UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

BioAutoML: Democratizing Machine Learning in Life Sciences

Robson Parmezan Bonidia

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Robson Parmezan Bonidia

BioAutoML: Democratizing Machine Learning in Life Sciences

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

USP – São Carlos
February 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Bonidia, Robson Parmezan
B715b BioAutoML: Democratizing Machine Learning in Life Sciences / Robson Parmezan Bonidia; orientador André Carlos Ponce de Leon Ferreira de Carvalho. -- São Carlos, 2024.
165 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Machine Learning. 2. Bioinformatics. 3. Metalearning. 4. Biological Sequences. 5. Automated Feature Engineering. I. Carvalho, André Carlos Ponce de Leon Ferreira de, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Robson Parmezan Bonidia

BioAutoML: Democratizando Aprendizado de Máquina nas Ciências da Vida

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Advisor: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

USP – São Carlos
Fevereiro de 2024

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and assistance of many individuals who have contributed in various ways to the preparation and completion of this study. Firstly, I would like to thank God, the greatest teacher anyone could ever know, for providing me with ideas, health, and strength in difficult times to overcome and face challenges.

My sincere thanks go to my family, for their continued encouragement and support throughout my studies. Special appreciation is extended to the seven most influential people in my life: my wife, Amanda; my daughter, Maria Laura; my parents, Ivone and Osmar; my godchildren, Lorena and Davi; and my grandmother, Olga.

I also want to express my deep gratitude to my advisor, Dr. André Carlos Ponce de Leon Ferreira de Carvalho, for his invaluable guidance throughout the course of my thesis. Dr. Carvalho is not only a model of academic excellence, but also one of the most remarkable people I have ever met. His boundless energy and enthusiasm not only motivated me, but also profoundly shaped my approach to research.

To professors and friends, Anderson Avila-Santos, Breno Almeida, Dr. Ulisses Nunes da Rocha, and Dr. Danilo Sanches, for the valuable ideas, encouragement, companionship, and understanding. I also thank all the professors who accompanied us and passed their knowledge and experiences.

I am also grateful to Dr. Ulisses Nunes da Rocha, as well as to the Helmholtz Centre for Environmental Research – UFZ in Leipzig, for providing me with the opportunity to apply our research in real-world scenarios and refine its functionalities to meet the practical needs of biologists and researchers in the field.

Lastly, but certainly not least, I extend my gratitude to the University of São Paulo, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Google Latin America Research Awards (LARA), the Helmholtz Visiting Researcher Grant/Award, and the FEMS Research & Training Grant/Award, for their support of 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

BONIDIA, R. P. **BioAutoML: Democratizing Machine Learning in Life Sciences**. 2024. 165 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

Recent technological advances allowed an exponential expansion of biological sequence data, and the extraction of meaningful information through Machine Learning (ML) algorithms. This knowledge improved the understanding of the mechanisms related to several fatal diseases, e.g., Cancer and COVID-19, helping to develop innovative solutions, such as CRISPR-based gene editing, coronavirus vaccine, and precision medicine. These advances benefit our society and economy, directly impacting people's lives in various areas, such as health care, drug discovery, forensic analysis, and food analysis. Nevertheless, ML approaches applied to biological data require representative, quantitative, and informative features. Necessarily, as many ML algorithms can handle only numerical data, sequences need to be translated into a feature vector. This process, known as feature extraction, is a fundamental step for the elaboration of high-quality ML-based models in bioinformatics, by allowing the feature engineering stage, with the design and selection of suitable features. Feature engineering, ML algorithm selection, and hyperparameter tuning are often manual and time-consuming processes, requiring extensive domain knowledge, and performed manually by a human expert. To deal with this problem, we developed a new package, BioAutoML, which automatically runs an end-to-end ML pipeline. BioAutoML extracts numerical and informative features from biological sequence databases, automating feature selection, recommendation of ML algorithm(s), and tuning of hyperparameters, using Automated ML (AutoML). BioAutoML has two components, divided into four modules, (1) automated feature engineering (feature extraction and selection modules) and (2) Metalearning (algorithm recommendation and hyperparameter tuning modules). Our experimental results, assessing the relevance of our proposal, indicate robust results for different problem domains, such as SARS-CoV-2, anticancer peptides, HIV sequences, and non-coding RNAs. According to our systematic review, our proposal is innovative compared to available studies in the literature, being the first study to propose automated feature engineering and metalearning for biological sequences. BioAutoML has a high potential to significantly reduce the expertise required to use ML pipelines, aiding researchers in combating diseases, particularly in low- and middle-income countries. This initiative can provide biologists, physicians, epidemiologists, and other stakeholders with an opportunity for widespread use of these techniques to enhance the health and well-being of their communities.

Keywords: BioAutoML, Automated Feature Engineering, Metalearning, Biological Sequences, MathFeature, Mathematical Descriptors.

RESUMO

BONIDIA, R. P. **BioAutoML: Democratizando Aprendizado de Máquina nas Ciências da Vida**. 2024. 165 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

Avanços tecnológicos recentes permitiram uma expansão exponencial dos dados de sequências biológicas e a extração de informações significativas por meio de algoritmos de Aprendizado de Máquina (AM). Esse conhecimento aprimorou a compreensão dos mecanismos relacionados a várias doenças fatais, como o câncer e a COVID-19, contribuindo para o desenvolvimento de soluções inovadoras, como a edição de genes com base no CRISPR, vacinas contra o coronavírus e medicina de precisão. Esses avanços beneficiam nossa sociedade e economia, impactando diretamente a vida das pessoas em várias áreas, como cuidados de saúde, descoberta de medicamentos, análise forense e análise de alimentos. No entanto, abordagens de AM aplicadas a dados biológicos requerem características representativas, quantitativas e informativas. Necessariamente, uma vez que muitos algoritmos de AM só podem lidar com dados numéricos, as sequências precisam ser traduzidas em um vetor de características. Esse processo, conhecido como extração de características, é uma etapa fundamental para a elaboração de modelos de AM de alta qualidade em bioinformática, permitindo a etapa de engenharia de características, com o design e seleção de características adequadas. A engenharia de características, a seleção de algoritmos de AM e o ajuste de hiperparâmetros são frequentemente processos manuais e demorados, que requerem amplo conhecimento do domínio e são realizados manualmente por um especialista humano. Para lidar com esse problema, desenvolvemos um novo pacote, o BioAutoML, que executa automaticamente um pipeline de AM de ponta a ponta. O BioAutoML extrai características numéricas e informativas de bancos de dados de sequências biológicas, automatizando a seleção de características, a recomendação de algoritmos de AM e o ajuste de hiperparâmetros, usando o Aprendizado de Máquina Automatizado (AutoML). O BioAutoML possui dois componentes, divididos em quatro módulos: (1) engenharia de características automatizada (módulos de extração e seleção de características) e (2) Meta-Aprendizado (módulos de recomendação de algoritmos e ajuste de hiperparâmetros). Nossos resultados experimentais, ao avaliar a relevância de nossa proposta, indicam resultados robustos para diferentes domínios de problemas, como SARS-CoV-2, peptídeos anticancerígenos, sequências de HIV e RNAs não codificadores. De acordo com nossa revisão sistemática, nossa proposta é inovadora em comparação com estudos disponíveis na literatura, sendo o primeiro estudo a propor engenharia de características automatizada e metalearning para sequências biológicas. O BioAutoML tem um alto potencial para reduzir significativamente a expertise necessária para usar pipelines de AM, auxiliando os pesquisadores no combate a doenças, principalmente em países de baixa e média renda. Esta iniciativa pode oferecer aos biólogos, médicos, epidemiologistas e outras partes interessadas a oportunidade de utilizar amplamente essas técnicas para aprimorar a saúde e o bem-estar de suas comunidades.

Palavras-chave: BioAutoML, Engenharia de Características Automatizada, Meta-Aprendizado, Sequências Biológicas, MathFeature, Descritores Matemáticos.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION AND PROBLEM STATEMENT

Artificial Intelligence (AI), specifically Machine Learning (ML) algorithms, has enabled the development of innovative solutions in healthcare, agriculture, forensics, and climate change (PALLATHADKA *et al.*, 2023). Due to the expansion and inherent complexity of biological data, ML methods have also shown broad applicability in the biology field (LIU *et al.*, 2015; GREENER *et al.*, 2021; VOLKAMER *et al.*, 2023). ML algorithms can extract useful and meaningful knowledge from biological sequence data (CHEN *et al.*, 2021), accelerating discoveries, reducing research expenses, and increasing scientific efficiency (SHARMA *et al.*, 2021). These advances directly benefit society, the economy, and people's lives.

Furthermore, ML has been successfully used to mitigate the impact of health-related problems (SHARMA *et al.*, 2021; CANNATARO; HARRISON, 2021; GHANNAM; TECHTMANN, 2021a), e.g., COVID-19 pandemic (CANNATARO; HARRISON, 2021; KAMALOV *et al.*, 2023), cancer diagnosis (PAINULI; BHARDWAJ *et al.*, 2022), and CRISPR/Cas9-based gene-editing technology (LI; ZHANG; TROYANSKAYA, 2021; MITROFANOV *et al.*, 2020). Despite its wide application, designing robust and trustworthy ML solutions usually requires expertise not commonly found in health researchers, causing severe inequalities (AHMED; MULA; DHAVALA, 2020; RUBEIS; DUBBALA; METZLER, 2022). According to (RUBEIS; DUBBALA; METZLER, 2022; VANHORN; ÇOBANOĞLU, 2022), in this context, democratizing AI implies granting accessibility to ML for individuals who are not specialists in the domain, e.g., individuals without a background in data science, mathematics, or informatics.

Consequently, in an era where AI is present in various processes that impact society, it is essential to ensure that its contributions are distributed equitably (SEGER *et al.*, 2023). This democratization must empower each individual, community, or society to contribute proportionately to their aptitude, availability, dedication, and speed, requiring equal opportunities across the world (RUBEIS; DUBBALA; METZLER, 2022). In addition to these challenges, one of the main obstacles to the application of ML algorithms to biological sequences is the unstructured nature of many of these data, since most algorithms, including those that produce interpretable models, only work with structured data. This problem can be dealt with feature extraction techniques to represent the originally unstructured data in a structured format. Nevertheless, the features must capture the relevant information present in the biological sequence, since the predictive performance of the model induced by an ML algorithm strongly depends on the representativeness of the input feature vector (WARING; LINDVALL; UMETON, 2020). These processes often require extensive domain knowledge, performed manually by a human expert, being one of the most time-consuming steps in the ML pipeline (WARING; LINDVALL; UMETON, 2020; AMERIFAR; NOROUZI; GHANDI, 2022).

To mitigate this limitation, Automated Machine Learning (AutoML) methods are being used to democratize access and effective use of ML algorithms by non-experts (KARMAKER *et al.*, 2021). AutoML has been successfully used in biological sequence data, with robust solutions, such as autoBioSeqpy (JING *et al.*, 2020), AutoGenome (LIU *et al.*, 2021), iLearn (CHEN *et al.*, 2019), and iLearnPlus (CHEN *et al.*, 2021). Although these studies mention AutoML, most of them apply general-purpose AutoML tools that do not automate the whole process, known as end-to-end ML, nor take into account the specifics of sequence data. The first two of these tools cover only the data modeling step. The last two, iLearn and iLearnPlus include more steps but do not automate the feature extraction from unstructured data. Thus, this step must be performed by the user, who needs to know how to work with unstructured data, mainly for feature engineering and to have programming skills (NG *et al.*, 2021). However, according to International Data Corporation (IDC)[1], by 2025 about 80% of the data generated will be unstructured, e.g., biological sequences, text, images, audio, and video.

These limitations motivated the development of a novel open-source software package, called BioAutoML[2,3], that can extract features based on different aspects, and automate the feature selection, algorithm(s) recommendation, and hyperparameter tuning for multi-class and binary classification of biological data. BioAutoML is an end-to-end Automated Machine Learning (AutoML) tool for experiments using biological sequences. This thesis seeks answers to the following Research Questions (RQ):

- **RQ1:** How can we effectively represent biological sequences to capture the most relevant information from the original data for ML applications?

- **RQ2:** Can we develop an automated, robust, efficient feature engineering and metalearning pipeline specifically designed for biological sequence data?

Our hypothesis to answer these questions is:

[1] https://www.idc.com/

[2] https://github.com/Bonidia/BioAutoML

[3] https://bonidia.github.io/BioAutoML-WP/

- **Hypothesis:** BioAutoML can recommend efficient and robust pipelines for representing biological sequences, automating feature selection, algorithm recommendation, and hyperparameter tuning. This reduces the time-consuming preprocessing stage while maintaining or improving the performance of predictive models, consequently lowering the expertise required to use ML pipelines for biological sequence analysis.

This proposal reduces the barrier to applying automated feature engineering and metalearning in biological sequences for non-experts, industries, as well as in fields of biology, bioinformatics, or medicine, helping analyze and predict large volumes of sequence data faster.

1.1 Feature Engineering Problem

According to Chou's 5-step rule (CHOU, 2011; LIU *et al.*, 2015), numerically representing biological sequences with an efficient and adequate mathematical expression, is one of the most relevant steps to establish an effective statistical predictor for a biological system. In ML, biological sequences, e.g., DNA/RNA and Protein, must be represented by a fixed number of features (e.g., binary, categorical, or continuous), transforming originally unstructured data into a structured format. Feature extraction or feature encoding is a key step in the construction of high-quality ML-based models, determining the effectiveness of trained models in bioinformatics applications (MUHAMMOD *et al.*, 2019; CHEN *et al.*, 2019; KHATUN *et al.*, 2020). The feature engineering process is a time-intensive step and requires domain knowledge of experts (KHURANA *et al.*, 2016; CHEN *et al.*, 2019; WARING; LINDVALL; UMETON, 2020), being the most time-consuming step, as well as a complex exercise (CHEN *et al.*, 2019). Furthermore, feature extraction generally includes both a feature engineering and a feature selection task.

Therefore, this thesis considers the feature construction as a key step to ML application success, being an inevitable step, mainly in biological sequences preprocessing (MUHAMMOD *et al.*, 2019; ZHANG *et al.*, 2021). In terms of terminology, the feature is synonymous of an input variable or attribute. Nevertheless, several revised studies also use the *feature descriptor* terminology (the majority in our review – see Chapter 2), which is the reason we adopted this term, where a feature descriptor refers to the feature extraction method/technique that can present several measures/values. Finally, we define the automated feature engineering task formally explained as follows:

- Given a set of biological sequence data, D, divided into train (D_{train}) and test (D_{test}), a set of feature descriptors, F_d, where $F_d = [f_{d1}, f_{d2}, \ldots, f_{dn}]$, our goal is to select the best numerical representation, that is, feature vector (V_f), combining different feature descriptors in the training set (D_{train}), using for the evaluation of the best V_f, a heuristic function that considers the most important feature descriptor (I_{fd}).

1.2 Metalearning

One of the main difficulties in the application of ML algorithms to a new dataset is the selection of the most adequate algorithm. Each ML algorithm has an inductive bias, which can be defined by the way it searches for a robust model, i.e., (1) starting with simple models and gradually increasing the complexity of the models until an effective model is found, (2) and the format adopted for the representation, i.e., a model used by a decision tree. Although it can be seen as a limitation, bias is necessary for learning to occur. As a consequence, each algorithm fits better datasets with particular conformations. Thus, there is no champion ML algorithm, that performs better than all others in every situation, but each ML algorithm performs better than others on some datasets, which are not known beforehand (WOLPERT; MACREADY, 1997). An alternative to selecting the best ML algorithm for a new dataset is to use prior knowledge regarding the performance of a set of algorithms in previous learning experiences. This idea is behind a particular approach to metalearning, defined in (BRAZDIL *et al.*, 2022) as learning to learn. According to the authors, metalearning is a research area that investigates how to recommend the most suitable algorithm, or set of algorithms, for a new task. In this study, we use metalearning to:

- Given a set of selected features, recommend the ML algorithm(s) able to induce the best predictive model, which can be a set of algorithms, each one inducing a model, and combine these models into an ensemble (P_{ml}), recommending the best algorithm. Ensemble methods can boost the performance of simple classifiers (e.g., using multiple prediction models for solving the same problem) and have proven their effectiveness in bioinformatics (LIU *et al.*, 2020; HANCOCK; KHOSHGOFTAAR, 2020; HE *et al.*, 2022).

1.3 Objectives

In light of our hypothesis, the main objective is to develop a novel open-source package, BioAutoML, to extract pertinent numerical insights from biological sequences and, employing AutoML, establish an automated feature engineering and metalearning pipeline. As a result, the specific objectives that come to the forefront include:

- To conduct a systematic literature review in the field of feature engineering for biological sequences;

- To develop the first package (called MathFeature) to provide a large and comprehensive set of feature extraction techniques based on mathematical descriptors for DNA, RNA, and Proteins;

- To develop an automated feature construction and metalearning package (called BioAutoML);

- Investigate whether our proposal obtains competitive performance compared to other studies;
- Assess whether MathFeature and BioAutoML can achieve competitive performance with state-of-art methods;
- Apply MathFeature and BioAutoML to challenging problems such as CRISPR-Cas9 system, cancer, and COVID-19;
- To publish results through articles (conference and journal) and writing of the doctoral thesis.

1.4 Justification

To support our proposal, we conducted a systematic literature review (as detailed in Chapter 2), during which we identified 29 studies that developed feature extraction tools, including packages, web servers, and toolkits, to classify biological sequences. These studies collectively presented 173 feature extraction descriptors, categorized into 15 major groups, such as physicochemical properties, proteochemometrics, and amino acid composition. During this review, we identified two significant gaps in the existing research: Firstly, the available studies do not adequately cover mathematical descriptors (e.g., chaos game, Fourier transform, entropy, and graphs). Mathematical descriptors have been proven effective in extracting relevant features from biological sequences (MACHADO; COSTA; QUELHAS, 2011; HOANG; YIN; YAU, 2016; ITO *et al.*, 2018; BONIDIA *et al.*, 2021a), particularly in cases where the problem structure is not well understood (NAEEM *et al.*, 2021). This gap highlights a key area that our proposal aims to address. Secondly, a majority of the existing studies (19 out of 29, or 65.52%) are dedicated to a specific type of sequence. Our proposal intends to offer a more versatile approach that can be applied to various biological sequence types, making it a valuable addition to the field.

Additionally, we assessed whether the reviewed studies utilize AutoML methods, and we identified four packages related to our proposal: iLearn (CHEN *et al.*, 2019), iLearnPlus (CHEN *et al.*, 2021), autoBioSeqpy (JING *et al.*, 2020), and AutoGenome (LIU *et al.*, 2021). However, these packages do not include automated feature engineering. The most closely related package, iLearn, requires users to configure an initial file, including the selection of descriptors and classifiers, which demands domain knowledge. Even its more advanced version, iLearnPlus, requires manually inserting extracted features, lacking the essential feature engineering automation. Therefore, our proposal stands out as the most comprehensive automated solution, encompassing the entire pipeline for biological sequence analysis, from feature engineering to ML algorithm recommendation and hyperparameter tuning, making it accessible even to users with limited programming skills.

1.5 Navigating Equity and Ethical Challenges

ML solutions have been proposed in several domains, e.g., in October 2022, the Food and Drug Administration (FDA) reported 521 AI and ML-enabled medical devices (JOSHI *et al.*, 2022). Nevertheless, many studies have a black-box nature (decisions are not understandable on a human level) (RUDIN, 2019; BABIC *et al.*, 2021), which may reduce AI's trust, accountability, and acceptance. Another concern is that ML models can follow hidden social biases in the data, leading to unfair, harmful, or discriminatory decisions. Some examples of these problems are reported in the literature, e.g., in 2009, genome-wide association studies had more than 96% of participants of European descent (POPEJOY; FULLERTON, 2016), failing on diversity. Other studies reported differences and biases of sex and gender in AI (CIRILLO *et al.*, 2020), seeking your equity. In dermatology and diabetes management, studies have discussed the lack of racial diversity in ML algorithms, with possible risks of health disparities (ADAMSON; SMITH, 2018; PHAM *et al.*, 2021).

In search of responsible solutions, this project follows guidelines proposed in the literature, such as how to develop and use AI responsibly (DIGNUM, 2019), AI for all (RAMOS, 2021), Guidelines for Trustworthy AI (ZHANG; ZHANG, 2023), Ethics of AI (UNESCO), and others. We also adopt the principles of Data-Centric AI (DCAI) (ZHA *et al.*, 2023; WHANG *et al.*, 2023), putting data at the heart of an AI system development process. For such, we applied the FAIR (Findable, Accessible, Interoperable, and Reusable) data principles (WILKINSON *et al.*, 2016), guidelines to improve the Findability, Accessibility, Interoperability, and Reuse of data, aiding scientific advancement and promoting. In addition, this thesis provides documentation to replicate our ML workflows, including (1) libraries and their versions, (2) execution environment, (3) training runs, (4) samples, (5) measures, and (6) predictions made by the model.

1.6 Innovations and Contributions

This thesis represents a significant advance in the application of ML techniques to the analysis of biological sequences, addressing fundamental challenges in the field and offering innovative solutions. The proposal of this thesis not only automates complex tasks but also enables researchers without domain knowledge to apply ML algorithms for sequence data analysis. These studies generated applicable results, demonstrating the considerable potential to substantially decrease the expertise required to operate ML pipelines. The contributions of this research are multifaceted, extending from theoretical advances to practical applications. They are summarized as follows:

- A systematic literature review to present, summarize, and study ML-based feature extraction tools (or packages, web servers, and toolkits) that have as a proposal to provide several feature descriptors for biological sequences classification (DNA, RNA, or Protein);

- Identification of 170 distinct descriptors for the numerical representation of biological sequences;
- A novel feature extraction pipeline using mathematical features;
- A novel feature extraction technique based on Tsallis entropy;
- A novel open-source Python package, named MathFeature. This package provides 37 descriptors, 20 of them are mathematical, and organized into five categories. MathFeature is an extensive and comprehensive set of feature extraction techniques based on mathematical descriptors for encoding DNA, RNA, and Proteins (primary sequence of amino acids) sequences. MathFeature is the first package to provide a large set of features based on mathematical descriptors and also well-known descriptors from other studies with biological sequences;
- The first study to propose an automated feature engineering and metalearning pipeline for ncRNA sequences in bacteria;
- BioAutoML, which to the best of our knowledge, automates the most extensive pipeline to date, encompassing feature engineering, ML algorithm recommendation, and hyperparameter tuning. This comprehensive approach sets a new package in the application of ML to biological sequences.

1.7 Thesis Organization/Outline

Throughout the thesis, several scenarios have been experimentally evaluated and discussed, leading to the development of an efficient and robust automated feature engineering and metalearning package. The topics covered in this thesis are presented in the form of articles (Collection of Articles), except for Chapter 2 and 8. Each chapter of the thesis can be read independently, since the information necessary for its understanding is provided within the chapter itself. Therefore:

- Chapter 2, which follows this introduction, describes a systematic literature review to present, summarize, and study ML-based feature extraction tools (or packages, web servers, and toolkits) that have as a proposal to provide several feature descriptors for numerically represent biological sequences (DNA, RNA, or Protein);
- Chapter 3 proposes a new study of feature extraction approaches based on mathematical features (numerical mapping with Fourier, entropy, and complex networks);
- Chapter 4 presents a Tsallis entropy-based feature extraction approach;

- Chapter 5 proposes a novel open-source Python package, named MathFeature, the first package to provide a large set of features based on mathematical descriptors.

- Chapter 6 develops a new package, BioAutoML, which automatically runs an end-to-end ML pipeline (using AutoML), extracting numerical and informative features from biological sequence databases, and automating feature selection, ML algorithm(s) recommendation and hyperparameters tuning.

- Chapter 7 compiles a series of published articles, consisting solely of abstracts, that have been shaped by the influence of our research;

- Finally, Chapter 8 discusses our findings and further challenges.

CHAPTER

2

HOW TO NUMERICALLY REPRESENT BIOLOGICAL SEQUENCES?

In this chapter, we developed a systematic literature review to present, summarize, and study Machine Learning (ML)-based feature extraction tools (or packages, web servers, and toolkits) that have as a proposal to provide several feature descriptors for biological sequences classification (DNA, RNA, or Protein), that is, without a defined scope, therefore, generalist studies. We propose to answer the following problem:

- **Main Problem:** *How to numerically represent a biological sequence (such as DNA, RNA, or protein) in a numeric vector that can effectively reflect the most discriminating information in a sequence?*

Considering this, our review has two main contributions: (1) a systematic literature review of 25 studies and (2) the first study to compile into a single article 173 feature extraction descriptors, divided into 15 large groups.

2.1 Method

This study followed the Systematic Literature Review (SLR) Guidelines in Software Engineering (KEELE *et al.*, 2007), which according to Keele *et al.* (2007) and Brereton *et al.* (2007), allows a rigorous and reliable evaluation of primary studies within a specific topic. This type of review presents a summary of evidence using systematic research methods and information synthesis (KITCHENHAM *et al.*, 2009a). Moreover, this methodology has been widely used in several fields. We base our review on recommendations from previous studies (KEELE *et al.*, 2007; BRERETON *et al.*, 2007; KITCHENHAM *et al.*, 2009a), which divide this process into three stages: planning, conducting, and reporting.

2.1.1 Planning the Review

The planning stage covers the identification of the need for a review, the definition of scientific questions, the identification of databases, the definition of keywords, search strategies, inclusion, exclusion, and quality criteria (KEELE *et al.*, 2007; BRERETON *et al.*, 2007).

Identification of the Need for a Review

To define the need for a systematic review, we apply the following search string to the PubMed database.

- *("feature extraction" OR "extraction" OR "features" OR "feature generation" OR "feature vectors") AND ("tool" OR "web server" OR "package" OR "toolkit") AND ("biological sequence" OR "sequences") AND ("review" OR "systematic review" "OR overview" OR "state of the art" OR "systematic mapping")*

It is important to emphasize that we consider PubMed our main database, because, according to (FALAGAS *et al.*, 2008; TOBER, 2011; KHARE; LEAMAN; LU, 2014), this database is widely used in biomedical and life sciences literature (our field of application). As a result, the search string returned 97 studies, but without any secondary study related to the proposed theme.

Review Protocol

Our main aim is to present, summarize, and study ML-based feature extraction tools (or packages, web servers, and toolkits) that have as a proposal to provide several feature descriptors for biological sequences classification (DNA, RNA, or Protein), that is, without a defined scope, therefore, generalist tools. Considering this, we formulated the following questions:

- **RQ1:** Which are the feature extraction tools for biological sequences?
  - **RQ1-A:** Which is the overview of selected studies?
- **RQ2:** Which biological sequences are most covered by the tools (DNA, RNA, or Protein)?
- **RQ3:** Which descriptors or numerical representations are provided by the tools?
- **RQ4:** Do the tools only provide the feature extraction phase or cover other ML steps like feature selection, classification, and performance analysis?
  - **RQ4-A:** If the tools cover other phases of the biological sequence classification, what are they?

Generally, inclusion, exclusion, and quality criteria are determined after defining the research questions. Therefore, we have established the following criteria:

- **Inclusion criteria:**
    1. Studies in English
    2. Studies with different feature extraction methods for biological sequences
    3. Studies with generalist tools, that is, that do not have a specific problem
    4. Studies published in journal
- **Exclusion criteria:**
    1. Studies that do not use ML techniques and feature extraction
    2. Studies written in a language other than English
    3. Duplicate studies
    4. Studies that are outside the scope of the review
    5. Specific studies on some problems of biological sequences classification
- **Quality criteria**
    1. Are the study aims clearly specified?
    2. Are the feature extraction methods adequately detailed?
    3. Study with different proposals/results?
    4. Study with complete results?
    5. Does the study provide a tool (or package, web server, and toolkit)?
    6. Is the study within the scope, that is, generalist tools?

To guarantee the quality and reliability of the review, all articles found will be analyzed according to Title, Abstract, Keywords, Proposed Approach, Results, and Conclusion. Moreover, we use the following electronic databases:

- **ACM Digital Library:** dl.acm.org
- **IEEE Xplore Digital Library:** ieeexplore.ieee.org
- **PubMed:** https://pubmed.ncbi.nlm.nih.gov/
- **Scopus:** https://www.scopus.com

Finally, we chose the Boolean method (KARIMI *et al.*, 2010) to search primary studies in the literature databases. The standard search string was:

- *("feature extraction" OR "extraction" OR "features" OR "feature generation" OR "feature vectors") AND ("machine" OR "learning") AND ("tool" OR "web server" OR "package" OR "toolkit") AND ("biological sequence" OR "sequence")*

Due to different query languages and limitations between the scientific articles databases, there were some differences in the search strings.

2.1.2 Conducting the Review

In this section, three stages are presented, among them: (1) search in databases, (2) exclusion of repeated studies, and (3) application of inclusion, exclusion, and quality criteria (KEELE *et al.*, 2007; BRERETON *et al.*, 2007). The (4) evaluation of all selected studies and (5) data synthesis will be reported in Results and Discussion. For better understanding, Figure 1 shows our SLR workflow. So, our first step was to apply search keys to all databases, returning a set of 1404 studies. Furthermore, to assist our review and obtain better accuracy and reliability, we use the Parsifal tool.

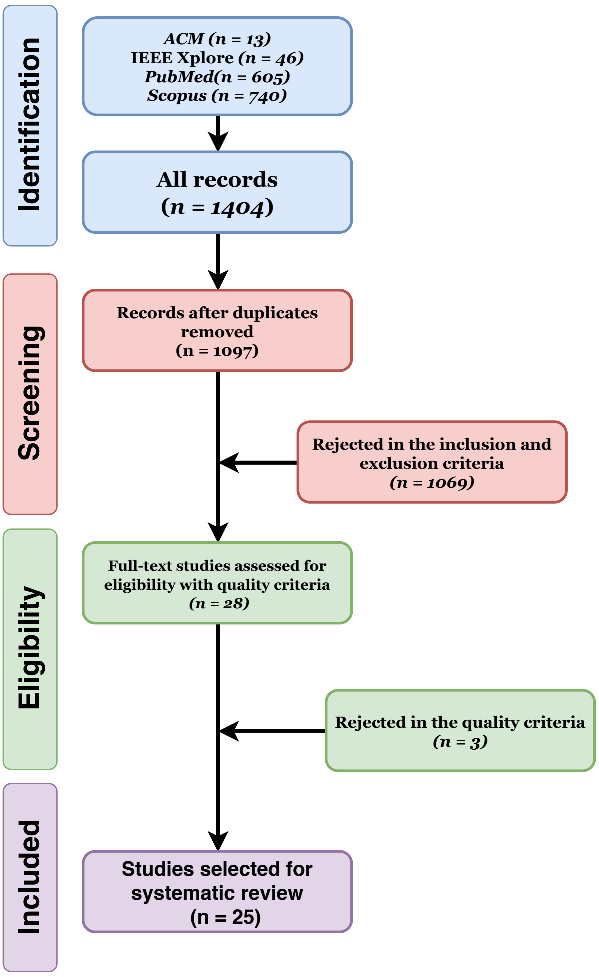

Identification
ACM (n = 13)
IEEE Xplore (n = 46)
PubMed(n = 605)
Scopus (n = 740)
All records
(n = 1404)
Screening
Records after duplicates removed
(n = 1097)
Rejected in the inclusion and exclusion criteria
(n = 1069)
Eligibility
Full-text studies assessed for eligibility with quality criteria
(n = 28)
Rejected in the quality criteria
(n = 3)
Included
Studies selected for systematic review
(n = 25)

Figure 1 – SLR Workflow. Based on Stafford *et al.* (2020), our study selection methodology was divided into four stages: Identification, Screening, Eligibility, and Inclusion.

Thereafter, duplicate studies were removed, returning an amount of 1097 titles (307 duplicate studies). Then, we perform a thorough analysis on the titles, keywords, and abstracts, according to the inclusion and exclusion criteria, in which we accept 28 studies (we reject 1069). Finally, after pre-selecting the studies, we performed a data synthesis, to apply an assessment based on the quality criteria. Hence, of the 28 studies, 3 were eliminated, leading to a final set of 25 studies.

2.1.3 Data extraction

To analyze the selected studies, we generated a data extraction form (see Table 1) in order to answer the research questions involved in this review.

Table 1 – Data Extraction Form.

Question	Description
RQ1	Title
RQ1	Journal
RQ1	Publication Year
RQ1	Citations
RQ1	Country (First Author)
RQ1	Web-page - Tool
RQ2	Application (e.g., DNA, RNA, or Protein)
RQ3	Feature descriptors (central review issue)
RQ4	Other ML steps

2.1.4 Threats to Validity

According to Wen *et al.* (2012), we have main threats to validity of our review, such as (1) study selection bias and (2) inaccurate data extraction. The selection of primary studies depends on the search strategy, the databases, and selection criteria. Thus, to overcome the problem of study selection bias, as suggested by Keele *et al.* (2007) and Brereton *et al.* (2007), we have elaborated a review protocol extremely linked to the research questions, using a control group with more than 10 studies. To overcome the threat of inaccurate data extraction, we elaborated, in a discussion among all researchers, specialized fields for data extraction, totally linked to research questions.

2.2 Results and Discussion

This section reports and discusses the findings of this review. Thereby, we present an overview of the selected studies, followed by a discussion of the findings according to the research questions (separated by subsections). Moreover, during the discussion, we report our results in a broader context related to the research questions.

2.2.1 Feature Extraction Tools for Biological Sequences - RQ1

Initially, after extracting the data fields exposed in Table 1, we generate a new table, presenting all selected studies, together with an ID for each study (e.g., *SN*), title, journal of publication, year and reference, as can be seen in Table 2.

Table 2 – Overview of selected studies

ID	Title	Journal	Year	Ref
S1	PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence	Nucleic Acids Research	2006	(LI *et al.*, 2006)
S2	PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition	Analytical biochemistry	2008	(SHEN; CHOU, 2008)
S3	Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence	Nucleic Acids Research	2011	(RAO *et al.*, 2011)
S4	propy: a tool to generate various modes of Chou's PseAAC	Bioinformatics	2013	(CAO; XU; LIANG, 2013)
S5	PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions	Bioinformatics	2014	(CHEN *et al.*, 2014b)
S6	PseKNC: A flexible web server for generating pseudo K-tuple nucleotide composition	Analytical Biochemistry	2014	(CHEN *et al.*, 2014a)
S7	SPiCE: a web-based tool for sequence-based protein classification and exploration	BMC bioinformatics	2014	(BERG *et al.*, 2014)
S8	protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences	Bioinformatics	2015	(XIAO *et al.*, 2015)
S9	ProFET: Feature engineering captures high-level protein functions	Bioinformatics	2015	(OFER; LINIAL, 2015)

S10	Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences	Nucleic Acids Research	2015	(LIU *et al.*, 2015)
S11	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	2015	(LIU *et al.*, 2014)
S12	Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions	Bioinformatics	2015	(CAO *et al.*, 2014)
S13	DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding	Bioinformatics	2015	(CHIU *et al.*, 2015)
S14	repRNA: a web server for generating various feature vectors of RNA sequences	Mol Genet Genomics	2016	(LIU *et al.*, 2016)
S15	Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences	Natural Science	2017	(LIU *et al.*, 2017)
S16	POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles	Bioinformatics	2017	(WANG *et al.*, 2017)
S17	BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches	Briefings in Bioinformatics	2017	(LIU, 2017)
S18	iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences	Bioinformatics	2018	(CHEN *et al.*, 2018)

S19	PROSES: A Web Server for Sequence-Based Protein Encoding	Journal of Comput. Biology	2018	(KÖSESOY; GÖK; ÖZ, 2018)
S20	PyBioMed: a python library for various molecular representations of chemicals, proteins and DNAs and their interactions	Journal of Cheminformatics	2018	(DONG *et al.*, 2018)
S21	PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences	Bioinformatics	2019	(MUHAMMOD *et al.*, 2019)
S22	BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches	Nucleic Acids Research	2019	(LIU; GAO; ZHANG, 2019)
S23	Seq2Feature: a comprehensive web-based feature extraction tool	Bioinformatics	2019	(NIKAM; GROMIHA, 2019)
S24	iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data	Briefings in Bioinformatics	2019	(CHEN *et al.*, 2019)
S25	Physicochemical n-Grams Tool: A tool for protein physicochemical descriptor generation via Chou's 5-step rule	Chemical Biology and Drug Design	2020	(VISHNOI; GARG; ARORA, 2020)

As can be seen, we present an overview of all selected studies that developed feature extraction tools (or packages, web servers, and toolkits). Moreover, in Figure 2, we plot a word cloud of the studies under review, where the size of each word reflects its frequency of occurrence. This cloud is based on the words contained in the titles of the selected studies. The preponderance of words refers to the purpose of this review (e.g., package, generating, web server, extraction, features, numerical, among others), indicating that the type of most discussed sequence in the literature is Protein, followed by DNA and RNA (see Figure 6). In addition, we generate bar graphs to observe the distribution of selected studies per year (see Figure 3), per journal (see Figure 4), and per country (see Figure 5).

In which, we can observe countless relevant information. For example, we note that 2015

Figure 2 – Words that frequently occur in studies under review.

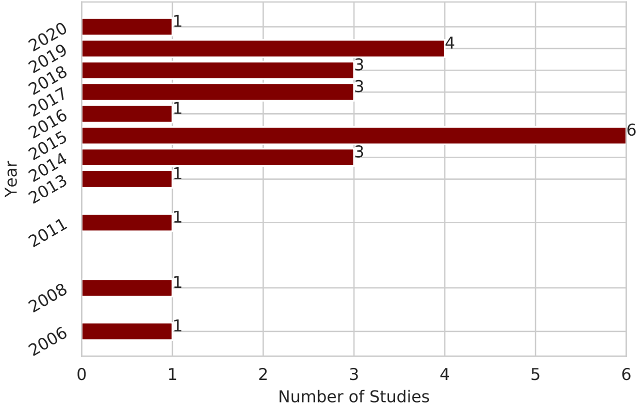


Figure 3 – Number of studies per year.

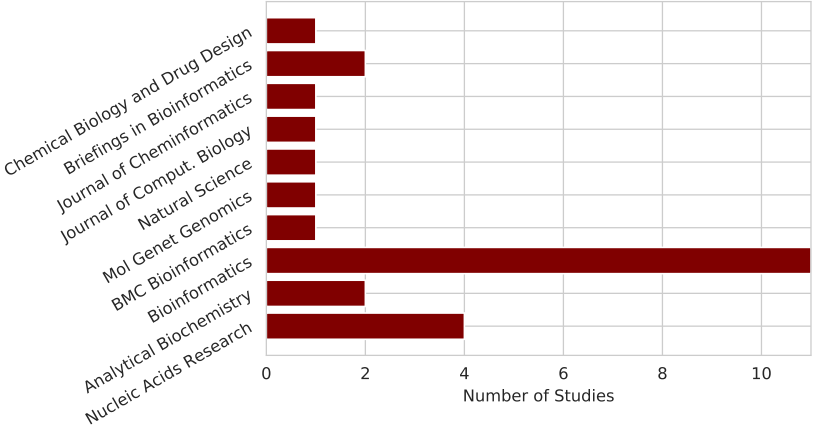


Figure 4 – Number of studies per journal.

and 2019 have the largest number of studies. Regarding the number of studies per journal, we noticed that the majority are contained in two, with an emphasis on Bioinformatics (Oxford Academic) with 44% of selected studies (11), followed by Nucleic Acids Research (4 studies),

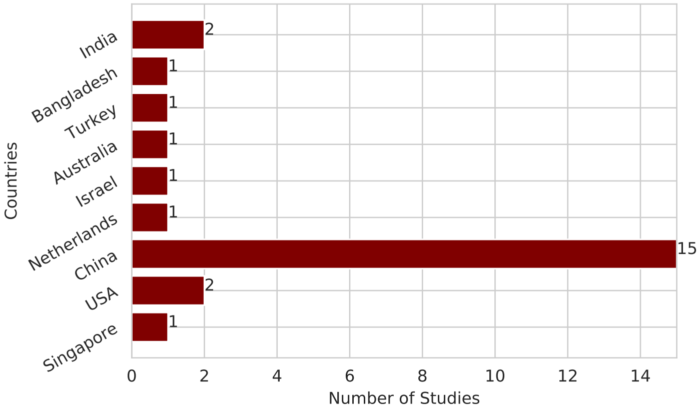


Figure 5 – Number of studies per country.

Analytical Biochemistry (2 studies), and Briefings in Bioinformatics (also 2 studies). Now, regarding the number of studies by country (considering the first author), we observe a domain of Chinese researchers, with 60% (15), followed by the USA and India (2 studies each). Finally, we present a final analysis of the tools available for all studies, as shown in Table 3, to check the availability of each contribution. Fundamentally, the tables were divided into the study, link, active (check if the tool is available), and source (where the active link was found).

As can be seen, all studies provide a tool (or package, web server, and toolkit), according to the main objective raised in this review. In which 92% of the links are available, with the exception of the study S1 and S3, where we did not find any reference there is an active link.

2.2.2 Biological Sequences Covered by the Studies - RQ2

Here, with the complete definition of a biological sequence, we can divide the selected studies into application categories (that is, DNA, RNA, or Protein), as exposed in Table 4. Furthermore, we also plotted a Venn Diagram with the union of the studies by application. As can be seen, the vast majority of studies are dedicated solely to generating various numerical representation schemes for protein sequences, representing 48% of the studies, followed by the application in all sequences with 6 studies. Also, looking at the Venn Diagram, we noticed that 20 studies are dedicated to Protein sequences, 12 for DNA sequences, and 9 for RNA sequences.

2.2.3 Feature Descriptors Provided by the Studies - RQ3

Fundamentally, as previously mentioned, our goal was to evaluate generalist tools for feature extraction, since this type of study would provide several approaches, presenting a general bias of ways to numerically represent biological sequences (which would not be possible by evaluating studies dedicated to some specific problem). As expected, we found hundreds of feature descriptors. Nevertheless, it is not feasible to individually analyze and describe each feature, so we chose, as observed in the studies, to divide into large groups (16 groups - these

Table 3 – Access link to tools.

Study	Link	Active	Source
S1	jing.cz3.nus.edu.sg/cgi-bin/prof/prof.cgi	No	Study
S2	www.csbio.sjtu.edu.cn/bioinf/PseAAC/	Yes	Internet
S3	bidd.cz3.nus.edu.sg/cgi-bin/prof/protein/profnew.cgi	No	Study
S4	code.google.com/archive/p/protpy/downloads	Yes	Study
S5	lin-group.cn/server/pseknc	Yes	Internet
S6	lin-group.cn/server/pseknc	Yes	Internet
S7	github.com/basvandenberg/spiceweb	Yes	Study
S8	protrweb.scbdd.com/	Yes	Study
S9	github.com/ddofer/ProFET	Yes	Study
S10	bioinformatics.hitsz.edu.cn/Pse-in-One/	Yes	Study
S11	bioinformatics.hitsz.edu.cn/repDNA/home	Yes	Study
S12	bioconductor.org/packages/release/bioc/html/Rcpi.html	Yes	Study
S13	tsupeichiu.github.io/DNAshapeR/	Yes	Study
S14	bioinformatics.hitsz.edu.cn/repRNA/	Yes	Study
S15	bioinformatics.hitsz.edu.cn/Pse-in-One2.0/	Yes	Study
S16	possum.erc.monash.edu/	Yes	Study
S17	bioinformatics.hitsz.edu.cn/BioSeq-Analysis	Yes	Study
S18	ifeature.erc.monash.edu/	Yes	Study
S19	proses.yalova.edu.tr/help.html	Yes	Study
S20	projects.scbdd.com/pybiomed.html	Yes	Study
S21	github.com/mrzResearchArena/PyFeat/	Yes	Study
S22	bliulab.net/BioSeq-Analysis2.0	Yes	Study
S23	iitm.ac.in/bioinfo/SBFE/index.html	Yes	Study
S24	ilearn.erc.monash.edu/	Yes	Study
S25	14.139.57.41/pngt/download.html	Yes	Study

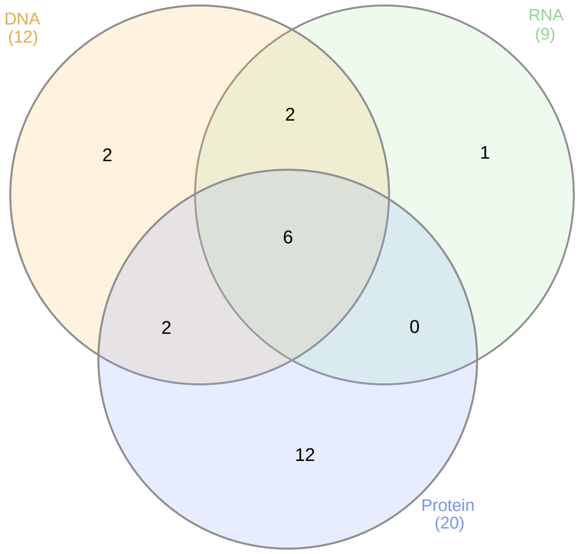


Figure 6 – Venn Diagram - Union of selected studies by application.

were defined based on all studies), as shown in Table 5.

Then, we elaborated three tables dividing all the feature descriptors found in the 25

Table 4 – Selected studies by application.

Application	Study
DNA	(CHIU *et al.*, 2015), (LIU *et al.*, 2014)
RNA	(LIU *et al.*, 2016)
Protein	(LI *et al.*, 2006), (SHEN; CHOU, 2008), (RAO *et al.*, 2011), (CAO; XU; LIANG, 2013), (BERG *et al.*, 2014), (XIAO *et al.*, 2015), (OFER; LINIAL, 2015), (CAO *et al.*, 2014), (CHEN *et al.*, 2018), (WANG *et al.*, 2017), (KÖSESOY; GÖK; ÖZ, 2018), (VISHNOI; GARG; ARORA, 2020)
DNA + RNA	(CHEN *et al.*, 2014a), (CHEN *et al.*, 2014b)
DNA + Protein	(DONG *et al.*, 2018), (NIKAM; GROMIHA, 2019)
DNA + RNA + Protein	(LIU *et al.*, 2015), (LIU *et al.*, 2017), (LIU, 2017), (MUHAMMOD *et al.*, 2019), (LIU; GAO; ZHANG, 2019), (CHEN *et al.*, 2019)

Table 5 – Descriptors group.

