Universidade de Aveiro
2021

Alexandre Emanuel
Monteiro Lourenço

# Reconstrução e classificação de sequências de ADN desconhecidas

# Reconstruction and classification of unknown DNA sequences

Universidade de Aveiro
2021

**Alexandre Emanuel Monteiro Lourenço**

# Reconstrução e classificação de sequências de ADN desconhecidas

# Reconstruction and classification of unknown DNA sequences

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Diogo Rodrigo Marques Pratas, Investigador Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Armando José Formoso de Pinho, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

**o júri / the jury**

| | |
|---|---|
| presidente / president | **Doutor Joaquim João Estrela Ribeiro Silvestre Madeira**<br>Professor Auxiliar, Universidade de Aveiro |
| vogais / examiners committee | **Doutora Susana de Almeida Mendes Vinga Martins**<br>Professora Associada, Instituto Superior Técnico, Universidade de Lisboa |
| | **Doutor Diogo Rodrigo Marques Pratas**<br>Investigador Auxiliar, Universidade de Aveiro (orientador) |

**agradecimentos / acknowledgements**

Quero expressar o meu mais sincero agradecimento ao Doutor Diogo Pratas e Doutor Armando J. Pinho pela oportunidade, suporte, e disponibilidade no desenvolvimento desta dissertação.

Gostava de agradecer especialmente ao Doutor Diogo Pratas que no decorrer deste desafio superou qualquer expectativa que pudesse ser imaginável. Fico muito grato por todos os ensinamentos e mentalidades transmitidas, tanto ao nível académico/profissional como ao nível pessoal. Tomou um papel fulcral durante toda a extensão deste ano, sempre com uma palavra amiga e incentivadora.

Ao Jorge M. Silva e ao João R. Almeida que contribuíram ativamente para a elevação da qualidade da solução aqui desenvolvida.

À minha familia, especialmente ao meu pai, à minha mãe, à minha irmã Camila, e aos meus avós, por todos os momentos em que estiveram sempre lá para mim. Foram muitos os momentos de indecisões e incertezas ao longo deste percurso e sem o apoio e ajuda deles nada disto teria sido possível.

Quero deixar também um agradecimento muito especial à Fátima Barbosa pela enorme paciência que teve comigo e por todo o suporte e motivação que me deu durante todo este processo.

Agradeço por fim a todos aqueles, amigos e colegas, que de alguma maneira influenciaram positivamente este resultado.

Fluffle.

**Palavras Chave** Classificação de Sequências, Inteligência Artificial, Compressão de Dados, Conjunto de Preditores, Metagenómica

**Resumo**

Os contínuos avanços em tecnologias de sequenciação de ADN e técnicas em metagenómica requerem metodologias de reconstrução confiáveis e de classificação precisas para o aumento da diversidade do repositório natural, contribuindo entretanto para a descrição e organização dos organismos. No entanto, após a sequenciação e a montagem *de-novo*, um dos desafios mais complexos advém das sequências de ADN que não correspondem ou se assemelham a qualquer sequência biológica da literatura. São três as principais razões que contribuem para essa exceção: uma irregularidade emergiu no processo de reconstrução, a sequência do organismo é altamente dissimilar dos organismos da literatura, ou um novo e diferente organismo foi reconstruído. A incapacidade de classificar com eficiência essas sequências desconhecidas aumenta a incerteza da constituição da amostra e desperdiça a oportunidade de descobrir novas espécies, uma vez que muitas vezes são descartadas.

Neste contexto, o principal objetivo desta tese é fornecer uma solução computacional eficiente para resolver este desafio com base em um conjunto de especialistas, nomeadamente preditores baseados em compressão, a distribuição de conteúdo de sequência e comprimentos de sequência normalizados. O método usa sequências de ADN e de aminoácidos e fornece classificação eficiente além das comparações referenciais padrão. Excepcionalmente, ele classifica as sequências de ADN sem recorrer diretamente a genomas de referência, mas sim às características que as sequências biológicas da espécie compartilham. Especificamente, ele usa apenas recursos extraídos individualmente de cada genoma sem usar comparações de sequência. Além disso, o pipeline é totalmente automático e permite a reconstrução sem referência de genomas a partir de *reads* FASTQ com a garantia adicional de armazenamento seguro de informações sensíveis.

O RFSC é então um pipeline de classificação de aprendizagem automática que se baseia em um conjunto de especialistas para fornecer classificação eficiente em contextos metagenómicos. Este pipeline foi aplicado em dados sintéticos e reais, alcançando em ambos resultados precisos e exatos que, no momento do desenvolvimento desta dissertação, não foram relatados na literatura. Especificamente, esta ferramenta desenvolvida, alcançou uma precisão de aproximadamente 97% na classificação de domínio/tipo.

**Keywords**

Sequence Classification, Machine Learning, Data Compression, Experts Ensemble, Metagenomics

**Abstract**

The continuous advances in DNA sequencing technologies and techniques in metagenomics require reliable reconstruction and accurate classification methodologies for the diversity increase of the natural repository while contributing to the organisms' description and organization. However, after sequencing and *de-novo* assembly, one of the highest complex challenges comes from the DNA sequences that do not match or resemble any biological sequence from the literature. Three main reasons contribute to this exception: the organism sequence presents high divergence according to the known organisms from the literature, an irregularity has been created in the reconstruction process, or a new organism has been sequenced. The inability to efficiently classify these unknown sequences increases the sample constitution's uncertainty and becomes a wasted opportunity to discover new species since they are often discarded.

In this context, the main objective of this thesis is the development and validation of a tool that provides an efficient computational solution to solve these three challenges based on an ensemble of experts, namely compression-based predictors, the distribution of sequence content, and normalized sequence lengths. The method uses both DNA and amino acid sequences and provides efficient classification beyond standard referential comparisons. Unusually, it classifies DNA sequences without resorting directly to the reference genomes but rather to features that the species biological sequences share. Specifically, it only makes use of features extracted individually from each genome without using sequence comparisons.

RFSC was then created as a machine learning classification pipeline that relies on an ensemble of experts to provide efficient classification in metagenomic contexts. This pipeline was tested in synthetic and real data, both achieving precise and accurate results that, at the time of the development of this thesis, have not been reported in the state-of-the-art. Specifically, it has achieved an accuracy of approximately 97% in the domain/type classification.

# Contents

# List of Figures

# List of Tables

# Acronyms

**A** Adenine. 4, 13

**AA** Amino Acid. 34–36

**AES** Advanced Encryption Standard. 21

**ANN** Artificial Neural Networks. 17

**BAM** Binary Alignment Map. 21

**BDM** Block Decomposition Method. 46

**C** Cytosine. 4, 13, 26

**CNN** Convolutional Neural Network. 17

**DDBJ** DNA Data Bank of Japan. 8

**DL** Deep Learning. 17

**DNA** Deoxyribonucleic Acid. i, iii, 1–7, 9, 10, 13, 15, 18, 19, 21, 23, 26–28, 33–36, 44, 45

**DNN** Deep Neural Networks. 17

**EMBL** European Molecular Biology Laboratory. 8

**ESA** European Space Agency. 8

**FN** False Negative. 41

**FP** False Positive. 41

**G** Guanine. 4, 13, 26

**GC** Guanine-Cytosine. iii, iv, 2, 6, 19, 23, 24, 26, 28, 29, 34–36, 43, 45

**GNB** Gaussian Naive Bayes. ii, v, 15, 16, 30, 35–37, 42, 44

**GWAS** Genome-Wide Association Studies. 18

**HGP** Human Genome Project. 6

**ISS** International Space Station. 8

**KNN** K-Nearest Neighbor. ii, v, 15, 16, 30, 31, 36, 40, 42–44

**LSTM** Long Short Term Memory Network. 17

**MLP** Multilayer Perceptrons. 17

**NB** Naive Bayes. 15, 16

**NC** Normalized Compression. iii, iv, 25, 26, 28, 29

**NCBI** National Center for Biotechnology Information. v, 8, 22, 24, 33–35, 37, 38, 40, 44, 45

**NGS** Next-Generation Sequencing. 1, 6

**NN** Neural Networks. 17

**ORF** Open Reading Frames. 23, 33

**PE** Paired-End. 10, 22, 33, 35

**RFSC** Reference-Free Sequence Classification Tool. iii, v, 1, 2, 5, 19–24, 30, 32, 37, 38, 41–43, 45

**RNA** Ribonucleic Acid. 4

**RNN** Recurrent Neural Network. 17

**SAM** Sequence Alignment Map. 10, 21

**SE** Single-End. 10, 22, 35

**SNP** Single Nucleotide Polymorphisms. 42

**SVM** Support Vector Machine. 15, 16

**T** Thymine. 4, 13

**TN** True Negative. 41

**TP** True Positive. 41

**U** Uracil. 4

**WGS** Whole Genome Sequencing. 5

**XGB** eXtreme Gradient Boosting. 32, 36, 37

# Chapter 1

# Introduction

## 1.1 Motivation

Metagenomics analyses are increasingly gaining importance in clinical, forensic, and exobiology fields [8, 9, 10, 11, 12]. One of the biggest drivers responsible for this growth was the emergence of next-generation sequencing technologies (NGS), which allow applications from gene expression quantification to genotyping and genome reconstruction [13]. NGS differentiates itself from traditional methods by introducing multiplex and analytical resolution capabilities, thus making it a more time and cost-efficient approach for fast screening [14, 15].

For the classification of new organisms in metagenomic samples, there is the need to follow well-defined laboratory and computation steps, namely from sequencing, passing by trimming and filtering, to *de-novo* assembly. This process can enter a loop between computation and laboratory through cloning and enrichment of specific reconstructed regions for achieving higher quality and completeness [16].

However, after sequence reconstruction, sometimes the results can be inconclusive when using referential comparison methods, specifically when irregularities are created in the reconstruction process, the divergence between the sequences of known organisms and the reconstructed sequence is too high, or a new organism has been sequenced [17, 18].

The vast majority of the classification pipelines available adopt a referential comparison method (e.g., Ganon [19], Fastv [20], VirTect [21], VIcaller [22], ViPR [23], GenomeDetective [24], RAST [25], MEGAN [26], MGS-Fast [27], MetaPhlAn [28]) where the reconstructed sequence is compared to a set of references present in a database [29]. This approach, however, becomes a disadvantage when faced with problems of the magnitude of the ones presented above.

The constant growth of the reference databases (e.g., GenBank [30], RefSeq [31], DDBJ [32], INSC [33], MIPS [34], MG-RAST Database [35]) and the increase in the size of sequencing datasets has also contributed to strategy changes regarding the comparison between each read with all sequences presented in those databases since the alignment used by some programs such as BLAST [36] becomes increasingly computationally expensive [29]. With this in mind, reference-based approaches, such as mapping *k-mers* (e.g. Kraken2 [37], centrifuge [38]), compression-based mapping (e.g. FALCON-meta [39]), or protein sequence alignments (e.g. Kaiju [40] and GOTTCHA [41]), have successfully emerged [29, 42, 43].

However, if the sequence has an extremely high level of dissimilarity or singularity, how can these reference-based programs discover and classify? How to distinguish the biological organism sequence from a sequencing or assembly exception? This challenge is an obvious limiting factor for extending the natural biological repository and discovering new pathogens.

## 1.2 Objectives

This thesis proposes RFSC, an alignment-free metagenomic classification tool, to fill the gaps resulting from referential comparison pipelines. Specifically, RFSC is a pipeline for the reconstruction and classification of DNA sequences without resorting directly to the sequence of the reference genomes. Instead, it utilizes an ensemble of five predictors, namely compression-based predictors and simple property characteristics, for probabilistic classification of reconstructed unknown DNA sequences. Specifically, the experts used are the individual compression proportion (or entropy) of the genome (normalized compression for DNA sequences) and proteome sequence (normalized compression for amino acid sequences), GC-content distribution [44], and normalized sequence lengths for the genome and proteome sequences [45].

It is counter-intuitive that knowing only how much the sequences from a genome and proteome can be compressed (represented by a single floating-point value between zero and one), their sequence length, and percentage of Guanine and Cytosine, it is possible to classify an organism. For example, if it is a virus, bacteria, archaea, fungi, protozoa, plant, or specific types of species sub-sequences, such as mitochondrial or plastid. This thesis aims at showing that it is possible, and provides a method to automatically allow this classification with very high accuracy. Another objective of this thesis is to perform a balanced and fair benchmark to the reconstruction and classification system. This benchmark includes the search for an accurate classifier according to the five predictors in use. For the purpose, synthetic and real DNA sequences are used.

