1566-2535. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1566253518304482>. Cited on page 17.