Group	Initials	Application Group
Amino Acid Composition	AAC	Protein
Pseudo-Amino Acid Composition	PseAAC	Protein
Composition, Transition, Distribution	CTD	Protein
Sequence-Order	SO	Protein
Topological Descriptors	TD	Protein
Conjoint Triad	CT	Protein
Proteochemometric Descriptors	PCM	Protein
Profile-based Features	PF	Protein
Nucleic Acid Composition	NAC	DNA, RNA
Pseudo Nucleic Acid Composition	PseNAC	DNA, RNA
Structure Composition	SC	DNA, RNA, Protein
Sequence Similarity	SS	DNA, RNA, Protein
Autocorrelation	AC	DNA, RNA, Protein
Numerical Mapping	NM	DNA, RNA, Protein
K-Nearest Neighbor	KNN	DNA, RNA, Protein
Physicochemical Property	PP	DNA, RNA, Protein

studies, totaling 170 descriptors to numerically represent biological sequences. Thereby, Table 6 represents DNA sequence descriptors, Table 7, RNA sequence descriptors, and Table 8, protein sequence descriptors. Each column in the tables refers to:

- **Group:** This column classifies the feature descriptor in each group shown in Table 5;
- **Descriptor:** Feature descriptors found in each study and classified in their respective

group;

- **Dimension:** Number of features generated by the descriptor (columns). This column is based on the information contained in the revised studies. The "$-$" symbol means that the dimension may vary according to the chosen parameter, or there is no such information in the studies.

- **Study:** Which study provides the descriptor.

As can be seen, for DNA sequences, we found 48 feature descriptors. The descriptors most provided by the studies are basic k-mer, reverse complementary k-mer, increment of diversity, nucleotide and dinucleotide composition, autocorrelation, and pseudo nucleic acid composition. For RNA sequences, 39 feature descriptors are presented, being that most provided by the studies are also nucleic acid composition, autocorrelation, and pseudo nucleic acid composition. Finally, we found 83 feature descriptors for protein. This number of descriptors is due to the number of specific studies for these sequences (48%). Moreover, unlike DNA/RNA (four nitrogenous bases), protein has 20 amino acids, so more information to extract features. The descriptors most provided by the studies are amino acid composition, pseudo-amino acid composition, autocorrelation, CTD, and sequence-order.

Table 6 – Descriptors found in all studies - DNA Sequences

Group	Descriptor	Dimension	Study
NAC	Nucleotide composition	4	S5, S22, S24
	Dinucleotide composition	16	S5, S22, S24
	Trinucleotide composition	64	S5, S22, S24
	Tetranucleotide composition	256	S5
	Pentanucleotide composition	1024	S5
	Hexanucleotide composition	4096	S5
	Basic kmer	4^k	S10, S11, S13, S15, S17, S20, S22, S24
	Reverse complementary kmer	-	S10, S11, S15, S17, S20, S22, S24
	Increment of diversity	$2k$	S11, S15, S17, S22
	Mismatch	-	S15, S17, S22
	Subsequence	-	S15, S17, S22
	GC-content	1	S21
	AT/GT Ratio	1	S21
	Cumulative skew	2	S21
	kGap	-	S21
	Position-specific nucleotide frequency	-	S22, S24
	Nucleotide Content	7	S23
	Conformational properties	18	S23
	Enhanced nucleic acid composition	18	S24
	Composition of k-spaced Nucleic Acid Pairs	-	S22, S24
AC	Normalized Moreau–Broto	240	S5, S15, S17, S22
	Moran	240	S5, S15, S17, S22

	Geary	240	S5, S15, S17, S22
	Dinucleotide-based auto covariance	$N \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
	Dinucleotide-based cross covariance	$N(N-1) \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
	Dinucleotide-based auto-cross covariance	$N^2 \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
	Trinucleotide-based auto covariance	$N \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
	Trinucleotide-based cross covariance	$N(N-1) \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
	Trinucleotide-based auto-cross covariance	$N^2 \cdot LAG$	S10, S11, S15, S17, S20, S22, S24
PseNAC	Type 1 Pseudo k-tuple nucleotide composition	$4^k + \lambda$	S5, S6
	Type 2 Pseudo k-tuple nucleotide composition	$4^k + \lambda \cdot N$	S5, S6
	Pseudo k-tuple nucleotide composition	$4^k + \lambda$	S10, S11, S15, S17, S20, S22, S24
	Pseudo dinucleotide composition	$16 + \lambda$	S10, S11, S15, S17, S20, S22, S24
	General parallel correlation pseudo dinucleotide composition	$16 + \lambda$	S10, S11, S15, S17, S20, S22, S24
	General parallel correlation pseudo trinucleotide composition	$64 + \lambda$	S10, S11, S15, S17, S20, S22, S24
	General series correlation pseudo dinucleotide composition	$16 + \lambda \cdot N$	S10, S11, S15, S17, S20, S22, S24
	General series correlation pseudo trinucleotide composition	$64 + \lambda \cdot N$	S10, S11, S15, S17, S20, S22, S24
SC	DNA shape features	-	S13
NM	Z-curve theory	-	S21, S22
	Nucleotide Chemical Property	-	S22, S24
	Accumulated Nucleotide Frequency	-	S22, S24
	Electron-ion interaction pseudopotential	-	S22, S24
	Pseudo electron-ion interaction pseudopotential	-	S22, S24
	Binary	-	S22, S24

PP	Dinucleotide physicochemical	-	S22, S23
	Trinucleotide physicochemical	-	S22
SS	BLAST-matrix	-	S22

Table 7 – Descriptors found in all studies - RNA Sequences

Group	Descriptor	Dimension	Study
NAC	Nucleotide composition	4	S5, S14, S22, S24
	Dinucleotide composition	16	S5, S14, S22, S24
	Trinucleotide composition	64	S5, S14, S22, S24
	Tetranucleotide composition	256	S5, S14
	Pentanucleotide composition	1024	S5, S14
	Hexanucleotide composition	4096	S5, S14
	Basic kmer	4^k	S10, S15, S17, S22, S24
	Reverse complementary kmer	-	S24
	Mismatch	-	S15, S17, S22
	Subsequence	-	S15, S17, S22
	GC-content	1	S21
	AT/GT Ratio	1	S21
	Cumulative skew	2	S21
	kGap	-	S21
	Position-specific nucleotide frequency	-	S22, S24
	Enhanced nucleic acid composition	-	S24
	Composition of k-spaced nucleic acid pairs	-	S22, S24
AC	Normalized Moreau–Broto	240	S5, S15, S17, S22
	Moran	240	S5, S15, S17, S22
	Geary	240	S5, S15, S17, S22
	Dinucleotide-based auto covariance	$N \cdot LAG$	S10, S15, S17, S22, S24
	Dinucleotide-based cross covariance	$N(N-1) \cdot LAG$	S10, S15, S17, S22, S24

	Dinucleotide-based auto-cross covariance	$N^2 \cdot LAG$	S10, S15, S17, S22, S24
PseNAC	Type 1 Pseudo k-tuple nucleotide composition	$4^k + \lambda$	S5, S6
	Type 2 Pseudo k-tuple nucleotide composition	$4^k + \lambda \cdot N$	S5, S6
	Pseudo k-tuple nucleotide composition	$4^k + \lambda$	S24
	Pseudo dinucleotide composition	$16 + \lambda$	S24
	General parallel correlation pseudo dinucleotide composition	$16 + \lambda$	S10, S14, S15, S17, S22, S24
	General series correlation pseudo dinucleotide composition	$16 + \lambda \cdot N$	S10, S14, S15, S17, S22, S24
SC	Triplet	32	S14, S15, S17, S22
	Pseudo-structure status composition	-	S14, S15, S17, S22
	Pseudo-distance structure status pair composition	-	S14, S15, S17, S22
	Secondary structure	-	S22
NM	Z-curve theory	-	S21, S22
	Nucleotide Chemical Property	-	S22, S24
	Accumulated Nucleotide Frequency	-	S22, S24
	Binary	-	S22, S24
PP	Dinucleotide physicochemical	-	S22

Table 8 – Descriptors found in all studies - Protein Sequences

Group	Descriptor	Dimension	Study
AAC	Amino acid composition	20	S1, S3, S7, S8, S9, S12, S18, S19, S20, S22, S24
	Dipeptide composition	400	S1, S3, S7, S8, S9, S12, S18, S19, S20
	Tripeptide composition	8000	S4, S8, S12, S18, S20, S22, S24
	Terminal end amino acid count	20	S7
	Amino acid pair	400	S19
	Secondary structure composition	3	S7
	Secondary structure - amino acid composition	60	S7
	Solvent accessibility composition	2	S7
	Solvent accessibility - amino acid composition	40	S7
	Codon composition	64	S7
	Protein length	1	S7
	Overlapping K-mers	-	S9
	Information-based statistics	-	S9
	Basic kmer	20^k	S10, S15, S17, S22
	Distance-based residue	-	S15, S17, S22
	Distance pair	-	S15, S17, S22
	Residue-Couple Model	-	S19
	Composition moment vector	-	S19
	Enhanced amino acid composition	-	S18, S24
	Composition of k-spaced amino acid pairs	2400	S18, S22, S24
	Dipeptide deviation from expected mean	400	S18
	Grouped amino acid composition	5	S18, S22, S24
	Enhanced grouped amino acid composition	-	S18, S24

	Composition of k-spaced amino acid group pairs	150	S18, S22, S24
	Grouped dipeptide composition	25	S18
	Grouped tripeptide composition	125	S18, S22, S24
	kGap	-	S21
	Position-specific nucleotide frequency	-	S22
PseAAC	Type 1 PseAAC	$20+\lambda$	S2, S3, S4, S7, S8, S10, S12, S15, S17, S18, S20, S22, S24
	Type 2 PseAAC	$20+\lambda \cdot N$	S2, S3, S4, S7, S8, S10, S12, S15, S17, S18, S20, S22, S24
	Dipeptide (or Type 3) PseAAC	420	S2
	General parallel correlation PseAAC	$20+\lambda$	S10, S15, S17, S22
	General series correlation PseAAC	$20+\lambda \cdot N$	S10, S15, S17, S22
	Pseudo K-tuple reduced AAC (type1 to type16)	-	S18, S24
AC	Normalized Moreau–Broto	240	S1, S3, S4, S7, S8, S12, S18, S20, S22, S24
	Moran	240	S1, S3, S4, S7, S8, S12, S18, S20, S22, S24
	Geary	240	S1, S3, S4, S7, S8, S12, S18, S20, S22, S24
	Auto covariance	-	S10, S15, S17, S22
	Cross covariance	-	S10, S15, S17, S22
	Auto-cross covariance	-	S10, S15, S17, S22
CTD	Composition	21	S1, S3, S4, S7, S8, S9, S12, S18, S19, S20, S22, S24
	Transition	21	S1, S3, S4, S7, S8, S9, S12, S18, S19, S20, S22, S24
	Distribution	105	S1, S3, S4, S7, S8, S9, S12, S18, S19, S20, S22, S24
SO	Sequence-order-coupling number	60	S1, S3, S4, S8, S12, S18, S20, S22, S24

	Quasi-sequence-order	100	S1, S3, S4, S7, S8, S12, S18, S20, S22, S24
TD	Topological descriptors	405	S3
PF	Signal average	-	S7
	Signal peaks area	-	S7
	PSSM (Position-Specific Scoring Matrix) profile	-	S8, S12, S15, S16, S17, S18, S22, S24
	Profile-based Physicochemical distance	-	S15, S17, S22
	Distance-based Top-n-gram	-	S15, S17, S22
	Top-n-gram	-	S15, S17, S22
	Sequence conservation score	-	S17, S22
	Frequency profiles matrix	-	S22
CT	Conjoint Triad	343	S8, S12, S18, S19, S20, S22, S24
	Conjoint k-spaced triad	$343 \cdot (k+1)$	S18, S24
PCM	Principal components analysis	175	S8, S12
	Principal components analysis (2D and 3D)	4025	S8
	Factor analysis	175	S8, S12
	Factor analysis (2D and 3D)	4025	S8
	Multidimensional scaling	175	S8, S12
	Multidimensional scaling (2D and 3D)	4025	S8
	BLOSUM and PAM matrix-derived	175	S8, S12, S18, S22, S24
	Biophysical quantitative properties	-	S9
	Amino acid properties	-	S12

	Molecular descriptors	-	S12
SS	Gene Ontology (GO) similarity	-	S12
	Sequence Alignment	-	S12
SC	Secondary structure	-	S17, S18, S22, S24
	Solvent accessible surface area	-	S17, S18, S22, S24
	Secondary structure binary	-	S18, S22, S24
	Disorder	-	S9, S18, S24
	Disorder content	-	S18, S24
	Disorder binary	-	S18, S24
	Torsional angles	-	S18, S24
NM	Binary	-	S18, S22, S24
	Orthonormal encoding	-	S19
	6-dimension One-hot method	-	S22
KNN	K-nearest neighbor for proteins	60	S18, S24
PP	AAindex	-	S9, S18, S22, S24
	Z-scale	-	S18, S22, S24
	Physicochemical n-Grams	-	S25

2.2.4 Other ML Steps Covered by Studies - RQ4

As previously mentioned, we must consider several stages to establish an effective statistical predictor for a biological system. Therefore, in this section, we investigate which studies cover another ML process (see Table 9), divided into 5 groups, as shown below.

Table 9 – Other ML steps covered by studies.

Study	DR	FS	Classification	Performance	Visualization
S1	-	-	-	-	-
S2	-	-	-	-	-
S3	-	-	-	-	-
S4	-	-	-	-	-
S5	-	-	-	-	-
S6	-	-	-	-	-
S7	-	-	**V**	**V**	**V**
S8	-	-	-	-	-
S9	**V**	**V**	**V**	**V**	**V**
S10	-	-	-	-	**V**
S11	-	-	-	-	-
S12	-	-	-	-	-
S13	-	-	-	-	**V**
S14	-	-	-	-	-
S15	-	-	-	-	**V**
S16	-	-	-	-	-
S17	-	**V**	**V**	**V**	**V**
S18	**V**	**V**	-	-	-
S19	-	-	-	-	-
S20	-	-	-	-	-
S21	-	-	**V**	**V**	**V**
S22	-	**V**	**V**	**V**	**V**
S23	-	-	-	-	-
S24	**V**	**V**	**V**	**V**	**V**
S25	-	-	-	-	-

- **Dimensionality Reduction (DR):** Algorithms that aim to reduce the dimensionality (features) of high-dimension data in a new subset with new features (low dimension) (YAN *et al.*, 2006).

- **Feature Selection (FS):** This group contains algorithms that address the feature selection problem.

- **Classification:** Classification task is used to predict events with a predefined number of targets (in the classification, there is a categorical or discrete target). Thus, this approach can be applied to analyze sequences, to find an indication of which class they belong to, consequently learning how to classify a new record (SUTHAHARAN, 2016).

- **Performance:** This group refers to tools that provide performance metrics (e.g., sensitivity, specificity, accuracy (ACC), among others.
- **Visualization:** Tools that provide data visualization, that uses static and interactive visuals within a specific context, to assist users in the interpretation of generated data or model performance.

Essentially, 76% of the studies are specifically dedicated to feature extraction, without considering other stages to establish an effective statistical predictor for a biological system. Considering this, some studies (24%) have generated the entire pipeline, focusing on several other processes, as presented above. Therefore, we explore these studies, analyzing which algorithms are provided for the phases of DR, FS, and classification, as exposed in Table 10. It is important to highlight that S9 did not present the complete list of algorithms, so we looked at its code.

Table 10 – Algorithms supported by the studies for DR, FS and classification.

Study	Description
S7	**Classification:** Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Gaussian Naive Bayes (GNB), Decision Tree (DT), Random Forest (RF).
S9	**DR:** Principal Component Analysis (PCA). **FS:** Recursive Feature Elimination (RFE), Univariate Feature Selection (UFS). **Classification:** SVM, RF, AdaBoost, Gradient Boosting (GB), Extra Trees, Logistic Regression (LR), KNN, GNB.
S17	**FS:** Chi-square test (CHI2), Mutual Information (MI). **Classification:** SVM, optimized evidence-theoretic KNN, RF, Covariance discriminant.
S18	**DR:** PCA, Latent Dirichlet allocation, t-Distributed Stochastic Neighbor Embedding (t-SNE). **FS:** CHI2, Information Gain (IG), MI, Pearson's Correlation Coefficient (PCC).
S21	**Classification:** LR, SVM, KNN, DT, GNB, Bagging, RF, AdaBoost, GB and LDA.
S22	**FS:** CHI2, MI **Classification:** SVM, RF, Conditional Random Field (CRF).
S24	**DR:** PCA, t-SNE, Latent Dirichlet allocation **FS:** CHI2, MI, PCC, IG, F-score. **Classification:** SVM, KNN, RF, LR, Artificial Neural Network (ANN).

We note that all studies address classic algorithms for DR and FS, mainly from statistical lines (e.g., PCA, CHI2, MI). While in the classification phase, they provide the most diverse algorithms, both linear and non-linear, classical, and ensembles. A highlight for the S24, which proves the largest number of algorithms (in general).

2.3 Systematic Review Updates and Chapter Remarks

So far, four new studies have been published since our review, iLearnPlus (CHEN *et al.*, 2021), BioSeq-BLM (LI; PANG; LIU, 2021b), autoBioSeqpy (JING *et al.*, 2020), and AutoGenome (LIU *et al.*, 2021). Thereby, we also decided to assess whether any revised studies apply Automated ML (AutoML) approaches, that according to (SÁ *et al.*, 2017), have a proposal similar to the field of hyper-heuristics, automatically recommend pipelines, algorithms, or parameters for specific tasks without much dependence on user knowledge. These tasks can include different ways of preprocessing or feature engineering, as well as algorithms and optimization of its parameters (hyper-parameter tuning) (SÁ *et al.*, 2017; SANTOS *et al.*, 2019). Considering this, Table 11 lists revised studies from Table 9, which cover another stage of ML, and recently published, checking if any follow the proposal of this thesis, automated feature engineering.

Table 11 – Use of AutoML for feature engineering, recommendation of ML algorithm, and hyper-parameter tuning.

Study	**Feature Engineering**	**ML algorithm**	**Tuning**
SPiCE	-	-	-
ProFET	-	-	-
BioSeq-Analysis	-	-	-
iFeature	-	-	-
BioSeq-Analysis2.0	-	-	-
iLearn	-	**V**	**V**
iLearnPlus	-	**V**	**V**
BioSeq-BLM	-	-	-
autoBioSeqpy	-	**V**	**V**
AutoGenome	-	**V**	**V**

The most similar packages to our proposal are iLearn, iLearnPlus, autoBioSeqpy, and AutoGenome, which apply AutoML to recommend ML algorithms, but they do not use automated feature engineering. The most similar package to our proposal, iLearn, requires an initial configuration file (choosing descriptors and classifiers), which needs domain knowledge from a human expert. Even in its most sophisticated version, iLearnPlus, a file needs to be inserted with the extracted features, instead of automatic feature engineering. The autoBioSeqpy and AutoGenome packages focus on recommending the best deep-learning architecture. Thus, to the best of our knowledge, BioAutoML automates the longest pipeline for biological sequence analysis, encompassing feature engineering, ML algorithm recommendation, and hyperparameter tuning.

CHAPTER

3

FEATURE EXTRACTION APPROACHES: A COMPARATIVE STUDY OF MATHEMATICAL FEATURES

As consequence of the various genomic sequencing projects, an increasing volume of biological sequence data is being produced. Although machine learning algorithms have been successfully applied to a large number of genomic sequence-related problems, the results are largely affected by the type and number of features extracted. This effect has motivated new algorithms and pipeline proposals, mainly involving feature extraction problems, in which extracting significant discriminatory information from a biological set is challenging. Considering this, our work proposes a new study of feature extraction approaches based on mathematical features (numerical mapping with Fourier, entropy and complex networks). As a case study, we analyze long non-coding RNA sequences. Moreover, we separated this work into three studies. First, we assessed our proposal with the most addressed problem in our review, e.g. lncRNA and mRNA; second, we also validate the mathematical features in different classification problems, to predict the class of lncRNA, e.g. circular RNAs sequences; third, we analyze its robustness in scenarios with imbalanced data. The experimental results demonstrated three main contributions: first, an in-depth study of several mathematical features; second, a new feature extraction pipeline; and third, its high performance and robustness for distinct RNA sequence classification.

3.1 Background

In recent years, an increasing number of biological sequences have been generated by thousands of sequencing projects (GUO; ZOU, 2019), creating a huge volume of data (HASHEMI *et al.*, 2018a). During the last decade, Machine Learning (ML) methods have shown broad applicability in computational biology and bioinformatics (MIN, 2010; SILVA *et al.*, 2019a). Consequently, several studies have been dedicated to investigating sequences of DNA and RNA molecules (BUDACH; MARSICO, 2018; MIN; LEE; YOON, 2016; CHEN *et al.*, 2019). Applying ML methods in these sequences has helped to extract important information from various datasets to explain biological phenomena (MIN, 2010). The development of efficient approaches benefits the mathematical understanding of the structure of biological sequences (LOU *et al.*, 2019), e.g., Precision cancer diagnostics (MAROS *et al.*, 2020), analytics in plants (MA; ZHANG; WANG, 2014), and Coronavirus epidemic (LI; LIU, 2020; BENVENUTO *et al.*, 2020).

Nevertheless, according to (MIN, 2010; XU; JACKSON, 2019), there are several challenging biological problems that motivated the emergence of proposals for new algorithms. Fundamentally, biological sequence analysis with ML presents one major problem, e.g., Feature Extraction (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015), an inevitable process, especially in the stage of biological sequence preprocessing (CHEN *et al.*, 2019; SAIDI *et al.*, 2012; BONIDIA *et al.*, 2020b). Necessarily, several methods in bioinformatics apply ML algorithms for sequence classification, and as many algorithms can deal only with numerical data, sequences need to be translated into sequences of numbers.

Thereby, modern applications extract relevant features from sequences based on several biological properties, e.g., physicochemical, Open Reading Frames (ORF)-based, usage frequency of adjoining nucleotide triplets, GC content, among others. This approach is common in biological problems, but these implementations are often difficult to reuse or adapt to another specific problem. For example, 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; PAN; XIONG, 2015) (e.g., in (PAN; XIONG, 2015), ORF score (feature importance) is less than 0.009 to classify circular RNA from other types of lncRNAs). Consequently, the feature extraction problem arises, in which extracting a set of useful features that contain significant discriminatory information becomes a fundamental step in the construction of a predictive model (MUHAMMOD *et al.*, 2019).

Therefore, these problems make the process of biological sequence classification a challenging task, creating a growing need to develop new techniques and methods to analyze sequences effectively and efficiently. In this paper, we have investigated the performance of different feature extraction methods for biological sequence analysis, using mathematical features, e.g., numerical mapping with Fourier transform, entropy, and graphs. As a case study, we have used lncRNA sequences, which are fundamentally unable to produce proteins (ABBAS *et al.*, 2016; SZCZEŚNIAK *et al.*, 2020) and have recently casted doubt on its functionality (SZCZEŚNIAK *et al.*, 2020). In addition, lncRNAs present several problem classes, such as: lncRNA vs. mRNA (KANG *et al.*, 2017; HAN *et al.*, 2018), lncRNA vs. circRNA (CHEN *et al.*, 2018), lncRNA vs. small non-coding RNAs, and lncRNA vs. noncoding antisense transcripts. Thus, enabling us to create a scenario to answer the questions raised in this work.

For that reason, the main objective of this paper is to evaluate the ability to generalize mathematical features in different lncRNA classification tasks. Moreover, we assess whether mathematical approaches do not have any dependencies from a specific problem when compared to biological approaches (e.g., those features that present a bias to the problem analyzed or some biological explanation, e.g., ORF for lncRNA vs. mRNA (Parmezan Bonidia *et al.*, 2019; BAEK *et al.*, 2018)). Thereby, we assume the following hypothesis:

- **Hypothesis:** Feature extraction approaches based on mathematical features are generalist, i.e., the ability to generalize mathematical features in different ncRNA types, such as the classification of lncRNA subclasses, being as efficient as biological approaches.

Considering this, our work presents new ideas and analysis for the feature extraction problem in biological sequences, with four main contributions: (1) A new feature extraction pipeline using mathematical features; (2) Study of 9 mathematical approaches; (3) Analysis of 6 numerical mappings with Fourier, proposing statistical measures; (4) The ability to generalize mathematical features in different ncRNA types, such as the classification of lncRNA subclasses.

3.2 Related Works

Essentially, as emphasized, we adopt lncRNA sequences as a case study, a class of Non-Coding RNAs (ncRNAs). In this context, we have conducted an in-depth review of the lncRNAs classification methods, in which several approaches have been developed, such as: CPC (versions 1 and 2) (KONG *et al.*, 2007; KANG *et al.*, 2017), 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), PlantRNA_Sniffer (VIEIRA *et al.*, 2017), PLncPRO (SINGH *et al.*, 2017), RNAplonc (NEGRI *et al.*, 2018), BASiNET (ITO *et al.*, 2018), LncFinder (HAN *et al.*, 2018), CREMA (SIMOPOULOS; WERETILNYK; GOLDING, 2018), LncRNAnet (BAEK *et al.*, 2018), CNIT (GUO *et al.*, 2019), PLIT (DESHPANDE *et al.*, 2019), PredLnc-GFStack (LIU *et al.*, 2019), LGC (WANG *et al.*, 2019) and DeepCPP (ZHANG *et al.*, 2020). For better understanding, Figure 7 presents theses works divided into Mathematical, Biological, and Hybrid approaches.

In general, the aforementioned studies apply supervised learning methods using binary classification (two classes - lncRNAs and protein-coding genes (mRNA)). There is a considerable amount of research on humans, followed by animals and plants. Regarding feature extraction, we observed a full domain of ORF and sequence-structure descriptors. As seen in Figure 7, there is a frequent use of biological features. On the other hand, some works have explored mathematical approaches for feature extraction, such as Genomic Signal Processing (GSP), DNA Numerical Representation (DNR) (PIAN *et al.*, 2016; HAN *et al.*, 2018), and Complex Networks (ITO *et al.*,

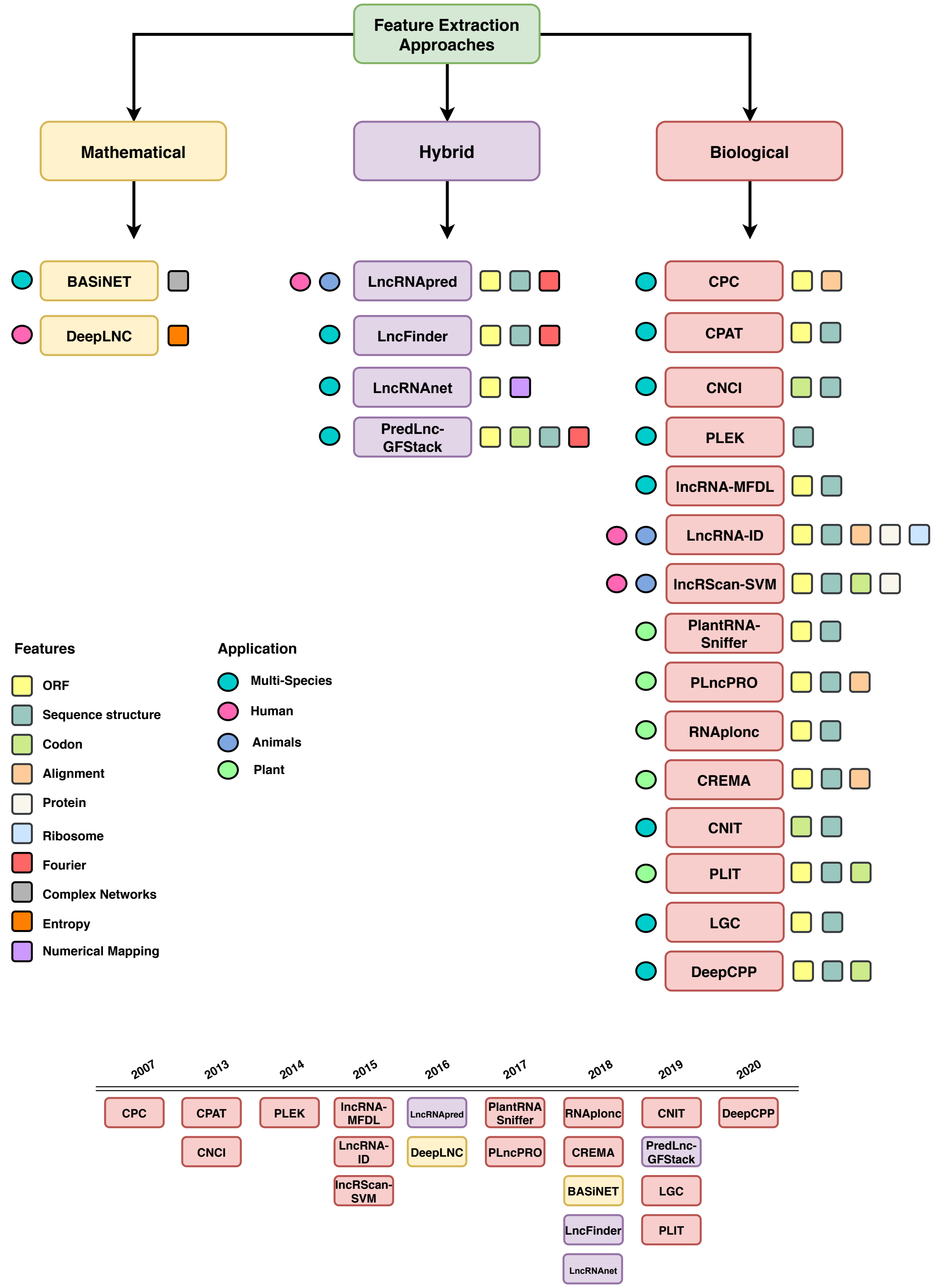


Figure 7 – Feature extraction approaches in our case study divided into: Mathematical, Biological, and Hybrid.

2018). Nevertheless, the authors used these attributes in conjunction with other biological feature extraction techniques or without testing other mathematical features. Practically no papers have focused on several mathematical approaches. Based on this, the objective of this section was to summarize the main methods of the literature and their characteristic descriptors. Therefore, we will not use the studies shown for comparison, but the most often used features.

3.3 Materials and Methods

In this section, we describe the methodological approach used to achieve the proposed objectives, as shown in Figure 8. Essentially, we divided our study into five stages: (1) Data selection and preprocessing; (2) Feature extraction; (3) Training; (4) Testing; (5) Performance analysis. Hence, each stage of the study is described, as well as information about the adopted process.

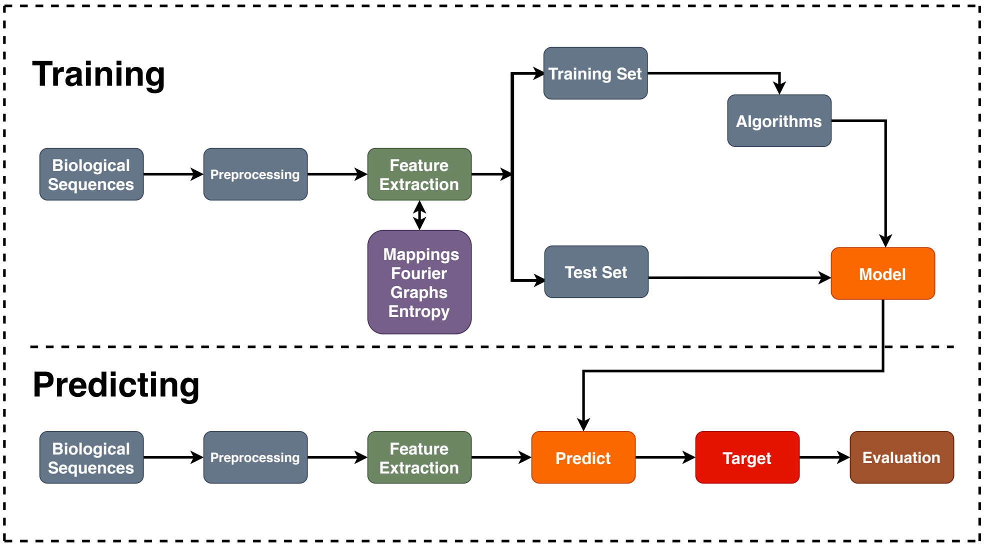


Figure 8 – Proposed Pipeline. Essentially, (1) datasets are preprocessed; (2) Feature extraction techniques are applied to each dataset; (3) Machine learning algorithms are executed in the training set to induce predictive models; (4) Induced models are applied to the test set; Finally, (5) the models are evaluated.

This work was also divided into three case studies: (I) We assessed our mathematical approaches with the most often addressed problem found in our review, e.g., lncRNA vs. mRNA; (II) We tested its generalization on different lncRNA classification tasks; (III) We analyze its robustness in scenarios with imbalanced data.

3.3.1 Data Selection

Recently, with a large number of transcribed sequences, the identification of ncRNAs has been a challenging task. For that reason, we have focused on lncRNAs classification problem, in special with ML algorithms, as described in section *Related Works*. However, we also adopt other datasets to assess the generalization of mathematical features. As preprocessing, we used only sequences longer than 200*nt* (LI; ZHANG; ZHOU, 2014), and we also removed sequence redundancy. Moreover, the sampling method was adopted in our dataset (case study I and II), since we are faced with the *imbalanced data problem* (BONIDIA *et al.*, 2019). Therefore, we applied random majority under-sampling, which consists of removing samples from the majority class (to adjust the class distribution) (LIU, 2004).

3.3.1.1 Case Study I

Sequences of five plant species were adopted to validate the proposed approaches. The summary of the dataset can be seen in Table 12. According to the literature approaches, this study also adopts two classes for the datasets: the positive class, with lncRNAs, and the negative class, with protein-coding genes (mRNAs).

Table 12 – Adopted species to create the datasets.

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

The mRNA data of the *Arabidopsis thaliana* (obtained from CPC2 (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` (*v*32) 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).

3.3.1.2 Case Study II

We applied the best mathematical features (according accuracy values) of the case study I to different classification problems with lncRNAs. In that case, we used only sequences from *Arabidopsis thaliana*, i.e., model species in plants. Thus, we have classified this study into three sub-problems, as shown in Table 13:

Table 13 – Datasets used in case study II.

Problems	Positive data	Negative data	Source
lncRNA vs. sncRNA	1291	1291	(KANG *et al.*, 2017)
lncRNA vs. Antisense	57	57	(CHEN *et al.*, 2011)
circRNA vs. lncRNA	2540	2540	(CHU *et al.*, 2017; KANG *et al.*, 2017)

3.3.1.3 Case Study III

In this step, we assessed the mathematical features in new sequences with imbalanced classes. According to (RAAD; STEGMAYER; MILONE, 2019; STEGMAYER *et al.*, 2019), this scenario simulates problems found in a real genome, allowing us to assess the robustness of our approaches. Therefore, we selected four datasets shown in Table 14. In this validation test, we extend the classification task introduced in case study II, circRNA vs. lncRNA, that has been approached by several works (PAN; XIONG, 2015; CHEN *et al.*, 2018; ZHANG *et al.*, 2020; CHAABANE *et al.*, 2020).

Table 14 – Datasets used in the validation test.

Dataset	**circRNA**	**Database**	**lncRNA**	**Database**
Human-1	6995	`circRNADb`	5000	`GENCODE`
Human-2	3280	`circBase`	2700	`LNCipedia`
Human-3	3280	`circBase`	60000	`LNCipedia`
C. sativus	4265	`PlantcircBase`	1730	`GreeNC`

We built these datasets, except C. sativus, based on (ZHANG *et al.*, 2020), which uses different databases. Fundamentally, the four datasets were generated using the combination of multiple bases, such as `circRNADb` (CHEN *et al.*, 2016), `GENCODE` (HARROW *et al.*, 2012), `circBase` (GLAŽAR; PAPAVASILEIOU; RAJEWSKY, 2014), `LNCipedia` (VOLDERS *et al.*, 2013), PlantcircBase (CHU *et al.*, 2017) and `GreeNC` (GALLART *et al.*, 2015). In addition, we introduced human specie data to assess the mathematical features with different sequences.

3.3.2 Feature Extraction

In this section, 9 feature extraction approaches are shown: 6 numerical mapping techniques with Fourier transform (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)), Entropy (Shannon (SHANNON, 1948) and Tsallis (TSALLIS; MENDES; PLASTINO, 1998)), and Complex Networks (ITO *et al.*, 2018). The theoretical and mathematical exploration of each approach can be found in our version published in Briefings in Bioinformatics (BONIDIA *et al.*, 2021a).

3.3.3 Normalization, Training and Evaluation Metrics

For data normalization, we used the min-max method (SOUTO *et al.*, 2008).Next, to assess our mathematical approaches, we investigate empirically three classification algorithms, such as Random Forest (RF) (BREIMAN, 2001), AdaBoost (HASTIE *et al.*, 2009) and CatBoost (PROKHORENKOVA *et al.*, 2018), in three datasets (*A. trichopoda*, *A. thaliana*, and *R. communis*). Thereby, to estimate the real accuracy, we applied 10-fold cross-validation, as shown in

(BONIDIA *et al.*, 2021a), in which all classifiers showed similar performance, with CatBoost being slightly better. Based on this, we chose for experimental tests, the CatBoost classifier. A wide variety of fields have employed CatBoost successfully (HANCOCK; KHOSHGOFTAAR, 2020; BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2020).

Moreover, this algorithm can induce interpretable predictive models when humans can easily understand the internal decision-making process (ZIHNI *et al.*, 2020). Consequently, domain experts can validate the knowledge used by the models for the classification of new sequences (Parmezan Bonidia *et al.*, 2019). Finally, to induce our classifier in experimental tests, we used Hold-out (70% of samples for *training* (with 10-fold cross-validation) and 30% for *testing*), as shown in Table 15. Nevertheless, for a better evaluation in case study II, we also apply the Leave-One-Out Cross-Validation (LOOCV), which according to (CHENG; GARRICK; FERNANDO, 2017) is an attractive resampling technique when the datasets are small.

Table 15 – Number of sequences used for training and testing.

Case Study	Dataset	Samples	Training	Testing
I	*A. trichopoda*	9112	6378	2734
	A. thaliana	5080	3556	1524
	C. sinensis	4430	3101	1329
	C. sativus	3460	2422	1038
	R. communis	6974	4881	2093
II	*lncRNA vs. sncRNA*	2582	1807	775
	lncRNA vs. Antisense	114	79	35
	circRNA vs. lncRNA	5080	3556	1524
III	*Human-1*	11995	8396	3599
	Human-2	5980	4186	1794
	Human-3	63280	44296	18984
	C. sativus	5995	4196	1799

We assess the effectiveness of our proposal using four measures for balanced datasets, such as Sensitivity (SE), Specificity (SPC), Accuracy (ACC), and Cohen's kappa coefficient (COHEN, 1960). For imbalanced datasets (case study III), we apply Balanced Accuracy (BACC), Geometric Mean (G-mean), and F1-score. These measures use True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. TP measures the correctly predicted positive labels; TN represents the correctly classified negative labels; FP describes all those negative entities that are incorrectly classified as positive and; FN represents the positive labels that are incorrectly classified as the negative labels.

3.4 Results

This section shows experimental results from 9 mathematical feature extraction approaches for biological sequences, divided into several case studies.

3.4.1 Case Study I

Initially, we induced models with the CatBoost classifier in the training sets (see Table 15). Thus, in Table 16, we present the results of all mathematical features using 4 evaluation metrics. As can be seen, all approaches presented interesting results, with the worst performance (ACC) of 0.8901 (*C. sinensis*) and the best of 0.9606 (*A. thaliana*). That is, all features were efficient in different datasets without a high loss of performance. Assessing each metric individually, we realized that in SE, the best performance was F-Real (3 datasets), followed by Tsallis (2 datasets) and F-Complex (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 F-Real and F-Complex (1 dataset). For each dataset, we can see in *A. trichopoda*, the best ACC was 0.9407 (F-Complex); *A. thaliana* with 0.9606 (F-Real); *C. sinensis* with 0.8901 (Tsallis); *C. sativus* with 0.8902 (Tsallis); and *R. communis* with 0.9513 (Tsallis). Highlight for Tsallis entropy, which evidenced the best results, mainly in ACC, showing to be more efficient in the case study I.

3.4.2 Case Study II

In this step, we selected the best three mathematical feature extraction approaches for generalization analysis, as follows: GSP (Fourier + complex numbers), entropy (Tsallis), and graphs (complex networks). Thereby, we assessed its generalization to classify sequences with different structures. For this, we used three new datasets established in section: *Case Study II*, as can be seen in Figure 9. Furthermore, as some datasets are small, we validate our approaches with two techniques, such as Hold-out and LOOCV, briefly described in the section: *Normalization, Training and Evaluation Metrics*.

Considering the results in case study II, we realized that graph-based features are the best in 2 of the 3 problems analyzed, followed by entropy and GSP. In the three datasets, with different validations, mathematical approaches have achieved interesting results with ACC, SE, and SPC, indicating an alternative and complementary approach to biological features. Furthermore, considering the classification task between circRNA and lncRNA, mathematical features were effective when compared to (PAN; XIONG, 2015) and (CHEN *et al.*, 2018), reaching 0.7780 and 0.7890 of ACC, respectively (using these comparisons as an (indirect) reference indicator), while our best approach (graph-based features) found 0.8307 (Hold-out) and 0.8272 (LOOCV) of ACC.

3.4.3 Statistical Significance Tests

The statistical significance was applied in both case studies (difference in ACC), using Friedman's statistical test and the Conover post-hoc test. Thereby, our null hypothesis ($H0$: there are no significant differences between the approaches compared) is tested against the alternative hypothesis (H_A: some approach has statistical significance ($\alpha = 0.05$, $p < \alpha$)). First,

Table 16 – Performance analysis. This table compares the sensitivity, specificity, accuracy and kappa metrics for each approach in the test sets using CatBoost classifier (Important: F = Fourier).

Dataset	Features	SE	SPC	ACC	Kappa
A. trichopoda	F-Zcurve	0.9744	0.8566	0.9155	0.8310
	F-Binary	0.9795	0.9005	0.9400	0.8800
	F-Real	**0.9802**	0.8837	0.9320	0.8639
	F-Integer	0.9773	0.8822	0.9298	0.8595
	F-EIIP	0.9781	0.8990	0.9386	0.8771
	F-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	F-Zcurve	0.9777	0.9383	0.9580	0.9160
	F-Binary	0.9619	0.9449	0.9534	0.9068
	F-Real	**0.9803**	0.9409	**0.9606**	**0.9213**
	F-Integer	0.9698	0.9436	0.9567	0.9134
	F-EIIP	0.9646	0.9449	0.9547	0.9094
	F-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	F-Zcurve	0.9021	**0.8707**	0.8864	0.7728
	F-Binary	0.8901	**0.8707**	0.8804	0.7607
	F-Real	0.9142	0.8571	0.8856	0.7713
	F-Integer	0.8825	0.8692	0.8758	0.7517
	F-EIIP	0.8840	0.8526	0.8683	0.7367
	F-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	F-Zcurve	0.8979	0.8478	0.8728	0.7457
	F-Binary	0.9056	0.8459	0.8757	0.7514
	F-Real	0.9268	0.8439	0.8854	0.7707
	F-Integer	0.9056	**0.8536**	0.8796	0.7592
	F-EIIP	0.8979	0.8459	0.8719	0.7437
	F-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	F-Zcurve	0.9446	0.9140	0.9293	0.8586
	F-Binary	0.9417	0.9589	0.9503	0.9006
	F-Real	**0.9589**	0.9408	0.9498	0.8997
	F-Integer	0.9465	0.9456	0.9460	0.8920
	F-EIIP	0.9455	0.9551	0.9503	0.9006
	F-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**

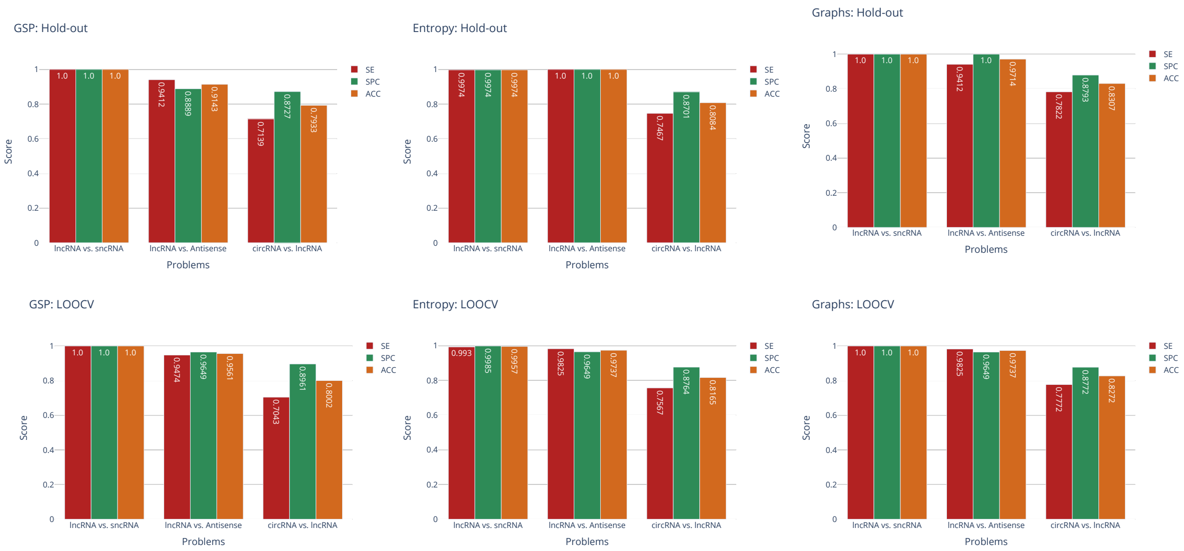


Figure 9 – Performance analysis of three mathematical features, GSP (fourier + complex numbers), entropy (Tsallis) and graphs (complex networks), for different problems (using Hold-out and LOOCV validation).

we apply the global test in the case study I, in which the Friedman test indicates significance ($\chi^2(8) = 17.34$, p-value = 0.0268), that is, we can reject $H0$, as $p < 0.05$. Thus, it is essential to execute the post-hoc statistical test. Conover statistics values were obtained, as well as p-values (see Table 17), using 5% of significance ($\alpha = 0.05$).

Table 17 – Conover statistics values - The rejected null hypothesis is in bold (p-values for $\alpha = 0.05$).