## 1.3 Dissertation Structure

In this thesis, the topic of bioinformatics will first be introduced, specifically its importance and applicability in the most diverse professional and scientific areas, moving on to a state-of-the-art analysis, considering the different existing paradigms, solutions and tools available. Next, the various themes that support the needs of developing a reference-free sequence classification tool will be addressed through the presentation of some of the key areas in which this tool could have a positive impact.

Then, the entire methodology added to this project will be presented, from the implemented architecture, always taking into account preferential factors of privacy, storage and data preparation, for handling and processing the data.

In the last part of the pipeline, referring to data extraction and processing, the used predictors will be introduced and explained, as well as all the implemented machine learning-based classification models.

Finally, a detailed analysis of the results will be carried out and conclusions will be inferred regarding the performance of the method and its implementation.

# Chapter 2

# Background

## 2.1 Bioinformatics

In recent years, many areas of study such as Bioinformatics, Biotechnology, Computational Biology and Biochemistry have gained importance in order to respond to new challenges arising from different fields [46].

In this thesis, the attention will be focused on the field of Bioinformatics that has been at the center of a massive scientific evolution in our days, influencing several fields of study and investigation, thus introducing new paradigms and dogmas.

This interdisciplinary field is supported by the coexistence of several research areas, including Biology, Medicine, Computer Science, Maths and Physics, thus allowing the capture and analysis of biological data [47], most often DNA and amino acid sequences.

In Bioinformatics, these DNA sequences are computationally treated, resulting from outputs of biochemical and computational methods [4], through the application of sequencing and assemble methods. These sequences, however, may not correspond exactly to the chemical compounds or their order due to multiple factors, including low-quality sequencing factors, contamination [4, 48] and unknown factors [4, 49]. The determination of the nucleic compounds and their respective order has been achieved with the Sanger sequencing method [50].

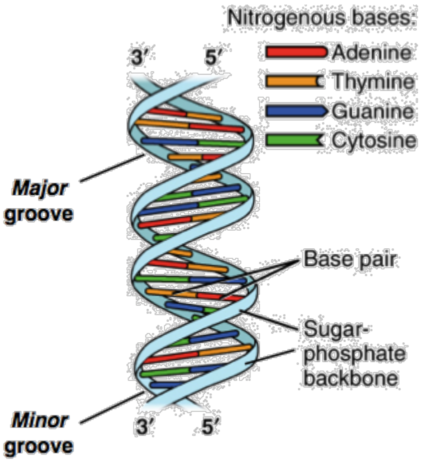


Figure 2.1: Double helix DNA representation. Source: Khan Academy.

Being the DNA and Amino Acid sequences fundamental in the bioinformatics environment, there is the need to extend this topic in order to understand what it is and where it came from.

The DNA present in a genome is a set of molecules that contains the unique genetic code of each organism, being each DNA molecule made of two twisting double-strands [51]. Each strand is constituted by four chemical units, namely Adenine (A), Thymine (T), Guanine (G) and Cytosine (C), the nucleotide bases. A representation of this structure can be found in Figure 2.1.

There are, however, exceptions since, for example in the case of viruses, both the DNA and the RNA molecule can be single-stranded, existing even cases in which both can be verified at the same time.

Approaching this issue from a different perspective, and making use of some of the concepts used in the last paragraph, the concept of nucleotide arises. Nucleotides are characterized by being the basic building block of nucleic acids, such as DNA or RNA, being composed of a nitrogenous base, a sugar, and a phosphate group. The nitrogenous base present in a nucleotide may be any of the aforementioned in the case of DNA or RNA, with the exception, in the latter case, of the Thymine (T), which is replaced in the transcription of DNA to RNA by Uracil (U) in order to be complementary to adenine.

The constitution of these organic molecules is represented in Figure 2.2.

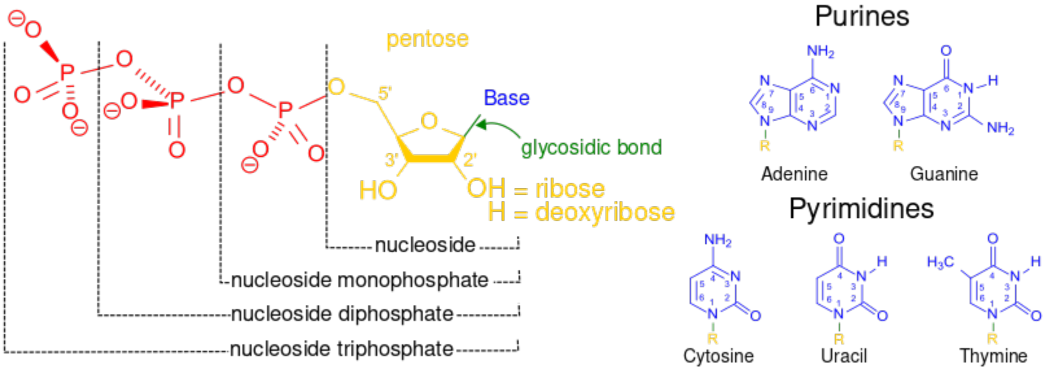


Figure 2.2: Nucleotide structural elements. Source: Wikipedia.

On the other hand, the codons (sets of three nucleotides) present in the DNA, when encoded, give rise to amino acids, existing however some directives that control this conversion. In addition to the fact that three nucleotides encode an amino acid, the code is nonoverlapping [52]. Therefore, from the DNA sequence it is possible to obtain the amino acid sequences, however, from the amino acid sequences one can only infer the possible DNA sequences.

Therefore, it is possible to conclude in this matter that a set of bases form DNA while a set of amino acids form a protein. In order to better understand the information contained in DNA and Proteins, it is necessary to deepen the studies both in genomics and proteomics fields, including the metagenomics field.

### 2.1.1 Metagenomics

Metagenomics is the study of the genomes present in samples directly extracted from the environment. In other words, concerns the study of collections of genomes from a mixed community of organisms, thus avoiding resorting to a single-isolate approach, which does not need to separate individual bacterial clones from complex microbial mixtures [53].

Therefore, and through the assignment of DNA fragments (*reads*, *contigs*, or *scaffolds*) to certain appropriate bioinformatic tools or pipelines, they will, based on different types of analysis, return a collection of genome designations or taxonomies that are present in those DNA samples [53].

Some of the techniques used in the WGS analysis, that is, the analysis of the entire genomic DNA sequence of a genome at a single time, are based on the compositions of nucleotide sequences, on the analysis of protein-coding open reading frames, among several others that will be addressed in the context of the RFSC architecture [1, 53, 54].

In order to provide an overview of the steps to be taken in a metagenomic analysis, an image that briefly illustrates a possible metagenomic workflow is provided in Figure 2.3, in which the five major steps in a typical shotgun metagenomics study are considered, namely the study design and experimental protocol (Experimental Pipeline), computational pre-processing (Preprocessing), sequence analysis, post-processing and validation. The shotgun sequencing method is an evolution of the Sanger method that adds capability to process larger genomes using substantial lower time, and introduces the notion of depth which is related with the average number of sequenced reads to cover a specific genome region [55]. For example, this technology enabled the sequencing of the human genome.

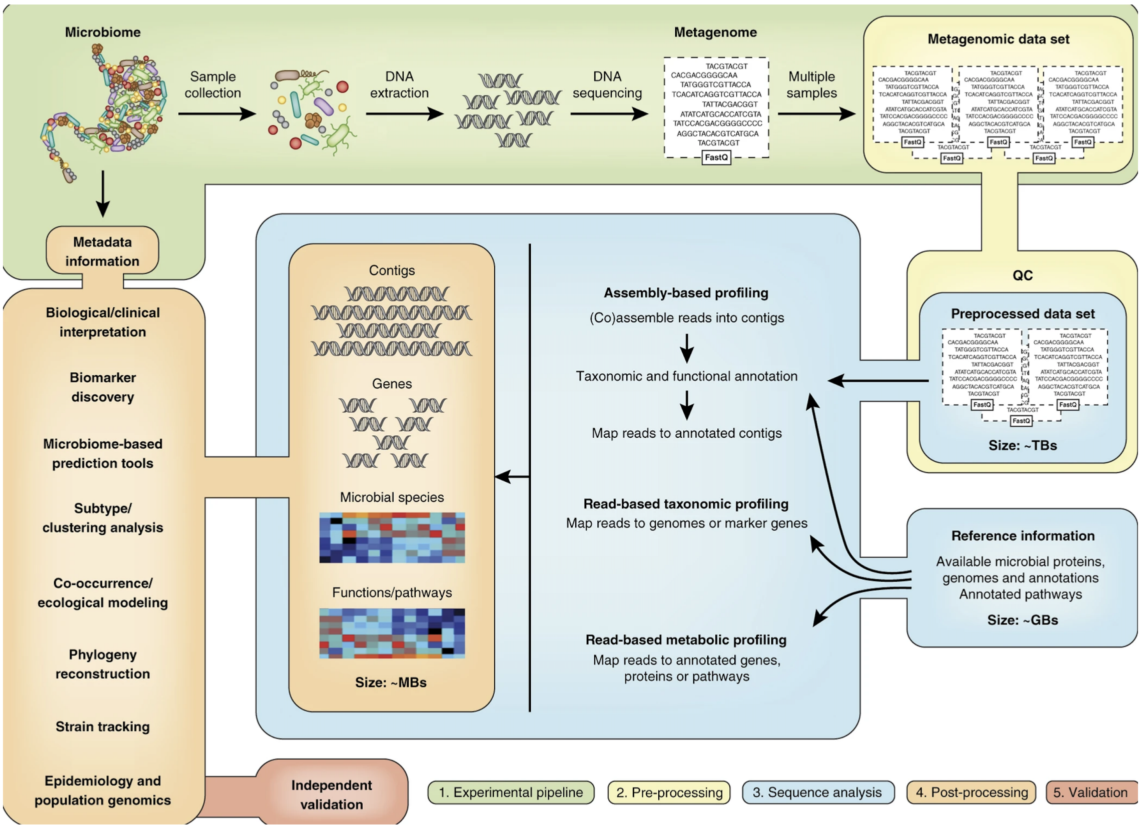


Figure 2.3: Brief representation of a metagenomics workflow. Source: Shotgun metagenomics, from sampling to analysis [1].

This workflow is thus initiated by a *Study design and experimental protocol* and passed directly to a set of steps responsible for quality control in which there is the removal of sequencing adaptors, sequence trimming and removal of sequencing duplicates (*Computational pre-processing*). In the next step, the *Sequence analysis*, a combination of *read-based* and *assembly-based* approaches are performed in order to enable a better analysis of sequences. Arriving at the *Post-processing* step, multiple statistical algorithms are implemented in order to interpret the information obtained. Finally, in *Validation*, it is possible to obtain conclusions about the biological data [1], namely at genomic and proteomic levels.

### 2.1.2 Genomics

In Bioinformatics, genome analysis takes a central position, as it provides a very important amount of data in understanding the instructions provided by the DNA at the genome sequence level.

One of the biggest milestones when it comes to genomic analysis concerns the first sequencing of the human genome back in 2000, when in the United States of America, under the direction of the National Institutes of Health and the U.S. Department of Energy, the Human Genome Project (HGP) successfully, and for the first time in human history, was sequenced close to 90% of the human genome [56]. This achievement made an immense contribution in the most varied areas, namely helping to better understand the evolution of the human, understanding hereditary pathologies, the contribution of environmental factors in human adaptation, the causation of several disease [56, 57], in nutrition researches [58], among many others. The genomics area as witness three main sequencing phases, namely the Sanger sequencing, the shotgun sequencing, and the Next-Generation sequencing (NGS).

The first two major sequencing technologies have already been mentioned while the NGS is the current technology used in most of the cases, mainly for large genome sequencing such as animals or plants. This technology offers a high-throughput while decreasing substantially the sequencing cost [59]. However, it also requires the development and constant improvement of dedicated and efficient pipelines for proper and accurate biological conclusions.

Genomics concerns the integral study of an organism's genome, being the genomic analysis responsible for the analysis of DNA sequences, identifying and studying their characteristics such as structural variation, sequence length, GC Content [45], among others.