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

Concerning to the Conover post-hoc test, only Tsallis and Shannon entropy-based features have statistical differences for specific cases, such as: F-Zcurve ($p < 0.0146$), F-Integer ($p < 0.0075$ (with Tsallis) and $p < 0.0390$ (with Shannon)), and F-EIIP ($p < 0.0128$). Moreover, there is no evidence of causal relationship among Fourier representations and Graphs, hence, based on statistical test we cannot ensure their efficiency in all datasets. In case study II, we realized that Friedman's statistical test is not significant, in which we obtained $\chi^2(2) = 1.64$, p-value = 0.4412, indicating that the three studied feature extraction techniques have a similar performance in all problems evaluated. However, based on statistical test, we cannot ensure the best mathematical models and their effectiveness and robustness in all datasets.

3.4.4 Feature Analysis

In this section, we introduce the feature importance analysis carried out in this study. In addition, we discuss a possible biological interpretation and offer some insights into the reasons why some features perform better than others. First, we build a list of features used by each approach (e.g., GPS, Entropy, and Graphs), as can be seen in Table 18. GSP has fixed attributes, with 19 statistical measures generated from the Fourier spectrum. Meanwhile, the quantity of features for entropy and graphs is defined by the parameters k and t, respectively. To better describe the feature analysis, we have elaborated a brief biological explanation of each approach (more details regarding theoretical issues can be found in the: *Feature Extraction*) section.

In GSP, we apply DFT with 6 numerical mapping techniques (or representations), when we extract statistical measures from the generated spectrum. Thereby, we can conjecture that RNA sequences are time-domain signals, given their biological properties linked to nucleotides, base periodicity and their interdependence. Therefore, the distinct mappings contain the same information organized in different shapes and dimensions. For this reason, the evaluation of different representations performance is a hard task and it is out of the scope of this research. In addition, as future work, we will evaluate the relation of mappings through complex and real numbers with signal processing based on Fourier transform.

Table 18 – Number of features for GSP (Fourier + Mapping), entropy (Shannon and Tsallis) and graphs (complex networks).

Approach	**Dimension**	**Features**
GSP	19	Peak to Average Power Ratio (2 features), average power spectrum, median, maximum, minimum, sample standard deviation, population standard deviation, percentile (15/25/50/75), range, variance, interquartile range, semi-interquartile range, coefficient of variation, skewness, and kurtosis.
Entropy	k	$k = 1, 2, \ldots, 24$
Graphs	$12 \cdot t$	Betweenness, assortativity, average degree, average path length, minimum degree, maximum degree, number of edges, degree standard deviation, frequency of motifs (size 3 and 4), clustering coefficient (local and global).

The entropy is considered an information measure able to recover the distribution of the k-mers and their amount of information. Thus, we have a pattern of k-mers distribution that is measured by the entropy of the sequence. In advance, if the k-mers distribution and location have a strong relationship with any classes, that information will be recovered by entropy as an information measure.

The mapping of RNA sequences to graphs (complex networks) takes into account the relationship between k-mers (with k = 3 in this study) and their neighborhood (step = 1) for each sequence. Thus, the neighborhood structural relationship between the k-mers is recovered and analyzed using complex network measurements (see Table 18). Besides, based on the structural neighborhood relationship between the k-mers, RNA classes produce networks with different topologies, which leads to representative and distinct features (measures) for each class of RNA.

Moreover, the application of thresholds generates features in different resolutions of the complex network. More specifically, starting with many edges and removing the less frequent edges at each iteration, soon we capture the dynamics of the topological change in the network. Essentially, identifying relevant features at different thresholds means that the dynamics caused by the removed edges are efficient to recover the most relevant structural relationships between the k-mers.

Finally, we also included a feature importance analysis, highlighting the most relevant features for a classification task. Feature importance is obtained by using a class of techniques for assigning scores to input features, which are used to induce predictive models. This analysis indicates the relative importance of each feature when making a prediction. Moreover, the relative scores can highlight which features may be most relevant to the target, and which features are the least important. This type of analysis may be interpreted by a domain expert and could support the decision for gathering more or different data. In this study, we analyzed the best features to classify lncRNA vs. mRNA and circRNA vs. lncRNA in *A. thaliana* (specie model in plant), as can be seen in Figure 10.

This analysis indicated that different features have a higher effect on the induction of a classifier. Additionally, we classified the highest importance score among sequence composition features, considering the best mathematical models, as follow: GSP (lncRNA vs. mRNA) percentile(25) and (lncRNA vs. circRNA) average; Entropy (lncRNA vs. mRNA) 1-mer and (lncRNA vs. circRNA) 19-mer; Graphs (lncRNA vs. mRNA) Number of edges-t9 and (lncRNA vs. circRNA) Number of edges-t1.

3.4.5 Computational Time and Complexity Analysis

We also assessed the computational cost and complexity of each tested approach (implemented in Python). For such, we applied GSP (Fourier + complex numbers), entropy (Tsallis), and graphs (complex networks) to 1291 randomly selected sequences. We performed the experiments using a machine with Intel Core i3-9100F CPU (3.60GHz), 16GB memory, and running in Debian GNU/Linux 10. The lowest computational cost was observed in the approaches based on GSP (0m7.183s) and entropy (0m51.427s), while graphs (3m58.208s) presented a much higher cost. These results demonstrated that, although the approaches report similar predictive performance, the computational costs are very different.

Regarding computational complexity, we derived worst-case asymptotic expressions

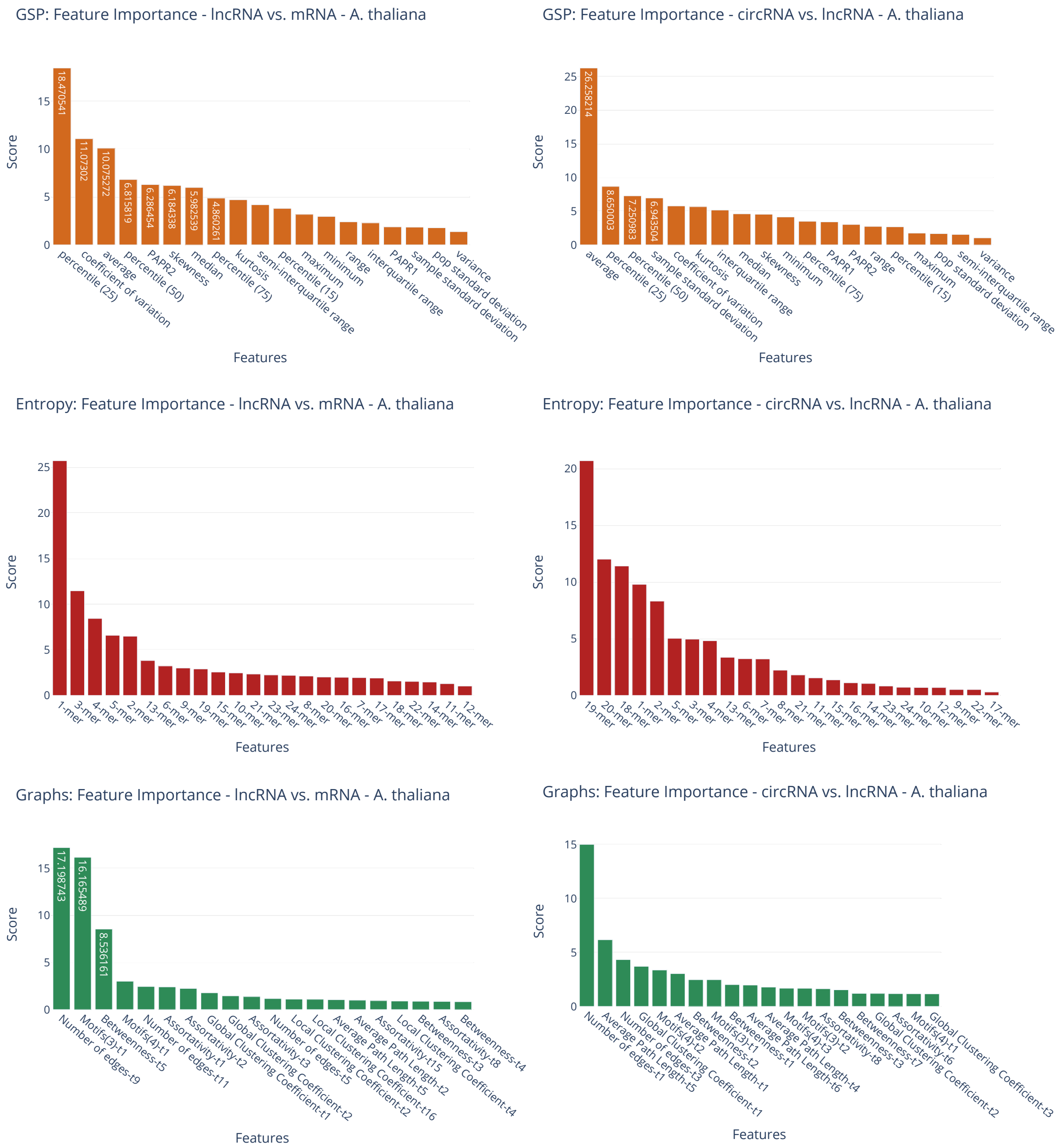


Figure 10 – Bar chart with relative importance score, GSP (fourier + complex numbers), entropy (Tsallis) and graphs (complex networks), considering lncRNA vs. mRNA and circRNA vs. lncRNA.

based on the Python implementation. which are presented in Table 19. Since the input variables for each developed approach are different, we first derived a general complexity expression based on these input parameters and further derived a set of simplified expressions whose dominant factor is the sequence size N.

According to (JACOBSEN; LYONS, 2003), the results for the GSP are straightforward, while the expression for the Entropy and Complex Networks depend on different subroutines, which may increase the asymptotic complexity when compared to GSP. The entropy expression depends directly on the parameter k from the k-mer methodology that generates 4^k features, hence,

Table 19 – Computational Complexity Analysis.

Approach	Complexity (Complete Expression)	Complexity
GSP	$O(N \log N)$	$O(N \log N)$
Entropy	$O\left(k\left(4^{(k^2)}+3\left(4^k\right)\right)\right)$	$O\left(2^k\right)$
Graphs	$O(k+(k*4k)+t*((4^k)^2+((4^k)^2)+c*((4^k)^2)))$	$O\left(2^k\right)$

*k = frequencies of k-mer, t = threshold - number of subgraphs,
*c = number of extracted features, GSP (FFT (JACOBSEN; LYONS, 2003)).

the asymptotic complexity is given by $O\left(k\left(4^{(k^2)}+3\left(4^k\right)\right)\right)$. This result can be observed on the application code[1]. Fundamentally, the first k represents the loop started at line 62, the first 4^{k^2} is the combination of the two loops in lines 66 and 67, finally, the last parcel, i.e. $3\left(4^k\right)$, is a result of lines 72, 77 and 78, which are three equally sized loops.

Finally, the computational complexity of the graphs is also derived according to the application code[2], which depends on the following parameters: k, the k-mer frequencies, t, the threshold, i.e., the number of subgraphs, and c, the number of extracted features. Therefore, according to lines 97 through 118 in the application code, the computational complexity is defined as $O(k+(k*4k)+t*((4^k)^2+((4^k)^2)+c*((4^k)^2)))$. Assuming that $k >> c$ and $k >> t$, the computational complexity is $O\left(2^k\right)$. To compare GSP to the other methods in terms of complexity, we consider that, in the worst-case scenario $k = N$, hence, asymptotically, the complex networks and the entropy method have the same computational complexity (exponential), while the GSP is the least complex (log-linear).

3.4.6 Case Study III - Validation Test With Imbalanced Datasets

In this section, we evaluated the proposed approach performance on new sequences with imbalanced data, using four datasets (as described in *Materials and Methods - Case Study III*). For this study, we selected the most challenging classification task used in case study II, circRNA vs. lncRNA (PAN; XIONG, 2015; CHEN *et al.*, 2018; ZHANG *et al.*, 2020; CHAABANE *et al.*, 2020). To assess the mathematical features (same approaches applied in case study II), we used imbalanced data metrics (e.g., BACC, G-mean, and F1-score), as exposed in Table 20.

As can be seen, graphs-based features presented the best overall predictive performance, with the best results in three of the four datasets analyzed, while Entropy had better results in the Human-2 dataset, with a minimum difference of 0.08. Nevertheless, we noticed that the performance obtained by graphs was similar to that observed in the case study II, even with imbalanced data, in particular for the datasets: Human-1 (F1-score: 0.8085, BACC: 0.7551, G-mean: 0.7505), Human-2 (F1-score: 0.8106, BACC: 0.8050, G-mean: 0.8043), and C. sativus

[1] For further reference, check: (BONIDIA, 2020) - EntropyClass.py
[2] For further reference, check: (BONIDIA, 2020) - ComplexNetworksClass.py

Table 20 – Performance analysis of three mathematical feature extraction approaches in imbalanced datasets.

Human-1				**Human-2**			
Approach	**F1-score**	**BACC**	**G-mean**	**Approach**	**F1-score**	**BACC**	**G-mean**
GSP	0.7402	0.6652	0.6559	GSP	0.8041	0.7990	0.7980
Entropy	0.7427	0.6656	0.6552	Entropy	**0.8120**	**0.8060**	**0.8051**
Graphs	**0.8085**	**0.7551**	**0.7505**	Graphs	0.8106	0.8050	0.8043

Human-3				**C. sativus**			
Approach	**F1-score**	**BACC**	**G-mean**	**Approach**	**F1-score**	**BACC**	**G-mean**
GSP	0.3464	0.7977	0.7950	GSP	0.8241	0.6711	0.6469
Entropy	0.3312	0.8060	0.8052	Entropy	0.8337	0.6852	0.6634
Graphs	**0.4447**	**0.8141**	**0.8127**	Graphs	**0.8487**	**0.7235**	**0.7097**

(F1-score: 0.8487, BACC: 0.7235, G-mean: 0.7097). Despite the good predictive performance obtained in the Human-3 dataset, with a BACC of 0.8141, the high data imbalance affected the quality of the models obtained by all approaches. Moreover, we noticed that GSP and Entropy-based features were more affected by the imbalanced data. Therefore, in the next section, we compared our study with biological and hybrid approaches, to assess the predictive performance when using other features in the same datasets.

3.5 Comparing Mathematical, Biological and Hybrid Approaches

In this section, we present our findings and discuss whether they support our hypothesis (*Feature extraction approaches based on mathematical features are generalist, i.e., the ability to generalize mathematical features in different ncRNA types, such as the classification of lncRNA subclasses, being as efficient as biological approaches*). According to the results obtained in the several experiments carried out, illustrated by Table 16, Figure 9, and Table 20, all mathematical feature extraction approaches positively affect the predictive performance obtained.

Nevertheless, to fully support our hypothesis, we also compared GSP, entropy, and graphs, when adopting a biological and hybrid approach, and applied to eight RNAs classification datasets (previously used), such as lncRNA vs. mRNA (case study I); lncRNA vs. sncRNA, lncRNA vs. Antisense, circRNA vs. lncRNA (case study II); Human-1, Human-2, Human-3, C. sativus (case study III). For a fair comparison, the new experiments follow the same methodology (70% of the dataset used for training, 30% for test, and using the CatBoost classifier), as shown in Table 21. The biological and hybrid models were trained and tested on the same dataset as the mathematical features. We generate our comparative approaches with some of the most frequently used features, shown in Figure 7. Figure 11 presents a feature ranking found in our review (specifically in biological approaches, see Figure 7). The x-axis refers to features categories and the y-axis to the number of studies.

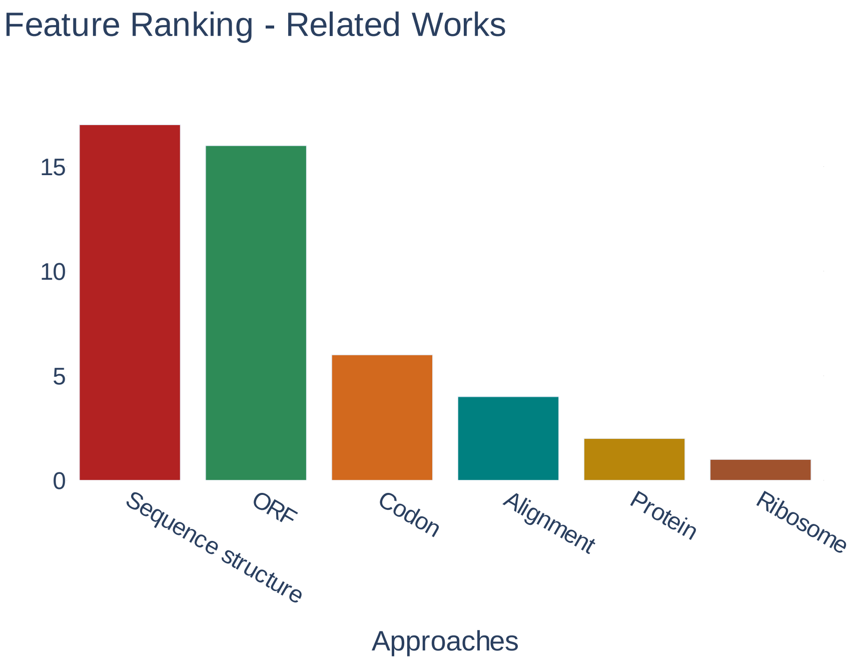


Figure 11 – Feature Ranking.

As can be seen, the feature group most often applied by the reported studies is sequence structure (17), followed by ORF (16), codon (6), alignment (4), protein (2), ribosome (1). Among these groups, we use the features of higher biological bias to generate our models, such as ORF coverage, ORF size, ORF integrity, ORF quality, Fickett score, peptide level features, and hexamer score. Several approaches have used some of these features, as shown in *Related Works*, e.g., CPC, CPAT, CREMA, PLIT, LGC, LncFinder, DeepCPP, among others (some works provide the option to extract these attributes, e.g., (HAN *et al.*, 2018; KANG *et al.*, 2017)). Thus, the hybrid models in this study are based on a combination of mathematical (proposed approaches) and biological features, e.g., GPS + Biological (Hybrid-1), Entropy + Biological (Hybrid-2), and Graphs + Biological (Hybrid-3). The results are shown in Table 21.

According to Table 21, the hybrid-1 and hybrid-2 models reported the best predictive performance (both 0.9915) in the first problem (lncRNA vs. mRNA datasets), followed by the biological (0.9888), and Entropy (mathematical approach - 0.9587), with a difference of 0.0328 and 0.0301, respectively. However, it is important to highlight that both biological and hybrid models have considered the ORF descriptor, which, according to (BAEK *et al.*, 2018), is an essential guideline for distinguishing lncRNAs from mRNA. Moreover, the ORF feature has a biological bias, being usually difficult to reuse or adapt to another specific problem (different than lncRNAs vs. mRNA). Besides, taking into account only the hybrid and biological model predictive performance, we observed that the gain was minimal when compared to the biological model (0.0027), highlighting the ORF descriptor importance.

Table 21 – Performance analysis of three mathematical feature extraction approaches against a biological and hybrid model for different sequence classification problems.

Balanced Datasets

lncRNA vs. mRNA - A. thaliana				**lncRNA vs. sncRNA - A. thaliana**			
Approach	**SE**	**SPC**	**ACC**	**Approach**	**SE**	**SPC**	**ACC**
GSP	0.9724	0.9409	0.9567	GSP	**1.0000**	**1.0000**	**1.0000**
Entropy	**0.9764**	**0.9409**	**0.9587**	Entropy	0.9974	0.9974	0.9974
Graphs	0.9685	0.9423	0.9554	Graphs	**1.0000**	**1.0000**	**1.0000**
Biological	**0.9895**	**0.9882**	**0.9888**	Biological	0.9768	0.9483	0.9626
Hybrid-1	**0.9948**	**0.9882**	**0.9915**	Hybrid-1	**1.0000**	0.9948	0.9974
Hybrid-2	**0.9961**	**0.9869**	**0.9915**	Hybrid-2	0.9768	0.9922	0.9845
Hybrid-3	0.9921	0.9856	0.9888	Hybrid-3	**1.0000**	0.9948	0.9974

lncRNA vs. Antisense - A. thaliana				**circRNA vs. lncRNA - A. thaliana**			
Approach	**SE**	**SPC**	**ACC**	**Approach**	**SE**	**SPC**	**ACC**
GSP	0.9412	0.8889	0.9143	GSP	0.7139	0.8727	0.7933
Entropy	**1.0000**	**1.0000**	**1.0000**	Entropy	0.7467	0.8701	0.8084
Graphs	0.9412	1.0000	0.9714	Graphs	**0.7822**	**0.8793**	**0.8307**
Biological	0.8889	0.9412	0.9143	Biological	0.7087	0.8635	0.7861
Hybrid-1	0.8889	**1.0000**	0.9429	Hybrid-1	0.7467	0.8780	0.8123
Hybrid-2	0.9444	0.9412	0.9429	Hybrid-2	0.7585	0.8963	0.8274
Hybrid-3	0.8889	**1.0000**	0.9429	Hybrid-3	**0.7769**	**0.8911**	**0.8340**

Imbalanced Datasets

circRNA vs. lncRNA - Human-1				**circRNA vs. lncRNA - Human-2**			
Approach	**F1-score**	**BACC**	**G-mean**	**Approach**	**F1-score**	**BACC**	**G-mean**
GSP	0.7402	0.6652	0.6559	GSP	0.8041	0.7990	0.7980
Entropy	0.7427	0.6656	0.6552	Entropy	**0.8120**	**0.8060**	**0.8051**
Graphs	0.8085	0.7551	0.7505	Graphs	0.8106	0.8050	0.8043
Biological	0.9383	0.9280	0.9280	Biological	0.7785	0.7617	0.7615
Hybrid-1	0.9561	0.9474	0.9473	Hybrid-1	0.8323	0.8212	0.8210
Hybrid-2	0.9566	0.9482	0.9481	Hybrid-2	**0.8448**	**0.8336**	**0.8334**
Hybrid-3	**0.9595**	**0.9510**	**0.9510**	Hybrid-3	0.8374	0.8267	0.8265

circRNA vs. lncRNA - Human-3				**circRNA vs. lncRNA - C. sativus**			
Approach	**F1-score**	**BACC**	**G-mean**	**Approach**	**F1-score**	**BACC**	**G-mean**
GSP	0.3411	0.7993	0.7971	GSP	0.8241	0.6711	0.6469
Entropy	0.3338	0.8093	0.8086	Entropy	0.8337	0.6852	0.6634
Graphs	**0.4242**	**0.8058**	**0.8045**	Graphs	0.8487	0.7235	0.7097
Biological	0.2876	0.7782	0.7775	Biological	0.8565	0.6852	0.6449
Hybrid-1	0.3406	0.8211	0.8209	Hybrid-1	0.8384	0.7094	0.6951
Hybrid-2	0.3539	0.8324	0.8323	Hybrid-2	**0.9159**	**0.8499**	**0.8467**
Hybrid-3	**0.4485**	**0.8343**	**0.8342**	Hybrid-3	0.8556	0.7387	0.7273

In addition, case study II extends the evaluation with three different sub-problems. The results obtained also confirm our hypothesis, since models induced with mathematical features performed better than those induced with biological features, especially in the classification tasks, e.g., lncRNA vs. Antisense (third dataset) and circRNA vs. lncRNA (fourth dataset), when the results obtained by graphs were 0.9714 and 0.8307 of ACC, respectively. In general, models based on graphs features were superior to those based on biological features by 0.0571 and 0.0446 of ACC. Regarding the hybrid model, the combination of biological and mathematical features helped to keep the model competitive in all datasets, mainly in circRNA vs. lncRNA, with the predictive performance of 0.8340 (ACC), indicating that merging features can improve the predictive performance of the induced classification models.

Regarding case study III (imbalanced datasets), there was a clear superiority in the predictive performance of models induced with hybrid features, followed by those induced with mathematical and biological features. It is possible to see a robust predictive performance with mathematical approaches in two datasets: Human-2 (Entropy - G-mean: 0.8106 and Hybrid-2 - G-mean: 0.8334) and Human-3 (Graphs - G-mean: 0.8045 and Hybrid-3 - G-mean: 0.8342). Furthermore, in all imbalanced datasets, models induced with hybrid features presented the best predictive performance, such as Human-1 (Hybrid-3, F1-score: 0.9595, BACC: 0.9510, G-mean: 0.9510), Human-2 (Hybrid-2, F1-score: 0.8448, BACC: 0.8336, G-mean: 0.8334), Human-3 (F1-score: 0.4485, BACC: 0.8343, G-mean: 0.8342), and C. sativus (F1-score: 0.9159, BACC: 0.8499, G-mean: 0.8467). These results show the effects of imbalanced data when using biological features, in particular in the Human-3 dataset, with a difference of 0.1609 (F1-score), compared to Hybrid-3.

Finally, we also assessed the statistical significance of the different predictive performances when comparing models induced with mathematical and biological features in the previously reported experiments. These tests showed the superiority of entropy ($p < 0.0468$), graphs ($p < 0.0105$), and all hybrid approaches (1: $p < 0.0019$, 2: $p < 0.0001$, and 3: $p < 4.2e-05$), compared with the use of biological features, supporting the previously mentioned hypothesis. Furthermore, we observed the high predictive performance of the hybrid models, suggesting that a combination of biological and mathematical features can lead to the induction of better predictive models, in particular when combining features from different approaches. Therefore, the proposed pipeline is efficient and robust in terms of generalization and predictive performance for different lncRNAs sequence classification problems.

3.6 Chapter Remarks

This work proposed to analyze feature extraction approaches for biological sequence classification. Specifically, we concentrated our work on the study of efficient and generalist mathematical features for different problems. As a case study, we used lncRNA sequences. In our experiments, as a starting point, nine mathematical approaches were analyzed, such as six numerical mapping techniques with Fourier Transform, Tsallis and Shannon entropy, and Graphs (complex networks). Thereby, we adopted several sequence classification scenarios to answer the questions raised in this work.

In our experiments, all mathematical features presented relevant and robust results with performances (ACC) between 0.8901-0.9606. In the second case study, once more, entropy-based features and graphs showing the best performance, followed by GSP. In the third case study, with imbalanced data, graphs-based features kept the best performance in three of the four datasets analyzed. Furthermore, we compared three mathematical approaches against biological and hybrid models, in eight datasets, in which we have presented suitable results, being superior, competitive, and robust in terms of generalization. We also verified that mathematical approaches perform as accurately as biological approaches and have a better generalization capacity since they outperform biological features in scenarios not designed for them. Finally, among the feature extraction approaches tested in this work, the combination of k-mer and entropy, as well as complex networks performs better than GSP at the cost of a significant increase in computational complexity.

CHAPTER

4

INFORMATION THEORY FOR BIOLOGICAL SEQUENCE CLASSIFICATION: A NOVEL FEATURE EXTRACTION TECHNIQUE BASED ON TSALLIS ENTROPY

The accelerated evolution of sequencing technologies has generated significant growth in the number of sequence data (HASHEMI *et al.*, 2018b), opening up new opportunities and creating new challenges for biological sequence analysis. To take advantage of the increased predictive power of machine learning (ML) algorithms, recent works have investigated the use of these algorithms to analyze biological data (SILVA *et al.*, 2019b; GREENER *et al.*, 2022).

The development of effective methods for sequence analysis, through ML, benefits the research advancement in new applications (LOU *et al.*, 2019; BONIDIA *et al.*, 2021a), such as understanding several problems (LOU *et al.*, 2019; BONIDIA *et al.*, 2021a), e.g., cancer diagnostics (MAROS *et al.*, 2020), development of CRISPR-Cas systems (EITZINGER *et al.*, 2020), drug discovery and development (VAMATHEVAN *et al.*, 2019) and COVID-19 diagnosis (Abubaker Bagabir *et al.*, 2022). Nevertheless, ML algorithms applied to the analysis of biological sequences present challenges, such as feature extraction (STORCHEUS; ROSTAMIZADEH; KUMAR, 2015). For non-structured data, as is the case of biological sequences, feature extraction is a key step for the success of ML applications (IUCHI *et al.*, 2021; CUI; ZHANG; ZOU, 2021; BONIDIA *et al.*, 2022).

Previous works have shown that universal concepts from Information Theory (IT), originally proposed by Claude Shannon (1948) (SHANNON, 1948), can be used to extract relevant information from biological sequences (VINGA, 2013; PRITIŠANAC *et al.*, 2019; VOPSON; ROBSON, 2021). According to Ré and Azad (2014), an IT-based analysis of symbolic sequences is of interest in various study areas, such as linguistics, biological sequence analysis, or image processing, whose relevant information can be extracted, for example, by Shannon's uncertainty theory (AKHTER *et al.*, 2013).

Studies have investigated the analysis of biological sequences with Shannon entropy in a wide range of applications (AKHTER *et al.*, 2013; MACHADO; COSTA; QUELHAS, 2011; TRIPATHI *et al.*, 2016). Given their large applicability, according to Yamano (2001), it is important to explore the possibility of generalized entropies, such as Tsallis (TSALLIS, 1988; TSALLIS; MENDES; PLASTINO, 1998), which was proposed to generalize the Boltzmann/Gibbs's traditional entropy to non-extensive physical systems (ALBUQUERQUE; ESQUEF; MELLO, 2004). This class of generalized entropy has been used for different problems, e.g., image analysis (ALBUQUERQUE; ESQUEF; MELLO, 2004; RAMÍREZ-REYES *et al.*, 2016), inference of gene regulatory networks (LOPES; OLIVEIRA; CESAR, 2011), DNA analysis (MACHADO; COSTA; QUELHAS, 2011) induction of decision trees (CRUZ-GARCíA; BORY-REYES; RAMIREZ-ARELLANO, 2022) and classification of epileptic seizures (THILAGARAJ; RAJASEKARAN; KUMAR, 2019).

In Albuquerque, Esquef and Mello (2004), the authors proposed a new image segmentation method using Tsallis entropy. Later, Ramírez-Reyes *et al.* (2016) showed a novel numerical approach to calculate the Tsallis entropic index feature for a given image. In Lopes, Oliveira and Cesar (2011), the authors introduced the use of generalized entropy for the inference of gene regulatory networks. DNA analysis using entropy (Shannon, Rényi, and Tsallis) and phase plane concepts were presented in (MACHADO; COSTA; QUELHAS, 2011), while (CRUZ-GARCíA; BORY-REYES; RAMIREZ-ARELLANO, 2022) used the concept of generalized entropy for decision trees. Recently, Thilagaraj, Rajasekaran and Kumar (2019) investigated a novel single feature based on Tsallis entropy to classify epileptic seizures. These studies report a wide range of contributions to the use of Tsallis entropy in different domains. To the best of our knowledge, this paper is the first work proposing its use as a feature (feature extraction) to represent distinct biological sequences. Additionally, it presents the first study of different Tsallis entropic indexes and their effects on classical classifiers.

A preliminary version of this proposal was presented in Bonidia *et al.* (2021a). Due to the favorable results obtained, we created a code to extract different descriptors available in a new programming package, called MathFeature (BONIDIA *et al.*, 2022), which implements mathematical descriptors for biological sequences. However, until now, we have not studied Tsallis entropy in depth, e.g., its effect, its application to other biological sequence datasets, and its comparison with other entropy-based descriptors, e.g., Shannon. Thus, in this paper, we investigate the answers to the following questions:

- **Question 1 (Q1):** Are Tsallis entropy-based features robust for extracting information from biological sequences in classification problems?
- **Question 2 (Q2):** Does the entropic index affect the classification performance?

- **Question 3 (Q3):** Is Tsallis entropy as robust as Shannon entropy for extracting information from biological sequences?

We are evaluating robustness in terms of performance, e.g., accuracy, recall, and F1 score, of the feature vectors extracted by our proposal on different biological sequence datasets. Finally, this study makes the following main research contributions: We propose an effective feature extraction technique based on Tsallis entropy, which is robust in terms of generalization, and also potentially representative for collecting information in fewer dimensions for sequence classification problems.

4.1 Literature Review

In this section, we develop a systematic literature review to present and summarize feature extraction descriptors for biological sequences (DNA, RNA, or protein). This review aims to report the need and lack of studies with mathematical descriptors, such as entropy, evidencing the contribution of this article. This section followed the Systematic Literature Review (SLR) Guidelines in Software Engineering (KEELE *et al.*, 2007), which, according to Keele *et al.* (2007), Brereton *et al.* (2007), allows a rigorous and reliable evaluation of primary studies within a specific topic. We base our review on recommendations from previous studies (KEELE *et al.*, 2007; BRERETON *et al.*, 2007; KITCHENHAM *et al.*, 2009b).

We propose to address the following problem: *How can we numerically represent a biological sequence (such as DNA, RNA, or protein) in a numeric vector that can effectively reflect the most discriminating information in a sequence?* To answer this question, we reviewed ML-based feature extraction tools (or packages, web servers, and toolkits) that aim, as a proposal, to provide several feature descriptors for biological sequences—that is, without a defined scope, and, therefore, generalist studies. Moreover, we used the following electronic databases: ACM Digital Library, IEEE Xplore Digital Library, PubMed, and Scopus. We chose the Boolean method (KARIMI *et al.*, 2010) to search primary studies in the literature databases. The standard search string was: *("feature extraction" OR "extraction" OR "features" OR "feature generation" OR "feature vectors") AND ("machine" OR "learning") AND ("tool" OR "web server" OR "package" OR "toolkit") AND ("biological sequence" OR "sequence").*

Due to different query languages and limitations between the scientific article databases, there were some differences in the search strings. Therefore, our first step was to apply search keys to all databases, returning a set of 1404 studies. Furthermore, we used the Parsifal tool to assist our review and obtain better accuracy and reliability. Thereafter, duplicate studies were removed, returning an amount of 1097 titles (307 duplicate studies). Then, we performed a thorough analysis of the titles, keywords, and abstracts, according to **inclusion and exclusion criteria**: (1) Studies in English, (2) Studies with different feature extraction techniques, (3) Studies with generalist tools and (4) Studies published in journals. We accepted 28 studies (we rejected, 1069). Finally, after pre-selecting the studies, we performed a data synthesis, to apply an assessment based on the **quality criteria**: (1) Are the study aims specified? (2) Study with different proposals/results? (3) Study with complete results?

Hence, of the 28 studies, 3 were eliminated, leading to a final set of 25 studies. As previously mentioned, we assessed generalist tools for feature extraction, since this type of study would provide several descriptors, presenting an overview of ways to numerically represent biological sequences (which would not be possible by evaluating studies dedicated to some specific problem). As expected, we found more than 100 feature descriptors. We chose to divide them into large groups (16 groups—these were defined based on all studies), as shown in Chapter 2. As can be seen, no study provides mathematical descriptors, such as Tsallis entropy, reinforcing the contribution of our proposal.

4.2 Information Theory and Entropy

According to Martignon (2001), IT can be defined as a mathematical treatment of the concepts, parameters, and rules related to the transmission and processing of information. The IT concept was first proposed by Claude Shannon (1948) in the work entitled "A Mathematical Theory of Communication" (SHANNON, 1948), where he showed how information could be quantified with absolute precision. The entropy originating from IT can be considered a measure of order and disorder in a dynamic system (ALBUQUERQUE; ESQUEF; MELLO, 2004; SHANNON, 1948). However, to define information and entropy, it is necessary to understand *random variables*, which, in probability theory, is a mathematical object that can take on a finite number of different states $x_1,\ldots,x_n$ with previously defined probabilities $p_1,\ldots,p_n$ (ADAMI, 2012). According to (BONIDIA *et al.*, 2021a), for a discrete random variable R taking values in $\{r[0],r[1],r[2],\ldots,r[N-1]\}$ with probabilities $\{p[0],p[1],p[2],\ldots,p[N-1]\}$, represented as $P(R=r[n])=p[n]$, we can define self-information or information as (RAMÍREZ-REYES *et al.*, 2016)

$$I = -log(p). \tag{4.1}$$

Thus, the Shannon entropy H_S is defined by

$$H_S = -\sum_{n=0}^{N-1} p[n]\; log_2\, p[n]. \tag{4.2}$$

Here, N is the number of possible events and $p[n]$ the probability that event n occurs. Fundamentally, with Shannon entropy, we can reach a single value that quantifies the information contained in different observation periods (LESNE, 2014). Furthermore, it is important to highlight that the Boltzmann/Gibbs entropy was redefined by Shannon as a measure of uncertainty

(ALBUQUERQUE; ESQUEF; MELLO, 2004). This formalism, known as Boltzmann–Gibbs–Shannon (BGS) statistics, has often been used to interpret discrete and symbolic data (RÉ; AZAD, 2014). Moreover, according to Albuquerque, Esquef and Mello (2004), Zhang and Wu (2011), if we decompose a physical system into two independent statistical subsystems A and B, the Shannon entropy has the extensive property (additivity)

$$H_S(A+B) = H_S(A) + H_S(B) \tag{4.3}$$

According to Maszczyk and Duch (2008), complementary information on the importance of specific events can be generated using the notion of generalized entropy, e.g., outliers or rare events. Along these lines, Constantino Tsallis (TSALLIS, 1988; TSALLIS; MENDES; PLASTINO, 1998) proposed a generalized entropy of the BGS statistics, which can be defined as follows:

$$H_T = \frac{1}{q-1}\left(1 - \sum_{n=0}^{N-1} p[n]^q\right). \tag{4.4}$$

Here, q is called the entropic index, which, depending on its value, can represent various types of entropy. Depending on the value of q, three different entropies can be defined (ALBUQUERQUE; ESQUEF; MELLO, 2004; ZHANG; WU, 2011):

- Superextensive entropy $(q < 1)$:

$$H_T(A+B) < H_T(A) + H_T(B) \tag{4.5}$$

- Extensive entropy $(q = 1)$:

$$H_T(A+B) = H_T(A) + H_T(B) \tag{4.6}$$

- Subextensive entropy $(q > 1)$:

$$H_T(A+B) > H_T(A) + H_T(B) \tag{4.7}$$

When $q < 1$, the Tsallis entropy is superextensive; for $q = 1$, it is extensive (e.g., leads to the Shannon entropy), and for $q > 1$, it is subextensive (TSALLIS, 1999). Therefore, based on these differences, it is important to explore the possibility of generalized entropies (YAMANO, 2001; CRUZ-GARCíA; BORY-REYES; RAMIREZ-ARELLANO, 2022; DéRIAN *et al.*, 2022). Another notable generalized entropy is the Rényi entropy, which generalizes the Shannon entropy, the Hartley entropy, the collision entropy and the min-entropy (FEHR; BERENS, 2014; RÉNYI *et al.*, 1961). The Rényi entropy can be defined as follows:

$$H_R = \frac{1}{1-q} log_2 \left(\sum_{n=0}^{N-1} p[n]^q\right). \tag{4.8}$$

As in the Tsallis entropy, $q = 1$ leads to Shannon entropy.

4.3 Materials and Methods

In this section, we describe the experimental methodology adopted for this study, which is divided into five stages: (1) data selection; (2) feature extraction; (3) extensive analysis of the entropic index; (4) performance analysis; and (5) comparative study.

4.3.1 A Novel Feature Extraction Technique

Our proposal is based on the studies of (MACHADO; COSTA; QUELHAS, 2011; BONIDIA *et al.*, 2021a). To generate our probabilistic experiment (VINGA, 2013), we use a known tool in biology, the k-mer. 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 (4.9).

$$\begin{aligned} P_k(\mathbf{s}) = \frac{c_i^k}{N-k+1} = \Bigg(& \frac{c_1^1}{N-1+1}, \ldots, \frac{c_4^1}{N-1+1}, \\ & \frac{c_{4+1}^2}{N-2+1}, \ldots, \frac{c_i^k}{N-k+1} \Bigg) \qquad k = 1, 2, \ldots, n. \end{aligned} \tag{4.9}$$

Here, each sequence ($\mathbf{s}$) was assessed with frequencies of $k = 1, 2, \ \ldots, 24$, in which c_i^k is the number of occurrences with length k in a sequence ($\mathbf{s}$) with length N; the index $i \in \{1, 2, \ldots, 4^1 + \ldots + 4^k\}$ refers to an analyzed substring (e.g., $[\{AAAA\}, \ldots, \{TTTT\}]$, for $k = 4$). Here, after counting the absolute frequencies of each k, we generate relative frequencies and then apply Tsallis entropy to generate the features. In the case of protein sequences, index i is $\{1, 2, \ldots, 20^1 + \ldots + 20^k\}$. For a better understanding, Algorithm 12 demonstrates our pseudocode.

This algorithm is divided into five steps: (1) each sequence is mapped to $k - mers$; (2) extraction of the absolute frequency of each $k - mer$; (3) extraction of the relative frequency of each $k - mer$ based on absolute frequency; (4) extraction of the Tsallis entropy, based on the relative frequency for each $k - mer$—see Equation (4.4); (5) generation, for each $k - mer$, of an entropic measure. Regarding interpretability, each entropic measure represents a $k - mer$, e.g., 1-mer = frequency of A, C, T, G. In other words, by analyzing the best measures—for example, through a feature importance analysis—we can determine which $k - mers$ are more relevant to the problem under study, providing an indication of which combination of nucleotides or amino acids contributes to the classification of the sequences.

```
Algorithm 1: Pseudocode of the Proposed Method
Inputs: S: Biological sequences; ksize: Range k-mer; q: entropic-index
Output: Features generated by Tsallis entropy
begin
    for seq in S do
        for k in range(ksize) do
            extract k combinations (N - k + 1);
            calculate absolute frequency;
            calculate relative frequency;
            calculate Tsallis entropy;
        end
    end
end
```

Figure 12 – Pseudocode of the Proposed Method.

4.3.2 Benchmark Dataset and Experimental Setting

To validate the proposal, we divided our experiments into five case studies:

- **Case Study I:** Assessment of the Tsallis entropy and the effect of the entropic index q, generating 100 feature vectors for each benchmark dataset with 100 different q parameters (entropic index). The features were extracted by Algorithm 12, with q varying from 0.1 to 10.0 in steps of 0.1 (except 1.0, which leads to the Shannon entropy). The goal was to find the best values for the parameter q to be used in the experiments. For this, three benchmark datasets from previous studies were used (BONIDIA *et al.*, 2021a; CHU *et al.*, 2017; MANAVALAN; SHIN; LEE, 2018). For the first dataset (D1), the selected task was long non-coding RNAs (lncRNA) vs. protein-coding genes (mRNA), as in (KLAPPROTH *et al.*, 2021), using a set with mRNA and lncRNA sequences (500 for each label—benchmark dataset (BONIDIA *et al.*, 2021a)). For the second dataset (D2), a benchmark set from (BONIDIA *et al.*, 2021a), the selected task was the induction of a classifier to distinguish circular RNAs (cirRNAs) from other lncRNAs using 1000 sequences (500 for each label). The third dataset (D3) is for Phage Virion Protein (PVP) classification, from (MANAVALAN; SHIN; LEE, 2018), with 129 PVP and 272 non-PVP sequences.

- **Case Study II:** We use the best parameters (q : entropic index—found in case study I) to evaluate its performance on new datasets: D4—Sigma70 Promoters (LIN *et al.*, 2017) (2141 sequences), D5—Anticancer Peptides (LI *et al.*, 2020a) (344 sequences) and D6—Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2, 24815 sequences) (BONIDIA *et al.*, 2022).

- **Case Study III—Comparing Tsallis with Shannon Entropy:** As a baseline of the comparison between methods, we use Shannon entropy, as we did not find any article studying the form of proposed classification with Tsallis entropy and the effect of the entropic parameter with different classifiers. In this experiment, we use D1, D2, D3, D4, D5 and D6.

- **Case Study IV—Comparing Generalized Entropies:** To better understand the effectiveness of generalized entropies for feature extraction, we evaluated Tsallis with the Rényi entropy. In this case, the evaluations of the two approaches were conducted by using the experiments from case study I, changing the entropic index for generating the datasets from 0.1 to 10.0 in steps of 0.1 and inducing the CatBoost classifier. In addition, the datasets used were D1, D2, and D3.

- **Case Study V—Dimensionality Reduction Analysis:** Finally, we assessed our proposal with other known techniques of feature extraction and dimensionality reduction, e.g., Singular Value Decomposition (SVD) (HALKO; MARTINSSON; TROPP, 2011) and Uniform Manifold Approximation and Projection (UMAP) (MCINNES *et al.*, 2018), using datasets D1, D2, D3 and D5. We also added three new benchmark datasets provided by (KHAN *et al.*, 2020) to predict recombination spots (D7) with 1050 sequences (it contained 478 positive sequences and 572 negative sequences) and for the HIV-1 M pure subtype against CRF classification (D8) with 200 sequences (it contained 100 positive and negative sequences) (REMITA *et al.*, 2017). In addition, we also used a multiclass dataset (D9) containing seven bacterial phyla with 488 small RNA (sRNA), 595 transfer RNA (tRNA) and 247 ribosomal RNA (rRNA) from (BONIDIA *et al.*, 2022a). Moreover, to apply SVD and UMAP, we kept the same feature descriptor by k-mer frequency.

For data normalization in all stages, we used the min-max algorithm. Furthermore, we investigated five classification algorithms, such as Gaussian Naive Bayes (GaussianNB), Random Forest (RF), Bagging, Multi-Layer Perceptron (MLP), and CatBoost. To induce our models, we randomly divided the datasets into ten separate sets to perform 10-fold cross-validation (case study I and case study V) and hold-out (70% of samples for training and 30% for testing—case study II, case study III, and case study IV). Finally, we assessed the results with accuracy (ACC), balanced accuracy (BACC), recall, F1 score, and Area Under the Curve (AUC). In D9, we considered metrics suitable for multiclass evaluation.

4.4 Results and Discussion

4.4.1 Case Study I

As aforementioned, we induced our classifiers (using 10-fold cross-validation) across all feature vectors generated with 100 different q parameters (totaling 300 vectors (3 datasets times 100 parameters)). Thereby, we obtained the results presented in Table 22. This table shows the best and worst parameters (entropic parameter q) of each algorithm in the three benchmark datasets, taking into account the ACC metric.

Table 22 – The best and worst parameter (q) of each benchmark dataset and classifier, taking into account the ACC metric.

Dataset	GaussianNB		RF		Bagging		MLP		CatBoost	
	q	ACC	q	ACC	q	ACC	q	ACC	q	ACC
D1	2.7	0.9370	0.4	0.9430	2.7	0.9400	2.2	0.9380	**2.3**	**0.9440**
	9.2	0.4760	9.6	0.7360	9.6	0.7270	10.0	0.5060	9.6	0.747
D2	1.5	0.7980	5.3	0.8220	5.7	0.8080	0.9	0.7800	**4.0**	**0.8300**
	9.6	0.5210	10.0	0.6510	10.0	0.6170	9.9	0.5060	9.2	0.6800
D3	8.7	0.7008	7.8	0.6910	2.0	0.7157	1.5	0.7184	**1.1**	**0.7282**
	1.3	0.6062	9.8	0.5985	9.5	0.5962	0.1	0.6860	5.7	0.6610

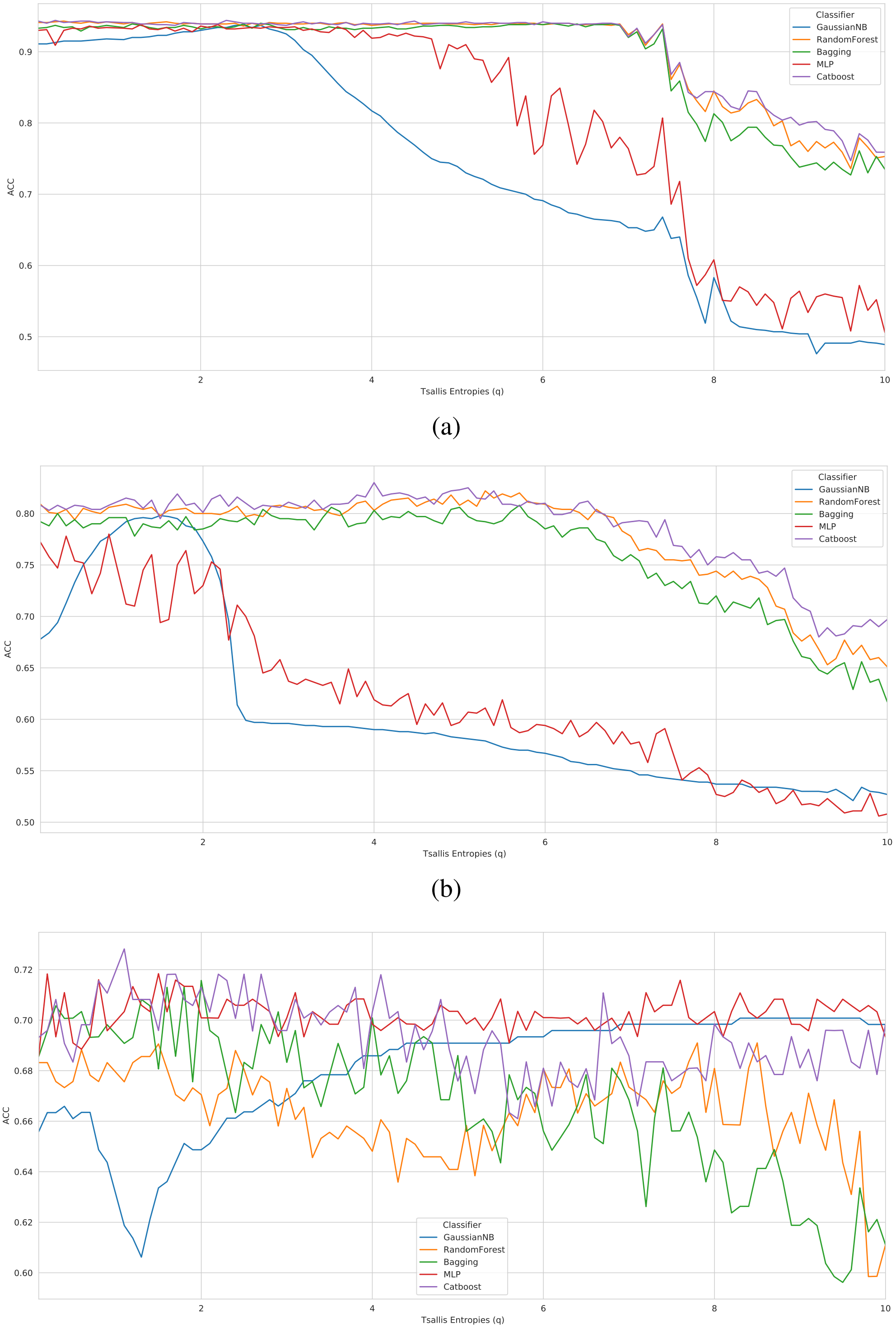
Thereby, evaluating each classifier, we observed that the CatBoost performed best in all datasets, with 0.9440 ($q = 2.3$), 0.8300 ($q = 4.0$), 0.7282 ($q = 1.1$) in D1, D2 and D3, respectively. The other best classifiers were RF, with 0.9430 ($q = 0.4 - \text{D1}$) and 0.8220 ($q = 5.3 - \text{D2}$), followed by Bagging, MLP, and GaussianNB. Furthermore, in general, we noticed that the best results presented parameters between $1.1 < q < 5.0$, i.e., when the Tsallis entropy was subextensive. Along the same lines, it can be observed in Table 22 that the worst parameters are between $9.0 < q < 10.0$ when the Tsallis entropy is also subextensive. However, for a more reliable analysis, we plotted graphs with the results of all tested parameters (0.1 to 10.0 in steps of 0.1), as shown in Figure 13.

A large difference can be observed in the entropy obtained by each parameter q, mainly in benchmark D3. Thereby, analyzing D1 and D2, we noticed a pattern of robust results until $q = 6$, for the best classifiers in both datasets. However, as the q parameter increases, the classifiers are less accurate. On the other hand, if we look at D3, the entropy obtained for each parameter q presents a much greater variation but follows the same drop with parameters close to $q = 10$. Regarding the superextensive entropy $(q < 1)$, some cases showed robust results; however, most classifiers behaved better with the subextensive entropy.

4.4.2 Case Study II

After substantially evaluating the entropic index, our findings indicated that the best parameters were among $1.1 < q < 5.0$. Thereby, we generated new experiments using five parameters to test their efficiency in new datasets, with $q = (0.5, 2.0, 3.0, 4.0, 5.0)$, as shown in Table 23 (sigma70 promoters—D4), Table 24 (anticancer peptides-D5) and Table 25 (SARS-CoV-2—D6). Here, we generated the results with the two best classifiers (RF and Catboost-best in bold).

Assessing each benchmark dataset, we note that the best results were ACC: 0.6687 and AUC: 0.6108 in D4 (RF, $q = 2.0$), ACC: 0.7212 and AUC: 0.7748 in D5 (RF, $q = 3.0$), and ACC:



(a)

(b)

(c)

Figure 13 – Performance analysis with five classifiers on 100 q parameters of three benchmark datasets (evaluation metric: ACC). (**a**) Benchmark D1—ACC; (**b**) Benchmark D2—ACC; (**c**) Benchmark D3—ACC.

1.0000 and AUC: 1.0000 in D6 (RF and CatBoost, $q = 5.0$). Once more, the results confirm that the best parameters are in the range of $1.1 < q < 5.0$, indicating a good choice when using Tsallis entropy.

Table 23 – Performance with different entropic index (q) values for the sigma70 promoter classification problem.

Dataset	q	Classifier	ACC	Recall	F1 Score	AUC	BACC
D4	**0.5**	RF	0.6594	0.2556	0.3423	0.6279	0.5647
		CatBoost	0.6563	0.1973	0.2848	0.6233	0.5487
	2.0	**RF**	**0.6687**	**0.3094**	**0.3932**	**0.6108**	**0.5845**
		CatBoost	0.6641	0.2063	0.2987	0.6301	0.5567
	3.0	RF	0.6672	0.3049	0.3886	0.6150	0.5822
		CatBoost	0.6625	0.2377	0.3282	0.6319	0.5629
	4.0	RF	0.6641	0.2825	0.3684	0.6163	0.5746
		CatBoost	0.6656	0.2466	0.3385	0.6415	0.5674
	5.0	RF	0.6641	0.2825	0.3684	0.6348	0.5746
		CatBoost	0.6734	0.2646	0.3598	0.6375	0.5775

Table 24 – Performance with different entropic index (q) values for the anticancer peptide classification problem.

Dataset	q	Classifier	ACC	Recall	F1 Score	AUC	BACC
D5	**0.5**	RF	0.7019	0.5952	0.6173	0.7437	0.6847
		CatBoost	0.6923	0.3810	0.5000	0.7488	0.6421
	2.0	RF	0.7019	0.5476	0.5974	0.7454	0.6770
		CatBoost	0.6538	0.4286	0.5000	0.7500	0.6175
	3.0	**RF**	**0.7212**	**0.5714**	**0.6234**	**0.7748**	**0.6970**
		CatBoost	0.6827	0.4286	0.5217	0.7385	0.6417
	4.0	RF	0.7019	0.5238	0.5867	0.7823	0.6732
		CatBoost	0.6923	0.4762	0.5556	0.7642	0.6575
	5.0	RF	0.7211	0.5476	0.6133	0.7813	0.6932
		CatBoost	0.6923	0.4762	0.5556	0.7600	0.6575

Table 25 – Performance with different entropic index (q) values for the SARS-CoV-2 (COVID-19) classification problem.

Dataset	q	Classifier	ACC	Recall	F1 Score	AUC	BACC
D6	**0.5**	RF	0.9989	0.9992	0.9994	1.0000	0.9985
		CatBoost	0.9982	1.0000	0.9990	0.9999	0.9947
	2.0	RF	0.9996	1.0000	0.9998	1.0000	0.9990
		CatBoost	0.9951	0.9996	0.9971	1.0000	0.9862
	3.0	**RF**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
		CatBoost	0.9996	1.0000	0.9998	1.0000	0.9990
	4.0	**RF**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
		CatBoost	0.9996	1.0000	0.9998	1.0000	0.9990
	5.0	**RF**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
		CatBoost	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

The perfect classification at D6 is supported by other studies in the literature (RAND-

HAWA *et al.*, 2020; NAEEM *et al.*, 2021; ARSLAN, 2021a). Nevertheless, after testing the Tsallis entropy on six benchmark datasets, we noticed an indication that this approach behaves better with longer sequences, e.g., D1 (mean length ≈ 751 bp), D2 (mean length ≈ 2799 bp), and D6 (mean length ≈ 10,870 bp) showed robust results, while D3 (mean length ≈ 268 bp), D4 (mean length ≈ 81 bp), and D5 (mean length ≈ 26 bp) showed less accurate results.

4.4.3 Case Study III—Comparing Tsallis with Shannon Entropy

Here, we used Shannon entropy as a baseline for comparison, according to Table 26. Various studies have covered the biological sequence analysis with Shannon entropy, in the most diverse applications. For a fair analysis, we reran the experiments on all datasets (case study I and II, six datasets), using hold-out, with the same train and test partition for both approaches. Once more, we used the best classifiers in case study II (RF and CatBoost), but, for a better understanding, we only show the best result in each dataset.

Table 26 – Performance of the proposed approach (Tsallis) vs. Shannon entropy (best results in bold). A tie counts one win for each approach.

Dataset	Classifier	Entropy	*q*	ACC	Recall	F1 Score	BACC
D1	CatBoost	Tsallis	2.3	**0.9420**	**0.9673**	**0.9437**	**0.9421**
		Shannon	-	**0.9420**	0.9651	0.9435	**0.9421**
D2	CatBoost	Tsallis	4.0	**0.8140**	**0.7760**	**0.8053**	**0.8153**
		Shannon	-	0.8080	0.7582	0.7970	0.8115
D3	CatBoost	Tsallis	1.1	**0.7231**	0.3869	**0.4724**	**0.6342**
		Shannon	-	0.7207	**0.3886**	0.4708	0.6334
D4	RF	Tsallis	2.0	**0.6687**	**0.3094**	**0.3932**	**0.5845**
		Shannon	-	0.6563	0.2556	0.3403	0.5623
D5	RF	Tsallis	3.0	**0.7212**	**0.5714**	**0.6234**	**0.6970**
		Shannon	-	0.7115	0.5476	0.6053	0.6851
D6	RF	Tsallis	5.0	0.9984	0.9846	0.9915	0.9922
		Shannon	-	**0.9985**	**0.9888**	**0.9922**	**0.9942**
Mean	-	Tsallis	-	0.8112	0.6659	0.7049	0.7776
		Shannon	-	0.8061	0.6507	0.6915	0.7714
Gain	-	-	-	**0.51%**	**1.52%**	**1.34%**	**0.62%**
Wins	-	Tsallis	-	**5**	**4**	**5**	**5**
		Shannon	-	2	2	1	2

According to Table 26, our proposal with Tsallis entropy showed better results of ACC (5 wins), recall (4 wins), F1 score (5 wins), and BACC (5 wins) than Shannon entropy in five datasets, falling short only on D6, with a small difference of 0.0002. Analyzing each metric individually, we observed that the best Tsallis parameters resulted in an F1 score gain compared to Shannon entropy of 5.29% and 1.81% in D4 and D5, respectively. Other gains were repeated in ACC, recall, and BACC. In the overall average, our proposal achieved improvements of 0.51%, 1.52%, 1.34%, and 0.62% in ACC, recall, F1 score, and BACC, respectively. Despite a lower accuracy in D3 and D4, this approach alone delivered a BACC of 0.6342 and 0.5845, i.e., it is a supplementary methodology to combine with other feature extraction techniques available in the literature. Based on this, we can state that Tsallis entropy is as robust as Shannon entropy for extracting information from biological sequences.

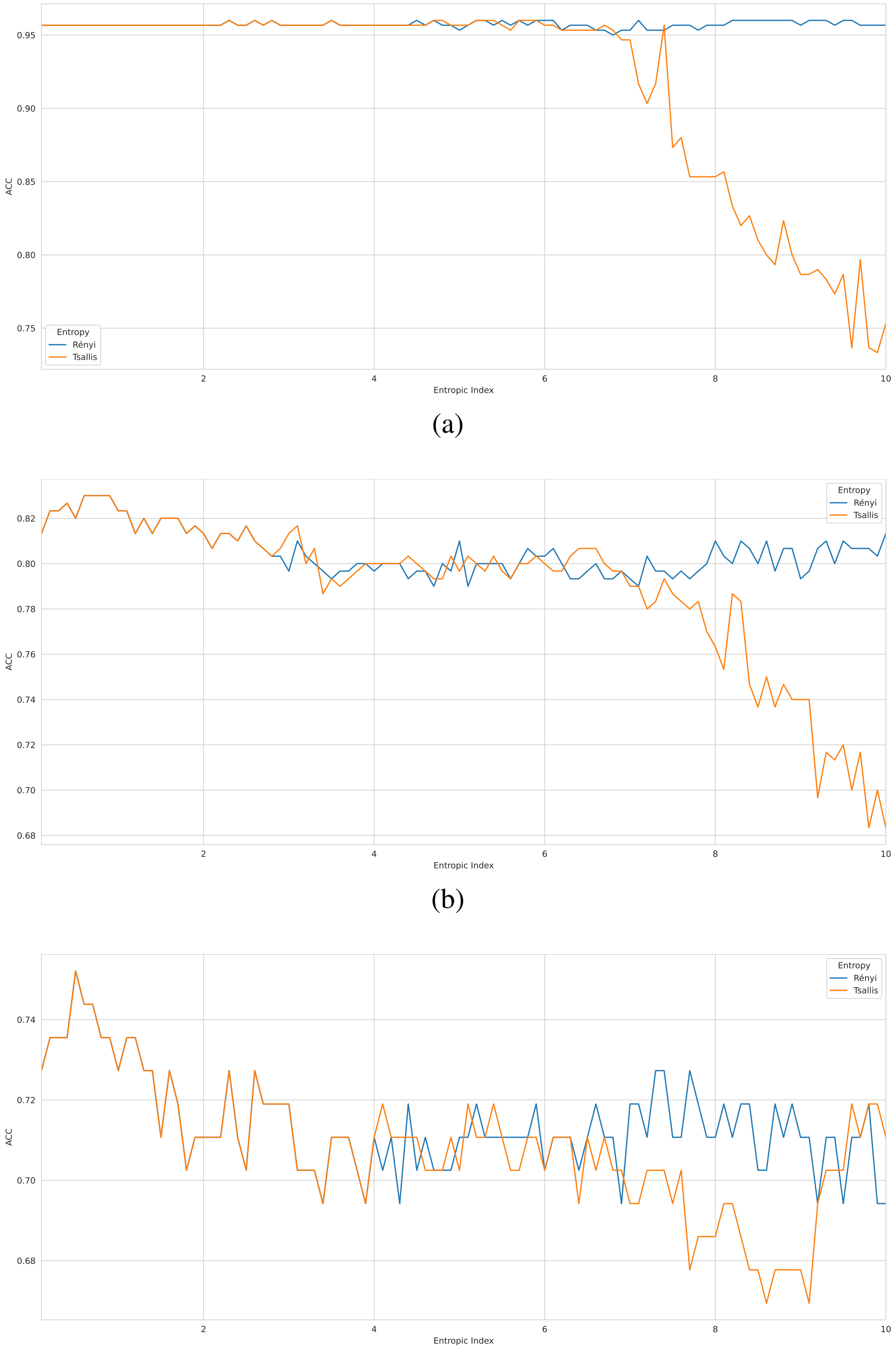
4.4.4 Case Study IV—Comparing Generalized Entropies

According to the Tsallis entropy results, wherein it overcame Shannon entropy, we realized the strong performance of generalized entropy as a feature descriptor for biological sequences. For this reason, we also evaluated the influence of another form of generalized entropy, such as Rényi entropy (RÉNYI *et al.*, 1961), as a good feature descriptor for biological sequences. Here, we investigated the performance of Tsallis and Rényi entropy, changing the entropic index for D1, D2, and D3. Moreover, we have chosen the best classifier from case study I (CatBoost).

When considering the same reproducible environment for the experiment, the performance peak was the same for both methods, as we can see in Figure 14, with graphs containing accuracy performance results for all the entropic index values (from 0.1 to 10.0). Regarding the best classification performance, for D1 (Figure 14a), we had ACC: 0.9600, recall: 0.9667, F1 score: 0.9603, and BACC: 0.9600; for D2 (Figure 14b), we obtained ACC: 0.8300, recall: 0.7733, F1 score: 0.8198, and BACC: 0.8300; and for D3 (Figure 14c), we had ACC: 0.7521, recall: 0.359, F1 score: 0.4828, and BACC: 0.649. As seen earlier, Tsallis entropy performs poorly from a specific entropy index onwards, but Rényi entropy demonstrates more consistent performance when compared to Tsallis, representing a possible alternative. Nevertheless, the results again highlight the promising use of generalized entropies as a feature extraction approach for biological sequences.

4.4.5 Case Study V—Dimensionality Reduction

In this last case study, we compared our proposal with other known techniques for feature extraction and dimensionality reduction in the literature, using the same representation of the biological sequences, the $k-mer$ frequency. In particular, for each DNA/RNA sequence, we generated $k-mers$ from $k=1$ to $k=10$, while, for proteins, we generated it until $k=5$, considering the high number of combinations with amino acids. All datasets used have around 1000 biological sequences, considering the prohibitive computational cost to deal with the $k-mer$ approach. In this study, our objective was to use SVD and UMAP to reduce the dimensionality of the $k-mer$ feature vector by extracting new features, as we did in our approach. However,



(a)

(b)

(c)

Figure 14 – Performance analysis with generalized entropies on 100 q parameters of three benchmark datasets (evaluation metric: ACC). (**a**) Benchmark D1—ACC; (**b**) Benchmark D2—ACC; (**c**) Benchmark D3—ACC.

high values of k present high computational costs, due to the amount of generated features, e.g., $k = 6$ in DNA (4096 features) and $k = 3$ in protein (8000 features).

From previous case studies, we realized that the feature extraction with Tsallis entropy provided interesting results. Thereby, we extended our study, applying SVD and UMAP in the datasets with $k - mer$ frequencies, reducing them to 24 components, comparable to the dimensions generated in our studies. Fundamentally, UMAP can deal with sparse data, as can SVD, which is known for its efficiency in dealing with this type of data (BERRY, 1992; RAJAMANICKAM, 2009; MCINNES; HEALY; MELVILLE, 2018). Both reduction methods can be used in the context of working with high-dimensional data. Although UMAP is widely used for visualization (BECHT *et al.*, 2019; DORRITY *et al.*, 2020), the reduction method can be used for feature extraction, which is part of an ML pipeline (LI *et al.*, 2022). UMAP can also be used with raw data, without needing to adopt another reduction technique before using it (MCINNES; HEALY; MELVILLE, 2018). We induced the CatBoost classifier using 10-fold cross-validation. We obtained the results listed in Table 27.

Table 27 – Performance of the proposed approach (Tsallis) vs. SVD vs. UMAP. A tie counts one win for each approach.

Dataset	Reduction	ACC	Recall	F1 Score	BACC
D1	Tsallis ($q = 2.3$)	**0.9430**	0.9650	**0.9438**	**0.9434**
	SVD	0.4980	0.0000	0.0000	0.4982
	UMAP	0.4980	**0.9963**	0.6632	0.4981
D2	Tsallis ($q = 4.0$)	**0.8120**	**0.7718**	**0.8030**	**0.8114**
	SVD	0.5004	0.0016	0.0032	0.5008
	UMAP	0.4994	0.0000	0.0000	0.5000
D3	Tsallis ($q = 1.1$)	**0.7307**	0.3538	0.4541	**0.6310**
	SVD	0.5389	0.7132	**0.4942**	0.5834
	UMAP	0.3191	**0.9933**	0.4825	0.4967
D5	Tsallis ($q = 3.0$)	0.6720	0.5181	0.5515	0.6508
	SVD	**0.7403**	**0.7630**	**0.7752**	**0.7261**
	UMAP	0.4021	0.0000	0.0000	0.5000
D7	Tsallis ($q = 3.0$)	**0.7371**	**0.6711**	**0.6947**	**0.7337**
	SVD	0.5438	0.0000	0.0000	0.4992
	UMAP	0.5143	0.1824	0.1147	0.4963
D8	Tsallis ($q = 1.1$)	0.6500	0.6111	0.6277	0.6525
	SVD	**0.8023**	**0.8575**	**0.7843**	**0.8171**
	UMAP	0.6326	0.7728	0.6544	0.6511
D9	Tsallis ($q = 9.2$)	**0.9489**	**0.9481**	**0.9507**	**0.9481**
	SVD	0.5586	0.6433	0.5517	0.6433
	UMAP	0.5992	0.6528	0.6167	0.6528
Mean	Tsallis	0.7848	0.6913	0.7179	0.7673
	SVD	0.5975	0.4255	0.3727	0.6097
	UMAP	0.4950	0.5139	0.3616	0.5421
Wins	Tsallis	**5**	**3**	**4**	**5**
	SVD	2	2	3	2
	UMAP	0	2	0	0

As can be seen, Tsallis entropy achieved five wins, against two for SVD and zero for UMAP, taking into account the ACC. In addition, in the general average, we obtained a gain of more than 18% in relation to SVD and UMAP in ACC, indicating that our approach can be potentially representative for collecting information in fewer dimensions for sequence classification problems.

4.5 Chapter Remarks

In this study, we evaluated the Tsallis entropy as a feature extraction technique, where we considered five case studies with nine benchmark datasets of sequence classification problems, as follows: (1) we assessed the Tsallis entropy and the effect of the entropic index; (2) we used the best parameters on new datasets; (3–4) we validated our study, using the Shannon and Rényi entropy as a baseline; and (5) we compared Tsallis entropy with other feature extraction techniques based on dimensionality reduction. In all case studies, we found that our proposal is robust for extracting information from biological sequences. Furthermore, the Tsallis entropy's performance is strongly associated with the length of sequences, providing better results when applied in longer sequences. The experiments also showed that Tsallis entropy is robust when compared to Shannon entropy. Regarding the limitations, we found that the entropic index (q) affects the performance of ML models, particularly when poorly parameterized. Finally, we highlighted good performance for the entropic index with q values between 1.1 and 5.0.

CHAPTER

5

MATHFEATURE: FEATURE EXTRACTION PACKAGE

One of the main challenges in the application of Machine Learning (ML) algorithms to biological sequence data is how to numerically represent a sequence in a numeric input vector. Feature extraction techniques capable of extracting numerical information from biological sequences have been reported in the literature. However, many of these techniques are not available in existing packages, such as mathematical descriptors. This paper presents a new package, MathFeature, which implements mathematical descriptors able to extract relevant numerical information from biological sequences, i.e., DNA, RNA, and Proteins (prediction of structural features along the primary sequence of amino acids). MathFeature makes available 20 numerical feature extraction descriptors based on approaches found in the literature, e.g., multiple numeric mappings, genomic signal processing, chaos game theory, entropy, and complex networks. MathFeature also allows the extraction of alternative features, complementing the existing packages. To ensure that our descriptors are robust and to assess their relevance, experimental results are presented in nine case studies. According to these results, the features extracted by MathFeature showed high performance (0.6350-0.9897, accuracy), both applying only mathematical descriptors, but also hybridization with well-known descriptors in the literature. Finally, through MathFeature, we overcome several studies in eight benchmark datasets, exemplifying the robustness and viability of the proposed package. MathFeature advances in the area by bringing descriptors not available in other packages, as well as allowing non-experts to use feature extraction techniques.

5.1 Background

Machine learning (ML) algorithms have been successfully applied to genomics, transcriptomics, and proteomics problems (DINIZ; CANDURI, 2017; SOUZA *et al.*, 2018). Nevertheless, their predictive performance depends on the representation of the sequences by relevant features, able to extract important aspects present in the original sequences. In Chou (2011), Liu *et al.* (2015), the authors address the relevance of using an appropriate mathematical expression to extract features from biological data, which has been adopted by several studies (BONIDIA *et al.*, 2019; LIU; GAO; ZHANG, 2019; CHEN *et al.*, 2019), e.g., non-classical secreted proteins (ZHANG *et al.*, 2020), phage virion proteins (MANAVALAN; SHIN; LEE, 2018), SARS-CoV-2 (NAEEM *et al.*, 2020; ARSLAN, 2021b), sigma70 promoters (LIN *et al.*, 2017), Long Non-Coding RNAs (HAN *et al.*, 2018; BONIDIA *et al.*, 2020a).

As result, many techniques have been proposed and experimentally investigated (CHEN *et al.*, 2014a; CHEN *et al.*, 2014b), and several of them were made available in public software packages, such as PROFEAT (LI *et al.*, 2006), PseAAC (SHEN; CHOU, 2008), propy (CAO; XU; LIANG, 2013), PseKNC-General (CHEN *et al.*, 2014b), SPiCE (BERG *et al.*, 2014), protr/ProtrWeb (XIAO *et al.*, 2015), ProFET (OFER; LINIAL, 2015), Pse-in-One (LIU *et al.*, 2015), repDNA (LIU *et al.*, 2014), Rcpi (CHIU *et al.*, 2015), repRNA (LIU *et al.*, 2016), BioSeq-Analysis (LIU, 2017), iFeature (CHEN *et al.*, 2018), PyBioMed (DONG *et al.*, 2018), Seq2Feature (NIKAM; GROMIHA, 2019), PyFeat (MUHAMMOD *et al.*, 2019), iLearn (CHEN *et al.*, 2019), periodicDNA (SERIZAY; AHRINGER, 2021), and iLearnPlus (CHEN *et al.*, 2021).

These software packages have been used to extract features from sequences. However, there are some aspects present in the sequences that the features extraction techniques included in these tools cannot extract. These features, which were shown to be relevant in previous studies (MACHADO; COSTA; QUELHAS, 2011; HOANG; YIN; YAU, 2016; MENDIZABAL-RUIZ *et al.*, 2017; BONIDIA *et al.*, 2021b), describe mathematical aspects observed in biological sequences and will be named here mathematical descriptors (NGUYEN; CANG; WEI, 2020). These descriptors are based on several techniques, such as multiple numeric mappings, Fourier transform, chaos game theory, entropy, and complex networks. To allow the extraction of these descriptors as features for the study of biological sequences, but also including conventional descriptors available in other packages, we created a novel open-source Python package, named MathFeature.

This package provides, in a single environment, many of the mathematical descriptors previously proposed for feature extraction from biological sequences (MACHADO; COSTA; QUELHAS, 2011; HOANG; YIN; YAU, 2016; MENDIZABAL-RUIZ *et al.*, 2017; BONIDIA *et al.*, 2021b). MathFeature contains 37 descriptors, in which, 20 of them are mathematical, organized into five groups (numerical mapping, chaos game, Fourier transform, entropy, and graphs). Additionally, MathFeature extends our preliminary investigation (BONIDIA *et al.*, 2021b), where we investigated nine sets of mathematical features. MathFeature also includes descriptors for Protein sequences, i.e., prediction of structural features along the primary sequence of amino acids. To the best of our knowledge, MathFeature is the first package to provide such a large and comprehensive set of feature extraction techniques based on mathematical descriptors for DNA, RNA, and Proteins.

5.2 Related Works

Fundamentally, we consider feature engineering a key step to ML application success (GUYON *et al.*, 2008; VISHNOI; GARG; ARORA, 2020; GHANNAM; TECHTMANN, 2021b), mainly in the biological sequences preprocessing (CHOU, 2011; SAIDI *et al.*, 2012; ZHANG *et al.*, 2021). In terms of terminology, according to (GUYON *et al.*, 2008), feature is synonymous of an input variable or attribute. Nevertheless, studies also use the *feature descriptor* terminology (the majority in our review - 15 studies), being the reason why we adopt this term, where a feature descriptor refers to the feature extraction method/technique that can present several measures/values.

In this section, we described 17 studies (cited in Background Section) related to feature extraction packages (tools, web servers, toolkits, etc), providing several feature descriptors for biological sequence analyzes. We organized the selected studies into application categories (that is, DNA, RNA, or Protein), as exposed in Table 28. Furthermore, we also plotted a Venn Diagram (Figure 15) with the composition of all studies by application. In general, most studies are focused on the representation of proteins (eight studies), while DNA and RNA studies had one application each. Moreover, considering the intersection of applications, we found four studies of applications combining DNA, RNA, and Protein, while DNA+Protein with two studies and DNA+RNA with one study, respectively.

Table 28 – Selected studies by application.

Application	**Study**
DNA	(LIU *et al.*, 2014)
RNA	(LIU *et al.*, 2016)
Protein	(LI *et al.*, 2006), (SHEN; CHOU, 2008), (CAO; XU; LIANG, 2013), (BERG *et al.*, 2014), (XIAO *et al.*, 2015), (OFER; LINIAL, 2015), (CHIU *et al.*, 2015), (CHEN *et al.*, 2018)
DNA + RNA	(CHEN *et al.*, 2014b)
DNA + Protein	(DONG *et al.*, 2018), (NIKAM; GROMIHA, 2019)
DNA + RNA + Protein	(LIU *et al.*, 2015), (LIU, 2017), (MUHAMMOD *et al.*, 2019),(CHEN *et al.*, 2019)

In our literature review, we found 173 feature descriptors. It is not feasible to individually analyze and describe each descriptor. For this reason, we have divided, based on our review, these descriptors into 15 large groups, as shown in Table 29. The group column classifies the feature descriptors based on the reviewed studies, and the study column includes packages that have at least one descriptor from the related group.

Considering the groups introduced in Table 29, we realized that most descriptors are based on AAC, PseAAC, CTD, and SO for proteins, while NAC and PseNAC descriptors for

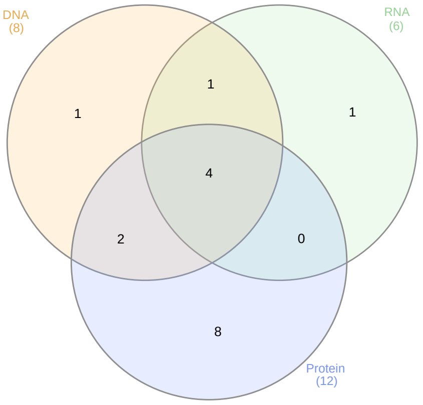


Figure 15 – Venn Diagram - Intersection of selected studies by application.

Table 29 – Descriptor groups in reviewed studies.

Group	Initials	Application Group	Study
Amino Acid Composition	AAC	Protein	(CHEN *et al.*, 2019) (LIU *et al.*, 2015) (LI *et al.*, 2006) (CAO; XU; LIANG, 2013) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (OFER; LINIAL, 2015) (CHIU *et al.*, 2015) (LIU, 2017) (CHEN *et al.*, 2018) (DONG *et al.*, 2018) (NIKAM; GROMIHA, 2019) (MUHAMMOD *et al.*, 2019)
Pseudo-Amino Acid Composition	PseAAC	Protein	(CHEN *et al.*, 2019) (LIU *et al.*, 2015) (SHEN; CHOU, 2008) (CAO; XU; LIANG, 2013) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (CHIU *et al.*, 2015) (LIU, 2017) (CHEN *et al.*, 2018) (DONG *et al.*, 2018)
Composition, Transition, Distribution	CTD	Protein	(CHEN *et al.*, 2019) (LI *et al.*, 2006) (CAO; XU; LIANG, 2013) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (OFER; LINIAL, 2015) (CHIU *et al.*, 2015) (CHEN *et al.*, 2018) (DONG *et al.*, 2018)
Sequence-Order	SO	Protein	(CHEN *et al.*, 2019) (LI *et al.*, 2006) (CAO; XU; LIANG, 2013) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (CHIU *et al.*, 2015) (CHEN *et al.*, 2018) (DONG *et al.*, 2018)
Conjoint Triad	CT	Protein	(CHEN *et al.*, 2019) (XIAO *et al.*, 2015) (CHIU *et al.*, 2015) (CHEN *et al.*, 2018) (DONG *et al.*, 2018)
Proteochemometric Descriptors	PCM	Protein	(CHEN *et al.*, 2019) (XIAO *et al.*, 2015) (CHIU *et al.*, 2015) (CHEN *et al.*, 2018)
Profile-based Features	PF	Protein	(CHEN *et al.*, 2019) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (CHIU *et al.*, 2015) (LIU, 2017) (CHEN *et al.*, 2018)
Nucleic Acid Composition	NAC	DNA, RNA	(CHEN *et al.*, 2019) (LIU *et al.*, 2015) (CHEN *et al.*, 2014b) (LIU *et al.*, 2014) (LIU *et al.*, 2016) (LIU, 2017) (DONG *et al.*, 2018) (MUHAMMOD *et al.*, 2019)
Pseudo Nucleic Acid Composition	PseNAC	DNA, RNA	(CHEN *et al.*, 2019) (LIU *et al.*, 2015) (CHEN *et al.*, 2014b) (LIU *et al.*, 2014) (LIU *et al.*, 2016) (LIU, 2017) (DONG *et al.*, 2018)
Structure Composition	SC	DNA, RNA, Protein	(CHEN *et al.*, 2019) (LIU *et al.*, 2016) (LIU, 2017) (CHEN *et al.*, 2018)
Sequence Similarity	SS	DNA, RNA, Protein	(CHIU *et al.*, 2015)
Autocorrelation	-	DNA, RNA, Protein	(CHEN *et al.*, 2019) (LI *et al.*, 2006) (CAO; XU; LIANG, 2013) (CHEN *et al.*, 2014b) (BERG *et al.*, 2014) (XIAO *et al.*, 2015) (LIU *et al.*, 2015) (LIU *et al.*, 2014) (CHIU *et al.*, 2015) (LIU, 2017) (CHEN *et al.*, 2018) (DONG *et al.*, 2018)
Numerical Mapping	-	DNA, RNA, Protein	(CHEN *et al.*, 2019) (CHEN *et al.*, 2018)
K-Nearest Neighbor	KNN	DNA, RNA, Protein	(CHEN *et al.*, 2019) (CHEN *et al.*, 2018)
Physicochemical Property	PP	DNA, RNA, Protein	(CHEN *et al.*, 2019) (OFER; LINIAL, 2015) (CHEN *et al.*, 2018) (NIKAM; GROMIHA, 2019)

Table 30 – Descriptors calculated by MathFeature compared to the available feature extraction packages. This table shows the number of MathFeature descriptors that existing packages have implemented.

Package	Mathematical Descriptors	Conventional Descriptors	Number of Descriptors Calculated
MathFeature	20	17	37
PROFEAT	0	2	2
PseAAC	0	2	2
propy	0	5	5
PseKNC-General	0	5	5
SPiCE	0	4	4
ProtrWeb	0	5	5
ProFET	2	3	5
Pse-in-One	0	5	5
repDNA	0	5	5
Rcpi	0	3	3
repRNA	0	5	5
BioSeq-Analysis	0	9	9
iFeature	1	4	5
PyBioMed	0	7	7
Seq2Feature	0	0	0
PyFeat	1	8	9
iLearn	2	13	15

DNA/RNA, and AC (Autocorrelation) for DNA, RNA, and protein. Nevertheless, MathFeature overcomes other packages in different types of mathematical descriptors (e.g., chaos game, Fourier transform, entropy and graphs), except two descriptors in numerical mapping, available in only two packages (CHEN *et al.*, 2018; CHEN *et al.*, 2019). In addition, to better illustrate the advantages of MathFeature compared with other studies, we included the Table 30, which shows the number of MathFeature descriptors that can also be found in other tools. In that case, it can be noticed that only iLearn has 15 descriptors from a total of 37 descriptors available on MathFeature. Also, we found only a few sets (2 up to 9) of similar descriptors from other packages compared to our study. Based on this analysis, we realized the novelty of MathFeature for providing different descriptors in biological sequences, which we believe be an important contribution. Moreover, most studies (13, 76.47%) were dedicated to evaluating only one type of sequence, while 4 (23.53%) studies cover multiple types of sequences, including MathFeature. Finally, our package is also competitive in terms of descriptors number (total of 37).

5.3 Package Description

MathFeature is a user-friendly package that covers 20 mathematical descriptors, as illustrated by Figure 16. We also elaborate the MathFeature execution workflow, which can be divided into four simple steps, as shown in Figure 18. In Table 31, we organized the 20 descriptors into 5 groups[1] (numerical mapping (7), chaos game (2), Fourier transform (7), entropy (2), and graphs (2)), according to their structure. MathFeature can be run on the console, but we also

[1] $L =$ length of the longest sequence, $k =$ frequencies of k-mer, $t =$ threshold - number of subgraphs.

provide a GUI-based platform, as shown in Figure 17. We briefly describe each of the 5 groups representing the 20 descriptors:

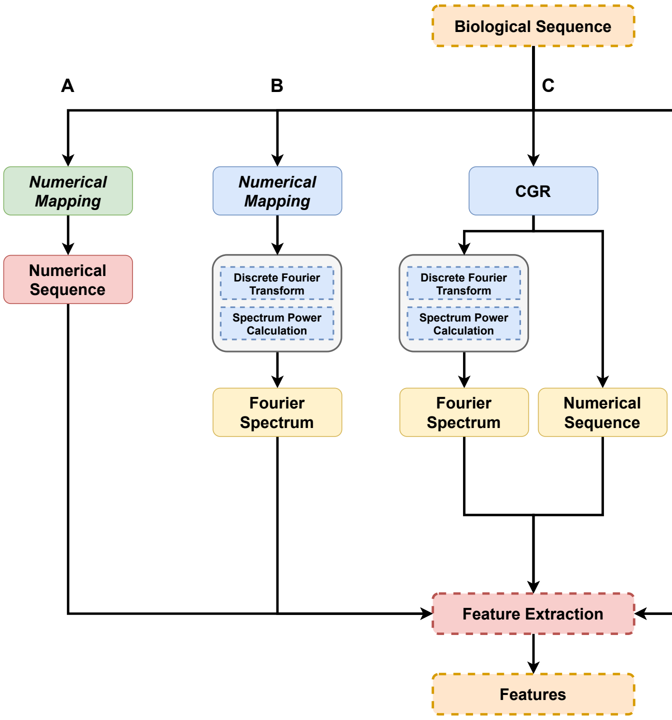
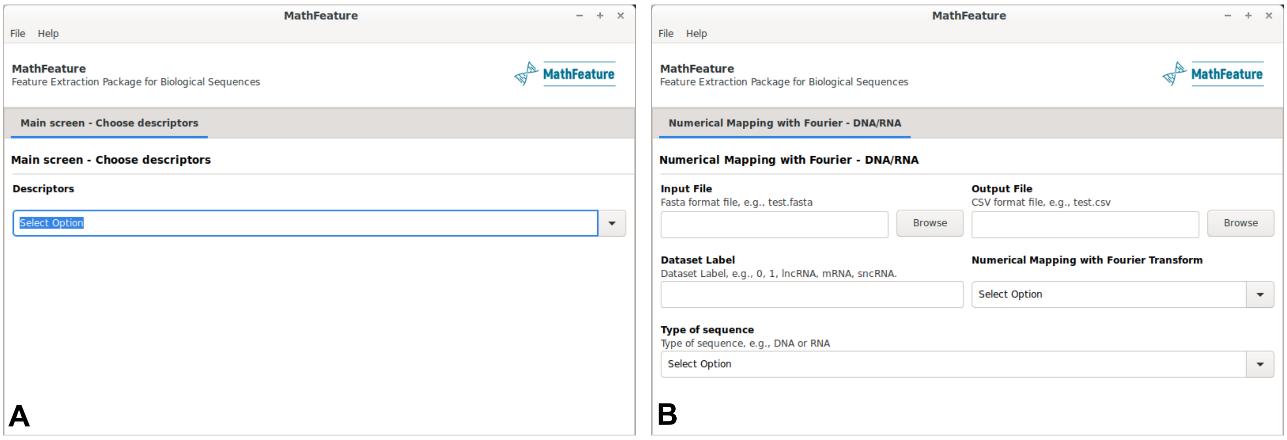


Figure 16 – Pipeline of descriptors calculated by MathFeature. **A:** Numerical Mapping; **B:** Fourier Transform; **C:** Chaos Game Representation; **D:** Entropy; **E:** Complex Networks.


MathFeature
File Help
MathFeature
Feature Extraction Package for Biological Sequences
MathFeature
Main screen - Choose descriptors
A
MathFeature
File Help
MathFeature
Feature Extraction Package for Biological Sequences
MathFeature
Numerical Mapping with Fourier - DNA/RNA
Browse
Browse
B

Main screen - Choose descriptors

Descriptors

Select Option

Numerical Mapping with Fourier - DNA/RNA

Input File
Fasta format file, e.g., test.fasta

Output File
CSV format file, e.g., test.csv

Dataset Label
Dataset Label, e.g., 0, 1, lncRNA, mRNA, sncRNA.

Numerical Mapping with Fourier Transform
Select Option

Type of sequence
Type of sequence, e.g., DNA or RNA
Select Option

Figure 17 – MathFeature - GUI-based platform. **A:** Home screen and **B:** Fourier-based descriptor.

- **Numerical Mapping:** Several sequence analysis studies require converting a biological sequence to a numeric sequence. Previous studies Zhang and Zhang (1994), Anastassiou (2001), Cristea (2002) have proposed descriptors for such, which are able to represent important aspects of these sequences. This group contains 7 descriptors for numerical mapping: Voss (VOSS, 1992) (known as binary mapping), Integer (CRISTEA, 2002), Real (CHAKRAVARTHY *et al.*, 2004), Z-curve (ZHANG; ZHANG, 1994), Electron-Ion Interaction Potential (EIIP) (NAIR; SREENADHAN, 2006; BLOCH; ARCE, 2006), Complex

Table 31 – Mathematical descriptors calculated by MathFeature for DNA, RNA, and Protein sequences.

Descriptor groups	**Descriptor**	**Dimension**	**Biological Sequence**
Numerical Mapping	Binary	$L \cdot 4$	DNA/RNA
	Z-curve	$L \cdot 3$	DNA/RNA
	Real	L	DNA/RNA
	Integer	L	DNA/RNA/Protein
	EIIP	L	DNA/RNA/Protein
	Complex Number	L	DNA/RNA
	Atomic Number	L	DNA/RNA
Fourier Transform	Binary + Fourier	19	DNA/RNA
	Z-curve + Fourier	19	DNA/RNA
	Real + Fourier	19	DNA/RNA
	Integer + Fourier	19	DNA/RNA/Protein
	EIIP + Fourier	19	DNA/RNA/Protein
	Complex Number + Fourier	19	DNA/RNA
	Atomic Number + Fourier	19	DNA/RNA
Chaos Game	Chaos Game Representation	$L \cdot 2$	DNA/RNA
	Chaos Game Signal (with Fourier)	19	DNA/RNA
Entropy	Shannon	k	DNA/RNA/Protein
	Tsallis	k	DNA/RNA/Protein
Graphs	Complex Networks (with threshold)	$12 \cdot t$	DNA/RNA/Protein
	Complex Networks (without threshold)	$26 \cdot k$	DNA/RNA/Protein

Numbers (Anastassiou, 2001; YU; LI; YU, 2018) and Atomic Number (HOLDEN *et al.*, 2007; MENDIZABAL-RUIZ *et al.*, 2017).

- **Fourier Transform (FT):** This group consists of feature extraction methods which generate sequence features based on Genomic Signal Processing (GSP), using FT, a widely applied approach in several biological sequence analysis problems (YIN; CHEN; YAU, 2014; HOANG; YIN; YAU, 2016; MENDIZABAL-RUIZ *et al.*, 2017; BONIDIA *et al.*, 2021b). To implement GSP techniques, we use all numerical mappings. A mathematical exploration can be seen in Bonidia *et al.* (2021b).

- **Chaos Game Representation (CGR):** This approach is also a mapping to a numerical sequence, but scale-independent and iterative for geometric representation of DNA sequences (JEFFREY, 1990). Based on available CGR representations, MathFeature package considers classical CGR (JEFFREY, 1990; HOANG; YIN; YAU, 2016), frequency CGR (ALMEIDA *et al.*, 2001), and CGR signal with Fourier Transform (FT) (HOANG; YIN; YAU, 2016).

- **Entropy:** Different studies have applied concepts from information theory for sequence feature extraction, mainly Shannon's Entropy (SE) (AKHTER *et al.*, 2013; MACHADO; COSTA; QUELHAS, 2011). According to (YAMANO, 2001), Tsallis Entropy (TE)

(TSALLIS; MENDES; PLASTINO, 1998) has been successfully explored in several studies. Moreover, Tsallis entropy try to generalize the Boltzmann/Gibbs's traditional entropy. This group includes these two descriptors (BONIDIA *et al.*, 2021b).

- **Graphs:** This group has descriptors based on graph theory (Complex Networks (CN)) which has been successfully used to represent biological sequence for classification tasks (PAVLOPOULOS *et al.*, 2011; AITTOKALLIO; SCHWIKOWSKI, 2006). The descriptors implemented in this group include techniques proposed in Ito *et al.* (2018) and explored in Bonidia *et al.* (2021b).

MathFeature also provides well-known descriptors from other studies with biological sequences (here named conventional descriptors, see Table 32, due to the large number of implementations in the revised packages, see Table 29) such as Nucleic acid composition (NAC), dinucleotide composition (DNC), trinucleotide composition (TNC), pseudo K-tuple nucleotide composition (PseKNC) (CHEN *et al.*, 2014b), accumulated nucleotide frequency (ANF - DNA, RNA, and protein) (NARAYAN *et al.*, 1994), basic k-mer (DNA, RNA, and protein) (MAPLESON *et al.*, 2016), AAC, dipeptide composition (DPC), tripeptide composition (TPC), and Xmer k-Spaced Ymer composition frequency (kGap - DNA, RNA, and protein) (MUHAMMOD *et al.*, 2019). In addition, we have also implemented two widely known descriptors in coding sequence studies, e.g., ORF (open reading frame) or coding features (BONIDIA *et al.*, 2021b) and Fickett score (WANG *et al.*, 2013). Finally, we summarized the set of features generated by each descriptor investigated in this study (mathematical and conventional), as described in Table 33. MathFeature is freely available at https://github.com/Bonidia/MathFeature, and its documentation is provided at https://bonidia.github.io/MathFeature/.