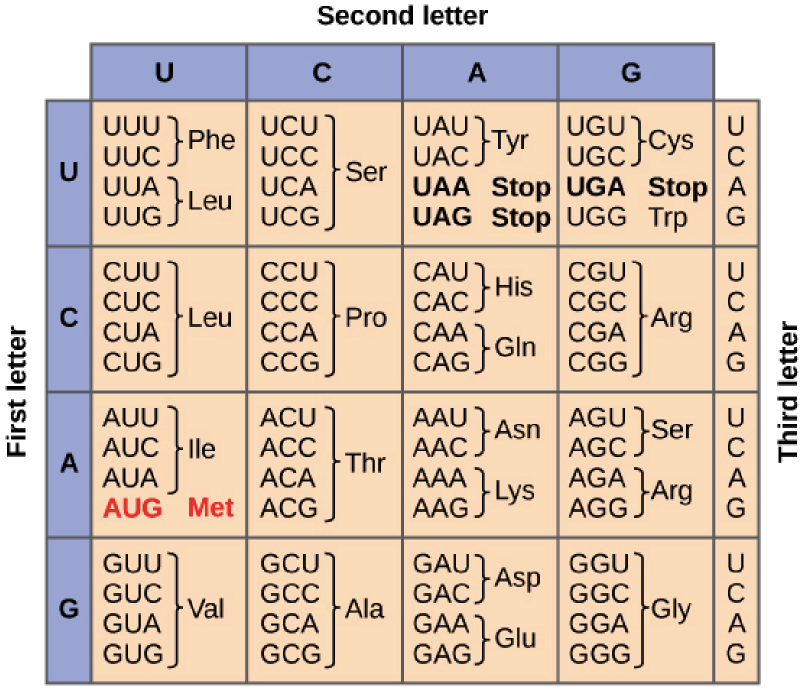
Genomics has grown substantially in the past years thanks to the multiple opportunities and applications it offers in the most varied areas, with special attention to the fields of evolutionary study and clinical applications, such as the study of complex diseases [60], gene therapy [61], and genome editing [62].

### 2.1.3 Proteomics

Proteomics is the large-scale study of proteins. Proteomics is another of the *omics* technologies, where the proteomic-based technologies emerge as a complement to genomic-based technologies, which in turn focus on the study of proteomes, that is, sets of proteins encoded by the genome present in cells, tissues or organisms [63, 64].

Proteomics thus takes on a dominant status in the study of genetic sequences alongside genomics, and can be considered “the tools that make living machines work” [64].

A protein is made of sequences of amino acids. Each amino acid is set according to triplets of DNA bases. For synchronizing the phase of the triplets, specific DNA triples provide the initial start, known as the starting codons. On the other hand, to end a protein sequence, one of the stop codons must be reached. Figure 2.4 shows the correspondence table between DNA triplets (three nucleotides) and amino acids.



Second letter

| First letter | U | C | A | G | Third letter |
|---|---|---|---|---|---|
| U | UUU Phe | UCU Ser | UAU Tyr | UGU Cys | U |
| | UUC | UCC | UAC | UGC | C |
| | UUA Leu | UCA | **UAA Stop** | **UGA Stop** | A |
| | UUG | UCG | **UAG Stop** | UGG Trp | G |
| C | CUU Leu | CCU Pro | CAU His | CGU Arg | U |
| | CUC | CCC | CAC | CGC | C |
| | CUA | CCA | CAA Gln | CGA | A |
| | CUG | CCG | CAG | CGG | G |
| A | AUU Ile | ACU Thr | AAU Asn | AGU Ser | U |
| | AUC | ACC | AAC | AGC | C |
| | AUA | ACA | AAA Lys | AGA Arg | A |
| | **AUG Met** | ACG | AAG | AGG | G |
| G | GUU Val | GCU Ala | GAU Asp | GGU Gly | U |
| | GUC | GCC | GAC | GGC | C |
| | GUA | GCA | GAA Glu | GGA | A |
| | GUG | GCG | GAG | GGG | G |

Figure 2.4: Correspondence table between DNA triplets for amino acids. Source: The Genetic Code by OpenStax College, Biology.

In proteomes, unlike the genetic code where the sequences are made up of sets of four nucleotides, the proteins can be built from twenty different amino acids in their alphabet. In addition to these amino acids, post-translational modifications can inflict protein modifications through the addition of other chemical constituents such as sugars, fats, or phosphates [64].

Proteomics, as all areas at the heart of bioinformatics, have been in great demand and have evolved substantially in recent years. Its capabilities expand into many sectors, with some of its most outstanding applications being the discovery of new protein markers for diagnostic purposes with the aim of developing new vaccines and drugs [65], understanding pathogenicity mechanisms [63], tumor classification [66], cancer research [67], among others.

## 2.2 Biological Sequence Reconstruction and Classification

As previously mentioned, genomes are possible to be found in all organisms, having all its genetic material and being very rich in information.

One of the most powerful dataset for biomedical research contains the sequencing of the human genome, which has between 20,000 to 25,000 genes that each one of them encodes millions of proteins [68]. Analysing and understanding this information can be used to greatly benefit human health [68]. The protection of this intellectual property has proven to be essential in several professional fields, such as biotechnology, pharmaceutical [68], evolutionary, among many others, having a noble objective of supporting the biological discovery, as well as the medical research and the conservation of biodiversity [69].

The study of environmental genomics also manages to bring together a panoply of multiple areas of interest, which, through multiple nucleotide, proteomic, metagenomic, transcriptomic and metatranscriptomic technologies, allows the extraction of a large amount of information regarding the taxonomy (of current and fossil organisms), the phylogeny, the evolution and the adaptation of organisms taking into account the environmental conditions surrounding them [70].

Nowadays, there are many large repositories online, where large quantities of sequences from the most diverse organisms can be accessed, to be used in various research projects [71]. Some of these databases are NCBI's GenBank [30] in the U.S., the EMBL [72] in Europe, and the DDBJ [32] in Japan. The existence of those databases shows the substantial work related to the reconstruction of the genomes provided in the last years. However, the complexity related to both the sequencing and the reconstruction depends on the context and area where the sample has been extracted. In the following subsections, several areas and contexts are described.

### 2.2.1 Exobiology

Exobiology is a recent interdisciplinary scientific field that has gain momentum with the increase of space exploration missions, that seeks to study the origin and evolution of life in the universe [73].

Over the past few years, several studies have been conducted in the area of exobiology, such as the ESA Exobiology Team Study from 1997-1998, in which they focused their attention on the study of exobiology in the solar system as well as the search for life in Mars.

Some examples of application of these technologies in Exobiology may concern, monitoring microbial communities aboard the International Space Station (ISS) as a way for maintaining astronaut health and the integrity of life-support systems [74], and the analysis of biological species to Mars-like environments [75].

It is not, however, strictly necessary to go very far into outer space to find evidence of alien genetic material, since on several occasions there are observations of several extraterrestrial nucleobases present on meteorite surfaces [76, 77, 78].

Until now, the few nucleobases found in meteorites are considered biologically common, which often puts into question the possibility of contamination of samples from sources on Earth. There are, however, exceptions, as discovered in the case of the Murchison and Lonewolf Nunataks 94102 meteorites, which among them had three unusual and terrestrially rare nucleobase analogs [76].

On the other hand, the existence of chemical components necessary for life in meteorites provides an additional reason for screening of exobiology metagenomics [79]. Several celestial corpora have contacted the Earth, which opens the possibility of having ancient or extraterrestrial genomes in this planet's most challenging environmental conditions. Because the standard assembly and classification methods rely on references, there is the possibility of missing these organisms. Curiously, there are organisms from regions with hard conditions that have been used as models for exobiology studies [80, 81]. Therefore, in this area it is usually expected a high dissimilarity or singularity in the genomes sequences.

All these discoveries have been revolutionizing the fields of exobiology and astrobiology, creating a huge expectation and curiosity around the area [77]. The presence of these amino acids and nucleobases in meteorites may allow a better understanding of the emergence of life on our planet as well as explore new forms of life in environments other than the earth.

Therefore, the development of reconstruction and classification methods for dissimilar sequences have an important role in current and following years.

### 2.2.2 Ancient DNA

Another of the complex analysis subclasses is related to Ancient DNA reconstruction and classification. This branch allows us to expand our knowledge on evolutionary issues that would otherwise be very difficult to reach reliable conclusions given the obvious barrier of time.

When applied to the DNA of human ancestors, it allows the learning of new facts about their lives, its migratory routes, as well as analyzing the evolutionary path over time to the present in genomic terms [82], even though the human lineage remains largely unexplored [83]. Another of the widely recognized case studies within the Ancient DNA environment concerns the woolly mammoths (*Mammuthus primigenius*), a species of mammoths of which it was possible in 2015 to perform a complete genome sequencing for two specimens from very different time periods (while one of the specimens it is estimated to be from 45000 years ago, while the other should be only 4300 years old) [84]. The large age difference between the specimens is an excellent opportunity to study different models of genome architecture evolution within a single species [84] or, for example, in the faunal history [85].

All these areas of study share deep connections and are, to a certain degree, intrinsically interconnected, as is the case with Ancient DNA and Exobiology, which given the fact that the high degree of antiquity of an asteroid makes it possible for the exobiological material that is in it deposited to be also antique. These conditions allow, in some cases, the use of this genomic material as a form of reconstruction of the past.

### 2.2.3 Human interaction and hostile terrestrial environments

Several factors, both at the environmental level and due to human consequences, transform certain environments into hostile situations to the existence of life. However, and even these environments having few properties that are suitable for the existence of life, there are certain organisms that, against all odds, manage to thrive.

The study of these organisms and enzymes produces great value for the pharmaceutical industry, allowing, in the case of the xenobiotic field, to analyze the effect of certain chemicals in our lives, and its effect in our well-being [86].

Occurrences of this caliber appear with some regularity, such as the discovery of a bacteria that can be found in adverse conditions in which requires acids and dissolved metals in order to function [87], or even bacteria acting as natural decontamination agents such as *Aeromonas veronii*, a tributyltin (TBT)-degrading bacterium isolated from Ria de Aveiro in Portugal which act as decontamination agents in this polluted areas [88].

On the other hand, antibiotic resistance is an emerging global problem with impact on genome evolution for both pathogenic and host species [89]. Therefore, antibiotic resistance is causing changes in the genomes of bacteria and certain hosts that are not adapted for such rapid evolution, forcing an evolution based on natural selection.

### 2.2.4 *De-novo* assemblers and reference-based assemblers

One of the major challenges in handling DNA sequences concerns the reconstruction of the original DNA sequence from fragment *reads* (DNA *assembly*) that are usually randomly generated from a long DNA molecule [90].

The reconstruction process, given its immense complexity in terms of the amount of data to be processed, becomes quite heavy, time-consuming and costly, depending on the complexity of the organism under analysis, the sequencing methodology, and the characteristics of the samples as previously addressed.

To overcome this assembly DNA challenge, there are several proposed solutions, which are generically divided into three groups: DNA sequence reconstruction without resorting to any previously reconstructed reference sequence, called *de-novo* assemblers, reference-based reconstruction, called reference-based assemblers [90], and hybrid reconstruction that relies on both approaches, for example, TRACESPipe [91].

Regarding the reference-based approaches, there is a need to distinguish an aligner from an assembler, since for an aligner as is the case of BWA [92] and Bowtie2 [93], in order to perform assemblies it needs to be adapted using other programs developed for that purpose [94]. While the sequence aligners have the purpose of verifying the sequence identity or similarity between two or more different sequences, the sequence assemblers have the objective of creating a long consensus sequence from short fragments of the same sequence in order to reconstruct the original DNA sequence. Therefore, aligners based on BWA [92] and Bowtie2 [93] can behave as an assembler when used together with, for example, SAMtools [95], a tool that implements various utilities for post-processing alignments in the SAM format [95], namely variant call.

In order to mitigate operating costs, the choice of the methods, taking into account the analysis situation (that is, whether there is a previously reconstructed reference sequence from a similar organism or not), takes an even more important role than would be expected. Other factors that need to be taken into account focus on the size of the genomes to be assembled, the redundancy (a factor that makes it especially difficult to analyze in the plants domain, since they have a high redundancy [96]), the acceptance of paired-end (PE) or single-end (SE) *reads*, among others.

There are thus several tools that, depending on the needs, may provide a much better solution when compared to others. To perform a *de-novo* assembly in small genomes, tools such as metaSPAdes [97], Velvet [98] and Hinge [99], when the size becomes considerable (up to 130MB) HGAP [100], DNASTAR Lasergene Genomics [101] and ABySS [102] are good options. On the other hand, if the assembly in question becomes based on a reference, tools such as RaGOO [103], Ragout [104], RECORD [105], MIRA [106], or aligner-based approaches, may be more adequate.

The PacBio High-Fidelity (HiFi) sequencing is a recent technology that produces reads with length 10 to 25 Kpb with very high accuracy (>99,9%). These are the assemblers that are able to efficiently deal with the size of these reads: HiCanu [107], hifiasm [108] and Falcon [109]. The HiFi assemblers are able to resolve more segmental duplications than common approaches and additionally are also able to produce haplotype-resolved assemblies.

In the next tables, a summary of tools for *de-novo* assembly, Table 1, and reference-based assembly, Table 2, is available, categorized by their applicability in genomic samples size.

On the other hand, in a metagenomic context, where multiple genomes usually coexist, there are a number of situations that should be taken into account that greatly hinder the assembly process. These factors relate to the depth of sequencing that is not uniform among

Table 1: *De-novo* assemblers categorized by their applicability in small and long size genomic samples, with the respective URL and reference.

| De-Novo Assemblers | | | |
|---|---|---|---|
| **Tool** | **Specification** | **URL** | **Reference** |
| metaSPAdes | Small Genomes | `https://cab.spbu.ru/software/meta-spades/` | [97] |
| Velvet | | `https://www.ebi.ac.uk/$\sim$zerbino/velvet/` | [98] |
| HGAP | Large Genomes | `https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/HGAP-in-SMRT-Analysis` | [100] |
| DNASTAR | | Requires a Comercial License | [101] |
| AFEAP | | Requires a Comercial License | [110] |
| MaSuRCA | Small and Large Genomes | `https://github.com/alekseyzimin/masurca` | [111] |
| ABySS | | `https://github.com/bcgsc/abyss` | [102] |
| HiCanu | Large and Redundant Genomes with Large Reads | `https://github.com/marbl/canu` | [107] |
| Hifiasm | | `https://github.com/chhylp123/hifiasm` | [108] |
| Hinge | | `https://github.com/HingeAssembler/HINGE` | [99] |
| Falcon | | `https://github.com/PacificBiosciences/FALCON` | [109] |

all of the genome, being most of the time highly unequal between different organisms, the nonclonal nature of the organisms within a single sample since between-strain differences become very similar to variation between repeats, and finally, the fact that the depth of a coverage of a particular species is rarely very high [112].

Taking this into account, there exists a set of assembly tools that have a better adaptation to analyze this type of metagenomic samples, such as MetaSPAdes [97], an assembler that uses multiple k-mers with different abundances, conserved regions and strain mixtures, or IDBA-UD [113], an assembler that also makes use of multiple k-mer sizes and coverages between paths [112]. Moreover, this selection is made with the assumption that the sequencing reads are short and provenient of the previous mentioned challenging contexts.

Table 2: Reference-based assemblers categorized by their applicability in small and long size genomic samples, with the respective URL and reference.

| **Reference-Based Assemblers** | | | |
|---|---|---|---|
| **Tool** | **Specification** | **URL** | **Reference** |
| RaGOO | Small and Large Genomes | `https://github.com/malonge/RaGOO` | [103] |
| Ragout | | `https://github.com/fenderglass/Ragout` | [104] |
| RECORD | | `https://sourceforge.net/projects/record-genome-assembler/` | [105] |
| MIRA | | `https://github.com/bachev/mira` | [106] |
| Amos | | `http://amos.sourceforge.net/wiki/index.php/AMOS` | [114] |
| RACA | | `https://github.com/ma-compbio/RACA` | [115] |
| ARACHNE | | `https://wi.mit.edu/` | [116] |
| IMR/DENOM | | `http://mtweb.cs.ucl.ac.uk/mus/www/19genomes/IMR-DENOM/` | [117] |
| AlignGraph | | `https://github.com/baoe/AlignGraph` | [118] |

## 2.3 Compression-based Analysis

Data compression-based analysis has become increasingly common when working with large amounts of data, offering good precision and accurate results when efficient and optimized compression models are applied.

Efficient compression methods have been applied in the most distinct areas of study, having had a special relevance in bioinformatics, in which the results adjacent to them have been proven to be very promising both in terms of clustering, classification, anomaly detection, singularity, among others [119].

There are, however, several risks that must be taken into account when starting to use this type of methodology. The choice of the data compressor to the data type is fundamental. Efficient data compressors are demanding for achieving higher accuracy in data analysis.

Assuming that, for this type of study, the only method in interest is the *Lossless compression*, since in bioinformatics it is essential to be able to reconstruct the complete original input from the compressed output, not risking the detection of false positive or misleading patterns [119, 120].

Looking at some of the techniques used in compression algorithms for genome sequences, they can, generally be divided into reference-free and reference-based methods [121].

In the case of reference-free methods, those tools make use of the structural properties of the sequences in order to enable their compression using, for example, the palindromes existing in the sequences under analysis [121, 122]. Some of the basic techniques used in reference-free compression methods concern the naive bit encoding, dictionary-based and statistical approaches [123].

In the Naive bit encoding, it is usual to perform the encoding of four bases within one byte via bit encoding, thus replacing each input symbol by two bits using the replacement $\{A \rightarrow 00, C \rightarrow 01, G \rightarrow 10, T \rightarrow 11\}$ [123].

In the Dictionary-based methodologies, it is usual to replace DNA subsequences that are repeated throughout the input sequence with references to a dictionary. However it does not need to be stored together with the compressed information, as it can be reconstructed in the decompression process [123, 124].

A statistical compression algorithm, also known as entropy encoding algorithm, is usually represented by a probabilistic or prefix tree data structure, which is created from a statistical model of the compressor's input data. It is a variable length code algorithm in which subsequences with a higher frequency are represented with shorter codes [123, 125].

As described in Figure 2.5 there are several genome sequence compression tools using reference-free methods in its core [126, 127], such as, biocompress series [128, 129], Cfast [130], CDNA [131], ARM [132], GenCompress [133], Off-line [134], CTW+LZ [135], DNA-Compress [136], NMLComp [137], DNA-X [138], DNAC [139], DNASequitur [140], DNA-Pack [141], GeNML [142], 2D [143], DNASC [144], GBC [145], POMA [146], DNAEnc3 [147], DNAEnc4v2 [148], DNACompact [149], BIND [150], LUT [151], GenCodex [152], SeqCompress [153], HighFCM [154], OCW [155], OBComp [156], Jarvis [157] and GeCo series [158, 159, 4].

On the other hand, the reference-based compression methods, also known as referential compression algorithms, are also algorithms that benefit from the use of dictionary-based methodologies, differing from the reference-based methodology since they encode sequences with respect to an external set of reference sequences [123, 160, 161]. There are several tools that make use of methodologies based on reference-based compression, such as DNAzip [162], RLZ [163], GRS [164], GReEn [165], GDC [166], COMRAD [167], FRESCO [168] and iDoComp [169]. There are however, at this reference-based compression level, two possible modes, being them the relative compression and the conditional compression [170].

The reference-based relative compression mode is characterized by using exclusively information/models from an auxiliary sequence and never from the sequence itself. On the other hand, the reference-based conditional compression mode uses, in addition to the reference-free compression models, models over one or more additional sequences [170].

Consequently, the vast majority of genomic sequence reference-based compression algorithms makes use of relative compression, since when sequences are very similar they have much less computational complexity and require much less resources when compared to conditional compression, being sufficient however to model the sequence accurately enough when there is a very high similarity between the sequences. Conditional compressors are nonetheless more recommended when the sequences have less dissimilarity [171, 170]. Moreover, they can be adapted to perform similarity analysis based on information distances [172]. There exist some reference-based compression tools that support any of those approaches by parameterization, as is the case of the GeCo series [158, 159, 4].

Besides the compressor tools already mentioned in each of the techniques, there is also a set of tools, which, in order to try to improve the performance in terms of sequence compression, makes use of both reference-free and reference-based methods in different stages of processing, as is the case GeCo [158], GeCo2 [159], GeCo3 [4], XM [173], DNA-COMPACT [174] and CoGI [175].

The diagram in Figure 2.5 provides a visual representation of the appearance of the main lossless and reference-free genomic sequence compressors over the latest 3 decades.

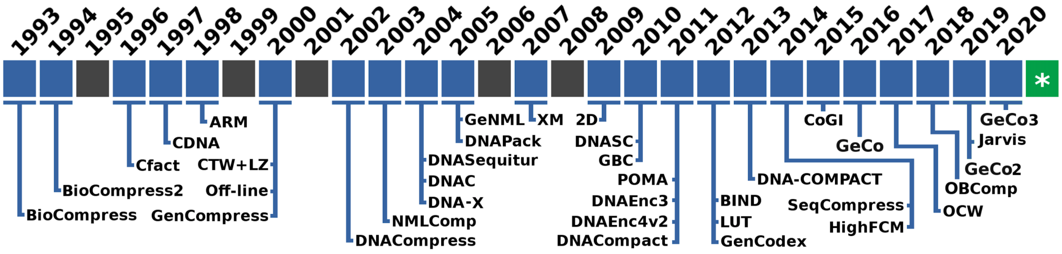


Figure 2.5: Proposed reference-free genomic sequence compressors sorted by release until 2021. Adapted from [2]. Source: A Reference-Free Lossless Compression Algorithm for DNA Sequences Using a Competitive Prediction of Two Classes of Weighted Models [2].

One of the most popular application of the reference-free sequence compression of genomic sequences is the Kolmogorov complexity estimation [3]. However, to approximate the Kolmogorov complexity there is the need to use a normal compressor. A normal compressor is a compressor described by a set of properties, including idempotency, monotonicity, symmetry and distributivity [176]. In these compressors, the idempotency arises from the fact that if they are compressed together and through an approach based on concatenation the information of a sequence and a copy of that same sequence, the result of this compression will have to be approximately equal to the number of bits that the compressor needs to describe the compressed version of one of them. As for monotonicity, this property states that if a certain sequence is compressed together with any other type of information, the number of bits must always be greater than or equal to the number of bits in the sequence. In symmetry the order between large digital objects to be compressed, can be arbitrary. The distributivity property is related to the triangle inequality, which essentially shows that the shortest distance between two objects is a straight line [176].

There are a few studies that demonstrate the quality of inter-domain classification using lossless normal compressor techniques, being one of them applied in the approximation of the Kolmogorov complexity [177], where it is possible, from a visual representation point of view, group the various sequences by domains, using for this case, a normalized compression for each sequence as a function of its logarithmic size [3], as it is demonstrated in Figure 2.6.

Therefore, this capability as a reference-free approach provides an interesting insight to explore this measure along with other experts for the development of an automatic taxonomic classification application using machine learning.

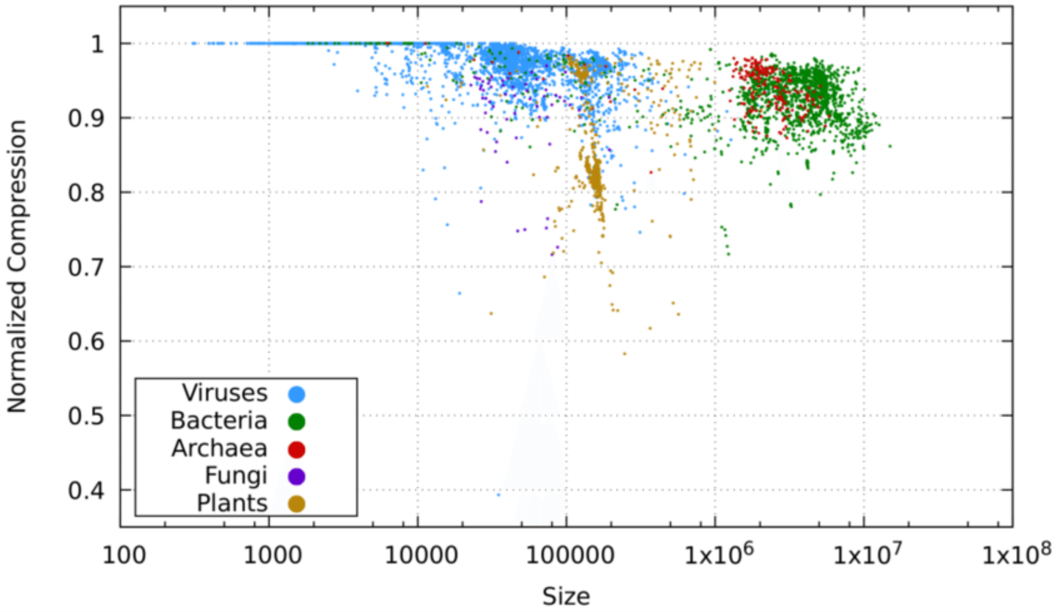


Figure 2.6: Normalized compression for each sequence of a specif domain, as a function of its logarithmic size [3]. Source: On the Approximation of the Kolmogorov Complexity for DNA Sequences [3].