Table 32 – Conventional descriptors calculated by MathFeature for DNA, RNA, and Protein sequences.

Descriptor groups	Descriptor	Dimension	Biological Sequence
Other descriptors	Basic k-mer	4^k or 20^k	DNA/RNA/Protein
	Customized k-mer	4^k or 20^k	DNA/RNA/Protein
	NAC	4	DNA/RNA
	DNC	16	DNA/RNA
	TNC	64	DNA/RNA
	ORF Features or Coding Features	10	DNA/RNA
	Fickett score	2	DNA/RNA
	PseKNC	-	DNA/RNA
	ANF	L	DNA/RNA/Protein
	kGap	$4^X \cdot 4^Y$ or $20^X \cdot 20^Y$	DNA/RNA/Protein
	AAC	20	Protein
	DPC	400	Protein
	TPC	8000	Protein

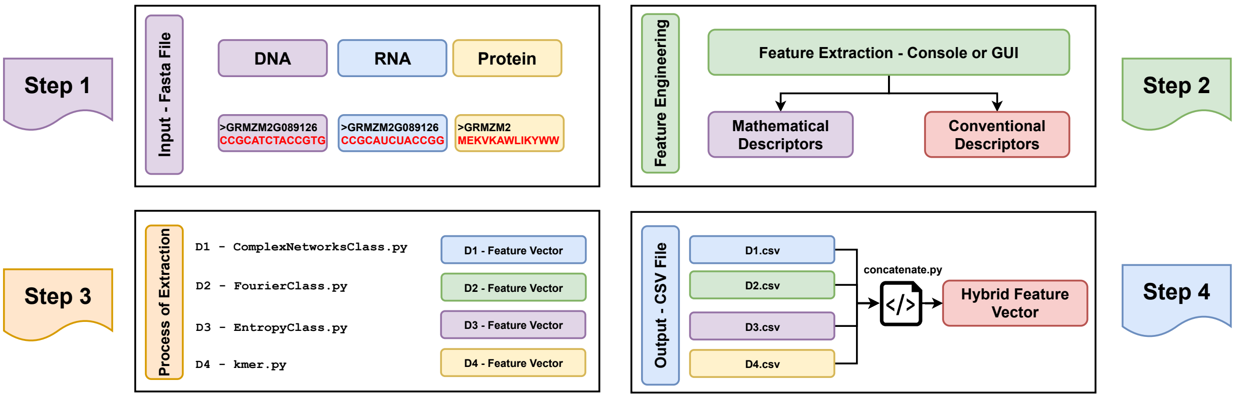


Figure 18 – MathFeature Execution Workflow. **Step 1:** Select input sequence (DNA/RNA/Protein - MathFeature only accepts fasta format); **Step 2:** Choose the descriptor (mathematical or conventional); **Step 3:** It is necessary to run each descriptor separately; **Step 4:** With the generated vectors, you can use them separately or hybridize them in a single vector.

5.4 Results

The main of this paper is to make publicly available a large set of feature extraction techniques for biological sequences, including mathematical descriptors not found in similar packages. These descriptors have been successfully applied to extract relevant features from biological sequences, as can be seen in Bonidia *et al.* (2021b), Hoang, Yin and Yau (2016), Yin, Chen and Yau (2014), Machado, Costa and Quelhas (2011), and Ito *et al.* (2018). For this reason, to assess the relevance of MathFeature descriptors, we provide case studies, which are detailed and presented in the experimental scenario section.

5.4.1 Experimental Setting

We run experiments for nine case studies with distinct scenarios for the classification of DNA, RNA, and protein sequences, as shown in Table 34. These case studies compare the use of several descriptors in distinct problem domains. Furthermore, we did not include any feature selection or hyperparameter optimization technique. Hence, for a fair comparison, we have selected descriptors using stratified random sampling (choosing descriptors in each group defined in the article, e.g., numerical mapping, Fourier transform, chaos game, entropy, graphs, and conventional) in all case studies to avoid any biased choices according to the problem domain. In addition, to compare our results with state-of-the-art studies, we use different ML algorithms, performance measures, and dataset partitions to adapt our pipeline to the benchmark dataset. Finally, we also select hybridized features using stratified random sampling, to assess how these feature sets can improve the machine learning model prediction.

5.4.2 Case Study I - Non-Classical Secreted Proteins

Here, we induced a classifier for the non-classical secreted proteins using benchmark datasets provided by Zhang *et al.* (2020) (training: 141 positive and 446 negative samples; test:

Table 33 – Features generated by each mathematical and conventional descriptor calculated by MathFeature.

Descriptors	Features
Binary, Z-curve, Real, Integer, EIIP, Complex Number, Atomic Number, CGR, ANF	Convert a biological sequence into a numerical sequence, e.g., Integer representation: GAGAGTGACCA == 3, 2, 3, 2, 3, 0, 3, 2, 1, 1, 2.
Binary + Fourier, Z-curve + Fourier, Real + Fourier, Integer + Fourier, EIIP + Fourier, Complex Number + Fourier, Atomic Number + Fourier, Chaos Game Signal (with Fourier)	Peak to average power ratio (2 features), average power spectrum, median, maximum, minimum, sample standard deviation, population standard deviation, percentile (15/25/50/75), range, variance, interquartile range, semi-interquartile range, coefficient of variation (cv), skewness, and kurtosis.
Shannon, Tsallis	For each k-mer (e.g., 1-mer, 2-mers, . . . , k-mers), we generate an entropic measure.
Complex Networks (with threshold)	Betweenness, assortativity, average degree, average path length, minimum degree, maximum degree, number of edges, degree standard deviation, frequency of motifs (size 3 and 4), clustering coefficient (local and global).
Complex Networks (without threshold)	Betweenness, assortativity, average degree, average path length, minimum degree, maximum degree, number of edges, degree standard deviation, frequency of motifs (size 3 and 4), clustering coefficient (local and global), Kleinberg's authority centrality scores, closeness centralities, Burt's constraint scores, multiplicities, density, diameter, eccentricity, edge betweenness, Kleinberg's hub score, maximum degree of a vertex set, neighborhood size, radius, strength (weighted degree), number of vertices.
k-mer, Customized k-mer, NAC, DNC, TNC, AAC, DPC, TPC, kGap	Generation of nucleic acid or amino acid statistical information, e.g., NAC for DNA: relative frequency of A, C, T, G.
ORF Features or Coding Features	Maximum ORF length, minimum ORF length, std ORF length, average ORF length, cv ORF length, maximum GC content - ORF, minimum GC content - ORF, std GC content - ORF, average GC content - ORF, cv GC content - ORF.
Fickett score	Fickett:orf, Fickett:full:sequence
PseKNC	Modes of PseKNC with physicochemical properties

34 positive and 34 negative objects). We extracted features using integer mapping, FT + integer mapping and AAC. Afterwards, we applied the CatBoost algorithm to the new datasets and assessed the predictive performance using Accuracy (ACC), F1-score and Matthews Correlation Coefficient (MCC). Our performance (ACC: 0.8382, F1-score: 0.8070 and MCC: 0.7149) was superior to state-of-the-art tools, such as SecretomeP (BENDTSEN *et al.*, 2005) (ACC: 0.5880, F1-score: 0.4620, and MCC: 0.2000) and PeNGaRoo (ZHANG *et al.*, 2020) (ACC: 0.7790, F1-score: 0.7890, and MCC: 0.5610).

5.4.3 Case Study II - Phage Virion Proteins

In this study, we use the problem of Phage Virion Proteins (PVP), as reported in Manavalan, Shin and Lee (2018). For the experiments carried out, we used benchmark data provided by Charoenkwan *et al.* (2020b), with 500 sequences for training (250 PVP and 250 non-PVP) and 126 for test (63 PVP and 63 non-PVP). To numerically represent the sequences, we built a hybrid

Table 34 – Experimental setting in nine case studies.

Problem	Reference	Case Study	Application	Number of Sequences	Classifier
Non-Classical Secreted Proteins	(ZHANG *et al.*, 2020)	I	Protein	655	CatBoost
Phage Virion Proteins	(CHAROENKWAN *et al.*, 2020b)	II	Protein	626	Support Vector Machines
SARS-CoV-2 Sequences	(HATCHER *et al.*, 2016)	III	DNA	24815	Random Forest
Sigma70 Promoters	(LIN *et al.*, 2017)	IV	DNA	2141	Support Vector Machines
Anticancer Peptides	(LI *et al.*, 2020b)	V	Protein	344	Random Forest
Protein Lysine Crotonylation	(Zhao *et al.*, 2020)	VI	Protein	40587	Random Forest
Long Non-Coding RNAs	(HAN *et al.*, 2018)	VII	RNA	21000 and 12000	CatBoost
Long Non-Coding RNAs	(MENG *et al.*, 2021)	VIII	RNA	36000	Deep Learning
Sigma70 Promoters	(HAQUE *et al.*, 2021)	IX	DNA	2141	Random Forest

feature set with SE ($k = 12$), CN ($k = 1, t = 2$) and AAC. To generate our predictive model, a classifier was induced using an ensemble method (bagging) of Support Vector Machines (SVMs), assessing its predictive performance with F1-score, ACC, Area under the curve (AUC), and MCC. Experimental results showed high performance for F1-score: 0.7934, ACC: 0.8016, AUC: 0.8661 and MCC: 0.6051. The results using the hybrid set of features were superior to the performance obtained using conventional features extracted from the same dataset (CHAROENKWAN *et al.*, 2020b). The use of the hybrid feature set also improved the predictive performance, when compared with the feature set used by PVPred (DING *et al.*, 2014) (ACC: 0.7300, AUC: 0.8570 and MCC: 0.5050), PVP-SVM (MANAVALAN; SHIN; LEE, 2018) (ACC: 0.7460, AUC: 0.8440 and MCC: 0.5050), and PVPred-SCM (CHAROENKWAN *et al.*, 2020a) (ACC: 0.7140, AUC: - and MCC: 0.4320), and slightly worse than Meta-iPVP (CHAROENKWAN *et al.*, 2020b) (ACC: 0.8170, AUC: 0.8700 and MCC: 0.6420).

5.4.4 Case Study III - SARS-CoV-2 Sequences

For this case study, we conducted experiments using a dataset to differentiate SARS-CoV-2 from other viruses (e.g., HIV, Influenza, hepatitis, ebolavirus, SARS). We downloaded all available virus sequences (29135) from the NCBI Viral Genome database (HATCHER *et al.*, 2016) (complete genomic sequences (DNA), e.g., Nucleotide Completeness = "complete" AND host = "homo sapiens"). In a preprocessing phase, we removed sequences smaller than 2000bp and larger than 50000bp (RANDHAWA *et al.*, 2020) to eliminate any bias in the sequence size, since SARS-CoV-2 has an average length of 29838bp, resulting in a dataset with 22442 and 2373 sequences from other viruses and SARS-CoV-2, respectively. In this experiment, we extracted the TE-based features ($k = 12$ and $q = 6$). We applied the Random Forest (RF) algorithm to the dataset represented by TE-based features, using 10-fold cross-validation (mean). It is important to note that we continued with an unbalanced dataset, keeping performance metrics (e.g., F1-score, BACC, but also including Cohen's kappa coefficient). In the experimental results, the predictive performance of the RF model to discriminate SARS-CoV-2 from several other viruses with F1-score, BACC, and kappa of 0.9873, 0.9919, 0.9860, respectively. Moreover, we tested other conventional descriptors (e.g., k-mer, PseKNC, ORF features, Fickett score, and TNC). These descriptors performed between (0.9800-0.9900, BACC), and hence, we realized the classification task between SARS-CoV-2 and other viruses, are linearly separable even using different feature vectors. In addition, these results are supported by Naeem *et al.* (2020), Arslan (2021b).

5.4.5 Case Study IV - Sigma70 Promoters

In this case study, we trained a SVM classifier to induce a sigma70 promoters predictor based on the benchmark dataset from Lin *et al.* (2017). This dataset contains 741 positive samples (promoter) and 1400 negative samples (non-promoter). For the feature extraction, we used the CGR descriptor. The experiments were assessed partitioning the dataset with 5-fold cross-validation (same as in (LIN *et al.*, 2017)), when the following mean performance values were obtained: 0.8594, 0.8346, 0.7872, and 0.6852 for ACC, BACC, F1-score, and MCC, respectively. In Lin *et al.* (2017), the authors report the performance of their tool, iPro70-PseZNC, also using SVM, for 2 of this metrics, ACC: 0.8450 and MCC: 0.6630. Thus, by using the mathematical descriptors, the results improved by 0.0144 (1.44%), for ACC, and 0.0222 (2.22%), for MCC.

5.4.6 Case Study V - Anticancer Peptides

In this case study, our goal is to identify anticancer peptides based on Li *et al.* (2020b). For such, we extracted the features CN ($k = 2$, $t = 1$) and AAC from the benchmark dataset provided by the authors (206 non-anticancer peptides and 138 anticancer peptides). The RF algorithm was applied to the transformed dataset using 10-fold cross-validation. The mean predictive performance of the trained model was assessed using ACC, F1-score, and MCC. The performance of this model was superior to the performance reported in Li *et al.* (2020b), (ACC: 0.9300, F1-score: 0.9061 and MCC: 0.8563 against ACC: 0.9273, F1-score: 0.9270 and MCC: 0.8490).

5.4.7 Case Study VI - Protein Lysine Crotonylation

Based on Zhao *et al.* (2020), we induced and assessed the RF algorithm for the identification of protein lysine crotonylation sites. The benchmark data provided by the author contains 32418 sequences for training (2742 positive and 29676 negative peptides - papaya) and 8169 sequences for test (711 positive and 7458 negative peptides - papaya). For feature extraction, we applied numerical mapping with EIIP. We assess the predictive performance with BACC and MCC, which were 0.6450 and 0.1652, respectively. These results were better than those obtained with the some feature extraction techniques used in Zhao *et al.* (2020), e.g., RF_{AAC} (MCC: 0.1030) and RF_{CKSAAP} (MCC: 0.1110).

5.4.8 Case Study VII - Long Non-Coding RNAs (lncRNA)

In this case study, we trained the CatBoost algorithm to classify lncRNAs sequences from protein-coding genes (mRNAs), using two datasets made available by Han *et al.* (2018): Human (training set: 16000 sequences and test set: 5000 sequences) and Wheat (training set:

8000 sequences and test set: 4000 sequences). From these datasets, we extracted the FT + real mapping, TNC and coding descriptors. Essentially, we follow the same pipeline of previous case studies. Once again, the predictive model induced using our descriptors showed high predictive performance in the datasets, e.g., Human (ACC: 0.9652, F1-score: 0.9646, MCC: 0.9309) and Wheat (ACC: 0.8870, F1-score: 0.8907, MCC: 0.7757). Our results were better than several tools shown in Han *et al.* (2018), e.g., CPC (KONG *et al.*, 2007) (Human - ACC: 0.8304; Wheat - ACC: 0.9595), CNCI (SUN *et al.*, 2013) (Human - ACC: 0.9450; Wheat - ACC: 0.6158), CPAT (WANG *et al.*, 2013) (Human - ACC: 0.9642; Wheat - ACC: 0.8743), PLEK (LI; ZHANG; ZHOU, 2014) (Human - ACC: 0.9274; Wheat - ACC: 0.8773), CPC2 (KANG *et al.*, 2017) (Human - ACC: 0.9614; Wheat - ACC: 0.7870).

5.4.9 Case Study VIII - Using MathFeature with Deep Learning

According to Min, Lee and Yoon (2017), Deep Learning (DL) is a field of ML responsible for several advances, due to its high predictive performance in big data (TANG *et al.*, 2019). Therefore, we assess our descriptors with a DL architecture, using the same case study problem VII (lncRNAs versus mRNAs - feature vector (FT + real mapping and coding descriptors)), but with a benchmark dataset from Meng *et al.* (2021) (*Zea mays* dataset (36000 sequences: 18000 lncRNA and 18000 mRNA), who dedicates his article to a DL approach. Our classifier was generated using Keras (CHOLLET,) (default parameters). Furthermore, we compared our model with three DL tools used in Meng *et al.* (2021) (PlncRNA-HDeep (MENG *et al.*, 2021), lncRNAnet (BAEK *et al.*, 2018) and LncADeep (YANG *et al.*, 2018)), using the same pipeline (hold-out (80% of samples for training and 20% for testing), ACC, Recall, and F1-score). Our model showed high predictive performance in the dataset, e.g., ACC: 0.9605, Recall: 0.9917, and F1-score: 0.9616, overcoming lncRNAnet (ACC: 0.7290, Recall: 0.7200, F1-score: 0.7260), LncADeep (ACC: 0.8000, Recall: 0.6660, F1-score: 0.7690) and PlncRNA-HDeep (Recall: 0.9790), but with a small decimal loss in relation (ACC: 0.0045 and F1-score: 0.0034) to PlncRNA-HDeep (ACC: 0.9650 and F1-score: 0.9650). Therefore, based on our results, MathFeature can also generate robust and efficient feature vectors for DL approaches.

5.4.10 Case Study IX - MathFeature versus other packages

So far, we have evaluated MathFeature with eight experiments in well-established problems. Nevertheless, in this last case study, we also compared MathFeature with five packages, e.g., BioSeq-Analysis (LIU, 2017), Seq2Feature (NIKAM; GROMIHA, 2019), PyFeat (MUHAMMOD *et al.*, 2019), iLearn (CHEN *et al.*, 2019), and SubFeat (HAQUE *et al.*, 2021). The experiments were carried out using the dataset provided by Haque *et al.* (2021), the same dataset used in the case study IV (Sigma70 Promoters). For this study, we considered 741 positive samples (promoter) and 1400 negative samples (non-promoter) and three metrics (ACC, AUC, MCC), evaluating the RF classifier using 10-fold cross-validation (as our reference). We kept our

CGR descriptor. MathFeature (ACC: 0.8576, AUC: 0.9252, and MCC: 0.6797) outperformed all packages, BioSeq-Analysis (ACC: 0.7637, AUC: 0.8297, and MCC: 0.4726), Seq2Feature (ACC: 0.7197, AUC: 0.7637, and MCC: 0.3723), PyFeat (ACC: 0.7842, AUC: 0.8589, and MCC: 0.5064), iLearn (ACC: 0.7597, AUC: 0.8173, and MCC: 0.5275), and SubFeat (ACC: 0.8098, AUC: 0.9232, and MCC: 0.5664). Moreover, based on results obtained comparing MathFeature and Seq2Feature, we generated a hybrid vector with features from both packages (MathFeature: CGR and Seq2Feature: Nucleotide content, random choice), which provided the best result (ACC: 0.8627, AUC: 0.9332, and MCC: 0.6927). Therefore, we achieved high predictive performance, applying only MathFeature or a hybrid combination of packages.

5.5 Discussion

We have assessed the MathFeature package in nine case studies grouped by protein and DNA/RNA sequences. We considered four protein problems and three DNA/RNA problems in the experiments. The classification problems in each case were chosen based on recent articles with distinct domains. For example, for protein molecules, we use the following datasets: (1) non-classical secreted proteins, that according to Zhang *et al.* (2020), are important for understanding pathogenesis mechanisms of Gram-positive bacteria; (2) The PVP identification, e.g., to develop new antibacterial drugs (MANAVALAN; SHIN; LEE, 2018); (3) anticancer peptides that present a new direction in the treatment of cancer (CHEN *et al.*, 2016; LI *et al.*, 2020b); and (4) protein lysine crotonylation, type of post-translational modification (Zhao *et al.*, 2020; WANG *et al.*, 2020). In these studies, we noticed that the hybrid combination of mathematical and conventional descriptors (available at MathFeature) improves the performance of the models, mainly applying CN, FT, numerical mapping (e.g., EIIP and integer), and AAC, varying the ACC/BACC of 0.6450-0.9300 in all problems. For DNA/RNA molecules, the problems used are (1) SARS-CoV-2, hot topic in bioinformatics (NAEEM *et al.*, 2020; ARSLAN, 2021b); (2) detection of sigma70 promoters to study the dynamics of gene expression (LIN *et al.*, 2017; CASSIANO; SILVA-ROCHA, 2020); (3) lncRNA sequences, that can play essential roles in biological processes, e.g., transcriptional regulation (PISIGNANO; LADOMERY, 2021; MENG *et al.*, 2021). For these problems, we obtained highly robust results (varying the ACC/BACC of 0.8594-0.9900), both applying only mathematical descriptors or a hybrid combination, highlighting TE-based features, CGR, FT, TNC, and coding descriptors. Finally, our findings report the relevance of MathFeature descriptors in several applications, e.g., humans, plants, and bacteria data.

5.6 Chapter Remarks

In this study, we have described a new package, named MathFeature, composed of an extensive and comprehensive set of 37 feature descriptors for biological sequences. From these 37 descriptors, 20 are based on mathematical approaches and are not available in other feature extraction packages. Other 17 descriptors, named conventional descriptors, were selected from those often used in the literature. The main motivation for this new package was that, despite the relevance of the features extracted by mathematical descriptors, they are not available in the current packages. Thus, MathFeature extends the existing packages, including mathematical techniques. To experimentally assess the descriptors implemented in this package, we conducted nine case studies, using several biological scenarios, e.g., DNA, RNA, and Proteins (primary sequence of amino acids), applied in different problem domains. Also, we avoid including any type of bias from selected features, and hence, the quality assessment of each feature can be done by the community with regards to the specific problem of interest. In the experiments, we obtained high predictive performance, both applying only mathematical descriptors (e.g., case studies II, III, VI) and applying a hybrid combination of them with well-known conventional descriptors found in the literature (e.g., AAC, TNC, Coding). Finally, through MathFeature, we outperformed several studies in benchmark datasets, indicating that all descriptors within MathFeature can improve the performance of predictive models induced by ML algorithms. Regarding the limitations, we observed that some of these descriptors (e.g., Fourier, Shannon, and Tsallis) have a low performance for short sequences. However, when mathematical are combined with conventional descriptors, in hybrid sets, there is a clear improvement in the predictive performance. Finally, as future work, we intend to investigate descriptors for short sequences, especially in prokaryotic organisms, and also include more protein descriptors. **Some key points:**

- A novel open-source Python package, named MathFeature;
- MathFeature provides 37 descriptors, 20 of them are mathematical, organized into five categories;
- MathFeature can be run on the console, but also provide a GUI (Graphical User Interface)-based platform;
- MathFeature is an extensive and comprehensive set of feature extraction techniques based on mathematical descriptors for encoding DNA, RNA and Proteins (primary sequence of amino acids) sequences;
- MathFeature is the first package to provide a large set of features based on mathematical descriptors and also well-known descriptors from other studies with biological sequences.

CHAPTER

6

BIOAUTOML: AUTOMATED FEATURE ENGINEERING AND METALEARNING FOR THE PREDICTION OF NON-CODING RNAS IN BACTERIA

Considering advances in sequencing, an increasing number of biological sequences have been generated (HASHEMI *et al.*, 2018a; LOU *et al.*, 2019). With the expansion in volume and complexity of biological data, ML algorithms have been successfully applied to their analysis (LIU *et al.*, 2015; GREENER *et al.*, 2021; CHEN *et al.*, 2019). ML algorithms can extract new and useful knowledge from biological data (CHEN *et al.*, 2021), allowing complex analyses, speeding up new findings and reducing research costs (SHARMA *et al.*, 2021). These advances bring important social and economical benefits, such as improving diagnosis, treatment and the design of new medications (SHARMA *et al.*, 2021; CANNATARO; HARRISON, 2021; GHANNAM; TECHTMANN, 2021a), e.g., COVID-19 pandemic (CANNATARO; HARRISON, 2021; RANDHAWA *et al.*, 2020), cancer diagnosis (MAROS *et al.*, 2020), and CRISPR/Cas9-based gene-editing technology (LI; ZHANG; TROYANSKAYA, 2021; MITROFANOV *et al.*, 2020).

Moreover, with the advancement of next generation sequencing technologies and multi-omics analysis (TURNER *et al.*, 2019), studies have focused on discovering and characterizing small non-coding RNAs (sRNAs) in bacteria and archaea, expanding the understanding of gene regulation and elucidating new biological mechanisms (STAV *et al.*, 2019). Moreover, Non-coding RNAs (ncRNAs) have distinct classes with specific functions, depending on their spatial structure, sequence composition, and length (COSTA *et al.*, 2021). Regarding genome annotation, the identification of protein-coding and non-protein-coding sequences is the first and most crucial step (WASHIETL *et al.*, 2011). In addition, sRNAs controls gene expression in prokaryotes, regulating processes, e.g., stress responses, nutrient acquisition, virulence, and biofilm formation (DAR; SOREK, 2018). According to Ahmed, Zheng and Liu (2016), there is a large number of regulatory ncRNAs, highlighting their potential links to bacterial pathogenesis.

Nevertheless, one of the main difficulties for applying ML algorithms to ncRNAs and other sequences are the categorical and unstructured nature of biological sequence data. A frequent alternative to deal with this problem is to apply, in a feature engineering process, feature extraction techniques (e.g., DNA sequences: A, C, T, G), to transform biological sequences into numerical data (e.g., GC content and k-mers) with a structured format. Feature extraction techniques based on various aspects have been proposed to extract numbers from these sequences, including physicochemical, biological and mathematical features (CHEN *et al.*, 2021; GHANNAM; TECHTMANN, 2021a). No matter the aspect, the features must capture the relevant information present in the biological sequence, as the predictive performance of the model induced by a ML algorithm strongly depends on the representativeness of the input feature vector (WARING; LINDVALL; UMETON, 2020). A common approach to increase the representativeness of the features is to select, among the extracted features, the subset that leads to the best predictive performance of a model induced by a ML algorithm. This approach, known as feature selection using wrappers, is also part of the feature engineering process.

The feature engineering process often requires extensive domain knowledge, performed manually by a human expert, and is one of the most time-consuming steps in the ML pipeline (WARING; LINDVALL; UMETON, 2020). Furthermore, according to (STAVRIDIS *et al.*, 2018), ncRNAs are divided into categories based on their cellular functionality and their sequential, thermodynamic, and structural properties, assuming that their sequence can provide robust discriminative features. However, the same sequences can act as more than one type of ncRNAs, e.g., mature microRNA can also be transfer RNA fragments. Consequently, most computational approaches can predict only the presence of ncRNAs. Even those designed to classify more than one type (class) of ncRNAs do not work well with more than 3 types (STAVRIDIS *et al.*, 2018; CHEN; QIAN; YOON, 2018).

These limitations motivated the development of a novel open-source software package, called BioAutoML, that can extract features based on different aspects, and automate the feature selection, algorithm(s) recommendation and algorithm(s) tuning steps for multi-class classification of biological data. BioAutoML is an end-to-end Automated Machine Learning (AutoML) tool for experiments using biological sequences, BioAutoML is able to deal with different categories of ncRNA in bacteria, such as: small RNA (sRNA), transfer RNA (tRNA), ribosomal RNA (rRNA), precursor-microRNA (pre-miRNA), microRNA (miRNA), small nucleolar RNA (snoRNA), small nuclear RNA (snRNA), transfer-messenger RNA (tmRNA) (STAVRIDIS *et al.*, 2018; CONSORTIUM, 2020). According to Sá *et al.* (2017), He, Zhao and Chu (2021), AutoML has a proposal similar to the area of hyper-heuristics, automatically recommending pipelines, algorithms, or hyper-parameters for specific tasks, reducing dependence on on user knowledge. These tasks can include different ways of preprocessing or feature engineering, as well as algorithms and optimization of its parameters (hyper-parameter tuning) (SÁ *et al.*, 2017; HE; ZHAO; CHU, 2021; SANTOS *et al.*, 2019). In this study, BioAutoML calls the MathFeature package (BONIDIA, 2020; BONIDIA *et al.*, 2022) to extract feature descriptors representing relevant numerical information from ncRNA sequences (**Feature Extraction module**). After receiving the feature values, BioAutoML, automatically recommends, using Bayesian Optimization (FRAZIER, 2018), the best pair of selected features and predictive model.

To select the features (**Feature Selection module**), BioAutoML follows the wrapper approach, using a predictive model to assess how good a feature set is. The feature extraction and feature selection model are part of the Feature Engineering process. To recommend the best model, BioAutoML recommends the best number of predictive models, and the ML algorithm to be used for the induction of each model (Algorithm Recommendation module). These two tasks are carried simultaneously, whereby one feeds information to the other in the Bayesian optimization process, until the predictive performance obtained by the pair {selected features and recommended algorithm} is not further improved. It is important to point out that the predictive model used in the wrapper and induced by the recommended algorithm can be an ensemble of predictive models.

This occurs when the Algorithm recommendation model recommends more than one ML algorithm. In this case, each recommended algorithm induces a predictive model, and the induced models are combined in an ensemble. As an ensemble of predictive models is, by itself, a predictive model, for the sake of simplicity, we will name the ensemble also a predictive model. Having finished the **Feature Engineering module** and the **Algorithm Recommendation module**, BioAutoML goes to the fourth module, which uses AutoML to fine tune the hyper-parameters of the recommended ML algorithms (SÁ *et al.*, 2017; HE; ZHAO; CHU, 2021; SANTOS *et al.*, 2019), aiming to improve the predictive performance of the model (**Tuning and Combination module**). If the predictive model is an ensemble, the algorithm that induces each model in the ensemble has its hyper-parameters tuned. Afterwards, the predictive models induced by the tuned algorithms are combined in the ensemble.

To make the role clearer of each module in the whole process, the part of the pipeline with the **Feature Extraction module** and the **Feature Selection module**, as they mainly work at the feature level, is named here **Feature Engineering**. The other part of the pipeline, with the **Algorithm Recommendation module** and the **Tuning and Combination module**, as their work at the ML level, is called henceforth Metalearning. Thus, BioAutoML creates an automated pipeline working at the data feature and algorithm level. In this research, we have investigated several insights to support our hypothesis, as follow:

- **Hypothesis:** Automated feature engineering and metalearning provides an efficient mechanism to extract features based on different aspects, and automates the feature selection, algorithm(s) recommendation and tuning steps, and hence, a high-quality prediction of categories of ncRNAs in bacteria.

The approach will help us to answer our Research Question (RQ), and consequently be used to confirm or deny the hypothesis, described as follows:

- **RQ:** Is it possible to predict different categories of bacterial ncRNAs using automated feature engineering and metalearning pipelines?

Finally, to support our hypothesis and research question, we have validated BioAutoML into three different case studies using several bacterial families.

6.1 Feature Engineering

According to Chou's 5-step rule Liu *et al.* (2015), Chou (2011), numerically representing biological sequences with an efficient and adequate mathematical expression is one of the most relevant steps to establish an effective statistical predictor for a biological system. In ML, biological sequences must be represented by a fixed number of features (e.g., binary, categorical, or continuous), transforming originally unstructured data into a structured format. Feature extraction or feature encoding is a key step in the construction of high-quality ML-based models, determining the effectiveness of trained models in bioinformatics applications, such as biological sequence classification (CHEN *et al.*, 2019; MUHAMMOD *et al.*, 2019; KHATUN *et al.*, 2020). Nevertheless, the feature engineering process is a time-intensive step and requires domain knowledge of experts (WARING; LINDVALL; UMETON, 2020; KHURANA *et al.*, 2016; CHEN *et al.*, 2019), which is a complex exercise (CHEN *et al.*, 2019). Therefore, to develop our proposal and answer our research question, we define the automated feature engineering task formally explained as follows:

- Given a set of sequence data, D, divided into train (D_{train}) and test (D_{test}), a set of feature descriptors, F_d, where $F_d = [f_{d1}, f_{d2}, \ldots, f_{dn}]$, our aim is to select the best numerical representation, that is, the feature vector (V_f), combining different feature descriptors in the training set (D_{train}), using an objective function that considers the most important feature descriptor (I_{fd}) to evaluate the best V_f.

6.2 Metalearning

One of the main difficulties in applying ML algorithms to a new dataset is selecting the most adequate algorithm for this dataset. Each ML algorithm has an inductive bias, which can be defined by the way it searches for a robust model, e.g. starting with simple models and gradually increasing the complexity of the models, until a robust model is found, and the format adopted to represent the models, e.g. a model represented by a decision tree. Although it can be seen as a limitation, the bias is necessary for learning to occur. As consequence, each algorithm fits better datasets with particular conformations. Thus, there is no champion ML algorithm, that performs better than all the others in every situation, but each ML algorithm performs better than the others on some datasets, which are not known beforehand (WOLPERT; MACREADY, 1997). A good alternative to select the best ML algorithm for a new dataset is to use previous knowledge regarding the performance of a set of algorithms in previous learning experiences. This is the idea behind a particular approach for metalearning, defined in (BRAZDIL *et al.*, 2022) as learning to learn. According to the authors, metalearning is a research area that investigates how to recommend the most suitable algorithm, or set of algorithms, for a new task. In this study, we use metalearning to do the following:

- Given a set of selected features, recommend the ML algorithm(s) able to induce the best predictive model, which can be a set of algorithms, each one inducing a model, and combine these models into an ensemble (P_{ml}), recommending the best algorithm. Ensemble methods can boost the performance of simple classifiers (e.g., using multiple prediction models for solving the same problem) and has proven its effectiveness in bioinformatics (LIU *et al.*, 2020; HANCOCK; KHOSHGOFTAAR, 2020; HE *et al.*, 2022).

6.3 Related Works

6.3.1 Feature Engineering

After a carrying out systematic literature review, we found 14 related studies proposing packages that use feature engineering (feature extraction and selection) and ML algorithms for biological sequence classification: PseAAC (SHEN; CHOU, 2008), propy (CAO; XU; LIANG, 2013), PseKNC-General (CHEN *et al.*, 2014b), SPiCE (BERG *et al.*, 2014), Pse-in-One (LIU *et al.*, 2015), repDNA (LIU *et al.*, 2014), Rcpi (CHIU *et al.*, 2015), BioSeq-Analysis (LIU, 2017), PyFeat (MUHAMMOD *et al.*, 2019), iLearn (CHEN *et al.*, 2019), iLearnPlus (CHEN *et al.*, 2021), BioSeq-BLM (LI; PANG; LIU, 2021a), autoBioSeqpy (JING *et al.*, 2020), and AutoGenome (LIU *et al.*, 2021). For each package, we checked if it uses AutoML for feature engineering, ML algorithms and, when they use these algorithms, tune their hyper-parameters. Table 35 summarizes our findings.

The most similar packages to BioAutoML are iLearn, iLearnPlus, autoBioSeqpy, and AutoGenome, which apply AutoML to recommend ML algorithms, but they do not use automated feature engineering. The most similar package to our proposal, iLearn, requires an initial configuration file (choosing descriptors and classifiers), which needs domain knowledge from a human expert. Even in its most sophisticated version, iLearnPlus, a file needs to be inserted with the extracted features, instead of automatic feature engineering. The autoBioSeqpy and

Table 35 – Use of AutoML for feature engineering, recommendation of ML algorithm and hyper-parameter tuning.

Study	**Feature Engineering**	**ML algorithm**	**Tuning**
PseAAC	-	-	
propy	-	-	-
PseKNC-General	-	-	-
SPiCE	-	-	-
Pse-in-One	-	-	-
repDNA	-	-	-
Rcpi	-	-	-
BioSeq-Analysis	-	-	
PyFeat	-	-	-
iLearn	-	V	V
iLearnPlus	-	V	V
BioSeq-BLM	-	-	-
autoBioSeqpy	-	V	V
AutoGenome	-	V	V
BioAutoML	V	V	V

AutoGenome packages focus on recommending the best deep learning architecture. Thus, to the best of our knowledge, BioAutoML automates the longest pipeline for biological sequence analysis, encompassing feature engineering, ML algorithm recommendation and hyper-parameter tuning. Furthermore, BioAutoML is a user-friendly tool for non-experts.

Looking at more general applications of AutoML, we can cite RECIPE (SÁ *et al.*, 2017) and TPOT (LE; FU; MOORE, 2020). Tree-based Pipeline Optimization Tool (TPOT) is an AutoML tool that optimizes ML pipelines using genetic programming. REsilient ClassifIcation Pipeline Evolution (RECIPE) is an AutoML framework with grammar-based genetic programming. One of the most notable methods of RECIPE is how it uses grammar to organize the knowledge acquired from the literature (SÁ *et al.*, 2017). RECIPE can also be an alternative to TPOT, as TPOT can create ML pipelines that are arbitrary, failing to solve a classification problem, therefore leading to a waste of computational resources (SÁ *et al.*, 2017). The major difference compared to BioAutoML is the lack of a feature extraction module for biological sequences. These two packages are for any application domain, requiring a previously selected feature vector.

6.3.2 Prediction Techniques of ncRNAs in Bacteria

Many ML-based techniques have been proposed to identify ncRNAs in bacteria (EPPENHOF; PEÑA-CASTILLO, 2019; ALMEIDA *et al.*, 2021; HE *et al.*, 2018; XIE; ZHANG; XIAO, 2020; BARIK; DAS, 2018; BAR *et al.*, 2021). In Barik and Das (2018), the authors compare the predictive performance of different techniques for RNAs classes, such as tRNAs, rRNAs, and mRNAs. For such, they use normalized minimum free energy of folding, motif frequency, and several RNA-folding parameters, such as base-pairing propensity, Shannon entropy, and base-pair distance. The model induced by the Random Forest algorithm presented 89.5% of predictive accuracy. Another related study, Eppenhof and Peña-Castillo (2019) constructed ML models to discriminate bona fide sRNAs applying five ML algorithms to random genomic sequences from five bacterial species. Seven features were used, including secondary structure. In He *et al.* (2018) the support vector machine (SVM) algorithm was applied to a Non-Coding DNA (ncDNA) benchmark dataset, collected from *Saccharomyces cerevisiae*. SVM was also used in Barman, Mukhopadhyay and Das (2017) to identify sRNAs in bacteria, particularly *Salmonella* Typhimurium LT2, *Escherichia coli (E. coli)* K-12, and *Salmonella* Typhi (*S.* Typhi). Some features are combined to achieve better results with accuracy of 81.25% and 88.82% for *E. coli* K-12 and *S.* Typhi Ty2. Unlike BioAutoML, these approaches did not apply an end-to-end ML pipeline.

6.4 BioAutoML Package

BioAutoML is a user-friendly multi-class and binary classification package that allows the use of automated feature engineering and metalearning, as illustrated by Figure 19. Its use does not require specialized human assistance. BioAutoML only needs a training dataset of biological sequences (FASTA files) to perform an end-to-end ML experiment, from the feature engineering to generating of the predictive model induced by tuned ML algorithms. Nevertheless, the modules implemented in the BioAutoML package can be run independently, i.e. users can just generate the best numerical representation and send it to another ML model generation package, or they can use features extracted from other packages to generate a predictive model. For such, BioAutoML has two components with two modules each (1) automated feature engineering (feature extraction and selection) and (2) Metalearning (algorithm recommendation and hyperparameters tuning). In the next sections, we briefly describe each component and module.

6.4.1 Feature Extraction

This module, which is the first feature engineering stage, extracts feature descriptors using the MathFeature package (BONIDIA *et al.*, 2022), e.g., Mathematical descriptors (Fourier, Shannon, Tsallis, among others) and Conventional descriptors (Nucleic Acid Composition (NAC), dinucleotide composition (DNC), trinucleotide composition (TNC), ORF Features, Xmer k-Spaced Ymer composition frequency (kGap), Fickett score, among others). As a result, more than 15 feature extraction techniques can numerically represent information found in biological sequences.

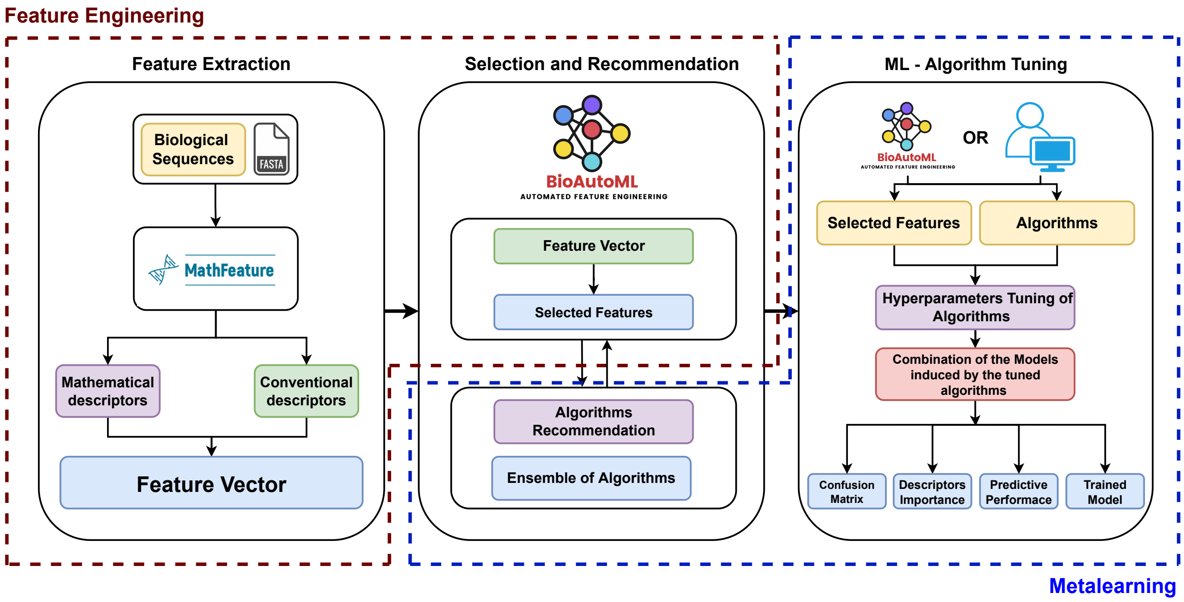


Figure 19 – Components implemented in the BioAutoML package: (1) Automated Feature Engineering (feature extraction and selection) and (2) Metalearning (algorithm recommendation and hyper-parameters tuning).

6.4.2 BioAutoML - Selection and Recommendation

The second module carries out automated feature engineering, selecting the best feature vector and ML algorithm to induce predictive models, which can be an ensemble of predictive models, as shown in Figure 20. For such, it uses the Bayesian optimization technique (FRAZIER, 2018). We use this technique because there is a large number of alternatives for the types and number of feature descriptors, characterizing an NP-hard problem. This module receives the following as input:

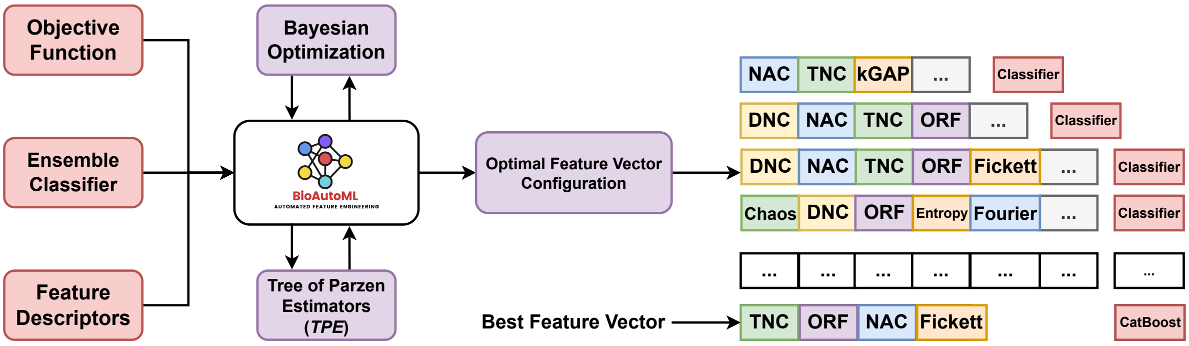

Objective Function
Ensemble Classifier
Feature Descriptors
Bayesian Optimization
BioAutoML
AUTOMATED FEATURE ENGINEERING
Tree of Parzen Estimators (TPE)
Optimal Feature Vector Configuration
NAC TNC kGAP ... Classifier
DNC NAC TNC ORF ... Classifier
DNC NAC TNC ORF Fickett ... Classifier
Chaos DNC ORF Entropy Fourier ... Classifier
... ...
Best Feature Vector
TNC ORF NAC Fickett CatBoost

Figure 20 – Illustration of how BioAutoML works: Selection and recommendation module.

- (1) All feature descriptors extracted by the first module;
- (2) An objective function, e.g., in our case, balanced accuracy for binary problems and F1-score (weighted) for multi-class problems;

- (3) ML algorithms (CatBoost (PROKHORENKOVA *et al.*, 2018), AdaBoost (SCHAPIRE, 2013), Random Forest (LIAW; WIENER *et al.*, 2002), and LightGBM (KE *et al.*, 2017)). These classifiers are responsible for analyzing the potential of the selected features. These algorithms are used for the wrapper-based feature selection, using different feature subsets as input. We chose these ML algorithms because they have good predictive performance and induce interpretable predictive models, allowing the understanding of the internal decision-making process (BONIDIA *et al.*, 2020a). The algorithms are widely adopted in the bioinformatics literature (LIU *et al.*, 2020; HANCOCK; KHOSHGOFTAAR, 2020; HE *et al.*, 2022).

To represent the search space (for selecting feature descriptors and recommending ML algorithms), we use a partially binary input vector, e.g. *[1, 0, 1, 0, 0, 1, [2]]*, when the last position can be a value from the set *0, 1, 2, 3*, representing each of the four ML algorithms. In the other position, value 0 means that the feature descriptor was not selected for the subset to be evaluated, and value 1 that was selected. Next, using Bayesian optimization (Hyperopt library - Tree of Parzen Estimators (BERGSTRA *et al.*, 2013)), BioAutoML selects a quasi-optimal feature vector, regarding the predictive performance of the model used in the wrapper. We chose Bayesian optimization based on studies in the literature (FRAZIER, 2018; VICTORIA; MARAGATHAM, 2021; ELSAYAD; NASSEF; AL-DHAIFALLAH, 2022), which demonstrate that it saves time and improves performance, presenting benefits over random search (TURNER *et al.*, 2021). As can be seen in Figure 20, BioAutoML generates combinations of features and ML algorithm(s) until it finds the best pair (selected feature set, recommended ML algorithm) to send to the fourth module, hyper-parameter tuning. We adopted as stopping criterion for the optimization when the predictive performance reaches a plateau or after assessing 50 pairs (this number can be changed by the user). Let us remember that this module can recommend one ML algorithm or a set of ML algorithms (when an ensemble model induced by ML algorithms is recommended).