## 2.4 Machine Learning

Machine learning becomes another of the key areas in this thesis, since after the acquisition of all predictors (inputs extracted from sequences for analysis), there is a need to handle and analyze them in order to obtain a prediction that is the most reliable possible, without using any referential method in this process and through an automatic classification system.

In this category, there are many interesting possibilities that deserve attention. From classical machine learning methods to classifying neural networks, there are many options that deserve reflection and analysis for application in this problem.

### 2.4.1 Classical Methods

In classical machine learning methods there are some algorithms that arouse greater interest, either because they are the most traditional or because, within this category, they are the ones that usually show the best behavior, among them are the *Naive Bayes* (NB), the *Gaussian Naive Bayes* (GNB), *K-Nearest Neighbor* (KNN), *Support Vector Machines* (SVM), decision trees and ensemble methods. In order to better understand the differences, advantages and disadvantages of these methods, it is necessary to carry out an individual analysis of them.

1. **Naive Bayes (NB)** Based on the Bayesian decision theory, NB assumes that each input predictor is independent, formulating naive assumptions [178]. This method makes use of conditional independence, being a popular method in multi-class prediction problems [178, 179]. However, considering that all predictors are independent translates into

a disadvantage, since for practical real-life purposes, this situation rarely occurs. Nevertheless, it usually provides acceptable results using lower computational time and code complexity.

2. **Gaussian Naive Bayes (GNB)** Based on the NB model analyzed above, the GNB assumes a normal distribution, which brings benefits when the predictors used are continuous [180]. Even though it is a simple classification technique, the predictions it obtains are usually of good quality, given the low complexity of the algorithm.

3. **K-Nearest Neighbor (KNN)** One of the best known and most straightforward classical machine learning methods widely used in classification problems [181]. In this method, the solution is achieved by identifying the $K$ nearest neighbors to the query that is being classified. The classification is assigned through the largest number of neighbors of a given domain. A parameter to take into account in this approach will be the selection of the value for $K$, since for each problem this variable may need adjustment.

4. **Support Vector Machines (SVM)** Similar to the other methods described earlier, SVM also allows the classification of a query among a set of domains. Its special feature is the use of different types of kernels (such as the Polynomial kernel, the Gaussian radial basis function kernel, the Sigmoid kernel, among many others) that allow transforming the input data in order to find the optimal boundary between the possible outputs. These boundaries are called hyperplanes and can exist in different dimensions, depending on the number of domains that one intends to delimit [182].

5. **Decision Trees** These types of algorithms are commonly referred to as binary trees in which classification is done by the splitting criteria. In these binary trees, the logical predicates at its nodes, and class labels in sheets are found [183, 184]. These methods tend to produce better predictions compared to the other classical methods analyzed here since they can work without loss of accuracy with sequences in which the predictors differ greatly at the level of orders of magnitude [183]. There are different ways to build decision trees that must be taken into account when using these algorithms, the most common being the top down greedy method partitioning [184], always seeking to maximize the initial criteria, in order to obtain a global maximum.

### 2.4.2 Ensemble Methods

These machine learning techniques have as main feature the combination of several base models in order to achieve more accurate and stable predictions, being the different methods that are part of the ensemble methods the Stacking (or Stacked Generalization), the BAGGing (or Bootstrap AGGregating) and the Boosting [185, 186].

1. **Stacking** Stacking is a method that applies several models to the original data, using logistic regressions to combine all single models, with the main objectives of minimizing the variance and increasing the predictive force.

2. **BAGGing** BAGGing has as its main focus decreasing the variance of the prediction, by generating additional data for training by introducing combinations with repetitions in the training datasets, using in turn weighted average functions to combine all single models, being very useful in random subspaces such as Random Forrests.

3. **Boosting** Finally, Boosting has the priority of increasing the predictive force making use of optimized distributed gradients, as is the case with XGBoost, using weighted majority votes techniques as a way to match all single models.

### 2.4.3 Artificial Neural Networks

In the domain of Deep Learning (DL) and Neural Networks (NN), there are the concepts of Deep Neural Networks (DNN) and Artificial Neural Networks (ANN).

Artificial Neural Networks (ANN) have proven to be very useful and powerful in recent decades, tending to be ideal for handling a huge amount of data, and getting excellent accuracy results in pattern recognition problems [187].

There are several advantages that can lead to opt for a neural network, such as the possibility of storing all the information of the entire network, the possibility of training the network, parallelism in processing, fault tolerance, among others. However, there are also some disadvantages that must be analyzed from project to project as they can put the benefits in question, such as the large hardware dependence and the monetary costs of hardware and time [188].

The choice of the neural network algorithm is also a fundamental step in this stage, since different algorithms are designed for different realities. Some of the most popular deep learning algorithms are Multilayer Perceptrons (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short Term Memory Networks (LSTM).

The following descriptions briefly contextualize a little more each of these Artificial Neural Networks (ANN).

1. **Multilayer Perceptrons (MLP)** This ANN is mainly suitable for classification or regression prediction problems in which the supervised learning technique called backpropagation for training is used. It is a network characterized by having three or more layers, having at least one input layer, the hidden layer and an output layer [189].

2. **Convolutional Neural Networks (CNN)** This ANN is a more focused neural network for image recognition and processing, operating on multiple layers (including convolutional layer, non-linearity layer, pooling layer and fully-connected layer) [187]. It was primarily designed to perform pixel data processing.

3. **Recurrent Neural Networks (RNN)** While CNN is a neural network designed for pixel data processing, RNN is commonly used in speech recognition and natural language processing, capturing time dynamics via cycles in the graph [190].

4. **Long Short Term Memory Networks (LSTM)** LSTM is an RNN based neural network better prepared for classifications and predictions of one or a set of queries, namely because these are networks that use special memory units, which allows them to have a greater storage of information for long periods of time, allowing better overall network performance [191].

For classification purposes, the choice of the algorithm is fundamental. However, this is a complex task, that requires extensive experience, if the algorithms are not all tested. Usually the choice of the algorithm is based on a balance between high precision/accuracy, computational resources, and easiness of implementation. Despite this choice, it is known that in recent competitions, the ensemble methods have provided the best scores.

# Chapter 3

# Reconstruction and Classification Methodology

The exponential growth of genome-wide association studies (GWAS) and its success in the identification of genetic risk variants and their biological functions has opened unprecedented opportunities for the introduction of genomic analysis in the most varied professional fields [192, 193, 194]. In the medical field, genomics and metagenomics analysis are starting to be the key in a new era of health care [195]. Since diagnosis plays a pivotal role in the patient's clinical progress, the introduction of these tools may provide personalized treatment, early detection, and disease prevention [196, 197]. For example, one of the significant contributions in this field can be the earlier detection of viral agents that can be associated with or cause of diseases [198, 199]. The emergence and re-emergence of new pathogens caused by climate changes [200], hybridization [201], evasive species [202], evolution-prone reservoirs [203, 204], and the abusive antibiotics usage [205] are continuous factors for sequence dissimilarity according to the extant species. Moreover, emergent discoveries of viral communities residing in different human organ samples provide additional purposes for screening intra-organ diversity genomes [206].

Another field where metagenomics analysis is considerably gaining ground is archaeogenomics. Archaeogenomics has been responsible for challenging fundamental themes of anthropological research such as human origins, migratory movements of ancient and modern populations, and infection agents [207, 208, 209]. Specifically, ancient DNA is usually distant from extant species that have accumulated evolutionary changes over the years; besides, ancient DNA is commonly characterized by damage patterns, namely fragmentation, deamination, and depurination [210]. Therefore, ancient genomes present higher dissimilarity to extant ones, which adds additional complexity to the classification process [211].

Since the specifications between the mentioned fields are different, it is essential to ensure that the reconstruction and classification pipelines used are customized, flexible, and robust to meet the criteria and their characteristics.

Different computer-assisted techniques have been implemented in the last few years in the development of reference-based classifiers, such as consensus searches, inductive learning/neural networks, and sequence alignments [212, 213, 214, 215, 216]. In the case of sequence alignments, the program aligns all the unknown sequences using one or more known database sequences to predict common portions [217]. This process results in good results for specific small DNA sequences classification, but it becomes increasingly complex when scaled for the largest datasets. In addition, these techniques have proven to be inaccurate in scenarios of low sequence identity, making evident the need for the emergence of new and more robust methods [215, 218]. As a way to mitigate these aspects, the pipeline proposed in this paper uses tools (e.g. metaSPAdes [97]) that rely on alignment-free sequence analysis methods, such as *k-mers* [219], where the fast detection of shared *k-mer* content strongly contributes to the reduction of the computational cost of assembly [220].

Methods that use different layers or levels of the chemical contents are more accurate in metagenomics classification. For instance, a multi-level analysis performing an additional proteome analysis can provide information that would otherwise be difficult to obtain. The proteome analysis of sequences allows the study of domains in which they are inserted, their structures, and functions [221]. Alternatively to these domains, proteins can also be found grouped into families based on their whole sequence [222, 223]. In this way, analysis can be carried out in order to identify protein-coding genes in metagenomic data as well as for grouping related sequences into families [223]. For this feature, tools such as BLASTP search [224] can be options to analyze metagenomic datasets (e.g. Swiss-Prot [225] and Pfam [226, 227]). However, problems related to the size of databases, scalability, and computational costs concerning the reference-based methods remain [222].

Despite all concentrated efforts, in the presence of a new or an extremely high dissimilarity genome sequence, these methods may not perform efficient classification specifically because they rely on references directly.

For providing reference-free metagenome distance estimation, fast methods recurring to Local Sensitive Hashing [228, 229] successfully emerged. These methods enable splitting the existing different nature sequences with higher accuracy. However, in these cases, the task of classification remains a challenge. Therefore, this thesis describes RFSC, a comprehensive solution designed for reference-free reconstruction and accurate classification combining reference-free and reference-based methods both at DNA and protein levels.

## 3.1 Architecture

The RFSC pipeline uses a set of tools in its composition distributed over distinct steps as shown in Figure 3.1, namely reference-free reconstruction, database creation, reference-based classification, and features-based classification.

Initially, reference-free reconstruction is characterized by the assembly of the genomes from FASTQ *reads*. Specifically, the *reads* go through a quality control process, entering into the process of genome reconstruction by building scaffolds from overlapping *reads* (*de-novo* assembly). After, the metagenomics database is built for reference-based classification. Here, a database is used because a low divergence level characterizes most of the genomes reconstructed. Only the organisms whose relative similarity or identity is below a specific threshold move on to the next stage for a deeper analysis, using for that purpose sequence features.

In the features-based classification (reference-free analysis), several tools and methods are used together, such as the entropy-based approach for analyzing sequences [230], GC-content analysis [44, 231], and sequence size, to create an ensemble of experts. These experts are then fed to a machine-learning algorithm to perform the classification of the sequence. The analysis of both protein and DNA analysis is carried through this process.

In any stage of the RFSC's workflow (Figure 3.1), the encryption/decryption of any files

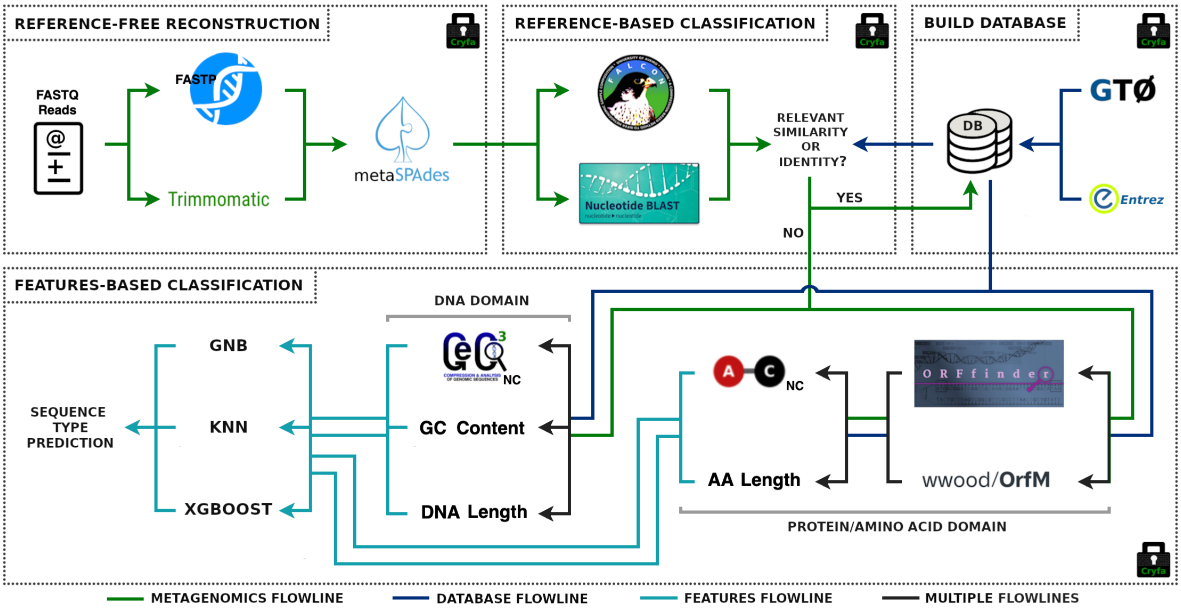


Figure 3.1: The architecture of RFSC for reference-free reconstruction, reference-based, and features-based classification of genomes. The tools are represented with the respective logos and names. There are four flowlines, namely the metagenomics, database, features, and multiple. The multiple flowlines stands for different flowlines that by space and color constrains have been represented as coincident. In any of the phases the cryfa tool can be used for secure storage of any file.

using Cryfa [232] may be performed. This process allows assigning a layer of security to the sensitive data processed in the pipeline [233], specially when they provide from clinical or exobiology scenarios.

The following sections will describe the functionalities, complementary options, and details of RFSC, paying particular attention to the pipeline tools, data privacy and storage, data preparation, de-novo assembly, reference-based classification, features-based classification (predictors and classifiers), testing methods, and evaluation methods.

### 3.1.1 Pipeline tools

In this subsection, the different approaches that RFSC uses are described. In many cases, already existing tools were used, given their efficiency and high quality. Table 1 presents all the tools integrated in RFSC with their respective references. In general, only the second stage required developing a specific tool for reference-free classification that is described and benchmarked in this thesis.

Notice that several tools may be substituted from the vast existing literature. Nevertheless, the choice of tools provided in this pipeline was based on multiple factors, such as the compatibility between tools, computational resources, diversity, and research aims.

Table 1: Tools integrated in RFSC and their respective references (Ref).

| **RFSC Integrated Tools** | | |
|---|---|---|
| **Tool** | **URL** | **Ref** |
| Trimmomatic | `http://www.usadellab.org/cms/?page=trimmomatic` | [234] |
| FASTP | `https://github.com/OpenGene/fastp` | [235] |
| metaSPAdes | `https://cab.spbu.ru/software/meta-spades/` | [97] |
| GTO | `https://cobilab.github.io/gto/` | [15] |
| Entrez | `https://www.ncbi.nlm.nih.gov/genome` | [236] |
| FALCON-meta | `https://github.com/cobilab/falcon` | [39] |
| Cryfa | `https://github.com/cobilab/cryfa` | [237] |
| Blastn | `https://blast.ncbi.nlm.nih.gov/Blast.cgi` | [238] |
| ORFfinder | `https://www.ncbi.nlm.nih.gov/orffinder/` | [239] |
| ORFM | `https://github.com/wwood/OrfM` | [240] |
| GeCo3 | `https://github.com/cobilab/geco3` | [4] |
| AC | `https://github.com/cobilab/ac` | [5] |

### 3.1.2 Data Privacy

With the great increase in the use of genomic tools in professional fields such as medicine and biological research, the amount of sensitive information pertaining to patients and scientific studies tends to become much greater over time. Therefore, it is imperative to have confidentiality, integrity, and authenticity in the information handled, which translates into a higher level of requirements in the field of data security [232]. In order to satisfy these conditions, RFSC provides secure encryption of genomic data through the Cryfa tool [232].

Cryfa is an industry-oriented tool, capable of encrypting files in FASTA, FASTQ, VCF, SAM, and BAM formats, using a fixed-block transformation followed by AES (Advanced Encryption Standard) [232, 237].

### 3.1.3 Data Storage

The large amount of sequencing information generated presents a challenge for long-term storage. As a way of trying to mitigate the impacts of storing large amounts of data, the use of data compression tools (e.g. Gzip) is one of the approaches followed by RFSC.

Cryfa also contributes to storage reduction, as it reduces storage approximately three times when compared to general encryption methods, without compromising security [237].

### 3.1.4 Data Preparation

#### Trimming Stage

The error-prone nature of high-throughput sequencing *reads* specially in ancient DNA and the exobiology areas, results in an additional layer of complexity for genomic analysis. Another factor to consider is related to Illumina sequencing, in which the error is distributed non-randomly over the length of the read [241].

In this way, before the *reads* are analyzed, they have to be trimmed and cleaned, removing eventual sequencing errors, and filter *reads* with low-quality scores [242]. For this purpose, RFSC makes use of two tools, namely Trimmomatic [234] and FASTP [235]. Both tools have similar behaviors, making it possible to perform quality control, adapter trimming, quality filtering, per-read quality pruning, among other operations that have the goal of providing clean data for downstream analysis [235]. As the read trimming stage is a fundamental process throughout this analysis, these two tools are available to the user choice to trim high-throughput sequences.

Although both tools support multi-threading and single-end (SE) and paired-end (PE) *reads*, some differentiating features exist. For example, FASTP allows a performance 2–5 times faster than other FASTQ preprocessing tools (such as Trimmomatic) [235].

In the Trimmomatic approach, some extra parameters are defined, namely choosing a minimum quality score (set to 3) in order to keep a base at the beginning and the end, a low-quality data filter with an average quality of 15, a set of thresholds defined for a palindrome, and simple clip with respective values of 30 and 15, as well as the disposal of all *reads* containing less than 25 bases [91, 241].

### Database

The database built for the multiple domains have the main objectives to offer datasets as extensive, diverse, and complete as possible. As such, both genomic and proteomic databases can be updated with the latest NCBI data whenever necessary. Moreover, other databases can be combined to increase diversity.

The datasets of the database used in the RFSC refer to the domains of viruses, bacteria, archaea, plants, fungi, protozoa, plastids, and mitochondria, as it can be seen in Table 2. This table shows the database containing the FASTA reference genomes, the number of sequences present in each dataset, and their respective compressed size.

Table 2: NCBI Database downloaded and built for RFSC. Each dataset size is provided as compressed size with Gzip (default level: -5).

| **NCBI Compressed Databases** | | |
|---|---|---|
| **Domain** | **Number of Sequences** | **Length** |
| Viruses | 10804 | 126.8 MB |
| Bacteria | 21372 | 26 GB |
| Archaea | 1125 | 4.4 GB |
| Fungi | 375 | 5.3 GB |
| Plant | 134 | 35.8 GB |
| Protozoa | 94 | 1.4 GB |
| Plastid | 6081 | 1.5 GB |
| Mitochondrion | 11345 | 314 MB |
| Total (DB) | 51330 | 74.9 GB |

Despite the recent increase in the quality of the NCBI reference genomes database, there is still some space for improvement as its quality has a direct impact on the ability to interpret a microbiome sample [243]. Therefore, tools for the extraction of contaminants are valuable approaches to complement this pipeline [244]. Moreover, the constant improvements of the reconstruction methodologies are enabling a substantial increase in the reference genomes quality, especially for harder genomes to assemble primarily because of higher repetitive nature [245]. Therefore, the current growth in data quality is expected to produce higher classification accuracy in the proximal future.

Notice that the RFSC database becomes almost obsolete in reference-free classification, having only direct applicability in referential analysis. This database's particular point in the features-based approach is updating the different domains' compression values in nucleotide and protein analysis and simple characteristics such as GC-content and sequence length. Therefore, by default, RFSC provides the features already computed. Nevertheless, this can also be re-built or added at any time automatically.

#### ORF Stage

The RFSC works automatically at both nucleotide and protein levels, enabling an additional layer of information to provide accurate results in the classification process. It is necessary to perform extra computations on the sequences to extract the amino acids sequences automatically, specifically detecting the Open Reading Frames (ORF).

RFSC detects the ORF by extracting the data between the start and stop codons followed by the translation. Accordingly, RFSC offers two possibilities to perform this analysis, namely ORFfinder [239], and OrfM [240]. Even though the results generated by both tools are very similar, there are some differences worth exploring.

In general, ORFfinder provides higher accuracy of true ORF, usually originating datasets with higher quality when compared to OrfM. However, it is much more computationally expensive, making its use often unfeasible in a timely period. In contrast, OrfM allows 4-5 times faster processing time when compared to similar tools (i.e. GetOrf [246] and Translate [247]) [240]. Producing good quality results but not as robust as ORFfinder, it is, therefore, best suited to large, high-quality datasets [240].

This step is essential for automatically translating the nucleotide sequences into protein sequences for further use in the experts' ensemble.

### 3.1.5 *De-Novo* Assembly

When analyzing the pipeline architecture shown in Figure 3.1, the *de-novo* (reference-free) assembly appears after the trimming stage (described previously). This step in the pipeline aims to reconstruct the genomes, starting from many *reads* without any prior knowledge about the correct sequence, order, abundancy, or composition. This step allows the reconstruction of the genomes, starting from a large number of *reads* without any kind of priori knowledge about the correct sequence or order of them.

To proceed with the *de-novo* assembly, the core meta-assembler metaSPAdes was used to assemble datasets with non-uniform coverage [97]. metaSPAdes was created primarily as a tool for metagenomic assembly rather than target-based assembly. metaSPAdes is used by activating a sequence of flags to improve the output data and reduce mismatches and short indels. Additionally, metaSPAdes also supports data consisting of *single-reads* and *paired-reads*, both of which are supported in RFSC. Depending on the DNA fragments used, the use of paired *reads* can become much more beneficial when compared to single *reads*, either in resolving structural rearrangements or in indicating the size of repetitive regions and how far apart contigs are from each other [248].

In the final part of the assembly, metaSPAdes creates scaffolds in a FASTA format with the reconstructed genomes or fragments of genomes. In several cases, when the depth of sequencing is low, several scaffolds are generated. As a complementary tool to the pipeline, the order and groups of these scaffolds can be predicted with comparative programs as smash++ [249].

After the reconstruction process, follows the classification using reference-based, reference-free, and features-based approaches. In the following subsections, these methodologies will be described in detail.

### 3.1.6 Reference-Based Classification

Despite RFSC being primarily directed at obtaining a features-based (reference-free) classification, it also supports a referential classification, mainly because most genomes present in a metagenomic sample have high levels of similarity/identity regarding the existing ones. As such, in the reference-based classification, there are two complementary possibilities implemented in the RFSC:

- Alignment-based classification using Blastn [36];
- Alignment-free classification using FALCON-meta [39].

These tools measure the identity/similarity between the input sequences (in this case, the FASTA files from the metaSPAdes) and any multi-FASTA database (RFSC offers support to download and build NCBI databases, although any other database can be used).

Regarding both reference-based classification tools, there are some differentiating factors between them. FALCON-meta, a fast and accurate tool, is used to measure the similarity against whole-genome reference databases, providing the score representing the similarity of the *reads* to each reference sequence. [39]. On the other hand, unlike FALCON-meta, Blastn [238] is a tool that performs sequence alignment to identify the species most likely resembling the input sequence. Furthermore, the Blastn database for reference analysis can be built locally or accessed remotely to measure the highest similarity rate against the reference database.

After using one or both of the reference-based classification tools, if the score representing the identity/similarity of the sequence is greater or equal to 70% (default value that can be changed as a parameter), the introduced genome is considered to have an identity/similarity factor very close to that reference. Otherwise, since the genome sequences are dissimilar or unknown, they follow to the next phase of the pipeline, namely the features-based (reference-free) classification.

## 3.2 Predictors

The features-based classification uses multiple predictors for feature extraction before the classifiers application phase. RFSC uses the following five predictors:

- Nucleotide sequences normalized compression;
- Amino acid sequences normalized compression;
- Nucleotide sequences GC-Content;
- Nucleotide sequences normalized length;
- Amino acid sequences normalized length.

These predictors will result in five floating-point values that will be redirected into a chosen classifier. Below, these predictors are going to de described in detail.

### 3.2.1 Nucleotide and Amino Acid sequences Normalized Compression

The Normalized Compression (NC) is a measure that quantifies the proportion of complexity (or information) that exists in a string [3]. The NC enables to provide a normalized upper-bound approximation to the Kolmogorov complexity [250, 177]. By knowing the proportion of complexity contained in strings, they can be compared independently from their sizes.

For computing the NC in nucleotide and amino acids, it is used efficient data compressors for each specific nature. The NC for the nucleotide sequences is computed using GeCo3 [4], while for the amino acid sequences with AC [5]. For finding the best compression levels for the GeCo3 and AC, a benchmark with the data compressors was created using the whole database, as it can be observed in Figure 3.2.