6.4.3 ML Algorithm(s) Hyper-parameter Tuning

The last module is tuning, where users can generate a predictive model using the recommendation of the feature vector and ML algorithms (among those whose implementation is available at BioAutoML). These algorithms will use the feature vector to induce a set of classification models, whose output will be combined using an ensemble-based approach. The quality of the classification models will be affected by the hyper-parameter values used for the recommended ML algorithms. In this work, Bayesian optimization is used to tune their hyper-parameters. For such, we separate part of the training set. The hyper-parameters tuned for each algorithm are defined by their official documentation, e.g., Random Forest (n_estimators, max_features, criterion, max_depth, min_samples_split, min_samples_leaf, and bootstrap). The optimization stops when the predictive performance reaches a plateau or after assessing 100 possible sets of values. In addition, this module generates important performance analysis files as outputs, e.g., best features, performance results, trained model, and feature importance, among others.

6.5 Experimental Results

The main purpose of this article was to provide a user-friendly and open access package that allows automated feature engineering and metalearning for the analysis of biological sequences. To assess the relevance of the proposed package, we evaluate its predictive performance in three case studies, described in the following sections: (1) Case Study I - Genomic Pipeline, (2) Case Study II - Pipeline with Annotated Bacterial Sequences, and (3) Case Study III - BioAutoML versus other proposed packages for automated experiments.

6.5.1 Case Study I - Genomic Pipeline

We designed an experiment to classify ncRNA families in bacteria, using three known types of bacterial RNA: sRNA, tRNA, and rRNA. These RNAs are often considered and studied to analyze ncRNA sequences, e.g., (1) tRNAs and rRNA can contaminate sRNA samples isolated from cytoplasmic total RNA extracts (LOONG; MISHRA, 2007), and (2) sRNAs in bacteria, key actors in transcriptional and post-transcriptional regulation (BARIK; DAS, 2018), emphasizing the importance of accurate prediction of these sequences. To further demonstrate the usefulness of our package, we generated our dataset using a standard bioinformatics pipeline, as shown in Figure 21, extracting sequences from genomes and then applying ML algorithms to predictive models. Our aim is to demonstrate that non-experts can easily connect their pipeline to BioAutoML.

To collect the RNAs from genomes, we used the Infernal application (NAWROCKI; EDDY, 2013). First, in our genomic pipeline, we accessed the Rfam Public MySQL Database obtaining a list of families for each RNA type (KALVARI *et al.*, 2021), using the Rfam database in its 14.7 version. Next, with the lists and the complete Rfam Covariance Model (CM), we generated three CM files using cmfetch, i.e., one for each RNA type. We use cmsearch considering the gathering cutoff (GA) selected by the Rfam curators to extract the sequences for the RNA types (KALVARI *et al.*, 2018), given the CM files and a genome. Once the sequences are extracted, they are passed as input to BioAutoML. Thereby, we selected *Escherichia coli* K-12 genome for an initial experiment with the genomic pipeline. In Table 36, we show the sequences generated for training and testing (Hold-out 80% training and 20% test).

BioAutoML returned a combination of three feature descriptors that considered to better numerically represent this dataset, kGap, Fourier, and Tsallis entropy. After automated feature engineering, our package select a feature vector and recommended an ML algorithm to be finely tuned. The final results are shown in Table 37 and Figure 22-A. As our problem is multi-class,

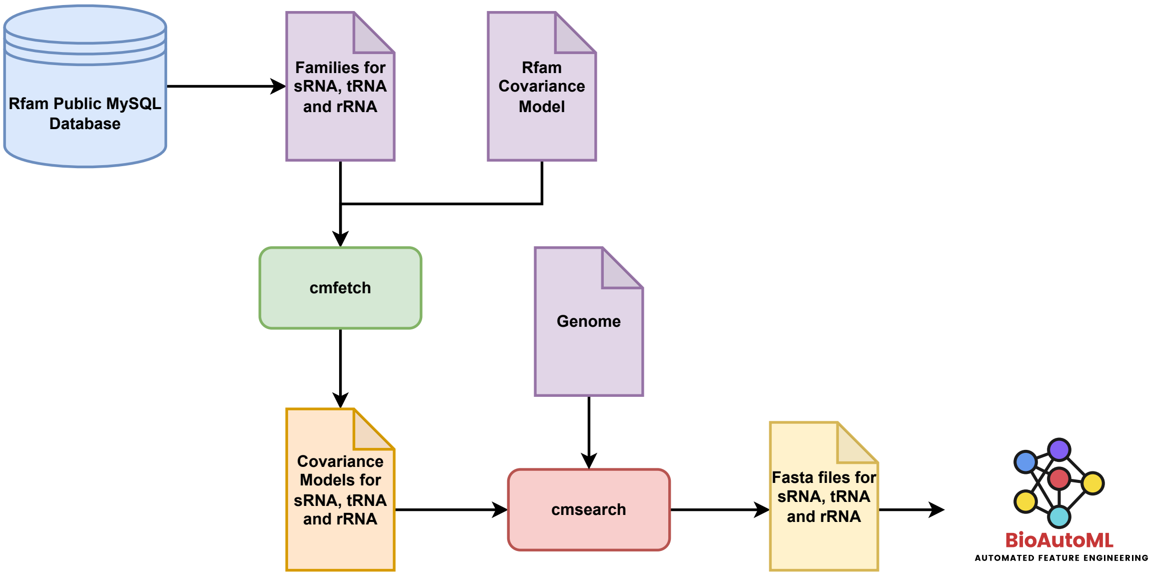

Rfam Public MySQL Database
Families for sRNA, tRNA and rRNA
Rfam Covariance Model
cmfetch
Genome
Covariance Models for sRNA, tRNA and rRNA
cmsearch
Fasta files for sRNA, tRNA and rRNA
BioAutoML
AUTOMATED FEATURE ENGINEERING

Figure 21 – Case Study I - Genomic Pipeline

Table 36 – Number of sequences from *E. coli* K-12 used for training and test.

RNA type	Samples	Training	Test
sRNA	166	133	33
tRNA	50	40	10
rRNA	40	32	8

we report the main results using precision (Macro and Weighted), recall (Macro and Weighted), F1-score (Macro and Weighted), and confusion matrix (GRANDINI; BAGLI; VISANI, 2020).

Table 37 – Results: *E. coli* K-12 - case study I.

	Precision	Recall	F1-Score
sRNA	1.00	0.97	0.98
tRNA	1.00	1.00	1.00
rRNA	0.89	1.00	0.94
Macro Average	0.96	0.99	0.98
Weighted Average	0.98	0.98	0.98

BioAutoML was performed between 0.96-0.98 (macro and weighted average) in this initial experiment, showing a robust numerical representation for the input genome. Next, we used the recommended algorithm to induce a predictive model to classify new unknown sequences. To test the potential of our package, we did a more complex experiment using bacterial phyla, as shown in Table 38. We analyzed the generalization potential for the classification of new bacterial genomes as new organisms will not be in the training set, e.g., training with *Chloracidobacterium* and classifying a new genome as *Terriglobus roseus*. Moreover, according to Lu and Salzberg (2020), the different GC content skew patterns throughout bacterial phylogenetic groups could change relevant characteristics of the sequences' primary structure used for the generation of descriptors. The bacterial phyla used for this experiment are shown in Table 38 and Figure 22-B.

Table 38 – Number of sequences for bacterial phyla by RNA type.

Phylum	Bacteria	sRNA	tRNA	rRNA	NCBI Taxonomy ID	Used as	
Acidobacteria	*Chloracidobacterium*	21	50	7	Negative	2821542	Train
	Terriglobus roseus	4	47	11	Negative	926566	Test
Actinobacteria	*Corynebacterium diphtheriae*	-	43	14	Positive	1450520	Train
	Mycobacterium tuberculosis	11	45	8	Positive	83332	Test
Bacteroidota	*Flavobacterium sediminis*	-	41	19	Negative	2201181	Train
	Mucilaginibacter gossypii	-	46	20	Negative	551996	Test
Cyanobacteria	*Oscillatoria acuminata*	9	65	19	Negative	56110	Train
	Prochlorococcus marinus	2	39	7	Negative	167539	Test
Firmicutes	*Staphylococcus aureus*	84	35	31	Positive	93061	Train
	Staphylococcus epidermidis	44	38	11	Positive	1282	Test
Proteobacteria	*Escherichia coli*	166	50	40	Negative	83333	Train
	Salmonella enterica	118	52	39	Negative	99287	Test
Verrucomicrobia	*Akkermansia glycaniphila*	-	44	7	Negative	1679444	Test
	Luteolibacter ambystomatis	29	46	14	Negative	2824561	Train

We randomly selected one bacteria from each phylum for a fair split, as shown in Table 38. We used seven bacteria for training and testing. The number of sequences generated by RNA type is also presented. The sequences were extracted using the same pipeline exposed in Figure 21. The performance metrics can be seen in Table 39.

Table 39 – Results: Bacterial phyla - case study I.

	Precision	**Recall**	**F1-Score**
sRNA	0.97	0.97	0.97
tRNA	0.98	1.00	0.99
rRNA	0.99	0.95	0.97
Macro Average	0.98	0.97	0.98
Weighted Average	0.98	0.98	0.98

Our package recommended six feature descriptors that best represent this new scenario, NAC, TNC, kGap, Fourier, and ORF. Two of these descriptors were in the initial experiment (kGap, and Fourier). Again, BioAutoML showed good predictive results, between 0.97-0.98 (macro and weighted average). However, we were classifying new bacterial sequences that were not in training, indicating the package's ability to recommend robust feature descriptors for the input problem.

6.5.2 Case Study II - Pipeline with Annotated Bacterial Sequences

For this case, we extracted annotated bacterial sequences from databases, standard pipeline in several studies (LIU *et al.*, 2020; JING *et al.*, 2020; LIU *et al.*, 2021). We used eight classes for this analysis: pre-miRNA, miRNA, snoRNA, snRNA, tmRNA, tRNA, rRNA and mRNA. Compared to case study I, we worked with specific types of small RNAs to study

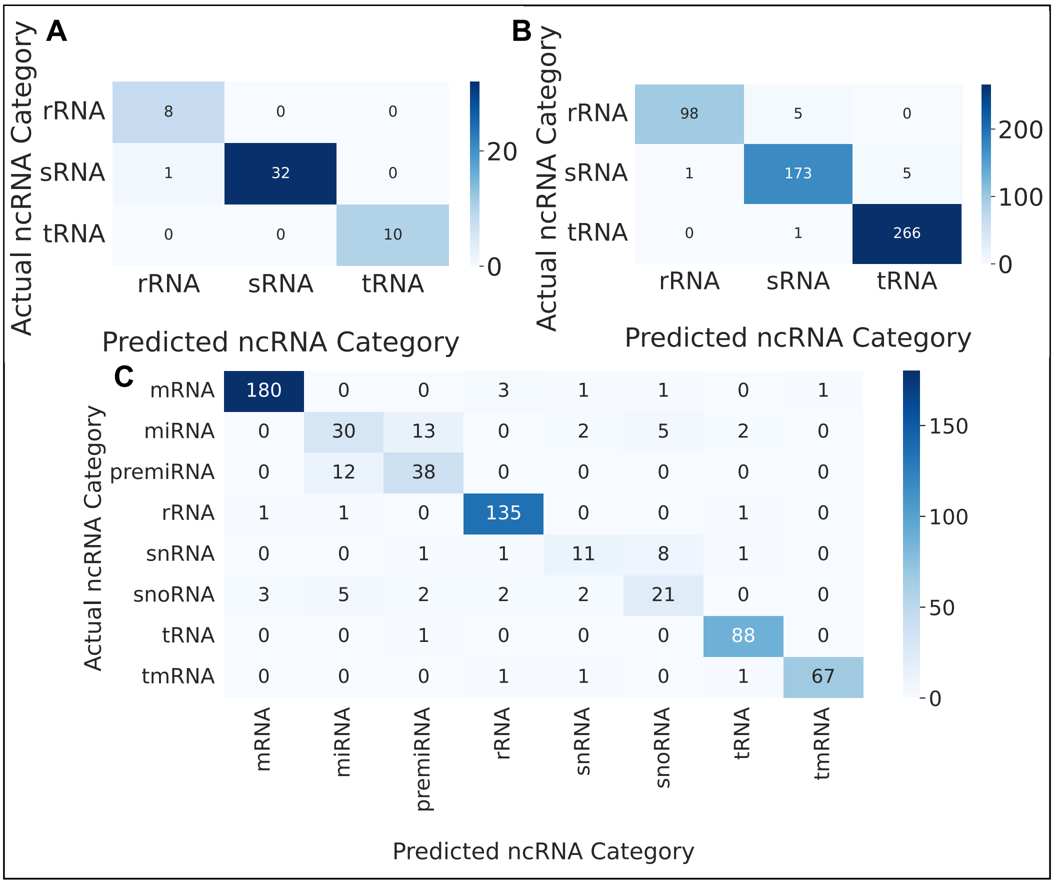


Figure 22 – Confusion matrix of the experiments. (**A**) Case Study I - *E. Coli* K-12. (**B**) Case Study I - Bacterial phyla. (**C**) Case Study II - Annotated Bacterial Sequences.

BioAutoML capacity for dealing with more classes. In addition, we used mRNA as a counterpoint by containing coding regions compared to the ncRNAs. These classes can be separated into regulatory, and housekeeping ncRNAs (ZHANG *et al.*, 2019). We also demonstrate the performance metrics for the application considering recurrent problems such as the classification of pre-miRNA between miRNA (STAVRIDIS *et al.*, 2018; TASDELEN; SEN, 2021; FU *et al.*, 2019), and the prediction of miRNA by itself (WANG; ZHANG; ZHAO, 2017). There are few studies related to the prediction of miRNAs in bacteria (DANG *et al.*, 2019) as the number of these annotated sequences is still small (CARDIN; BORCHERT, 2017).

We collected ncRNA sequences from RNAcentral, a database of non-coding RNA sequences that provides a single access point to at least 44 RNA resources in its last version (CONSORTIUM, 2020). We accessed the RNAcentral Public Postgres database running SQL queries to filter active cross-reference sequences by type, limited to 1,000 sequences, and restricting them to bacterial organisms. With the results from the queries, FASTA files for each class were created. Considering how collecting from diverse databases could bring some redundancy, we used CD-HIT Est (LI; GODZIK, 2006) to cluster the sequences, removing redundancy at 95% similarity. The same preprocessing pipeline was applied for mRNA, but we collected the sequences from GenBank (SAYERS *et al.*, 2019), filtering for bacterial organisms.

In Table 40, we show the numbers of examples collected from RNAcentral and GenBank, the numbers after applying CD-HIT Est with the preprocessing method used in BioAutoML, and how many of these sequences are used for training and testing (Hold-out 80% training and 20% test). The results generated by BioAutoML are presented in Table 41 and Figure 22-C.

Table 40 – Number of sequences used for training and test - case study II.

RNA type	**Samples**	**Preprocessing**	**Training**	**Test**
pre-miRNA	327	253	203	50
miRNA	464	263	211	52
snoRNA	331	178	143	35
snRNA	176	113	91	22
tmRNA	1,000	350	280	70
tRNA	1,000	445	356	89
rRNA	1,000	687	549	138
mRNA	1,000	702	514	188

Table 41 – Results generated by BioAutoML in the case study II.

	Precision	**Recall**	**F1-Score**
pre-miRNA	0.69	0.76	0.72
miRNA	0.63	0.58	0.60
snoRNA	0.60	0.60	0.60
snRNA	0.65	0.50	0.56
tmRNA	0.99	0.96	0.97
tRNA	0.95	0.99	0.97
rRNA	0.95	0.98	0.96
mRNA	0.98	0.97	0.97
Macro Average	0.80	0.79	0.80
Weighted Average	0.89	0.89	0.89

Again, our package presented robust performance, even for eight classes, ranging from 0.79-0.89 (macro and weighted average). By analyzing each class individually, we observed a better performance for rRNA (F1-Score: 0.96), tRNA (F1-Score: 0.97), tmRNA (F1-Score: 0.97), pre-miRNA (F1-Score: 0.72), and mRNA (F1-Score: 0.97), but lower performance for miRNA, snoRNA and snRNA (F1-Score: both around 0.60). However, multi-class classification problems present more challenges than binary classification problems, e.g., an imbalanced dataset. Even so, BioAutoML recommended a good feature vector formed by the descriptors NAC, DNC, TNC, kGAP, ORF feature, Fourier, and Tsallis entropy.

6.5.3 Case Study III - Comparing BioAutoML with other AutoML packages

In this last case study, we compared BioAutoML with well-known AutoML packages used in different classification tasks (BALAJI; ALLEN, 2018; ZÖLLER; HUBER, 2021). In our literature review, we did not find any tool for biological sequence classification with automatic feature engineering, characterizing the innovative nature of BioAutoML. To allow the experimental comparison, we chose two packages using AutoML: RECIPE (SÁ *et al.*, 2017) and TPOT (LE; FU; MOORE, 2020). The major difference compared to BioAutoML is the lack of a feature extraction module for biological sequences.

Similar to BioAutoML, they include feature selection, algorithm recommendation and hyper-parameter tuning. These two packages are for any application domain, requiring a previously selected feature vector. For a fair comparison, we used the same output from the feature extraction module in the AutoML packages (all feature descriptors), feature descriptors recommended by BioAutoML, and datasets from the previous case studies. These experiments assess whether BioAutoML can build predictive models with recommended feature vectors and ML algorithms as robust as RECIPE and TPOT, which are well known for the quality of their pipelines (BALAJI; ALLEN, 2018; ZÖLLER; HUBER, 2021). The BioAutoML results shown in Table 42 are the average of 10 runs. All experiments, package configurations (default parameters) and datasets can be consulted in our repository[1]. We performed the experiments using a machine with Intel Core i3-9100F CPU (3.60GHz), 16GB memory, and running in Debian GNU/Linux 10.

Table 42 – Case study III - BioAutoML versus other AutoML packages

Dataset	Version	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)	Time (minutes)
CS-I-phyla	**BioAutoML**	0.97	0.97	0.97	**16.34**
Recommended Feature Descriptors	RECIPE	0.97	0.97	0.97	32.48
Recommended Feature Descriptors	**TPOT**	0.99	0.99	0.99	72.41
All Feature Descriptors	RECIPE	0.96	0.96	0.96	30.46
All Feature Descriptors	**TPOT**	0.98	0.98	0.98	46.39
CS-II	**BioAutoML**	0.88	0.88	0.88	**85.02**
Recommended Feature Descriptors	RECIPE	0.87	0.61	0.68	272.46
Recommended Feature Descriptors	**TPOT**	0.89	0.89	0.89	416.28
All Feature Descriptors	RECIPE	0.77	0.36	0.38	151.12
All Feature Descriptors	**TPOT**	0.90	0.89	0.89	338.55

As can be seen, we observed similar performance between BioAutoML and other tools (TPOT and RECIPE) in CS-I and CS-II, considering two different types of experiments: (i) with all feature descriptors, and (ii) with the vector recommended by BioAutoML. We also noted the improvement prediction of TPOT when the input was provided by vector recommended from BioAutoML (gain of 2% and 1% for CS-I and CS-II, respectively). Another interesting result is related to the computational time to generate an ML model when both TPOT and RECIPE spent a huge computational effort (416.26 and 272.46 minutes, respectively in CS-II) while BioAutoML spent 85.02 minutes. BioAutoML also recommends the best vector be extracted automatically. It's important to highlight that both TPOT and RECIPE do not have any mechanism to recommend

[1] https://github.com/Bonidia/BioAutoML - Case Studies

the best vector to be automatically extracted for biological sequences. Finally, the statistical significance was applied in this case study (difference in F1-Score (Weighted)), using Friedman's test, indicating that there is no statistical significance in performance ($P - value = 0.156$, using $\alpha = 0.05$), suggesting that our proposal is as robust as known methods in the literature.

6.6 Discussion

We assessed BioAutoML in three case studies with ncRNA sequences. We consider different ncRNA categories for multi-class classification tasks using ncRNA bacteria data. For case study I, we used Infernal, which builds statistical models of RNA secondary structure and sequence consensus called Covariance Models (CMs) (NAWROCKI; EDDY, 2013). Infernal is still widely used for genome annotation, especially for detecting non-coding RNA (LI *et al.*, 2021; CHAN *et al.*, 2021). However, one of its limitations for creating CM is the need for a secondary structure model for the RNA families. The experimental results from this case study show the success of BioAutoML in using only primary structure features to predict what we found with Infernal.

For case study II, we considered eight classes, including miRNAs. Although some studies in the literature consider that prokaryotes do not have true miRNA as in eukaryotes (CLARK; PAZDERNIK; MCGEHEE, 2019; WATKINS; ARYA, 2019), recently, many similarities between the non-coding sequences were observed, indicating miRNA-like mechanisms in prokaryotes, which resulted in the annotated sequences used in our study (WATKINS; ARYA, 2019; SOLTANI-FARD *et al.*, 2021). Prokaryotic miRNAs can also accumulate in the nucleolus as pre-miRNAs, and mature miRNAs (SOLTANI-FARD *et al.*, 2021), emphasizing the challenge for an accurate classification in these two classes. Other classes used, such as snoRNA (STREIT *et al.*, 2020) and snRNA (LINDSAY *et al.*, 2013), are also relatively rare in prokaryote organisms. Nevertheless, it is relevant to discover more of these non-coding sequences in bacteria with the advancements in RNA sequencing technology and ML-based algorithms.

Finally, in case study III, we observed the robust predictive performance of BioAutoML when compared with AutoML tools found in the literature, mainly due to the quality of their pipelines (BALAJI; ALLEN, 2018; ZÖLLER; HUBER, 2021). The experimental results indicated the efficiency of the feature extraction module, which can extract features based on different aspects, automated feature selection, algorithm(s) recommendation, and tuning steps. Together, they predicted the categories of ncRNAs in bacteria with high predictive accuracy, even when the number of classes was increased.

6.7 Chapter Remarks

In this article, we propose and experimentally evaluate a new package, BioAutoML, to classify biological sequences. BioAutoML uses AutoML to select the best feature vector from a set of descriptors extracted by the MathFeature package, to recommend the best ML algorithms, and tune the hyper-parameters of the recommended algorithm. For such, it initially performs automated feature engineering and metalearning for non-coding sequences in bacteria, which has the potential to accelerate new studies in bioinformatics. We develop a package that does not require specialized human assistance, supporting research on challenging problems in biological sequence analysis. Our findings support our hypothesis, showing the benefits of using automated feature engineering and metalearning. Although in this study, BioAutoML is applied only to ncRNA sequences in bacteria, it can be used in other DNA/RNA sequence scenarios. We focused exclusively on bacteria, due to the biotechnological potential existing in the investigated strains. Nevertheless, the first module of BioAutoML is an important task for providing feature descriptors for different types of sequences (nucleotides or proteins, i.e., prediction of structural features along the primary sequence of amino acids). We also used our previous framework, MathFeature, to extract features for BioAutoML. BioAutoML can be used for binary and multi-class classification problems, allowing its integration with many existing packages. Finally, in future work, we intend to expand the BioAutoML to proteins and add new feature extraction packages, e.g., iLearn, BioSeq-Analysis, and BioSeq-BLM, testing other feature selection methods such as combining Bayesian Optimization and Lipschitz Optimization, Genetic Algorithm and Genetic programming. **Key Points:**

- The first study to propose an automated feature engineering and metalearning pipeline for ncRNA sequences in bacteria;
- BioAutoML can be used in multi-class and binary problems;
- BioAutoML can be employed in other DNA/RNA sequences scenarios;
- BioAutoML can accelerate new bioinformatics studies, reducing the feature engineering time-consuming stage and improving the design and performance of ML pipelines;
- BioAutoML reduces the requirement of human expert assistance.

CHAPTER

7

EMPOWERING SCIENTIFIC DISCOVERY

In this section, we have curated a collection of articles, whether published or not, each exclusively comprised of abstracts that demonstrate the utilization or influence of our research. These abstracts collectively serve as a testament to the impact of our study. As you review these summaries, our aim is for you to recognize the diverse ways our research can be employed in the field of biological sequences.

7.1 MathPIP: Classification of Proinflammatory Peptides

Proinflammatory peptide (PIP) is a relevant part of the inflammatory response, often the first response of our immune system to strange bodies, i.e., inflammatory-inducing infection, such as COVID-19. Thus, it is essential to have reliable ways to classify and analyze new instances of PIPs. Machine learning (ML) models have been widely employed for the classification of biological sequences, being the basis for most studies in extensive databases of biological information. Most ML algorithms have difficulty directly dealing with these sequences. Thereby, relevant features are extracted from these sequences, making feature extraction one of the key steps in the application of ML algorithms to biological data. Different features have been proposed, many of them based on prior knowledge, such as molecular structures. However, many biological sequences publicly available do not come with prior knowledge. To deal with this limitation, we propose to investigate the use of mathematical descriptors to extract features from PIP sequences. To assess how relevant the features extracted using mathematical descriptors, we run experiments where we apply three ML algorithms. In these experiments, we obtained a predictive accuracy of 0.7034, which is on par with current PIP classifiers.

CAVALCANTE, João Pedro Uchôa; GONÇALVES, Anderson Cardoso; BONIDIA, Robson Parmezan; SANCHES, Danilo Sipoli; DE CARVALHO, André Carlos Ponce de Leon Ferreira. **MathPIP: classification of proinflammatory peptides using mathematical descriptors.** In:

Advances in Bioinformatics and Computational Biology: 14th Brazilian Symposium on Bioinformatics, BSB 2021, Virtual Event, November 22–26, 2021, Proceedings 14. Springer International Publishing, 2021. p. 131-136.

7.2 Feature Importance Analysis of Non-Coding Sequences

Non-coding sequences have gained increasing space in scientific areas related to bioinformatics, due to essential roles played in different biological processes. Elucidating the function of these non-coding regions is a relevant challenge, which has been addressed by several Machine Learning (ML) studies in various fields of ncRNA, e.g., small non-coding RNAs (sRNAs) and Circular RNAs (circRNAs). The identification of these biological sequences is possible through feature engineering techniques, which can help point out specifics in different types of problems with ML. Thereby, there are recent studies focusing on interpretable computational methods, i.e., the best features based on feature importance analysis. For that reason, in this study we have proposed to explore different feature descriptors and the degree of importance involved in classification tasks, using two case studies: (1) prediction of sRNAs in Bacteria and (2) prediction of circRNA in Humans. We developed a general pipeline using hybrid feature vectors with mathematical and conventional descriptors. In addition, these vectors were generated with MathFeature package and feature selection techniques in both case studies. Finally, our experiment results reported high predictive performance and the relevance of combining conventional and mathematical descriptors in different organisms.

DE ALMEIDA, Breno Lívio Silva; QUEIROZ, Alvaro Pedroso; SANTOS, Anderson Paulo Avila; BONIDIA, Robson Parmezan; DA ROCHA, Ulisses Nunes; SANCHES, Danilo Sipoli; DE CARVALHO, André Carlos Ponce de Leon Ferreira. **Feature importance analysis of non-coding dna/rna sequences based on machine learning approaches**. In: Advances in Bioinformatics and Computational Biology: 14th Brazilian Symposium on Bioinformatics, BSB 2021, Virtual Event, November 22–26, 2021, Proceedings 14. Springer International Publishing, 2021. p. 81-92.

7.3 Fatectídeos: Prediction of Antiviral Peptides

Since the beginning of time, humanity has been grappling with various types of biologically significant viral threats, including tropical pathogens, some highly lethal or pandemic in nature, such as Ebola and H1N1. Recently, we experienced one of the largest global pandemics in history, caused by the Sars-CoV-2 virus, also known as the Novel Coronavirus, posing a challenging problem. Despite advancements in health research, viral infections continue to yield a high mortality rate. Research and projects for the development of new antiviral medications are ongoing. Consequently, peptide-based drugs have become increasingly important to explore, aiding in vaccine development. Antiviral peptides (AVPs) are a subset of antimicrobial peptides (AMPs) that act as the first line of defense in the innate immune response in many organisms, and they are defense peptides produced in response to pathogenic diseases that have a significant impact on humanity. AVPs can be obtained through different methods: computational methods, natural sources, and biological sources. One of the relatively unexplored methods for predicting antiviral peptides is the development of tools using machine learning. However, despite these contributions, there are still opportunities for improvement in various prediction tools, with a focus on prediction. One of the major challenges is dealing with unstructured biological sequence data, as many algorithms only handle numerical data. Thus, it is necessary to translate biological sequences into numerical vectors. Based on this, we propose a tool that will perform the classification of antiviral peptides using the BioAutoML tool as a foundation, which extracts information from biological sequences for conversion into numerical data. This will simplify this step for biologists and researchers developing various methods using AVPs, impacting not only Brazilian science but also society and health.

MEDEIROS, Beatriz Leite; INÁCIO, Gabriele de Campos; BONIDIA, Robson Parmezan. **Fatectídeos: Prediction of Antiviral Peptides.** In: Iniciação Científica - Fatec Ourinhos - Ciência de Dados.

7.4 CRISPRloci: CRISPR–Cas systems

CRISPR–Cas systems are adaptive immune systems in prokaryotes, providing resistance against invading viruses and plasmids. The identification of CRISPR loci is currently a non-standardized, ambiguous process, requiring the manual combination of multiple tools, where existing tools detect only parts of the CRISPR-systems, and lack quality control, annotation, and assessment capabilities of the detected CRISPR loci. Our CRISPRloci server provides the first resource for the prediction and assessment of all possible CRISPR loci. The server integrates a series of advanced Machine Learning tools within a seamless web interface featuring: (i) prediction of all CRISPR arrays in the correct orientation; (ii) definition of CRISPR leaders for each locus; and (iii) annotation of cas genes and their unambiguous classification. As a result, CRISPRloci is able to accurately determine the CRISPR array and associated information, such as: the Cas subtypes; cassette boundaries; accuracy of the repeat structure, orientation, and leader sequence; virus-host interactions; self-targeting; as well as the annotation of cas genes, all of which have been missing from existing tools. This annotation is presented in an interactive interface, making it easy for scientists to gain an overview of the CRISPR system in their organism of interest. Predictions are also rendered in GFF format, enabling in-depth genome browser inspection. In summary, CRISPRloci constitutes a full suite for CRISPR–Cas system characterization that offers annotation quality previously available only after manual inspection.

ALKHNBASHI, Omer S; MITROFANOV, Alexander; BONIDIA, Robson et al. **CRISPRloci: comprehensive and accurate annotation of CRISPR–Cas systems**. Nucleic Acids Research, v. 49, n. W1, p. W125-W130, 2021.

7.5 Stability of Gut Microbial Communities

Beta-diversity dispersion based on Bray-Curtis distances of a set of samples calculated using 16S amplicon sequencing data can indicate the stability in microbial communities. We hypothesize that a genome-centric analysis of the biodiversity of viruses and prokaryotes can predict the stability in the gut microbiome of children with and without atopic eczema (AE). Bray-Curtis distances to centroids of two 10-year-old children groups (AE: 17 children with AE; non-AE: 13 healthy children) were calculated. We also sequenced metagenomes from the gut DNA of the two groups to test our hypothesis. After, we recovered metagenome-assembled genomes (MAGs) and uncultivated virus genomes (UViGs) using MuDoGeR. We generated 33 new samples to balance our dataset using Synthetic Minority Oversampling Technique (SMOTE). We predicted the stability of the community using a random forest regression model (RF) based on Bray-Curtis distances to centroids (target) and MAG and UViG coverages (features). We evaluated our RF model's accuracy using the root mean squared errors (RMSE) and R^2 from a linear curve of the observed and predicted values. The stability of the gut microbial community in non-AE was higher than that observed for the AE children, based on the average higher interquartile ranges of Bray-Curtis distance of each group's samples to their centroids (t-test, $p < 0.05$). 2255 MAGs and 2024 UViGs were recovered from our metagenome dataset. The MAGs were affiliated with 11 phyla. UViG Taxonomic analysis indicated, 1633 UViGs affiliated with the Phylum Uroviricota; however, 391 UViGs were not affiliated with any known taxa. The 0.075 RMSE and 0.799 R^2 of our RF showed that our bioindicators were good predictors of stability in the gut microbial communities. We identified 24 MAGs and 59 UViGs as AE bioindicators of community stability using the RF's mean decrease Gini coefficient analysis. These 24 MAGs belonged to Firmicutes A (17), Bacteroidota (4), Actinobacteriota (3). The 59 UViGs were distributed to 10 families, the most dominant being Peduoviridae (26). Most of these UViGs (46) were temperate viruses, and we could assign hosts for 47 UViGs, from which 5 are also bioindicators (4 Clostridia, 1 Bacteroidia). Our study demonstrated that the gut microbiome of children with AE is a good model for exploring analysis stability in human gut microbiomes. Our bioindicators may be used to define omics-based diagnostic tools for AE, and an in-depth analysis of their genetic potential may open doors for novel microbiome-based treatments for AE.

Hu, Die; . . . ; da Roch, Ulisses Nunes. **Predicting the Stability of Gut Microbial Communities using Viral-Prokaryotic Genome-Centric Analysis Machine Learning in Children**

with Atopic Eczema. Department of Environmental Microbiology, Helmholtz Centre for Environmental Research, 04318 Leipzig, Saxony, Germany.

7.6 BioDeepFuse: A Hybrid Deep Learning Approach

The accurate classification of non-coding RNA (ncRNA) sequences is pivotal for advanced non-coding genome annotation and analysis, a fundamental aspect of genomics that facilitates understanding of ncRNA functions and regulatory mechanisms in various biological processes. While traditional machine learning approaches have been employed for distinguishing ncRNA, these often necessitate extensive feature engineering. Recently, deep learning algorithms have provided advancements in ncRNA classification. This study presents BioDeepFuse, a hybrid deep learning framework integrating convolutional neural networks (CNN) or bidirectional long short-term memory (BiLSTM) networks with handcrafted features for enhanced accuracy. This framework employs a combination of k-mer one-hot, k-mer dictionary, and feature extraction techniques for input representation. Extracted features, when embedded into the deep network, enable optimal utilization of spatial and sequential nuances of ncRNA sequences. Using benchmark datasets and real-world RNA samples from bacterial organisms, we evaluated the performance of BioDeepFuse. Results exhibited high accuracy in ncRNA classification, underscoring the robustness of our tool in addressing complex ncRNA sequence data challenges. The effective melding of CNN or BiLSTM with external features heralds promising directions for future research, particularly in refining ncRNA classifiers and deepening insights into ncRNAs in cellular processes and disease manifestations. In addition to its original application in the context of bacterial organisms, the methodologies and techniques integrated into our framework can potentially render BioDeepFuse effective in various and broader domains.

SANTOS, Anderson Paulo Avila; DE ALMEIDA, Breno Lívio Silva; BONIDIA, Robson Parmezan et al. **BioDeepFuse: A Hybrid Deep Learning Approach with Integrated Feature Extraction Techniques for Enhanced Non-coding RNA Classification**. In: Submitted in RNA Biology, 2023, accepted with minor reviews.

7.7 BioPrediction: Study of Molecular Interactions

Given the increasing number of biological sequences stored in databases, there is a large source of information that can benefit several sectors such as agriculture and health. Machine Learning (ML) algorithms can extract useful and new information from these data, increasing social and economic benefits, in addition to productivity. However, the categorical and unstructured nature of biological sequences makes this process difficult, requiring ML expertise. In this paper, we propose and experimentally evaluate an end-to-end automated ML-based framework, named BioPrediction, able to identify implicit interactions between sequences, e.g., long non-coding

RNA and protein pairs, without the need for end-to-end ML expertise. Our experimental results show that the proposed framework can induce ML models with high predictive accuracy, between 77% and 91%, which are competitive with state-of-the-art tools.

FLORENTINO, Bruno Rafael; SANCHES, Natan Henrique; BONIDIA, Robson Parmezan; CARVALHO, André C. P. L. F. de. **BioPrediction: Democratizing Machine Learning in the Study of Molecular Interactions.** In: Anais do XX Encontro Nacional de Inteligência Artificial e Computacional. SBC, 2023. p. 525-539.

7.8 BioAutoML: End-to-End Machine Learning Package for Life Sciences

Humanity has faced several challenges related to healthcare, epidemiological problems, climate change, energy consumption, and water resources. Consequently, with advances in sequencing, an increasing number of biological data has been generated in the post-genomic age, where approaches have been developed for genomics, transcriptomics, and proteomics problems. Due to this large amount of data, opportunities arise to change these challenging scenarios using Machine Learning (ML) algorithms. ML can extract useful and meaningful knowledge from biological data, reducing research expenses and increasing scientific efficiency. These advances benefit our society and economy, impacting people's lives in various areas, such as health care, the environment, pollution, and water treatment. Nevertheless, many studies usually neglected FAIR data principles for software development in ML. Furthermore, other challenges are that ML approaches applied to biological data also require quality steps related to feature engineering, algorithm selection, and hyperparameter tuning. These processes are manual and require extensive knowledge of ML. To address this concern, we developed BioAutoML, which automatically runs an end-to-end ML pipeline. To the best of our knowledge, our proposal automates the longest pipeline for biological sequence analysis, encompassing feature engineering, ML algorithm recommendation, and hyperparameter tuning. So far, we have achieved promising results on several problems, such as SARS-CoV-2, anticancer peptides, pro-inflammatory peptides, HIV-1 sequences, and phage virion proteins. BioAutoML lowers the barrier to applying feature engineering and metalearning in biological sequences for non-experts, democratizing ML in life sciences.

BONIDIA, Robson et al. **BioAutoML: End-to-End Machine Learning Package for Life Sciences.** In: 10th FEMS Congress of European Microbiologists, 2023, Hamburg - Germany. 10th FEMS Congress of European Microbiologists, 2023.

BONIDIA, Robson. **BioAutoML:Helmholtz Visiting Researcher Grant/Award** - Helmholtz Information & Data Science Academy (HIDA), 2023.

BONIDIA, Robson. **BioAutoML:FEMS Research & Training Grant/Award** - Federation of European Microbiological Societies (FEMS), 2023.

7.9 BioAutoML-API

In this article, we present a case study on democratization, especially Machine Learning (ML) resources in the field of biology, and the importance of these advancements for society. Technology has contributed to the advancement of human life. However, recent occurrences of recurring biological issues on the planet have taught us a lesson and accelerated an important process, the democratization of technology. Motivated by these factors, we have developed an API to support the data pipeline responsible for the processes of the BioAutoML application, which is an end-to-end ML tool designed for sequence data processing. Considering BioAutoML's capability to optimize scientific research in microbiology and biotechnology, we have recognized the need to build a scalable application to support the execution of multiple concurrent processes. This contributes to a more user-friendly environment and, consequently, a more democratic one. The goal is to facilitate access and encourage the equitable and responsible development and provision of ML-based tools.

Rampazzo, Felipe César S.; Souza, Octávio Onofre A.; BONIDIA, Robson Parmezan. **BioAutoML-API.** In: Iniciação Científica - Fatec Ourinhos - Ciência de Dados.

7.10 AutoAI-Pandemics

Infectious diseases, transmitted directly or indirectly, are among the main causes of epidemics, or even pandemics. Despite recent achievements, there are several open challenges in predicting epidemic outbreaks, detecting variants, contact tracing, discovering new drugs, and fighting misinformation. Artificial Intelligence (AI) can provide tools to deal with these scenarios, demonstrating promising results in the fight against the COVID-19 pandemic. Although AI creates new opportunities, its proper use requires advanced knowledge of computing, statistics, and mathematics, restricting its use by public health professionals working with infectious diseases. Our objective is to develop an integrated and user-friendly platform that can be effectively employed by non-experts working with infectious diseases. This platform, named AutoAI-Pandemics, will provide robust solutions using Automated Machine Learning for (T1) epidemiological analysis to detect possible epidemic scenarios and corresponding interventions to suppress disease spread with minimal social impact safely; (T2) bioinformatics analysis, supporting pathogen genome mining, and (T3) fighting misinformation by assisting the search for reliable information sources. This platform will be able to work on various critical stages of an epidemic/pandemic. Thus, it can be used by policymakers and other stakeholders, healthcare professionals, pharmaceutical industries, genomic surveillance organizations, and for combating disinformation. AutoAI-Pandemics will comply with what is expected of responsible AI solutions, which include fairness, privacy protection, sustainability, and respect for legislation. To deal with the complex aspects of this project, we assembled an interdisciplinary team of researchers with expertise in computer science, AI, bioinformatics, and infectious diseases epidemiology.

BONIDIA, Robson et al. **AutoAI-Pandemics: Democratizing Machine Learning for Analysis, Study, and Control of Epidemics and Pandemics**. This work was selected as one of the most promising proposals (a total of 221 proposals from 47 countries following a rigorous review process (142 from Africa, 40 from Asia, 26 from LAC, and 12 from MENA)) in a global competition, held by the Global South Artificial Intelligence for Pandemic and Epidemic Preparedness and Response Network - AI4PEP - 2023.

7.11 Democratizing Artificial Intelligence in LAC

Artificial Intelligence (AI) is becoming increasingly integrated into various aspects of society. However, it is crucial to ensure, not only, that AI benefits are distributed equitably, but also, its responsible use. Multiple countries are creating regulations to address these concerns, but the borderless nature of AI requires global cooperation to define regulatory and guideline consensus. Considering this, The Global South AI for Pandemic & Epidemic Preparedness & Response Network (AI4PEP) has developed an initiative comprising 16 projects across 16 countries in the Global South, seeking to strengthen equitable and responsive public health systems that leverage Southern-led responsible AI solutions to improve prevention, preparedness, and response to emerging and re-emerging infectious disease outbreaks. This paper introduces our branches in Latin American and Caribbean (LAC) countries and discusses AI governance in LAC. Our network in LAC has high potential to help fight infectious diseases, particularly in low– and middle-income countries, generating opportunities for the widespread use of AI techniques to improve the health and well-being of their communities.

CARVALHO, André; BONIDIA, Robson et al. **Democratising Artificial Intelligence for Pandemic Preparedness and Global Governance in Latin American and Caribbean Countries.** This work was submitted to the United Nations Call for Papers on Global AI Governance[1].

[1] https://www.un.org/techenvoy/ai-advisory-body)

CHAPTER 8

CONCLUSIONS AND FUTURE CHALLENGES

Artificial Intelligence (AI) offers valuable tools to mitigate the impact of numerous challenges that affect society. Advances in all knowledge domains show how AI can not only accelerate scientific discoveries and the design of innovative solutions but also be one of the most valuable tools for improving the quality of life on Earth. Nevertheless, it is crucial to ensure that these benefits are distributed equitably. To this end, we should establish plans to democratize AI and benefit all regions of the globe. The democratization of AI, especially Machine Learning (ML), is a transformative journey towards inclusion, innovation, and scientific excellence that should not just be a theoretical concept; but a practical effort.

In this thesis, we focus on one of the numerous definitions of AI democratization, which, according to (RUBEIS; DUBBALA; METZLER, 2022; VANHORN; ÇOBANOĞLU, 2022), involves granting accessibility to ML for individuals who are not specialists in the domain, such as those without a background in data science, mathematics, or informatics. To address this concern, we developed BioAutoML, which automatically runs an end-to-end ML pipeline for biological sequence data. To the best of our knowledge, our proposal automates the longest pipeline, encompassing feature engineering, ML algorithm recommendation, and hyperparameter tuning.

We chose the field of life sciences because ML algorithms can extract valuable and meaningful knowledge from biological data, accelerating discoveries, reducing research expenses, and enhancing scientific efficiency. These advancements can directly benefit society, the economy, and people's lives. Furthermore, other challenges are that ML approaches applied to biological data also require quality steps related to feature engineering, algorithm selection, and hyperparameter tuning. These processes often require extensive domain knowledge, performed manually by a human expert, being one of the most time-consuming steps.

As a result, the proposal of this thesis not only automates complex tasks but also enables researchers without domain knowledge to apply ML algorithms for sequence data analysis. The ability to generate an end-to-end automated ML pipeline reduces the burden of laborious manual data preprocessing. Our studies have generated results applicable to the analysis of biological sequences, demonstrating the considerable potential for substantially decreasing the expertise needed to operate AI/ML pipelines. This support aids researchers in addressing diverse issues, including diseases that profoundly affect human lives, giving biologists and other stakeholders an opportunity for the widespread use of these techniques.

Furthermore, we have achieved robust results with successful cases across multiple problem domains, such as SARS-CoV-2, anticancer peptides, pro-inflammatory peptides, HIV-1 sequences, phage virion proteins, non-classical secreted proteins, sigma70 promoters, protein lysine crotonylation, recombination spots, small non-coding RNAs, long non-coding RNAs, circular RNAs, and others. The results of this thesis also generated awards, grants, and publications in high-impact scientific journals. The papers and tools associated with the thesis, up until its completion, have garnered 104 stars on GitHub and approximately 119 citations. The principal papers derived from this thesis (five in total) have an accumulated impact factor of 59.319.

In addition, during my time in Germany at the Helmholtz Centre for Environmental Research — UFZ, Leipzig, we conducted tests with BioAutoML using real-world problems to fine-tune its functionalities, adapting them to address the practical challenges encountered by biologists, microbiologists, and virologists in their everyday work. Regarding the limitations, we are in the final stages of releasing a user-friendly and comprehensible web version, code-free, to broadly apply the benefits of BioAutoML. Additionally, we are also carrying out a code refactoring process to optimize computational execution costs.

Therefore, it is no longer acceptable for ML applications in Life Sciences, or any other field, to remain confined to the domain of specialists and data scientists. With our research, it has been possible to witness how democratization opens doors, breaks down barriers, and welcomes all who wish to harness its potential. This means a shift from exclusivity to accessibility, making ML a shared resource for the collective improvement of science and society. However, the power that comes with democratization requires an unwavering commitment to ethical conduct. For this reason, it is extremely important to address ethical considerations, data privacy, and scientific integrity as we democratize knowledge and tools.

To democratize AI knowledge, it is crucial to maximize the dissemination of results and products among target communities where motivating issues arise, as well as to train human resources. We must also educate young academics, health professionals, policymakers, journalists, and invested citizens. To this end, we participate in an initiative called AI4PEP (The Global South AI for Pandemic & Epidemic Preparedness & Response Network), which proposes various approaches, such as (1) Engagement with under-represented communities; (2) Special short courses; (3) Public awareness campaigns on AI; (4) Collaboration with Industry; (5) International Collaboration; (6) Open-source initiatives; and (7) Community development.