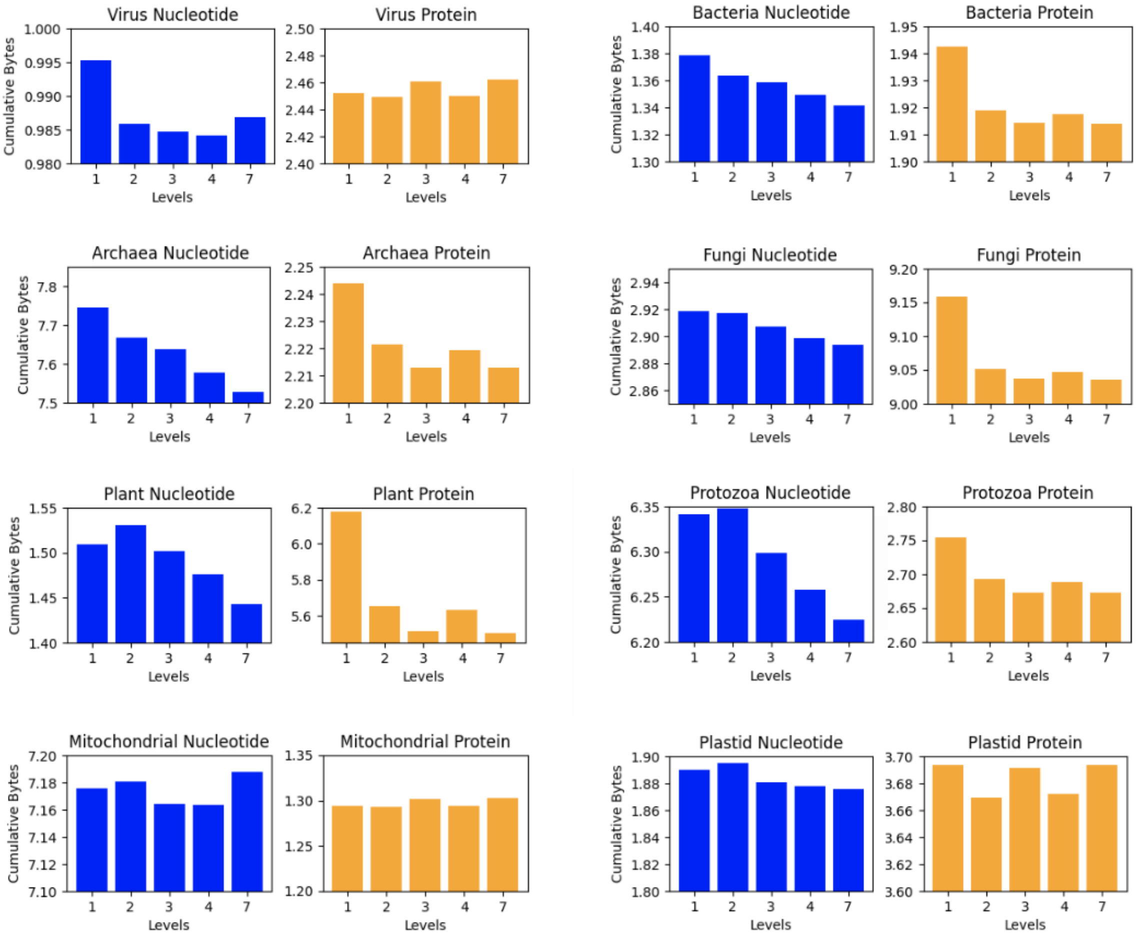


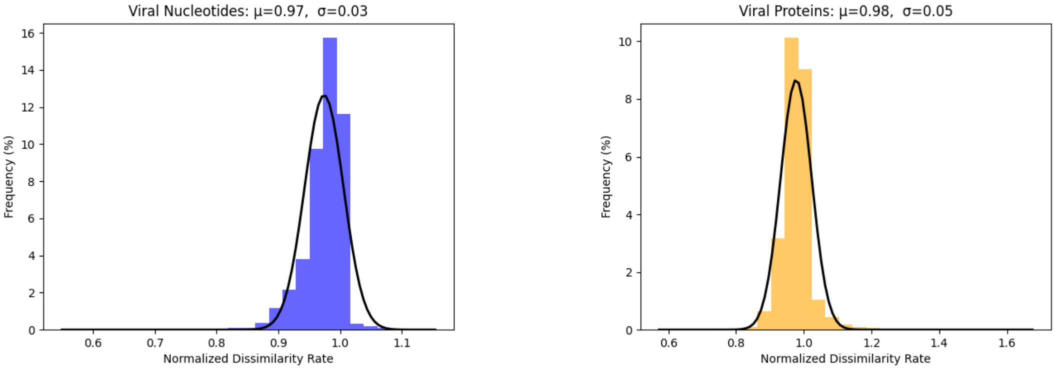
Figure 3.2: Compression level benchmark with cumulative sizes for all tested levels (namely levels 1, 2, 3, 4 and 7) of each domain or type considered (Viral, Bacteria, Archaea, Fungi, Plant, Protozoa, Mitochondrial, and Plastid).

Specifically, the NC is calculated according to

$$\mathrm{NC}(x) = \frac{C(x)}{|x| \log_2 |\mathrm{A}|}, \tag{3.1}$$

where $x$ is a string, $C(x)$ is the compressed size of $x$ in bits, $|\mathrm{A}|$ the number of different elements in $x$ (size of the alphabet) and $|x|$ the length of $x$. In the case of the DNA sequences, $|x| = 4$, while for the amino acid sequences $|x| = 20$.

Figure 3.3 depicts the histograms of the sequences as a function of the Normalized Compression, where Figure 3.3 (a) plot represents the nucleotide domain and Figure 3.3 (b) the amino acid domain. The mean ($\mu$) and standard deviation ($\sigma$) values are also represented in both cases. The histograms have been computed using the sequences described in the database subsection.



(a) Viral Nucleotide NC

(b) Viral Amino Acid NC

Figure 3.3: Histograms of the Normalized Compression (NC) computed with GeCo3 [4] for genomic sequences **(a)** and AC [5] for amino acid sequences **(b)**.

### 3.2.2 GC-Content

The GC percentage is given by the number of cytosine (C) and guanine (G) bases in a string $z$ with length $|z|$ according to

$$\mathcal{GC}(z) = \frac{100}{|z|} \sum_{i=1}^{|z|} \mathcal{N}(z_i || z_i \in \Xi), \tag{3.2}$$

where $z_i$ is each symbol of $z$ (assuming causal order), $\Xi$ is a subset alphabet containing the symbols $\{G, C\}$ and $\mathcal{N}$ the program that counts the numbers of symbols in $\Xi$. Complementary, $\mathcal{AT}(z) = 100 - \mathcal{GC}(z)$.

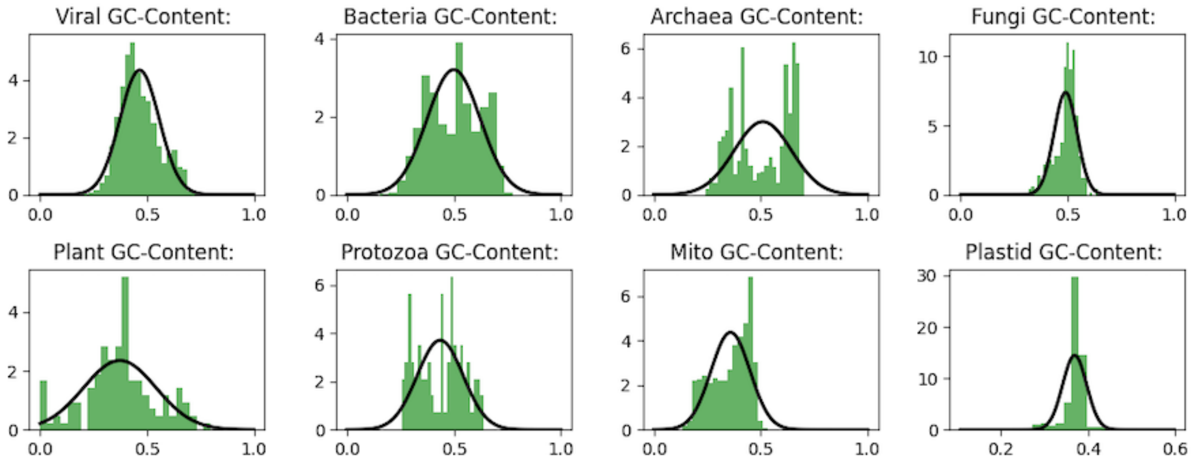
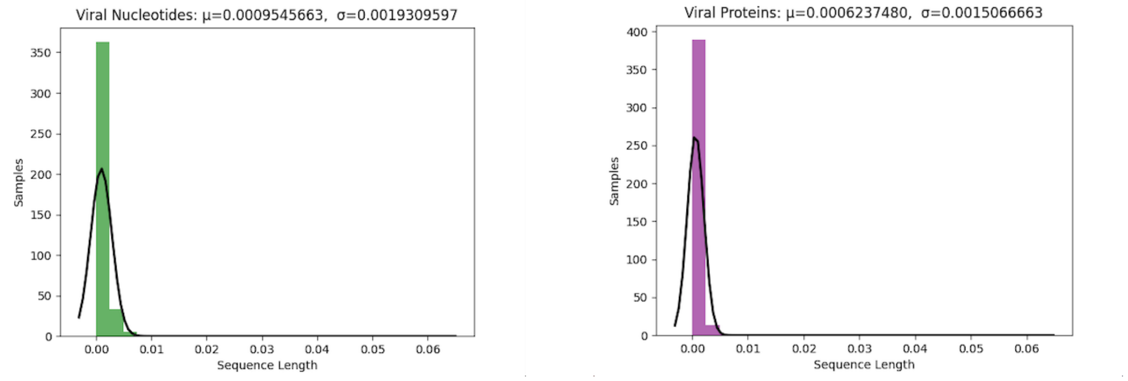


Figure 3.4: The GC-content Gaussian distribution histograms per life or type domain.

Figure 3.4 depicts the GC-content histograms, computed with the values of the division of Equation 3.2 by 100, for several sequence domains and types, namely viral, bacteria, archaea, plant, protozoa, and mitochondrial. The histograms have been computed using the sequences described in the database subsection.

### 3.2.3 Nucleotide and Amino Acid sequences normalized length

Another predictor used in the classification process is the normalized length. The normalized length of a sequence is defined as the number of symbols that the sequence contains according to the largest sequence from the database. Both the normalized lengths for the DNA sequences and amino acid sequences have been used.



(a) Viral Nucleotide Length

(b) Viral Amino Acid Length

Figure 3.5: Histograms of the normalized lengths for DNA **(a)**, and amino acids **(b)** sequences.

Figure 3.5 depicts the nucleotide and amino acid lengths histograms for the viral domain. The mean ($\mu$) and standard deviation ($\sigma$) values are also represented in both cases. The histograms have been computed using the sequences described in the database subsection.

## 3.3 Classifiers

Figure 3.6 provides a spatial representation of a portion of the values present in the database of the different domains, making use of the DNA NC, amino acid NC, and GC-Content values of the sequences as substitutes for the *xx*, *yy* and *zz* axes, respectively. For training and testing the algorithms, the full database previously referred in Table 2 was used.

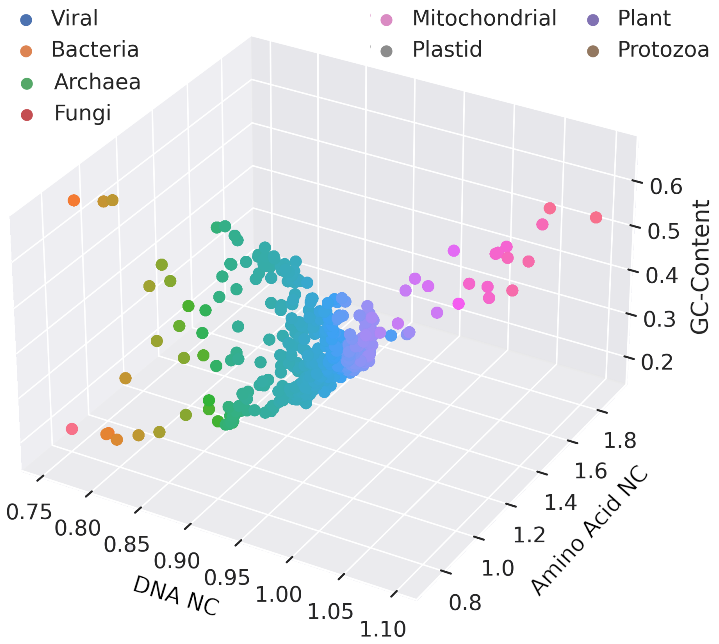


Figure 3.6: Sample distribution of a training dataset sample considering the DNA NC (Normalized Compression), Amino Acid NC, and GC-Content.