I recognize that our journey continues, ensuring that the benefits of democratization reach researchers, communities, and societies around the world. To achieve this, I will continue to dedicate myself to three new fronts: **(1) AutoAI-Pandemics**[1] (Democratizing Machine Learning for Analysis, Study, and Control of Epidemics and Pandemics); **(2) BioPrediction**[2] (Democratizing Machine Learning in the Study of Molecular Interactions), and **(3) Drug Discovery** (Democratizing Machine Learning in the Study of Drug Discovery). Finally, the democratization of AI is more than just a possibility – it is a necessity. By utilizing tools such as BioAutoML, and engaging in initiatives like AI4PEP, we are actively working to dismantle barriers and promote inclusion for a future in which the power of AI is accessible to the many, not just the privileged few.

8.1 Grants, Awards, and Books

- **Google Latin America Research Awards (LARA)**, Google, 2021. Project: BioAutoML: Automated Feature Engineering for Classification of Biological Sequences. Elected by LARA-Google among the 24 most promising ideas in Latin America (24 awarded projects, from a base of 700 submissions)[3,4,5].

- **AutoAI-Pandemics, which was selected as one of the most promising proposals** (a total of 221 proposals from 47 countries following a rigorous review process (142 from Africa, 40 from Asia, 26 from LAC, 12 from MENA)) in a global competition, held by the Global South Artificial Intelligence for Pandemic and Epidemic Preparedness and Response Network – AI4PEP – CAN$362,500[6,7,8,9].

- CARVALHO, André; MENEZES, Angelo; BONIDIA, Robson. **Ciência de Dados – Fundamentos e Aplicações**. GEN – Grupo Editorial Nacional, v. 1, 2023 — In launch phase.

- **Helmholtz Visiting Researcher Grant/Award** – Helmholtz Information & Data Science Academy (HIDA), 2023. Project Title: BioAutoML-Fast: End-to-End Multi-Threaded Machine Learning Package for Life Sciences.

[1] http://autoaipandemics.icmc.usp.br/
[2] https://github.com/Bonidia/BioPrediction
[3] https://blog.google/intl/pt-br/novidades/iniciativas/conheca-os-vencedores-do-premio-lara-2021-o-programa-de-bolsas-de-pesquisa-do-google/
[4] https://cemeai.icmc.usp.br/programa-de-bolsas-do-google-premia-trabalhos-orientados-pelo-cemeai/
[5] https://npdiario.com.br/cidades/norte-pioneirense-e-premiado-pelo-google/
[6] http://autoaipandemics.icmc.usp.br/
[7] https://veja.abril.com.br/tecnologia/plataforma-virtual-ajuda-a-combater-epidemias-com-inteligencia-artificial/
[8] http://www.saocarlos.usp.br/cemeai-apoia-projeto-selecionado-em-chamada-global-para-ia-na-saude-publica/
[9] https://ai4pep.org/our-projects/

- **FEMS Research & Training Grant/Award** – Federation of European Microbiological Societies (FEMS), 2023 (€: 5,000.00).

- **BioPrediction – Project selected to participate in Prototypes for Humanity 2023, during COP28-Dubai**, chosen from 3000 entries, from more than 100 countries, standing out among the 100 best, Prototypes for Humanity – COP28-Dubai[10].

- **Finalists (Top 15 of 82) – Falling Walls Lab Brazil 2022**, DWIH São Paulo, Falling Walls Foundation, DAAD The German Center for Science and Innovation[11].

- **Winning Team (Advisor), 1st place, "Breaking the Wall of Fake News"**, Falling Walls Lab Brazil 2023, DWIH São Paulo, Falling Walls Foundation, DAAD, The German Center for Research and Innovation [12].

- **Supervisor of the ÁGUEDA Project** (Artificial Intelligence for Early Detection of Breast Cancer), recognized as the best ongoing research project in the field of Exact and Earth Sciences in Brazil for undergraduate scientific initiation by Conic-Semesp (the Largest Scientific Initiation Congress in Brazil), among more than 1,400 registered projects.[13].

- **Finalist Team (Advisor), "Breaking the Wall of Alzheimer's Detection"**, Falling Walls Lab Brazil 2023, DWIH São Paulo, Falling Walls Foundation, DAAD, The German Center for Research and Innovation [14].

- **Certificate of excellence** for the distinction achieved at Falling Walls Lab Brazil 2023 as Advisor Professor, 2023.

- **Motion No. 405/2023 (Ourinhos-SP City Council, Brazil)** — Congratulations to the students and professors of Fatec Ourinhos for developing the 'ITT – Is That True' app, a platform to combat fake news, 2023.

8.2 Papers Published During the Ph.D.

- **IEEE Access (IF 2019: 3.745):** BONIDIA, Robson Parmezan et al. A Novel Decomposing Model With Evolutionary Algorithms for Feature Selection in Long Non-Coding RNAs – (BONIDIA *et al.*, 2020a)

- **Briefings in Bioinformatics (IF 2020: 11.622):** BONIDIA, Robson Parmezan et al. Feature extraction approaches for biological sequences: a comparative study of mathematical features – (BONIDIA *et al.*, 2021a)

[10] https://www.prototypesforhumanity.com/
[11] https://www.youtube.com/watch?v=H5C_UIgVeQM
[12] http://biofatecou.fatecourinhos.edu.br
[13] https://www.semesp.org.br/conic/resultados/
[14] http://biofatecou.fatecourinhos.edu.br

- **Nucleic Acids Research (IF 2020: 16.971):** CRISPRloci: comprehensive and accurate annotation of CRISPR–Cas systems – (ALKHNBASHI *et al.*, 2021)

- **Briefings in Bioinformatics (IF 2021: 13.994):** BONIDIA, Robson Parmezan et al. MathFeature: feature extraction package for DNA, RNA, and protein sequences based on mathematical descriptors – (BONIDIA *et al.*, 2022)

- **BSB – Brazilian Symposium on Bioinformatics:** MathPIP: Classification of Proinflammatory Peptides Using Mathematical Descriptors – (CAVALCANTE *et al.*, 2021)

- **BSB – Brazilian Symposium on Bioinformatics:** Feature Importance Analysis of Noncoding DNA/RNA Sequences Based on Machine Learning Approaches – (ALMEIDA *et al.*, 2021)

- **Entropy (IF 2021: 2.738):** BONIDIA et al. Information Theory for Biological Sequence Classification: A Novel Feature Extraction Technique Based on Tsallis Entropy, 2022 – (BONIDIA *et al.*, 2022b)

- **Briefings in Bioinformatics (IF 2021: 13.994):** BONIDIA et al. BioAutoML: Automated Feature Engineering and Metalearning for the Prediction of Non-Coding RNAs in Bacteria, 2022 – (BONIDIA *et al.*, 2022c)

- **20th Encontro Nacional de Inteligência Artificial e Computacional:** BioPrediction: Democratizing Machine Learning in the Study of Molecular Interactions, 2023 – (FLORENTINO *et al.*, 2023).

- **This work was submitted to the United Nations Call for Papers on Global AI Governance:** Democratizing Artificial Intelligence for Pandemic Preparedness and Global Governance in Latin American and Caribbean Countries.

- **RNA Biology (IF 2022: 4.1000):** SANTOS, Anderson Paulo Avila; DE ALMEIDA, Breno Lívio Silva; BONIDIA, Robson Parmezan et al. BioDeepFuse: A Hybrid Deep Learning Approach with Integrated Feature Extraction Techniques for Enhanced Non-coding RNA Classification, 2023 — accepted with minor reviews.

- **Nucleic Acids Research (IF 2022: 14.900):** BONIDIA, Robson Parmezan et al. BioAutoML: Democratizing Machine Learning in Life Sciences, 2023 – In preparation.

8.3 Other Recognitions Obtained During the Ph.D.

- **Finalist in the Higher Education Category (Among the 10 finalists in the Higher Education Category – 2897 subscribers - BioFatecou Project), Transformer Educator**

Award - Sebrae, Instituto Significare and Bett Brasil, which aims to select the most transformative projects in Brazil, 2023[15,16].

- Honorable mention to the BioFatecou Project (more than 200 submissions) – **25th edition of the Professor Mário Palmério Top Educational Award** - ABMES - Associação Brasileira de Mantenedoras de Ensino Superior - Brazil - 2023[17,18].
- **Hollie's Hub for Good – DigitalOcean**. BioFatecou: Introducing Undergraduates to Academic Research, 2023 (US$: 2,500.00)[19].
- Grace: Resume Recommendation System with Artificial Intelligence — Article published at the IV Workshop for Undergraduate Student Works, held during the Brazilian Symposium on Databases (SBBD) 2023.
- Predicting Playoff Winners: A Case Study with Machine Learning in American Football Matches — Article published at the IV Workshop for Undergraduate Student Works, held during the Brazilian Symposium on Databases (SBBD) 2023.
- Advisor to undergraduate students Ana Clara B. Medeiros et al., 2nd place at the 1st Cambará Ideas Fair, Grace: Resume Recommendation System with Artificial Intelligence — 1st Cambará Ideas Fair - Norte Pioneiro - Paraná, 2023.
- Advisor of undergraduate student Wagner Lopes Cardozo, 3rd place at the 1st Cambará Ideas Fair, Águeda: Artificial Intelligence for Early Detection of Breast Cancer – 1st Cambará Ideas Fair - Norte Pioneiro - Paraná, 2023.

8.4 Supplementary Endeavors Throughout the Ph.D.

- **Organizer – I Workshop de Soluções de Problemas com Ciência de Dados (2022)[20]:** (1) Event with 210 registrations; (2) Meeting with 10 companies; (3) Presentation of projects; (4) Expert lectures.
- **1st Data Science Challenge - Fatec Ourinhos**[21] National Football League (NFL) Results Prediction.

[15] https://www.cps.sp.gov.br/professor-da-fatec-ourinhos-chega-a-final-do-premio-educador-transformador/

[16] https://conteudo.significare.org.br/finalistas_premio-educador-transformador

[17] https://top.abmes.org.br/noticias-2/54-projeto-que-oferece-formacao-on-line-e-gratuita-sobre-competencias-empreendedoras-vence-a-25-edicao-do-premio-top-educacional

[18] https://www.linkedin.com/feed/update/urn:li:activity:7094726276795502592/

[19] https://www.digitalocean.com/community/pages/hollies-hub-for-good

[20] https://bonidia.github.io/workshop-cd/

[21] https://www.kaggle.com/competitions/1-desafio-cd-fatec-ourinhos

- **Organizer – BioFatecou Deep Talks (2023)[22]:** (1) Twenty presentations of projects; (2) Event with 120 registrations.

- **BioFatecou[23]:** Introducing Undergraduate Students to Academic Research and Responsible Use of Artificial Intelligence — The project aims to contribute to the development of excellent professionals, with ethical awareness of AI usage, who will work directly in the field, often impacting people's lives with their solutions. Among other contributions, we can mention: (I) Personal and professional growth; (II) Cultivation of critical thinking, impacting all areas of their lives; and (III) Interaction with other professionals in the field, promoting networking. The main competencies emphasized in the project include argumentation, communication, knowledge, empathy, planning, organization, teamwork, responsibility, citizenship, and scientific, critical, and creative thinking. Additionally, BioFatecou's goal is to encourage participants to develop solutions that align with the Sustainable Development Goals (SDGs) [9], such as (SDG 3) Good Health and Well-being; (SDG 4) Quality Education; (SDG 5) Gender Equality; (SDG 10) Reduced Inequalities; and (SDG 13) Climate Action. The project has been underway in the Data Science program at Faculdade de Tecnologia de Ournhos (FATEC) since August 2021, integrated into the courses of Integrative Project (III, IV, and V). Currently, it involves the participation of over 54 students, a number projected to exceed 100 in the upcoming months, with prospects for growth.

[22] https://drive.google.com/drive/u/1/folders/1OXHM79q5bIdPiNV1oFdWAep-_s-moH05

[23] http://biofatecou.fatecourinhos.edu.br/

BIBLIOGRAPHY

ABBAS, Q.; RAZA, S. M.; BIYABANI, A. A.; JAFFAR, M. A. A review of computational methods for finding non-coding rna genes. **Genes**, Multidisciplinary Digital Publishing Institute, v. 7, n. 12, p. 113, 2016. Citation on page 56.

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. Citation on page 61.

Abubaker Bagabir, S.; IBRAHIM, N. K.; Abubaker Bagabir, H.; Hashem Ateeq, R. Covid-19 and artificial intelligence: Genome sequencing, drug development and vaccine discovery. **Journal of Infection and Public Health**, v. 15, n. 2, p. 289–296, 2022. ISSN 1876-0341. Citation on page 75.

ACHAWANANTAKUN, R.; CHEN, J.; SUN, Y.; ZHANG, Y. Lncrna-id: Long non-coding rna identification using balanced random forests. **Bioinformatics**, Oxford University Press, v. 31, n. 24, p. 3897–3905, 2015. Citation on page 57.

ADAMI, C. The use of information theory in evolutionary biology. **Annals of the New York Academy of Sciences**, v. 1256, n. 1, p. 49–65, 2012. Citation on page 78.

ADAMSON, A. S.; SMITH, A. Machine learning and health care disparities in dermatology. **JAMA dermatology**, American Medical Association, v. 154, n. 11, p. 1247–1248, 2018. Citation on page 26.

AHMED, S.; MULA, R. S.; DHAVALA, S. S. A framework for democratizing ai. **arXiv preprint arXiv:2001.00818**, 2020. Citation on page 21.

AHMED, W.; ZHENG, K.; LIU, Z. F. Small non-coding RNAs: New insights in modulation of host immune response by intracellular bacterial pathogens. **Frontiers in Immunology**, Frontiers Media S.A., v. 7, n. OCT, p. 431, oct 2016. ISSN 16643224. Citation on page 108.

AITTOKALLIO, T.; SCHWIKOWSKI, B. Graph-based methods for analysing networks in cell biology. **Brief Bioinformatics**, v. 7, n. 3, p. 243–255, sep 2006. ISSN 14675463. Available: <https://pubmed.ncbi.nlm.nih.gov/16880171/>. Citation on page 98.

AKHTER, S.; BAILEY, B. A.; SALAMON, P.; AZIZ, R. K.; EDWARDS, R. A. Applying shannon's information theory to bacterial and phage genomes and metagenomes. **Scientific reports**, Nature Publishing Group, v. 3, p. 1033, 2013. Citations on pages 76 and 97.

ALBUQUERQUE, M. P. D.; ESQUEF, I. A.; MELLO, A. G. Image thresholding using tsallis entropy. **Pattern Recognition Letters**, Elsevier, v. 25, n. 9, p. 1059–1065, 2004. Citations on pages 76, 78, and 79.

ALKHNBASHI, O. S.; MITROFANOV, A.; BONIDIA, R. *et al.* CRISPRloci: comprehensive and accurate annotation of CRISPR–Cas systems. **Nucleic Acids Research**, v. 49, n. W1, p. W125–W130, 06 2021. ISSN 0305-1048. Citation on page 137.

ALMEIDA, B. L. S. d.; QUEIROZ, A. P.; SANTOS, A. P. A.; BONIDIA, R. P.; ROCHA, U. N. d.; SANCHES, D. S.; CARVALHO, A. C. P. d. L. F. d. Feature importance analysis of non-coding dna/rna sequences based on machine learning approaches. In: SPRINGER. **Brazilian Symposium on Bioinformatics**. [S.l.], 2021. p. 81–92. Citations on pages 112 and 137.

ALMEIDA, J. S.; CARRICO, J. A.; MARETZEK, A.; NOBLE, P. A.; FLETCHER, M. Analysis of genomic sequences by chaos game representation. **Bioinformatics**, Oxford University Press, v. 17, n. 5, p. 429–437, 2001. Citation on page 97.

AMERIFAR, S.; NOROUZI, M.; GHANDI, M. A tool for feature extraction from biological sequences. **Briefings in Bioinformatics**, Oxford University Press, v. 23, n. 3, p. bbac108, 2022. Citation on page 22.

Anastassiou, D. Genomic signal processing. **IEEE Signal Processing Magazine**, v. 18, n. 4, p. 8–20, July 2001. ISSN 1558-0792. Citations on pages 61, 96, and 97.

ARSLAN, H. Machine learning methods for covid-19 prediction using human genomic data. In: **Multidisciplinary Digital Publishing Institute Proceedings**. [S.l.: s.n.], 2021. v. 74, n. 1, p. 20. Citation on page 86.

____. Machine learning methods for covid-19 prediction using human genomic data. **Proceedings**, v. 74, n. 1, 2021. ISSN 2504-3900. Available: <https://www.mdpi.com/2504-3900/74/1/20>. Citations on pages 92, 102, and 104.

BABIC, B.; GERKE, S.; EVGENIOU, T.; COHEN, I. G. Beware explanations from ai in health care. **Science**, American Association for the Advancement of Science, v. 373, n. 6552, p. 284–286, 2021. Citation on page 26.

BAEK, J.; LEE, B.; KWON, S.; YOON, S. lncrnanet: Long non-coding rna identification using deep learning. **Bioinformatics**, Oxford University Press, v. 1, p. 9, 2018. Citations on pages 56, 57, 71, and 103.

BALAJI, A.; ALLEN, A. Benchmarking automatic machine learning frameworks. **arXiv preprint arXiv:1808.06492**, 2018. Citations on pages 120, 121, and 122.

BAR, A.; ARGAMAN, L.; ALTUVIA, Y.; MARGALIT, H. Prediction of novel bacterial small rnas from ril-seq rna–rna interaction data. **Frontiers in microbiology**, Frontiers Media SA, v. 12, 2021. Citation on page 112.

BARIK, A.; DAS, S. A comparative study of sequence-and structure-based features of small rnas and other rnas of bacteria. **RNA biology**, Taylor & Francis, v. 15, n. 1, p. 95–103, 2018. Citations on pages 112 and 116.

BARMAN, R. K.; MUKHOPADHYAY, A.; DAS, S. An improved method for identification of small non-coding rnas in bacteria using support vector machine. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 1–8, 2017. Citation on page 113.

BECHT, E.; MCINNES, L.; HEALY, J.; DUTERTRE, C.-A.; KWOK, I. W.; NG, L. G.; GINHOUX, F.; NEWELL, E. W. Dimensionality reduction for visualizing single-cell data using umap. **Nature biotechnology**, Nature Publishing Group, v. 37, n. 1, p. 38–44, 2019. Citation on page 89.

BENDTSEN, J. D.; KIEMER, L.; FAUSBØLL, A.; BRUNAK, S. Non-classical protein secretion in bacteria. **BMC microbiology**, BioMed Central, v. 5, n. 1, p. 1–13, 2005. Citation on page 100.

BENTÉJAC, C.; CSÖRGŐ, A.; MARTÍNEZ-MUÑOZ, G. A comparative analysis of gradient boosting algorithms. **Artificial Intelligence Review**, Springer, p. 1–31, 2020. Citation on page 62.

BENVENUTO, D.; GIOVANETTI, M.; CICCOZZI, A.; SPOTO, S.; ANGELETTI, S.; CICCOZZI, M. The 2019-new coronavirus epidemic: Evidence for virus evolution. **Journal of Medical Virology**, v. 92, n. 4, p. 455–459, 2020. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jmv.25688>. Citation on page 56.

BERG, B. A. van den; REINDERS, M. J.; ROUBOS, J. A.; RIDDER, D. D. Spice: a web-based tool for sequence-based protein classification and exploration. **BMC bioinformatics**, Springer, v. 15, n. 1, p. 93, 2014. Citations on pages 34, 40, 92, 93, 94, and 111.

BERGSTRA, J.; YAMINS, D.; COX, D. D. *et al.* Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: CITESEER. **Proceedings of the 12th Python in science conference**. [S.l.], 2013. v. 13, p. 20. Citation on page 115.

BERRY, M. W. Large-scale sparse singular value computations. **The International Journal of Supercomputing Applications**, SAGE Publications Sage UK: London, England, v. 6, n. 1, p. 13–49, 1992. Citation on page 89.

BLOCH, K. M.; ARCE, G. R. Analyzing protein sequences using signal analysis techniques. In: **Computational and Statistical Approaches to Genomics**. [S.l.]: Springer, 2006. p. 137–161. Citation on page 96.

BONIDIA, R. P. **Feature Extraction Approaches for Biological Sequences: A Comparative Study of Mathematical Features**. [S.l.]: GitHub, 2020. Https://github.com/Bonidia/FeatureExtraction_BiologicalSequences. Citations on pages 69 and 109.

BONIDIA, R. P.; DOMINGUES, D. S.; SANCHES, D. S.; CARVALHO, A. C. de. Mathfeature: feature extraction package for dna, rna and protein sequences based on mathematical descriptors. **Briefings in Bioinformatics**, Oxford University Press, v. 23, n. 1, p. bbab434, 2022. Citations on pages 75, 76, 81, 109, 113, and 137.

BONIDIA, R. P.; MACHIDA, J. S.; NEGRI, T. C.; ALVES, W. A. L.; KASHIWABARA, A. Y.; DOMINGUES, D. S.; CARVALHO, A. D.; PASCHOAL, A. R.; SANCHES, D. S. A novel decomposing model with evolutionary algorithms for feature selection in long non-coding rnas. **IEEE Access**, v. 8, p. 181683–181697, 2020. Citations on pages 92, 115, and 136.

BONIDIA, R. P. *et al.* **Feature extraction and selection analysis in biological sequence: a case study with metaheuristics and mathematical models**. Master's Thesis (Master's Thesis) — Universidade Tecnológica Federal do Paraná, 2020. Citation on page 56.

BONIDIA, R. P.; SAMPAIO, L. D. H.; DOMINGUES, D. S.; PASCHOAL, A. R.; LOPES, F. M.; CARVALHO, A. C. P. L. F. de; SANCHES, D. S. Feature extraction approaches for biological sequences: a comparative study of mathematical features. **Briefings in Bioinformatics**, 02 2021. ISSN 1477-4054. Bbab011. Citations on pages 25, 61, 62, 75, 76, 78, 80, 81, and 136.

____. Feature extraction approaches for biological sequences: a comparative study of mathematical features. **Briefings in Bioinformatics**, 02 2021. ISSN 1477-4054. Bbab011. Available: <https://doi.org/10.1093/bib/bbab011>. Citations on pages 92, 97, 98, and 99.

BONIDIA, R. P.; SAMPAIO, L. D. H.; LOPES, F. M.; SANCHES, D. S. 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. Citations on pages 56, 59, and 92.

BONIDIA, R. P.; SANTOS, A. P. A.; ALMEIDA, B. L. de; STADLER, P. F.; ROCHA, U. N. da; SANCHES, D. S.; CARVALHO, A. C. de. Bioautoml: automated feature engineering and metalearning to predict noncoding rnas in bacteria. **Briefings in Bioinformatics**, Oxford University Press, v. 23, n. 4, p. bbac218, 2022. Citation on page 82.

BONIDIA, R. P.; SANTOS, A. P. A.; ALMEIDA, B. L. de; STADLER, P. F.; ROCHA, U. Nunes da; SANCHES, D. S.; CARVALHO, A. C. D. Information theory for biological sequence classification: A novel feature extraction technique based on tsallis entropy. **Entropy**, MDPI, v. 24, n. 10, p. 1398, 2022. Citation on page 137.

BONIDIA, R. P.; SANTOS, A. P. A.; ALMEIDA, B. L. S. de; STADLER, P. F.; ROCHA, U. N. da; SANCHES, D. S.; CARVALHO, A. C. P. L. F. de. BioAutoML: automated feature engineering and metalearning to predict noncoding RNAs in bacteria. **Briefings in Bioinformatics**, v. 23, n. 4, p. bbac218, 06 2022. ISSN 1477-4054. Available: <https://doi.org/10.1093/bib/bbac218>. Citation on page 137.

BRAZDIL, P. B.; RIJN, J. N. van; SOARES, C.; VANSCHOREN, J. **Metalearning: Applications to Automated Machine Learning and Data Mining**. [S.l.]: Leiden University, Institute of Advanced Computer Science, 2022. Citations on pages 24 and 111.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citation on page 61.

BRERETON, P.; KITCHENHAM, B. A.; BUDGEN, D.; TURNER, M.; KHALIL, M. Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, Elsevier, v. 80, n. 4, p. 571–583, 2007. Citations on pages 29, 30, 32, 33, and 77.

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. Available: <https://doi.org/10.1093/bioinformatics/bty222>. Citation on page 56.

CANNATARO, M.; HARRISON, A. Bioinformatics helping to mitigate the impact of COVID-19 – Editorial. **Briefings in Bioinformatics**, v. 22, n. 2, p. 613–615, 03 2021. ISSN 1477-4054. Available: <https://doi.org/10.1093/bib/bbab063>. Citations on pages 21 and 107.

CAO, D.-S.; XIAO, N.; XU, Q.-S.; CHEN, A. F. Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions. **Bioinformatics**, v. 31, n. 2, p. 279–281, 09 2014. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btu624>. Citations on pages 35 and 40.

CAO, D.-S.; XU, Q.-S.; LIANG, Y.-Z. propy: a tool to generate various modes of Chou's PseAAC. **Bioinformatics**, v. 29, n. 7, p. 960–962, 02 2013. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btt072>. Citations on pages 34, 40, 92, 93, 94, and 111.

CARDIN, S.-E.; BORCHERT, G. M. Viral micrornas, host micrornas regulating viruses, and bacterial microrna-like rnas. **Bioinformatics in MicroRNA Research**, Springer, p. 39–56, 2017. Citation on page 119.

CASSIANO, M. H. A.; SILVA-ROCHA, R. Benchmarking bacterial promoter prediction tools: Potentialities and limitations. **Msystems**, Am Soc Microbiol, v. 5, n. 4, p. e00439–20, 2020. Citation on page 104.

CAVALCANTE, J. P. U.; GONÇALVES, A. C.; BONIDIA, R. P.; SANCHES, D. S.; CARVALHO, A. C. P. d. L. F. d. Mathpip: Classification of proinflammatory peptides using mathematical descriptors. In: SPRINGER. **Brazilian Symposium on Bioinformatics**. [S.l.], 2021. p. 131–136. Citation on page 137.

CHAABANE, M.; WILLIAMS, R. M.; STEPHENS, A. T.; PARK, J. W. circdeep: deep learning approach for circular rna classification from other long non-coding rna. **Bioinformatics**, Oxford University Press, v. 36, n. 1, p. 73–80, 2020. Citations on pages 61 and 69.

CHAKRAVARTHY, N.; SPANIAS, A.; IASEMIDIS, L. D.; TSAKALIS, K. Autoregressive modeling and feature analysis of dna sequences. **EURASIP Journal on Applied Signal Processing**, Hindawi Publishing Corp., v. 2004, p. 13–28, 2004. Citations on pages 61 and 96.

CHAN, P. P.; LIN, B. Y.; MAK, A. J.; LOWE, T. M. trnascan-se 2.0: improved detection and functional classification of transfer rna genes. **Nucleic acids research**, Oxford University Press, v. 49, n. 16, p. 9077–9096, 2021. Citation on page 122.

CHAROENKWAN, P.; KANTHAWONG, S.; SCHADUANGRAT, N.; YANA, J.; SHOOMBUATONG, W. Pvpred-scm: improved prediction and analysis of phage virion proteins using a scoring card method. **Cells**, Multidisciplinary Digital Publishing Institute, v. 9, n. 2, p. 353, 2020. Citation on page 101.

CHAROENKWAN, P.; NANTASENAMAT, C.; HASAN, M. M.; SHOOMBUATONG, W. Meta-ipvp: a sequence-based meta-predictor for improving the prediction of phage virion proteins using effective feature representation. **Journal of Computer-Aided Molecular Design**, Springer, v. 34, n. 10, p. 1105–1116, 2020. Citations on pages 100 and 101.

CHEN, C.-C.; QIAN, X.; YOON, B.-J. RNAdetect: efficient computational detection of novel non-coding RNAs. **Bioinformatics**, v. 35, n. 7, p. 1133–1141, 08 2018. ISSN 1367-4803. Citation on page 108.

CHEN, D.; YUAN, C.; ZHANG, J.; ZHANG, Z.; BAI, L.; MENG, Y.; CHEN, L.-L.; CHEN, M. PlantNATsDB: a comprehensive database of plant natural antisense transcripts. **Nucleic Acids Research**, v. 40, n. D1, p. D1187–D1193, 11 2011. ISSN 0305-1048. Available: <https://doi.org/10.1093/nar/gkr823>. Citation on page 60.

CHEN, L.; ZHANG, Y.-H.; HUANG, G.; PAN, X.; WANG, S.; HUANG, T.; CAI, Y.-D. 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. Citations on pages 56, 61, 63, and 69.

CHEN, W.; DING, H.; FENG, P.; LIN, H.; CHOU, K.-C. iacp: a sequence-based tool for identifying anticancer peptides. **Oncotarget**, Impact Journals, LLC, v. 7, n. 13, p. 16895, 2016. Citation on page 104.

CHEN, W.; LEI, T.-Y.; JIN, D.-C.; LIN, H.; CHOU, K.-C. Pseknc: A flexible web server for generating pseudo k-tuple nucleotide composition. **Analytical Biochemistry**, v. 456, p. 53 – 60, 2014. ISSN 0003-2697. Available: <http://www.sciencedirect.com/science/article/pii/S0003269714001249>. Citations on pages 34, 40, and 92.

CHEN, W.; ZHANG, X.; BROOKER, J.; LIN, H.; ZHANG, L.; CHOU, K.-C. 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. Available: <https://doi.org/10.1093/bioinformatics/btu602>. Citations on pages 34, 40, 92, 93, 94, 98, and 111.

CHEN, X.; HAN, P.; ZHOU, T.; GUO, X.; SONG, X.; LI, Y. circrnadb: a comprehensive database for human circular rnas with protein-coding annotations. **Scientific reports**, Nature Publishing Group, v. 6, n. 1, p. 1–6, 2016. Citation on page 61.

CHEN, X.; LIN, Q.; LUO, C.; LI, X.; ZHANG, H.; XU, Y.; DANG, Y.; SUI, K.; ZHANG, X.; QIAO, B. *et al.* Neural feature search: A neural architecture for automated feature engineering. In: IEEE. **2019 IEEE International Conference on Data Mining (ICDM)**. [S.l.], 2019. p. 71–80. Citations on pages 23 and 110.

CHEN, Z.; ZHAO, P.; LI, C.; LI, F.; XIANG, D.; CHEN, Y.-Z.; AKUTSU, T.; DALY, R.; WEBB, G.; ZHAO, Q.; KURGAN, L.; SONG, J. iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. **Nucleic Acids Research**, 02 2021. ISSN 0305-1048. Gkab122. Available: <https://doi.org/10.1093/nar/gkab122>. Citations on pages 21, 22, 25, 53, 92, 107, 108, and 111.

CHEN, Z.; ZHAO, P.; LI, F.; LEIER, A.; MARQUEZ-LAGO, T. T.; WANG, Y.; WEBB, G. I.; SMITH, A. I.; DALY, R. J.; CHOU, K.-C.; SONG, J. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. **Bioinformatics**, v. 34, n. 14, p. 2499–2502, 03 2018. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/bty140>. Citations on pages 35, 40, 92, 93, 94, and 95.

CHEN, Z.; ZHAO, P.; LI, F.; MARQUEZ-LAGO, T. T.; LEIER, A.; REVOTE, J.; ZHU, Y.; POWELL, D. R.; AKUTSU, T.; WEBB, G. I.; CHOU, K.-C.; SMITH, A. I.; DALY, R. J.; LI, J.; SONG, J. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. **Briefings in Bioinformatics**, v. 21, n. 3, p. 1047–1057, 04 2019. ISSN 1477-4054. Available: <https://doi.org/10.1093/bib/bbz041>. Citations on pages 22, 23, 25, 36, 40, 56, 92, 93, 94, 95, 103, 107, 110, and 111.

CHENG, H.; GARRICK, D. J.; FERNANDO, R. L. Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction. **Journal of animal science and biotechnology**, Springer, v. 8, n. 1, p. 38, 2017. Citation on page 62.

CHIU, T.-P.; COMOGLIO, F.; ZHOU, T.; YANG, L.; PARO, R.; ROHS, R. DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. **Bioinformatics**, v. 32, n. 8, p. 1211–1213, 12 2015. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btv735>. Citations on pages 35, 40, 92, 93, 94, and 111.

CHOLLET, F. Keras: https://keras. io, 2015. Citation on page 103.

CHOU, K.-C. Some remarks on protein attribute prediction and pseudo amino acid composition. **Journal of theoretical biology**, Elsevier, v. 273, n. 1, p. 236–247, 2011. Citations on pages 23, 92, 93, and 110.

CHU, Q.; ZHANG, X.; ZHU, X.; LIU, C.; MAO, L.; YE, C.; ZHU, Q.-H.; FAN, L. Plantcircbase: a database for plant circular rnas. **Molecular plant**, Elsevier, v. 10, n. 8, p. 1126–1128, 2017. Citations on pages 60, 61, and 81.

CIRILLO, D.; CATUARA-SOLARZ, S.; MOREY, C.; GUNEY, E.; SUBIRATS, L.; MELLINO, S.; GIGANTE, A.; VALENCIA, A.; REMENTERIA, M. J.; CHADHA, A. S. *et al.* Sex and gender differences and biases in artificial intelligence for biomedicine and healthcare. **NPJ digital medicine**, Nature Publishing Group UK London, v. 3, n. 1, p. 81, 2020. Citation on page 26.

CLARK, D. P.; PAZDERNIK, N. J.; MCGEHEE, M. R. Chapter 19 - noncoding rna. In: CLARK, D. P.; PAZDERNIK, N. J.; MCGEHEE, M. R. (Ed.). **Molecular Biology (Third Edition)**. Third edition. Academic Cell, 2019. p. 604–621. ISBN 978-0-12-813288-3. Available: <https://www.sciencedirect.com/science/article/pii/B9780128132883000197>. Citation on page 122.

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. Citation on page 62.

CONSORTIUM, R. RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. **Nucleic Acids Research**, v. 49, n. D1, p. D212–D220, 10 2020. Citations on pages 108 and 119.

COSTA, M. C.; OLIVEIRA, J. V. de A.; SILVA, W. M.; SEN, R.; FALLMANN, J.; STADLER, P. F.; WALTER, M. E. M. Machine learning studies of non-coding rnas based on artificially constructed training data. In: **BIOINFORMATICS**. [S.l.: s.n.], 2021. p. 176–183. Citation on page 107.

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. Citations on pages 61 and 96.

CRUZ-GARCíA, J. S. De la; BORY-REYES, J.; RAMIREZ-ARELLANO, A. A two-parameter fractional tsallis decision tree. **Entropy**, v. 24, n. 5, 2022. ISSN 1099-4300. Available: <https://www.mdpi.com/1099-4300/24/5/572>. Citations on pages 76 and 79.

CUI, F.; ZHANG, Z.; ZOU, Q. Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. **Briefings in Functional Genomics**, v. 20, n. 1, p. 61–73, 02 2021. ISSN 2041-2657. Available: <https://doi.org/10.1093/bfgp/elaa030>. Citation on page 75.

DANG, T. H. Y.; TYAGI, S.; D'CUNHA, G.; BHAVE, M.; CRAWFORD, R.; IVANOVA, E. P. Computational prediction of micrornas in marine bacteria of the genus thalassospira. **PloS one**, Public Library of Science San Francisco, CA USA, v. 14, n. 3, p. e0212996, 2019. Citation on page 119.

DAR, D.; SOREK, R. Bacterial noncoding RNAs excised from within protein-coding transcripts. **mBio**, American Society for Microbiology, v. 9, n. 5, sep 2018. ISSN 21507511. Available: <https://journals.asm.org/doi/abs/10.1128/mBio.01730-18>. Citation on page 108.

DESHPANDE, S.; SHUTTLEWORTH, J.; YANG, J.; TARAMONLI, S.; ENGLAND, M. Plit: An alignment-free computational tool for identification of long non-coding rnas in plant transcriptomic datasets. **Computers in Biology and Medicine**, v. 105, p. 169 – 181, 2019. ISSN 0010-4825. Citation on page 57.

DIGNUM, V. **Responsible artificial intelligence: how to develop and use AI in a responsible way**. [S.l.]: Springer, 2019. Citation on page 26.

DING, H.; FENG, P.-M.; CHEN, W.; LIN, H. Identification of bacteriophage virion proteins by the anova feature selection and analysis. **Molecular BioSystems**, Royal Society of Chemistry, v. 10, n. 8, p. 2229–2235, 2014. Citation on page 101.

DINIZ, W. J. d. S.; CANDURI, F. Bioinformatics: an overview and its applications. **Genet Mol Res**, v. 16, n. 1, 2017. Citation on page 91.

DONG, J.; YAO, Z.-J.; ZHANG, L.; LUO, F.; LIN, Q.; LU, A.-P.; CHEN, A. F.; CAO, D.-S. Pybiomed: a python library for various molecular representations of chemicals, proteins and dnas and their interactions. **Journal of cheminformatics**, BioMed Central, v. 10, n. 1, p. 16, 2018. Citations on pages 36, 40, 92, 93, and 94.

DORRITY, M. W.; SAUNDERS, L. M.; QUEITSCH, C.; FIELDS, S.; TRAPNELL, C. Dimensionality reduction by umap to visualize physical and genetic interactions. **Nature communications**, Nature Publishing Group, v. 11, n. 1, p. 1–6, 2020. Citation on page 89.

DéRIAN, N.; PHAM, H.-P.; NEHAR-BELAID, D.; TCHITCHEK, N.; KLATZMANN, D.; ERIC, V.; SIX, A. The tsallis generalized entropy enhances the interpretation of transcriptomics datasets. **PLOS ONE**, Public Library of Science, v. 17, n. 4, p. 1–16, 04 2022. Citation on page 79.

EITZINGER, S.; ASIF, A.; WATTERS, K. E.; IAVARONE, A. T.; KNOTT, G. J.; DOUDNA, J. A.; MINHAS, F. Machine learning predicts new anti-CRISPR proteins. **Nucleic Acids Research**, v. 48, n. 9, p. 4698–4708, 04 2020. ISSN 0305-1048. Citation on page 75.

ELSAYAD, A. M.; NASSEF, A. M.; AL-DHAIFALLAH, M. Bayesian optimization of multiclass svm for efficient diagnosis of erythemato-squamous diseases. **Biomedical Signal Processing and Control**, v. 71, p. 103223, 2022. ISSN 1746-8094. Citation on page 115.

EPPENHOF, E. J.; PEÑA-CASTILLO, L. Prioritizing bona fide bacterial small rnas with machine learning classifiers. **PeerJ**, PeerJ Inc., v. 7, p. e6304, 2019. Citations on pages 112 and 113.

FALAGAS, M. E.; PITSOUNI, E. I.; MALIETZIS, G. A.; PAPPAS, G. Comparison of pubmed, scopus, web of science, and google scholar: strengths and weaknesses. **The FASEB journal**, Federation of American Societies for Experimental Biology, v. 22, n. 2, p. 338–342, 2008. Citation on page 30.

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. Citation on page 57.

FEHR, S.; BERENS, S. On the conditional rényi entropy. **IEEE Transactions on Information Theory**, IEEE, v. 60, n. 11, p. 6801–6810, 2014. Citation on page 79.

FLORENTINO, B. R.; SANCHES, N. H.; BONIDIA, R. P.; CARVALHO, A. C. de. Bioprediction: Democratizing machine learning in the study of molecular interactions. In: SBC. **Anais do XX Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2023. p. 525–539. Citation on page 137.

FRAZIER, P. I. A tutorial on bayesian optimization. **arXiv preprint arXiv:1807.02811**, 2018. Citations on pages 109, 114, and 115.

FU, X.; ZHU, W.; CAI, L.; LIAO, B.; PENG, L.; CHEN, Y.; YANG, J. Improved pre-mirnas identification through mutual information of pre-mirna sequences and structures. **Frontiers in genetics**, Frontiers, v. 10, p. 119, 2019. Citation on page 119.

GALLART, A. P.; PULIDO, A. H.; LAGRÁN, I. Anzar Martínez de; SANSEVERINO, W.; CIGLIANO, R. A. Greenc: a wiki-based database of plant lncrnas. **Nucleic acids research**, Oxford University Press, v. 44, n. D1, p. D1161–D1166, 2015. Citations on pages 60 and 61.

GHANNAM, R. B.; TECHTMANN, S. M. Machine learning applications in microbial ecology, human microbiome studies, and environmental monitoring. **Computational and Structural Biotechnology Journal**, Elsevier, 2021. Citations on pages 21, 107, and 108.

____. Machine learning applications in microbial ecology, human microbiome studies, and environmental monitoring. **Computational and Structural Biotechnology Journal**, 2021. ISSN 2001-0370. Available: <http://www.sciencedirect.com/science/article/pii/S2001037021000325>. Citation on page 93.

GLAŽAR, P.; PAPAVASILEIOU, P.; RAJEWSKY, N. circbase: a database for circular rnas. **Rna**, Cold Spring Harbor Lab, v. 20, n. 11, p. 1666–1670, 2014. Citation on page 61.

GOODSTEIN, D. M.; SHU, S.; HOWSON, R.; NEUPANE, R.; HAYES, R. D.; FAZO, J.; MITROS, T.; DIRKS, W.; HELLSTEN, U.; PUTNAM, N. *et al.* Phytozome: a comparative platform for green plant genomics. **Nucleic acids research**, Oxford University Press, v. 40, n. D1, p. D1178–D1186, 2011. Citation on page 60.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **arXiv preprint arXiv:2008.05756**, 2020. Citation on page 117.

GREENER, J. G.; KANDATHIL, S. M.; MOFFAT, L.; JONES, D. T. A guide to machine learning for biologists. **Nature Reviews Molecular Cell Biology**, Nature Publishing Group, p. 1–16, 2021. Citations on pages 21 and 107.

____. A guide to machine learning for biologists. **Nature Reviews Molecular Cell Biology**, Nature Publishing Group, v. 23, n. 1, p. 40–55, 2022. Citation on page 75.

GUO, J.-C.; FANG, S.-S.; WU, Y.; ZHANG, J.-H.; CHEN, Y.; LIU, J.; WU, B.; WU, J.-R.; LI, E.-M.; XU, L.-Y.; SUN, L.; ZHAO, Y. CNIT: a fast and accurate web tool for identifying protein-coding and long non-coding transcripts based on intrinsic sequence composition. **Nucleic Acids Research**, v. 47, n. W1, p. W516–W522, 05 2019. ISSN 0305-1048. Citation on page 57.

GUO, M.; ZOU, Q. Perspectives of bioinformatics in big data era. **Current Genomics**, v. 20, n. 2, p. 79, 2019. Citation on page 55.

GUYON, I.; GUNN, S.; NIKRAVESH, M.; ZADEH, L. A. **Feature extraction: foundations and applications**. [S.l.]: Springer, 2008. Citation on page 93.

HALKO, N.; MARTINSSON, P.-G.; TROPP, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. **SIAM review**, SIAM, v. 53, n. 2, p. 217–288, 2011. Citation on page 82.

HAN, S.; LIANG, Y.; MA, Q.; XU, Y.; ZHANG, Y.; DU, W.; WANG, C.; LI, Y. Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property. **Briefings in Bioinformatics**, 2018. Citations on pages 56, 57, 71, 92, 101, 102, and 103.

HANCOCK, J.; KHOSHGOFTAAR, T. M. Catboost for big data: an interdisciplinary review. **Research Square**, 2020. Citations on pages 24, 62, 111, and 115.

HAQUE, H. F.; RAFSANJANI, M.; ARIFIN, F.; ADILINA, S.; SHATABDA, S. Subfeat: Feature subspacing ensemble classifier for function prediction of dna, rna and protein sequences. **Computational Biology and Chemistry**, Elsevier, v. 92, p. 107489, 2021. Citations on pages 101 and 103.

HARROW, J.; FRANKISH, A.; GONZALEZ, J. M.; TAPANARI, E.; DIEKHANS, M.; KOKOCINSKI, F.; AKEN, B. L.; BARRELL, D.; ZADISSA, A.; SEARLE, S. *et al.* Gencode: the reference human genome annotation for the encode project. **Genome research**, Cold Spring Harbor Lab, v. 22, n. 9, p. 1760–1774, 2012. Citation on page 61.

HASHEMI, F. S. G.; ISMAIL, M. R.; YUSOP, M. R.; HASHEMI, M. S. G.; SHAHRAKI, M. H. N.; RASTEGARI, H.; MIAH, G.; ASLANI, F. Intelligent mining of large-scale bio-data: Bioinformatics applications. **Biotechnology & Biotechnological Equipment**, Taylor & Francis, v. 32, n. 1, p. 10–29, 2018. Citations on pages 55 and 107.

HASHEMI, F. S. G.; ISMAIL, M. R.; YUSOP, M. R.; HASHEMI, M. S. G.; SHAHRAKI, M. H. N.; RASTEGARI, H.; MIAH, G.; ASLANI, F. Intelligent mining of large-scale bio-data: Bioinformatics applications. **Biotechnology & Biotechnological Equipment**, Taylor & Francis, v. 32, n. 1, p. 10–29, 2018. Citation on page 75.

HASTIE, T.; ROSSET, S.; ZHU, J.; ZOU, H. Multi-class adaboost. **Statistics and its Interface**, International Press of Boston, v. 2, n. 3, p. 349–360, 2009. Citation on page 61.

HATCHER, E. L.; ZHDANOV, S. A.; BAO, Y.; BLINKOVA, O.; NAWROCKI, E. P.; OSTAPCHUCK, Y.; SCHäFFER, A. A.; BRISTER, J. R. Virus Variation Resource – improved response to emergent viral outbreaks. **Nucleic Acids Research**, v. 45, n. D1, p. D482–D490, 11 2016. ISSN 0305-1048. Available: <https://doi.org/10.1093/nar/gkw1065>. Citation on page 101.

HE, S.; DOU, L.; LI, X.; ZHANG, Y. Review of bioinformatics in azheimer's disease research. **Computers in Biology and Medicine**, v. 143, p. 105269, 2022. ISSN 0010-4825. Citations on pages 24, 111, and 115.

HE, W.; JU, Y.; ZENG, X.; LIU, X.; ZOU, Q. Sc-ncdnapred: a sequence-based predictor for identifying non-coding dna in saccharomyces cerevisiae. **Frontiers in microbiology**, Frontiers, v. 9, p. 2174, 2018. Citations on pages 112 and 113.

HE, X.; ZHAO, K.; CHU, X. Automl: A survey of the state-of-the-art. **Knowledge-Based Systems**, Elsevier, v. 212, p. 106622, 2021. Citations on pages 108 and 109.

HOANG, T.; YIN, C.; YAU, S. S.-T. Numerical encoding of dna sequences by chaos game representation with application in similarity comparison. **Genomics**, Elsevier, v. 108, n. 3-4, p. 134–142, 2016. Citations on pages 25, 92, 97, and 99.

HOLDEN, T.; SUBRAMANIAM, R.; SULLIVAN, R.; CHEUNG, E.; SCHNEIDER, C.; JR, G. T.; FLAMHOLZ, A.; LIEBERMAN, D.; CHEUNG, T. Atcg nucleotide fluctuation of deinococcus radiodurans radiation genes. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Instruments, Methods, and Missions for Astrobiology X**. [S.l.], 2007. v. 6694, p. 669417. Citation on page 97.

ITO, E. A.; KATAHIRA, I.; VICENTE, F. F. d. R.; PEREIRA, L. F. P.; LOPES, F. M. Basinet—biological sequences network: a case study on coding and non-coding rnas identification. **Nucleic acids research**, 2018. Citations on pages 25, 57, 58, 61, 98, and 99.

IUCHI, H.; MATSUTANI, T.; YAMADA, K.; IWANO, N.; SUMI, S.; HOSODA, S.; ZHAO, S.; FUKUNAGA, T.; HAMADA, M. Representation learning applications in biological sequence analysis. **Computational and Structural Biotechnology Journal**, v. 19, p. 3198–3208, 2021. ISSN 2001-0370. Available: <https://www.sciencedirect.com/science/article/pii/S2001037021002208>. Citation on page 75.

JACOBSEN, E.; LYONS, R. The sliding dft. **IEEE Signal Processing Magazine**, IEEE, v. 20, n. 2, p. 74–80, 2003. Citations on pages 68 and 69.

JEFFREY, H. J. Chaos game representation of gene structure. **Nucleic acids research**, Oxford University Press, v. 18, n. 8, p. 2163–2170, 1990. Citation on page 97.

JING, R.; LI, Y.; XUE, L.; LIU, F.; LI, M.; LUO, J. autobioseqpy: a deep learning tool for the classification of biological sequences. **Journal of Chemical Information and Modeling**, ACS Publications, v. 60, n. 8, p. 3755–3764, 2020. Citations on pages 22, 25, 53, 111, and 118.

JOSHI, G.; JAIN, A.; ADHIKARI, S.; GARG, H.; BHANDARI, M. Fda approved artificial intelligence and machine learning (ai/ml)-enabled medical devices: An updated 2022 landscape. **medRxiv**, Cold Spring Harbor Laboratory Press, p. 2022–12, 2022. Citation on page 26.

KALVARI, I.; NAWROCKI, E. P.; ARGASINSKA, J.; QUINONES-OLVERA, N.; FINN, R. D.; BATEMAN, A.; PETROV, A. I. Non-coding rna analysis using the rfam database. **Current protocols in bioinformatics**, Wiley Online Library, v. 62, n. 1, p. e51, 2018. Citation on page 116.

KALVARI, I.; NAWROCKI, E. P.; ONTIVEROS-PALACIOS, N.; ARGASINSKA, J.; LAMKIEWICZ, K.; MARZ, M.; GRIFFITHS-JONES, S.; TOFFANO-NIOCHE, C.; GAUTHERET, D.; WEINBERG, Z. *et al.* Rfam 14: expanded coverage of metagenomic, viral and microrna families. **Nucleic Acids Research**, Oxford University Press, v. 49, n. D1, p. D192–D200, 2021. Citation on page 116.

KAMALOV, F.; CHERUKURI, A. K.; SULIEMAN, H.; THABTAH, F.; HOSSAIN, A. Machine learning applications for covid-19: a state-of-the-art review. **Data Science for Genomics**, Elsevier, p. 277–289, 2023. Citation on page 21.

KANG, Y.-J.; YANG, D.-C.; KONG, L.; HOU, M.; MENG, Y.-Q.; WEI, L.; GAO, G. 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. Citations on pages 56, 57, 60, 71, and 103.

KARIMI, S.; POHL, S.; SCHOLER, F.; CAVEDON, L.; ZOBEL, J. Boolean versus ranked querying for biomedical systematic reviews. **BMC medical informatics and decision making**, BioMed Central, v. 10, n. 1, p. 58, 2010. Citations on pages 31 and 77.

KARMAKER, S. K.; HASSAN, M. M.; SMITH, M. J.; XU, L.; ZHAI, C.; VEERAMACHANENI, K. Automl to date and beyond: Challenges and opportunities. **ACM Computing Surveys (CSUR)**, ACM New York, NY, v. 54, n. 8, p. 1–36, 2021. Citation on page 22.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, 2017. Citation on page 115.

KEELE, S. *et al.* **Guidelines for performing systematic literature reviews in software engineering**. [S.l.], 2007. Citations on pages 29, 30, 32, 33, and 77.

KHAN, F.; KHAN, M.; IQBAL, N.; KHAN, S.; KHAN, D. M.; KHAN, A.; WEI, D.-Q. Prediction of recombination spots using novel hybrid feature extraction method via deep learning approach. **Frontiers in Genetics**, v. 11, p. 1052, 2020. ISSN 1664-8021. Available: <https://www.frontiersin.org/article/10.3389/fgene.2020.539227>. Citation on page 82.

KHARE, R.; LEAMAN, R.; LU, Z. Accessing biomedical literature in the current information landscape. In: **Biomedical Literature Mining**. [S.l.]: Springer, 2014. p. 11–31. Citation on page 30.

KHATUN, M. S.; HASAN, M. M.; SHOOMBUATONG, W.; KURATA, H. Proin-fuse: improved and robust prediction of proinflammatory peptides by fusing of multiple feature representations. **Journal of Computer-Aided Molecular Design**, Springer, v. 34, n. 12, p. 1229–1236, 2020. Citations on pages 23 and 110.

KHURANA, U.; TURAGA, D.; SAMULOWITZ, H.; PARTHASRATHY, S. Cognito: Automated feature engineering for supervised learning. In: **2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)**. [S.l.: s.n.], 2016. p. 1304–1307. Citations on pages 23 and 110.

KITCHENHAM, B.; BRERETON, O. P.; BUDGEN, D.; TURNER, M.; BAILEY, J.; LINKMAN, S. Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009. Citation on page 29.

____. Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009. Citation on page 77.

KLAPPROTH, C.; SEN, R.; STADLER, P. F.; FINDEISS, S.; FALLMANN, J. Common features in lncrna annotation and classification: A survey. **Non-coding RNA**, MDPI, v. 7, n. 4, p. 77, 2021. Citation on page 81.

KONG, L.; ZHANG, Y.; YE, Z.-Q.; LIU, X.-Q.; ZHAO, S.-Q.; WEI, L.; GAO, G. 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. Citations on pages 57 and 103.

KÖSESOY, İ.; GÖK, M.; ÖZ, C. Proses: A web server for sequence-based protein encoding. **Journal of Computational Biology**, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New . . . , v. 25, n. 10, p. 1120–1122, 2018. Citations on pages 36 and 40.

LE, T. T.; FU, W.; MOORE, J. H. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. **Bioinformatics**, Oxford University Press, v. 36, n. 1, p. 250–256, 2020. Citations on pages 112 and 121.

LESNE, A. Shannon entropy: a rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. **Mathematical Structures in Computer Science**, Cambridge University Press, v. 24, n. 3, 2014. Citation on page 78.

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. Citations on pages 57, 59, and 103.

LI, H.-L.; PANG, Y.-H.; LIU, B. Bioseq-blm: a platform for analyzing dna, rna and protein sequences based on biological language models. **Nucleic acids research**, Oxford University Press, v. 49, n. 22, p. e129–e129, 2021. Citation on page 111.

____. BioSeq-BLM: a platform for analyzing DNA, RNA and protein sequences based on biological language models. **Nucleic Acids Research**, 09 2021. ISSN 0305-1048. Gkab829. Available: <https://doi.org/10.1093/nar/gkab829>. Citation on page 53.

LI, J.; LIU, W. Puzzle of highly pathogenic human coronaviruses (2019-ncov). **Protein & Cell**, Springer, p. 1–4, 2020. Citation on page 56.

LI, J.-Y.; LI, W.-X.; WANG, A.-T.; ZHANG, Y. Mitoflex: an efficient, high-performance toolkit for animal mitogenome assembly, annotation and visualization. **Bioinformatics**, Oxford University Press, v. 37, n. 18, p. 3001–3003, 2021. Citation on page 122.

LI, M.; SI, Y.; YANG, W.; YU, Y. Et-umap integration feature for ecg biometrics using stacking. **Biomedical Signal Processing and Control**, Elsevier, v. 71, p. 103159, 2022. Citation on page 89.

LI, Q.; ZHOU, W.; WANG, D.; WANG, S.; LI, Q. Prediction of anticancer peptides using a low-dimensional feature model. **Frontiers in bioengineering and biotechnology**, Frontiers, v. 8, p. 892, 2020. Citation on page 81.

____. Prediction of anticancer peptides using a low-dimensional feature model. **Frontiers in Bioengineering and Biotechnology**, v. 8, p. 892, 2020. ISSN 2296-4185. Available: <https://www.frontiersin.org/article/10.3389/fbioe.2020.00892>. Citations on pages 101, 102, and 104.

LI, V. R.; ZHANG, Z.; TROYANSKAYA, O. G. CROTON: an automated and variant-aware deep learning framework for predicting CRISPR/Cas9 editing outcomes. **Bioinformatics**, v. 37, p. i342–i348, 07 2021. ISSN 1367-4803. Citations on pages 21 and 107.

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. Citation on page 119.

LI, Z. R.; LIN, H. H.; HAN, L. Y.; JIANG, L.; CHEN, X.; CHEN, Y. Z. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. **Nucleic Acids Research**, v. 34, p. W32–W37, 07 2006. ISSN 0305-1048. Available: <https://doi.org/10.1093/nar/gkl305>. Citations on pages 34, 40, 92, 93, and 94.

LIAW, A.; WIENER, M. *et al.* Classification and regression by randomforest. **R news**, v. 2, n. 3, p. 18–22, 2002. Citation on page 115.

LIN, H.; LIANG, Z.-Y.; TANG, H.; CHEN, W. Identifying sigma70 promoters with novel pseudo nucleotide composition. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 16, n. 4, p. 1316–1321, 2017. Citations on pages 81, 92, 101, 102, and 104.

LINDSAY, M. A.; GRIFFITHS-JONES, S.; VALADKHAN, S.; GUNAWARDANE, L. S. Role of small nuclear rnas in eukaryotic gene expression. **Essays in biochemistry**, Portland Press, v. 54, p. 79–90, 2013. Citation on page 122.

LIU, A. C. The effect of oversampling and undersampling on classifying imbalanced text datasets. **The University of Texas at Austin**, Citeseer, 2004. Citation on page 60.

LIU, B. Bioseq-analysis: a platform for dna, rna and protein sequence analysis based on machine learning approaches. **Briefings in bioinformatics**, Oxford University Press, v. 20, n. 4, p. 1280–1294, 2017. Citations on pages 35, 40, 92, 93, 94, 103, and 111.

LIU, B.; GAO, X.; ZHANG, H. BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches. **Nucleic Acids Research**, v. 47, n. 20, p. e127–e127, 09 2019. ISSN 0305-1048. Available: <https://doi.org/10.1093/nar/gkz740>. Citations on pages 36, 40, and 92.

LIU, B.; LIU, F.; FANG, L.; WANG, X.; CHOU, K.-C. 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. Available: <https://doi.org/10.1093/bioinformatics/btu820>. Citations on pages 35, 40, 92, 93, 94, and 111.

____. reprna: a web server for generating various feature vectors of rna sequences. **Molecular Genetics and Genomics**, Springer, v. 291, n. 1, p. 473–481, 2016. Citations on pages 35, 40, 92, 93, and 94.

LIU, B.; LIU, F.; WANG, X.; CHEN, J.; FANG, L.; CHOU, K.-C. 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. Available: <https://doi.org/10.1093/nar/gkv458>. Citations on pages 21, 23, 35, 40, 92, 93, 94, 107, 110, and 111.

LIU, B.; WU, H.; CHOU, K.-C. *et al.* Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of dna, rna, and protein sequences. **Natural Science**, Scientific Research Publishing, v. 9, n. 04, p. 67, 2017. Citations on pages 35 and 40.

LIU, D.; XU, C.; HE, W.; XU, Z.; FU, W.; ZHANG, L.; YANG, J.; WANG, Z.; LIU, B.; PENG, G. *et al.* Autogenome: an automl tool for genomic research. **Artificial Intelligence in the Life Sciences**, Elsevier, v. 1, p. 100017, 2021. Citations on pages 22, 25, 53, 111, and 118.

LIU, S.; ZHAO, X.; ZHANG, G.; LI, W.; LIU, F.; LIU, S.; ZHANG, W. Predlnc-gfstack: A global sequence feature based on a stacked ensemble learning method for predicting lncrnas from transcripts. **Genes**, Multidisciplinary Digital Publishing Institute, v. 10, n. 9, p. 672, 2019. Citation on page 57.

LIU, Y.; YU, Z.; CHEN, C.; HAN, Y.; YU, B. Prediction of protein crotonylation sites through lightgbm classifier based on smote and elastic net. **Analytical Biochemistry**, v. 609, p. 113903, 2020. ISSN 0003-2697. Citations on pages 24, 111, 115, and 118.

LOONG, S. N. K.; MISHRA, S. K. Unique folding of precursor micrornas: quantitative evidence and implications for de novo identification. **Rna**, Cold Spring Harbor Lab, v. 13, n. 2, p. 170–187, 2007. Citation on page 116.

LOPES, F. M.; OLIVEIRA, E. A. de; CESAR, R. M. Inference of gene regulatory networks from time series by tsallis entropy. **BMC systems biology**, Springer, v. 5, n. 1, p. 61, 2011. Citation on page 76.

LOU, H.; SCHWARTZ, M.; BRUCK, J.; FARNOUD, F. Evolution of k-mer frequencies and entropy in duplication and substitution mutation systems. **IEEE Transactions on Information Theory**, IEEE, 2019. Citations on pages 56, 75, and 107.

LU, J.; SALZBERG, S. L. Skewit: The skew index test for large-scale gc skew analysis of bacterial genomes. **PLoS computational biology**, Public Library of Science San Francisco, CA USA, v. 16, n. 12, p. e1008439, 2020. Citation on page 117.

MA, C.; ZHANG, H. H.; WANG, X. Machine learning for big data analytics in plants. **Trends in Plant Science**, v. 19, n. 12, p. 798 – 808, 2014. ISSN 1360-1385. Citation on page 56.

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. Citations on pages 25, 76, 80, 92, 97, and 99.

MANAVALAN, B.; SHIN, T. H.; LEE, G. Pvp-svm: sequence-based prediction of phage virion proteins using a support vector machine. **Frontiers in microbiology**, Frontiers, v. 9, p. 476, 2018. Citations on pages 81, 92, 100, 101, and 104.

MAPLESON, D.; ACCINELLI, G. G.; KETTLEBOROUGH, G.; WRIGHT, J.; CLAVIJO, B. J. KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. **Bioinformatics**, v. 33, n. 4, p. 574–576, 11 2016. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btw663>. Citation on page 98.

MAROS, M. E.; CAPPER, D.; JONES, D. T.; HOVESTADT, V.; DEIMLING, A. von; PFISTER, S. M.; BENNER, A.; ZUCKNICK, M.; SILL, M. Machine learning workflows to estimate class probabilities for precision cancer diagnostics on dna methylation microarray data. **Nature Protocols**, Nature Publishing Group, p. 1–34, 2020. Citations on pages 56, 75, and 107.

MARTIGNON, L. Information theory. In: SMELSER, N. J.; BALTES, P. B. (Ed.). **International Encyclopedia of the Social & Behavioral Sciences**. Oxford: Pergamon, 2001. p. 7476 – 7480. ISBN 978-0-08-043076-8. Available: <http://www.sciencedirect.com/science/article/pii/B0080430767006082>. Citation on page 78.

MASZCZYK, T.; DUCH, W. Comparison of shannon, renyi and tsallis entropy used in decision trees. In: SPRINGER. **International Conference on Artificial Intelligence and Soft Computing**. [S.l.], 2008. p. 643–651. Citation on page 79.

MCINNES, L.; HEALY, J.; MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. **arXiv preprint arXiv:1802.03426**, 2018. Citation on page 89.

MCINNES, L.; HEALY, J.; SAUL, N.; GROßBERGER, L. Umap: Uniform manifold approximation and projection. **Journal of Open Source Software**, The Open Journal, v. 3, n. 29, p. 861, 2018. Available: <https://doi.org/10.21105/joss.00861>. Citation on page 82.

MENDIZABAL-RUIZ, G.; ROMÁN-GODÍNEZ, I.; TORRES-RAMOS, S.; SALIDO-RUIZ, R. A.; MORALES, J. A. On dna numerical representations for genomic similarity computation. **PloS one**, Public Library of Science, v. 12, n. 3, p. e0173288, 2017. Citations on pages 61, 92, and 97.

MENG, J.; KANG, Q.; CHANG, Z.; LUAN, Y. Plncrna-hdeep: plant long noncoding rna prediction using hybrid deep learning based on two encoding styles. **BMC bioinformatics**, BioMed Central, v. 22, n. 3, p. 1–16, 2021. Citations on pages 101, 103, and 104.

MIN, R. **Machine Learning Approaches to Biological Sequence and Phenotype Data Analysis**. [S.l.]: University of Toronto, 2010. Citations on pages 55 and 56.

MIN, S.; LEE, B.; YOON, S. Deep learning in bioinformatics. **Briefings in Bioinformatics**, v. 18, n. 5, p. 851–869, 07 2016. ISSN 1467-5463. Citation on page 56.

____. Deep learning in bioinformatics. **Briefings in bioinformatics**, Oxford University Press, v. 18, n. 5, p. 851–869, 2017. Citation on page 103.

MITROFANOV, A.; ALKHNBASHI, O. S.; SHMAKOV, S. A.; MAKAROVA, K.; KOONIN, E.; BACKOFEN, R. CRISPRidentify: identification of CRISPR arrays using machine learning approach. **Nucleic Acids Research**, v. 49, n. 4, p. e20–e20, 12 2020. ISSN 0305-1048. Citations on pages 21 and 107.

MUHAMMOD, R.; AHMED, S.; FARID, D. M.; SHATABDA, S.; SHARMA, A.; DEHZANGI, A. 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. Available: <https://doi.org/10.1093/bioinformatics/btz165>. Citations on pages 23, 36, 40, 56, 92, 93, 94, 98, 103, 110, and 111.

NAEEM, S. M.; MABROUK, M. S.; MARZOUK, S. Y.; ELDOSOKY, M. A. A diagnostic genomic signal processing (GSP)-based system for automatic feature analysis and detection of COVID-19. **Briefings in Bioinformatics**, v. 22, n. 2, p. 1197–1205, 08 2020. ISSN 1477-4054. Available: <https://doi.org/10.1093/bib/bbaa170>. Citations on pages 92, 102, and 104.

____. A diagnostic genomic signal processing (gsp)-based system for automatic feature analysis and detection of covid-19. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 2, p. 1197–1205, 2021. Citations on pages 25 and 86.

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. Citations on pages 61 and 96.

NARAYAN, P.; LUDWICZAK, R. L.; GOODWIN, E. C.; ROTTMAN, F. M. Context effects on n 6-adenosine methylation sites in prolactin mrna. **Nucleic acids research**, Oxford University Press, v. 22, n. 3, p. 419–426, 1994. Citation on page 98.

NAWROCKI, E. P.; EDDY, S. R. Infernal 1.1: 100-fold faster rna homology searches. **Bioinformatics**, Oxford University Press, v. 29, n. 22, p. 2933–2935, 2013. Citations on pages 116 and 122.

NEGRI, T. d. C.; ALVES, W. A. L.; BUGATTI, P. H.; SAITO, P. T. M.; DOMINGUES, D. S.; PASCHOAL, A. R. Pattern recognition analysis on long noncoding rnas: a tool for prediction in plants. **Briefings in bioinformatics**, 2018. Citation on page 57.

NG, S. L.; RABHI, F. A.; WHYTE, G.; ZENG, A. Introducing the brewai automl tool. In: SPRINGER. **International Conference on Internet of Things as a Service**. [S.l.], 2021. p. 198–207. Citation on page 22.

NGUYEN, D. D.; CANG, Z.; WEI, G.-W. A review of mathematical representations of biomolecular data. **Physical Chemistry Chemical Physics**, Royal Society of Chemistry, v. 22, n. 8, p. 4343–4367, 2020. Citation on page 92.

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. Available: <https://doi.org/10.1093/bioinformatics/btz432>. Citations on pages 36, 40, 92, 93, 94, and 103.

OFER, D.; LINIAL, M. ProFET: Feature engineering captures high-level protein functions. **Bioinformatics**, v. 31, n. 21, p. 3429–3436, 06 2015. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btv345>. Citations on pages 34, 40, 92, 93, and 94.

PAINULI, D.; BHARDWAJ, S. *et al.* Recent advancement in cancer diagnosis using machine learning and deep learning techniques: A comprehensive review. **Computers in Biology and Medicine**, Elsevier, v. 146, p. 105580, 2022. Citation on page 21.

PALLATHADKA, H.; MUSTAFA, M.; SANCHEZ, D. T.; SAJJA, G. S.; GOUR, S.; NAVED, M. Impact of machine learning on management, healthcare and agriculture. **Materials Today: Proceedings**, Elsevier, v. 80, p. 2803–2806, 2023. Citation 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. Citations on pages 56, 61, 63, and 69.

Parmezan Bonidia, R.; Ponce de Leon Ferreira de Carvalho, A. C.; Rossi Paschoal, A.; Sipoli Sanches, D. 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. Citations on pages 57 and 62.

PAVLOPOULOS, G. A.; SECRIER, M.; MOSCHOPOULOS, C. N.; SOLDATOS, T. G.; KOSSIDA, S.; AERTS, J.; SCHNEIDER, R.; BAGOS, P. G. Using graph theory to analyze biological networks. **BioData Min**, v. 4, n. 1, 2011. ISSN 17560381. Available: <https://pubmed.ncbi.nlm.nih.gov/21527005/>. Citation on page 98.

PHAM, Q.; GAMBLE, A.; HEARN, J.; CAFAZZO, J. A. The need for ethnoracial equity in artificial intelligence for diabetes management: review and recommendations. **Journal of Medical Internet Research**, JMIR Publications Toronto, Canada, v. 23, n. 2, p. e22320, 2021. Citation on page 26.

PIAN, C.; ZHANG, G.; CHEN, Z.; CHEN, Y.; ZHANG, J.; YANG, T.; ZHANG, L. 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. Citation on page 57.

PISIGNANO, G.; LADOMERY, M. Post-transcriptional regulation through long non-coding rnas (lncrnas). **Non-Coding RNA**, v. 7, n. 2, 2021. ISSN 2311-553X. Available: <https://www.mdpi.com/2311-553X/7/2/29>. Citation on page 104.

POPEJOY, A. B.; FULLERTON, S. M. Genomics is failing on diversity. **Nature**, Nature Publishing Group UK London, v. 538, n. 7624, p. 161–164, 2016. Citation on page 26.

PRITIŠANAC, I.; VERNON, R. M.; MOSES, A. M.; KAY, J. D. F. Entropy and information within intrinsically disordered protein regions. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 21, n. 7, p. 662, 2019. Citation on page 75.

PROKHORENKOVA, L.; GUSEV, G.; VOROBEV, A.; DOROGUSH, A. V.; GULIN, A. Catboost: unbiased boosting with categorical features. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2018. p. 6638–6648. Citations on pages 61 and 115.

RAAD, J.; STEGMAYER, G.; MILONE, D. H. Complexity measures of the mature miRNA for improving pre-miRNAs prediction. **Bioinformatics**, v. 36, n. 8, p. 2319–2327, 12 2019. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btz940>. Citation on page 61.

RAJAMANICKAM, S. **Efficient algorithms for sparse singular value decomposition**. [S.l.]: University of Florida, 2009. Citation on page 89.

RAMÍREZ-REYES, A.; HERNÁNDEZ-MONTOYA, A. R.; HERRERA-CORRAL, G.; DOMÍNGUEZ-JIMÉNEZ, I. Determining the entropic index q of tsallis entropy in images through redundancy. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 18, n. 8, p. 299, 2016. Citations on pages 76 and 78.

RAMOS, G. Ai for all. **New Scientist**, v. 252, n. 3363, p. 27, 2021. ISSN 0262-4079. Available: <https://www.sciencedirect.com/science/article/pii/S0262407921021667>. Citation on page 26.

RANDHAWA, G. S.; SOLTYSIAK, M. P.; ROZ, H. E.; SOUZA, C. P. de; HILL, K. A.; KARI, L. Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: Covid-19 case study. **Plos one**, Public Library of Science San Francisco, CA USA, v. 15, n. 4, p. e0232391, 2020. Citations on pages 86, 101, and 107.

RAO, H. B.; ZHU, F.; YANG, G. B.; LI, Z. R.; CHEN, Y. Z. Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. **Nucleic Acids Research**, v. 39, p. W385–W390, 05 2011. ISSN 0305-1048. Available: <https://doi.org/10.1093/nar/gkr284>. Citations on pages 34 and 40.

RÉ, M. A.; AZAD, R. K. Generalization of entropy based divergence measures for symbolic sequence analysis. **PloS one**, Public Library of Science, v. 9, n. 4, 2014. Citations on pages 75 and 79.

REMITA, M. A.; HALIOUI, A.; DAIGLE, B.; KIANI, G.; DIALLO, A. B. *et al.* A machine learning approach for viral genome classification. **BMC bioinformatics**, Springer, v. 18, n. 1, p. 1–11, 2017. Citation on page 82.

RÉNYI, A. *et al.* On measures of entropy and information. In: BERKELEY, CALIFORNIA, USA. **Proceedings of the fourth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1961. v. 1, n. 547-561. Citations on pages 79 and 87.

RUBEIS, G.; DUBBALA, K.; METZLER, I. "democratizing" artificial intelligence in medicine and healthcare: Mapping the uses of an elusive term. **Frontiers in Genetics**, Frontiers, v. 13, p. 902542, 2022. Citations on pages 21 and 133.

RUDIN, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. **Nature machine intelligence**, Nature Publishing Group UK London, v. 1, n. 5, p. 206–215, 2019. Citation on page 26.

SÁ, A. G. de; PINTO, W. J. G.; OLIVEIRA, L. O. V.; PAPPA, G. L. Recipe: a grammar-based framework for automatically evolving classification pipelines. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2017. p. 246–261. Citations on pages 53, 108, 109, 112, and 121.

SAIDI, R.; ARIDHI, S.; NGUIFO, E. M.; MADDOURI, M. 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. Citations on pages 56 and 93.

SANTOS, A.; CASTELO, S.; FELIX, C.; ONO, J. P.; YU, B.; HONG, S. R.; SILVA, C. T.; BERTINI, E.; FREIRE, J. Visus: An interactive system for automatic machine learning model building and curation. In: **Proceedings of the Workshop on Human-In-the-Loop Data Analytics**. [S.l.: s.n.], 2019. p. 1–7. Citations on pages 53 and 109.

SAYERS, E. W.; CAVANAUGH, M.; CLARK, K.; OSTELL, J.; PRUITT, K. D.; KARSCH-MIZRACHI, I. Genbank. **Nucleic acids research**, Oxford University Press, v. 47, n. D1, p. D94–D99, 2019. Citation on page 119.

SCHAPIRE, R. E. Explaining adaboost. In: **Empirical inference**. [S.l.]: Springer, 2013. p. 37–52. Citation on page 115.

SEGER, E.; OVADYA, A.; SIDDARTH, D.; GARFINKEL, B.; DAFOE, A. Democratising ai: Multiple meanings, goals, and methods. In: **Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society**. [S.l.: s.n.], 2023. p. 715–722. Citation on page 21.

SERIZAY, J.; AHRINGER, J. periodicdna: an r/bioconductor package to investigate k-mer periodicity in dna. **F1000Research**, Faculty of 1000 Ltd, v. 10, 2021. Citation on page 92.

SHANNON, C. E. A mathematical theory of communication. **The Bell system technical journal**, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948. Citations on pages 61, 75, and 78.

SHARMA, M. *et al.* Emerging trends of bioinformatics in health informatics. In: **Computational Intelligence in Healthcare**. [S.l.]: Springer, 2021. p. 343–367. Citations on pages 21 and 107.

SHEN, H.-B.; CHOU, K.-C. Pseaac: A flexible web server for generating various kinds of protein pseudo amino acid composition. **Analytical Biochemistry**, v. 373, n. 2, p. 386 – 388, 2008. ISSN 0003-2697. Available: <http://www.sciencedirect.com/science/article/pii/S000326970700663X>. Citations on pages 34, 40, 92, 93, 94, and 111.

SILVA, J. C. F.; TEIXEIRA, R. M.; SILVA, F. F.; BROMMONSCHENKEL, S. H.; FONTES, E. P. Machine learning approaches and their current application in plant molecular biology: A systematic review. **Plant Science**, Elsevier, v. 284, p. 37–47, 2019. Citations on pages 55 and 56.

____. Machine learning approaches and their current application in plant molecular biology: A systematic review. **Plant Science**, v. 284, p. 37–47, 2019. ISSN 0168-9452. Available: <https://www.sciencedirect.com/science/article/pii/S0168945218315802>. Citation on page 75.

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. Citation on page 57.

SINGH, U.; KHEMKA, N.; RAJKUMAR, M. S.; GARG, R.; JAIN, M. 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. Citation on page 57.

SOLTANI-FARD, E.; TAGHVIMI, S.; KICHI, Z. A.; WEBER, C.; SHABANINEJAD, Z.; TAHERI-ANGANEH, M.; KHATAMI, S. H.; MOUSAVI, P.; MOVAHEDPOUR, A.; NATARELLI, L. Insights into the function of regulatory rnas in bacteria and archaea. **International Journal of Translational Medicine**, Multidisciplinary Digital Publishing Institute, v. 1, n. 3, p. 403–423, 2021. Citation on page 122.

SOUTO, M. C. de; ARAUJO, D. S. de; COSTA, I. G.; SOARES, R. G.; LUDERMIR, T. B.; SCHLIEP, A. 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. Citation on page 61.

SOUZA, K. Padovani de; SETUBAL, J. C.; CARVALHO, A. C. Ponce de Leon F. de; OLIVEIRA, G.; CHATEAU, A.; ALVES, R. Machine learning meets genome assembly. **Briefings in Bioinformatics**, v. 20, n. 6, p. 2116–2129, 08 2018. ISSN 1477-4054. Available: <https://doi.org/10.1093/bib/bby072>. Citation on page 91.

STAFFORD, I.; KELLERMANN, M.; MOSSOTTO, E.; BEATTIE, R.; MACARTHUR, B.; ENNIS, S. A systematic review of the applications of artificial intelligence and machine learning in autoimmune diseases. **NPJ digital medicine**, Nature Publishing Group, v. 3, n. 1, p. 1–11, 2020. Citations on pages 13 and 32.

STAV, S.; ATILHO, R. M.; Mirihana Arachchilage, G.; NGUYEN, G.; HIGGS, G.; BREAKER, R. R. Genome-wide discovery of structured noncoding RNAs in bacteria. **BMC Microbiology**, BioMed Central Ltd., v. 19, n. 1, p. 1–18, mar 2019. ISSN 14712180. Available: <https://bmcmicrobiol.biomedcentral.com/articles/10.1186/s12866-019-1433-7>. Citation on page 107.

STAVRIDIS, M.; KORFIATI, A.; SAKELLAROPOULOS, G.; MAVROUDI, S.; THEOFILATOS, K. Non-coding rna sequences identification and classification using a multi-class and multi-label ensemble technique. In: SPRINGER. **IFIP International Conference on Artificial Intelligence Applications and Innovations**. [S.l.], 2018. p. 179–188. Citations on pages 108 and 119.

STEGMAYER, G.; PERSIA, L. E. D.; RUBIOLO, M.; GERARD, M.; PIVIDORI, M.; YONES, C.; BUGNON, L. A.; RODRIGUEZ, T.; RAAD, J.; MILONE, D. H. Predicting novel microrna: a comprehensive comparison of machine learning approaches. **Briefings in bioinformatics**, Oxford University Press, v. 20, n. 5, p. 1607–1620, 2019. Citation on page 61.

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. Citations on pages 56 and 75.

STREIT, D.; SHANMUGAM, T.; GARBELYANSKI, A.; SIMM, S.; SCHLEIFF, E. The existence and localization of nuclear snornas in arabidopsis thaliana revisited. **Plants**, Multidisciplinary Digital Publishing Institute, v. 9, n. 8, p. 1016, 2020. Citation on page 122.

SUN, L.; LIU, H.; ZHANG, L.; MENG, J. 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. Citation on page 57.

SUN, L.; LUO, H.; BU, D.; ZHAO, G.; YU, K.; ZHANG, C.; LIU, Y.; CHEN, R.; ZHAO, Y. 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. Citations on pages 57 and 103.

SUTHAHARAN, S. Machine learning models and algorithms for big data classification. **Integr. Ser. Inf. Syst**, Springer, v. 36, p. 1–12, 2016. Citation on page 51.

SZCZEŚNIAK, M. W.; WANOWSKA, E.; MUKHERJEE, N.; OHLER, U.; MAKAŁOWSKA, I. Towards a deeper annotation of human lncrnas. **Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms**, Elsevier, v. 1863, n. 4, p. 194385, 2020. Citation on page 56.

TANG, B.; PAN, Z.; YIN, K.; KHATEEB, A. Recent advances of deep learning in bioinformatics and computational biology. **Frontiers in Genetics**, v. 10, p. 214, 2019. ISSN 1664-8021. Available: <https://www.frontiersin.org/article/10.3389/fgene.2019.00214>. Citation on page 103.

TASDELEN, A.; SEN, B. A hybrid cnn-lstm model for pre-mirna classification. **Scientific reports**, Nature Publishing Group, v. 11, n. 1, p. 1–9, 2021. Citation on page 119.

THILAGARAJ, M.; RAJASEKARAN, M. P.; KUMAR, N. A. Tsallis entropy: As a new single feature with the least computation time for classification of epileptic seizures. **Cluster Computing**, Springer, v. 22, n. 6, p. 15213–15221, 2019. Citation on page 76.

TOBER, M. Pubmed, sciencedirect, scopus or google scholar–which is the best search engine for an effective literature research in laser medicine? **Medical Laser Application**, Elsevier, v. 26, n. 3, p. 139–144, 2011. Citation on page 30.

TRIPATHI, R.; PATEL, S.; KUMARI, V.; CHAKRABORTY, P.; VARADWAJ, P. K. 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. Citations on pages 57 and 76.

TSALLIS, C. Possible generalization of boltzmann-gibbs statistics. **Journal of statistical physics**, Springer, v. 52, n. 1-2, p. 479–487, 1988. Citations on pages 76 and 79.

____. Nonextensive statistics: theoretical, experimental and computational evidences and connections. **Brazilian Journal of Physics**, SciELO Brasil, v. 29, n. 1, p. 1–35, 1999. Citation on page 79.

TSALLIS, C.; MENDES, R.; PLASTINO, A. R. The role of constraints within generalized nonextensive statistics. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 261, n. 3-4, p. 534–554, 1998. Citations on pages 61, 76, 79, and 98.

TURNER, A. W.; WONG, D.; KHAN, M. D.; DREISBACH, C. N.; PALMORE, M.; MILLER, C. L. Multi-Omics Approaches to Study Long Non-coding RNA Function in Atherosclerosis. **Frontiers in Cardiovascular Medicine**, Frontiers Media S.A., v. 6, p. 9, feb 2019. ISSN 2297055X. Citation on page 107.

TURNER, R.; ERIKSSON, D.; MCCOURT, M.; KIILI, J.; LAAKSONEN, E.; XU, Z.; GUYON, I. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In: PMLR. **NeurIPS 2020 Competition and Demonstration Track**. [S.l.], 2021. p. 3–26. Citation on page 115.

VAMATHEVAN, J.; CLARK, D.; CZODROWSKI, P.; DUNHAM, I.; FERRAN, E.; LEE, G.; LI, B.; MADABHUSHI, A.; SHAH, P.; SPITZER, M. *et al.* Applications of machine learning in drug discovery and development. **Nature reviews Drug discovery**, Nature Publishing Group, v. 18, n. 6, p. 463–477, 2019. Citation on page 75.

VANHORN, K.; ÇOBANOĞLU, M. C. Democratizing ai in biomedical image classification using virtual reality. **Virtual Reality**, Springer, v. 26, n. 1, p. 159–171, 2022. Citations on pages 21 and 133.

VICTORIA, A. H.; MARAGATHAM, G. Automatic tuning of hyperparameters using bayesian optimization. **Evolving Systems**, Springer, v. 12, n. 1, p. 217–223, 2021. Citation on page 115.

VIEIRA, L. M.; GRATIVOL, C.; THIEBAUT, F.; CARVALHO, T. G.; HARDOIM, P. R.; HEMERLY, A.; LIFSCHITZ, S.; FERREIRA, P. C. G.; WALTER, M. E. M. 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. Citation on page 57.

VINGA, S. Information theory applications for biological sequence analysis. **Briefings in bioinformatics**, Oxford University Press, v. 15, n. 3, p. 376–389, 2013. Citations on pages 75 and 80.

VISHNOI, S.; GARG, P.; ARORA, P. Physicochemical n-grams tool: A tool for protein physicochemical descriptor generation via chou's 5-step rule. **Chemical Biology & Drug Design**, Wiley Online Library, v. 95, n. 1, p. 79–86, 2020. Citations on pages 36, 40, and 93.

VOLDERS, P.-J.; HELSENS, K.; WANG, X.; MENTEN, B.; MARTENS, L.; GEVAERT, K.; VANDESOMPELE, J.; MESTDAGH, P. Lncipedia: a database for annotated human lncrna transcript sequences and structures. **Nucleic acids research**, Oxford University Press, v. 41, n. D1, p. D246–D251, 2013. Citation on page 61.

VOLKAMER, A.; RINIKER, S.; NITTINGER, E.; LANINI, J.; GRISONI, F.; EVERTSSON, E.; RODRÍGUEZ-PÉREZ, R.; SCHNEIDER, N. Machine learning for small molecule drug discovery in academia and industry. **Artificial Intelligence in the Life Sciences**, Elsevier, v. 3, p. 100056, 2023. Citation on page 21.

VOPSON, M. M.; ROBSON, S. C. A new method to study genome mutations using the information entropy. **Physica A: Statistical Mechanics and its Applications**, v. 584, p. 126383, 2021. ISSN 0378-4371. Available: <https://www.sciencedirect.com/science/article/pii/S0378437121006567>. Citation on page 75.

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. Citations on pages 61 and 96.

WANG, D.; ZHANG, Y.; ZHAO, Y. Lightgbm: an effective mirna classification method in breast cancer patients. In: **Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics**. [S.l.: s.n.], 2017. p. 7–11. Citation on page 119.

WANG, G.; YIN, H.; LI, B.; YU, C.; WANG, F.; XU, X.; CAO, J.; BAO, Y.; WANG, L.; ABBASI, A. A.; BAJIC, V. B.; MA, L.; ZHANG, Z. Characterization and identification of long non-coding RNAs based on feature relationship. **Bioinformatics**, v. 35, n. 17, p. 2949–2956, 01 2019. ISSN 1367-4803. Citation on page 57.

WANG, J.; YANG, B.; REVOTE, J.; LEIER, A.; MARQUEZ-LAGO, T. T.; WEBB, G.; SONG, J.; CHOU, K.-C.; LITHGOW, T. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. **Bioinformatics**, v. 33, n. 17, p. 2756–2758, 05 2017. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btx302>. Citations on pages 35 and 40.

WANG, L.; PARK, H. J.; DASARI, S.; WANG, S.; KOCHER, J.-P.; LI, W. 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. Citations on pages 57, 98, and 103.

WANG, R.; WANG, Z.; WANG, H.; PANG, Y.; LEE, T.-Y. Characterization and identification of lysine crotonylation sites based on machine learning method on both plant and mammalian. **Scientific reports**, Nature Publishing Group, v. 10, n. 1, p. 1–12, 2020. Citation on page 104.

WARING, J.; LINDVALL, C.; UMETON, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. **Artificial Intelligence in Medicine**, Elsevier, v. 104, p. 101822, 2020. Citations on pages 22, 23, 108, and 110.

WASHIETL, S.; FINDEISS, S.; MÜLLER, S. A.; KALKHOF, S.; Von Bergen, M.; HOFACKER, I. L.; STADLER, P. F.; GOLDMAN, N. RNAcode: robust discrimination of coding and noncoding regions in comparative sequence data. **RNA.**, v. 17, n. 4, p. 578–594, apr 2011. ISSN 13558382. Citation on page 107.

WATKINS, D.; ARYA, D. P. Regulatory roles of small rnas in prokaryotes: Parallels and contrast with eukaryotic mirna. **Non-coding RNA Investig**, v. 3, p. 28, 2019. Citation on page 122.

WEN, J.; LI, S.; LIN, Z.; HU, Y.; HUANG, C. Systematic literature review of machine learning based software development effort estimation models. **Information and Software Technology**, Elsevier, v. 54, n. 1, p. 41–59, 2012. Citation on page 33.

WHANG, S. E.; ROH, Y.; SONG, H.; LEE, J.-G. Data collection and quality challenges in deep learning: A data-centric ai perspective. **The VLDB Journal**, Springer, p. 1–23, 2023. Citation on page 26.

WILKINSON, M. D.; DUMONTIER, M.; AALBERSBERG, I. J.; APPLETON, G.; AXTON, M.; BAAK, A.; BLOMBERG, N.; BOITEN, J.-W.; SANTOS, L. B. da S.; BOURNE, P. E. *et al.* The fair guiding principles for scientific data management and stewardship. **Scientific data**, Nature Publishing Group, v. 3, n. 1, p. 1–9, 2016. Citation on page 26.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE transactions on evolutionary computation**, IEEE, v. 1, n. 1, p. 67–82, 1997. Citations on pages 24 and 111.

XIAO, N.; CAO, D.-S.; ZHU, M.-F.; XU, Q.-S. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. **Bioinformatics**, v. 31, n. 11, p. 1857–1859, 01 2015. ISSN 1367-4803. Available: <https://doi.org/10.1093/bioinformatics/btv042>. Citations on pages 34, 40, 92, 93, and 94.

XIE, J.; ZHANG, L.; XIAO, M. A review of artificial intelligence applications in bacterial genomics. In: IEEE. **2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.], 2020. p. 1870–1876. Citation on page 112.

XU, C.; JACKSON, S. A. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. **Genome Biology**, v. 20, p. 1–4, 2019. ISSN 1474-760X. Available: <https://doi.org/10.1186/s13059-019-1689-0>. Citation on page 56.

YAMANO, T. Information theory based on nonadditive information content. **Physical Review E**, APS, v. 63, n. 4, p. 046105, 2001. Citations on pages 76, 79, and 97.

YAN, S.; XU, D.; ZHANG, B.; ZHANG, H.-J.; YANG, Q.; LIN, S. Graph embedding and extensions: A general framework for dimensionality reduction. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 29, n. 1, p. 40–51, 2006. Citation on page 51.

YANG, C.; YANG, L.; ZHOU, M.; XIE, H.; ZHANG, C.; WANG, M. D.; ZHU, H. Lncadeep: An ab initio lncrna identification and functional annotation tool based on deep learning. **Bioinformatics**, 2018. Citation on page 103.

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. Citations on pages 97 and 99.

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. Citations on pages 61 and 97.

ZHA, D.; BHAT, Z. P.; LAI, K.-H.; YANG, F.; HU, X. Data-centric ai: Perspectives and challenges. **arXiv preprint arXiv:2301.04819**, 2023. Citation on page 26.

ZHANG, G.; DENG, Y.; LIU, Q.; YE, B.; DAI, Z.; CHEN, Y.; DAI, X. Identifying circular rna and predicting its regulatory interactions by machine learning. **Frontiers in genetics**, Frontiers, v. 11, p. 655, 2020. Citations on pages 61 and 69.

ZHANG, J.; ZHANG, Z.-m. Ethics and governance of trustworthy medical artificial intelligence. **BMC Medical Informatics and Decision Making**, BioMed Central, v. 23, n. 1, p. 1–15, 2023. Citation on page 26.

ZHANG, P.; WU, W.; CHEN, Q.; CHEN, M. Non-coding rnas and their integrated networks. **Journal of integrative bioinformatics**, De Gruyter, v. 16, n. 3, 2019. Citation on page 119.

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. Citations on pages 61 and 96.

ZHANG, Y.; JIA, C.; FULLWOOD, M. J.; KWOH, C. K. DeepCPP: a deep neural network based on nucleotide bias information and minimum distribution similarity feature selection for RNA coding potential prediction. **Briefings in Bioinformatics**, 03 2020. ISSN 1477-4054. Citation on page 57.

ZHANG, Y.; WU, L. Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach. **Entropy**, Molecular Diversity Preservation International, v. 13, n. 4, p. 841–859, 2011. Citation on page 79.

ZHANG, Y.; YU, S.; XIE, R.; LI, J.; LEIER, A.; MARQUEZ-LAGO, T. T.; AKUTSU, T.; SMITH, A. I.; GE, Z.; WANG, J. *et al.* Pengaroo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. **Bioinformatics**, Oxford University Press, v. 36, n. 3, p. 704–712, 2020. Citations on pages 92, 99, 100, 101, and 104.

ZHANG, Z.-Y.; YANG, Y.-H.; DING, H.; WANG, D.; CHEN, W.; LIN, H. Design powerful predictor for mrna subcellular location prediction in homo sapiens. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 1, p. 526–535, 2021. Citations on pages 23 and 93.

Zhao, Y.; He, N.; Chen, Z.; Li, L. Identification of protein lysine crotonylation sites by a deep learning framework with convolutional neural networks. **IEEE Access**, v. 8, p. 14244–14252, 2020. Citations on pages 101, 102, and 104.

ZIHNI, E.; MADAI, V. I.; LIVNE, M.; GALINOVIC, I.; KHALIL, A. A.; FIEBACH, J. B.; FREY, D. Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome. **Plos one**, Public Library of Science San Francisco, CA USA, v. 15, n. 4, p. e0231166, 2020. Citation on page 62.

ZÖLLER, M.-A.; HUBER, M. F. Benchmark and survey of automated machine learning frameworks. **Journal of artificial intelligence research**, v. 70, p. 409–472, 2021. Citations on pages 120, 121, and 122.

ICMC USP

SÃO CARLOS