Notice that although there were used the normalized lengths as extra predictors, in Figure 3.6, a good distinction of the classes with the NC of the DNA and amino acids can already be identified. This characteristic enlightens the importance of efficient data compression in the sequence classification task. However, when a predictor is used individually, the accuracy is modest. This behavior can be observed in Figure 3.7, Figure 3.8 and Figure 3.9, which includes a study case for the success prediction percentages for each domain/type taking into account each predictor.

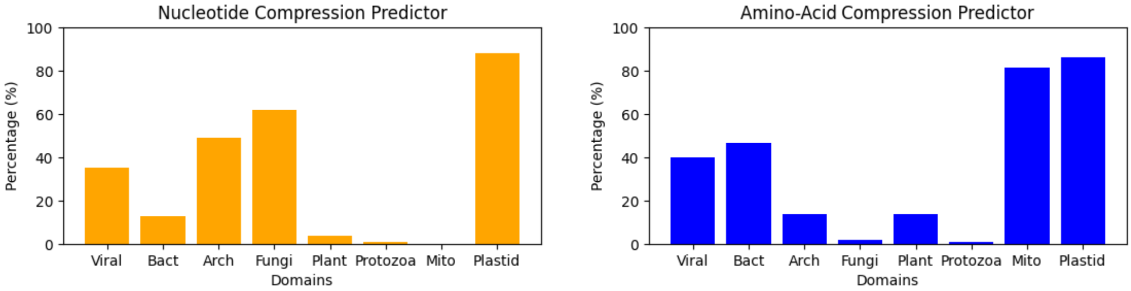


Figure 3.7: Study case for the success prediction percentages for each domain taking into account each individual predictor, more specifically the Nucleotide and Amino Acid Compression (NC) predictors, the Nucleotide and Amino Acid Lengths predictors, and the GC-Content predictor.

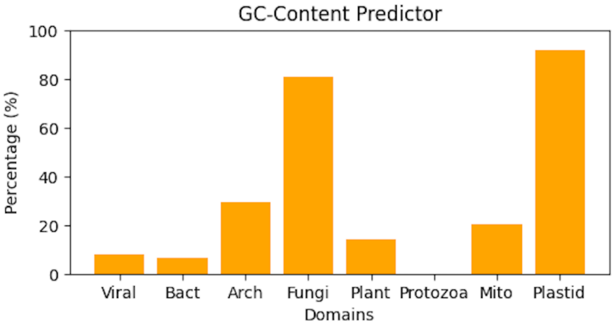


Figure 3.8: Study case for the success prediction percentages for each domain taking into account each individual predictor, more specifically the Nucleotide and Amino Acid Compression (NC) predictors, the Nucleotide and Amino Acid Lengths predictors, and the GC-Content predictor.

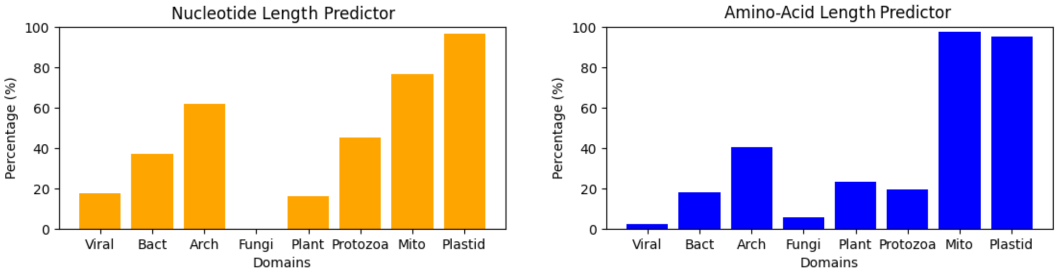


Figure 3.9: Study case for the success prediction percentages for each domain taking into account each individual predictor, more specifically the Nucleotide and Amino Acid Compression (NC) predictors, the Nucleotide and Amino Acid Lengths predictors, and the GC-Content predictor.

For automatic classification, it is required to have an automatic mechanism. According to Figure 3.1 in the features-based classification area, for the implementation of this mechanism, it was used different probabilistic, voting, and machine learning algorithms. These classifiers are the following:

- Gaussian Naive Bayes (GNB);
- K-Nearest Neighbors (KNN);
- eXtreme Gradient Boosting (Xgboost).

The following subsections provide the definitions and details for the three types of classifiers used in RFSC.

### 3.3.1 Gaussian Naive Bayes (GNB)

Gaussian Naive Bayes [251] is defined as the group of supervised machine learning classification algorithms based on the Bayes theorem following Gaussian normal distribution, where there is an assumption of independence between every pair of predictors.

The likelihood of each predictor recognizing the associated domain is calculated according to

$$\mathcal{L}(\alpha_i|\beta) = \frac{1}{\sqrt{2\pi\sigma_\beta^2}}\exp(-\frac{(\alpha_i-\mu_\beta)^2}{2\sigma_\beta^2}), \tag{3.3}$$

where the $\alpha_i$ and $\beta$ refer to the sequence to be analyzed and the domain to which it may belong, respectively. In turn, the parameters $\mu_\beta$ and $\sigma_\beta$ are estimated using maximum likelihood.

A visual representation of the method of operation of a GNB classifier is provided in Figure 3.10.

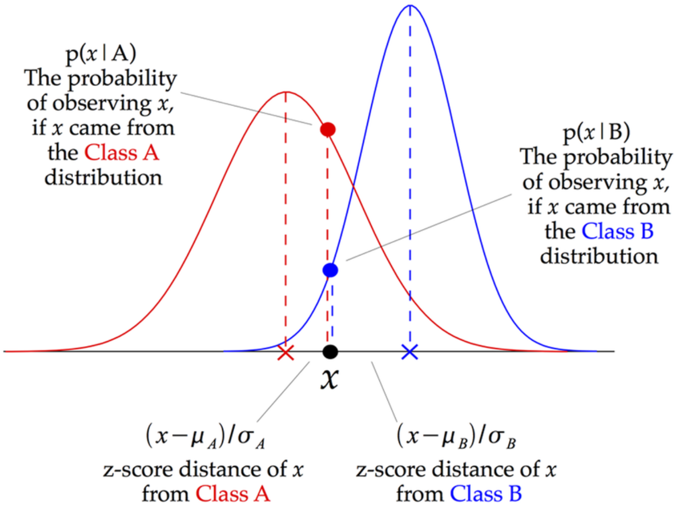

p(x | A)
The probability of observing x, if x came from the Class A distribution
p(x | B)
The probability of observing x, if x came from the Class B distribution
x
(x−μ_A)/σ_A
z-score distance of x from Class A
(x−μ_B)/σ_B
z-score distance of x from Class B

Figure 3.10: Gaussian Naive Bayes Classifier visual representation. Source: moothness without Smoothing: Why Gaussian Naive Bayes Is Not Naive for Multi-Subject Searchlight Studies [6].

### 3.3.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbors [252] is another approach to data classification, taking distance functions into account and performing classification predictions based on the majority vote of its $K$ neighbours.

In this KNN implementation, the Euclidean distance function between two points was used, calculated according to

$$\mathcal{D}_{Eucl} = \sqrt{\sum_{i=1}^{K}(x_i - y_i)^2}, \tag{3.4}$$

where $K$ is the representation of the number of predictors used in this algorithm.

In order to help understand this algorithm, a visual representation of the classification of an element between two categories using the K-Nearest neighbor is presented in Figure 3.11.

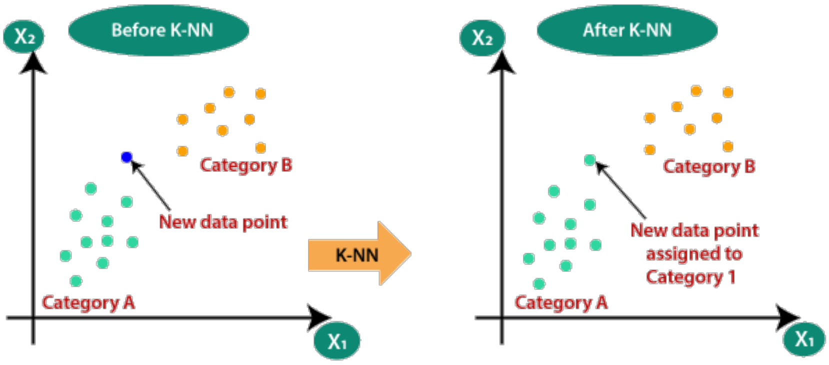


Figure 3.11: K-Nearest Neighbor Classifier visual representation. Source: javaTpoint.

### 3.3.3 eXtreme Gradient Boosting (Xgboost)

XGBoost [253] is a widespread and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm that predicts a target variable by combining the estimates of a set of simpler models. Specifically, new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. This task uses a gradient descent algorithm to minimize the loss when adding new models. This method can be used in both regression and classification predictive modeling problems.

Given a dataset $(X, Y)$, where $X$ is the data and $Y$ the labeled targets that belong to the interval $i \in [0, m]$, the gradient boost is computed as follows

$$F_i = F_{i-1}(X) + \alpha_i h_i(X, r_{i-1}), \tag{3.5}$$

where $\alpha_i$ and $r_i$ are the regularization parameters and residuals computed with the $i^{th}$ element. The $h_i$ is a function that is trained to predict the residuals $(r_i)$ using the data $X$ for the $i^{th}$ tree. To compute $\alpha_i$, the residuals $(r_i)$ are utilized in the function

$$\arg\min_{\alpha} \sum_{i=0}^{m} L(Y_i, F_{i-1}(X_i)) + \alpha h_i(X_i, r_{i-1}), \tag{3.6}$$

where $L(Y, F(X))$ is the differential loss function.

Compared to the previously presented classifiers, a visual representation of the operation of the XGBoost algorithm is provided in Figure 3.12.

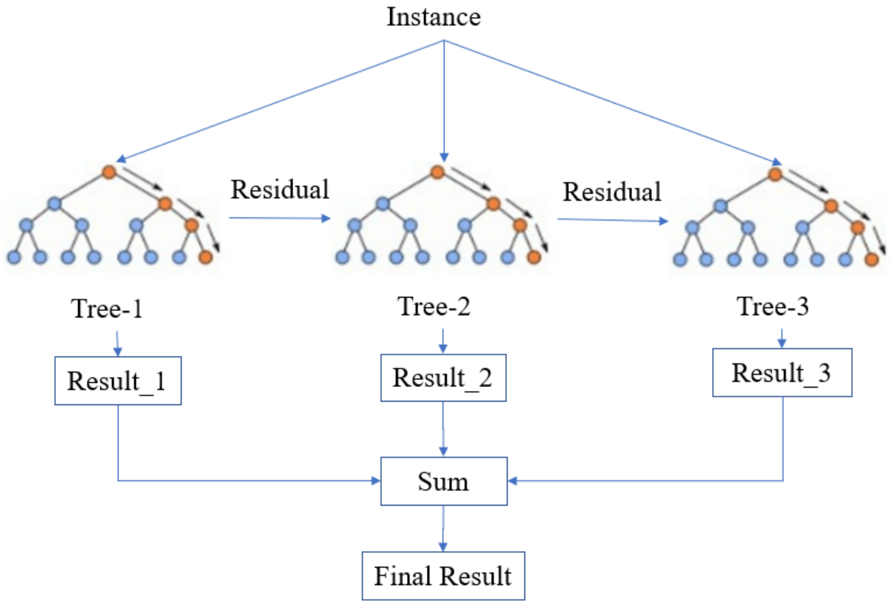


Figure 3.12: Structure of extreme gradient boosting algorithm. Source: Predicting the Risk of Chronic Kidney Disease (CKD) Using Machine Learning Algorithm [7].

## 3.4 Implementation

### 3.4.1 Program features

The functionalities of the RFSC are now presented below, exposing them and contextualizing the actions related to its applicability.

**System Options**

Basic installation actions, cleaning executables, help menus, among others.

- **-h, --help**
  Displays the help menu and exits the program.

- **-v, --version**
  Displays the version of the program and other important informations.

- **-i, --install**
  Installs all the necessary tools for the correct functioning of the program.

- **-clc, --clean**
  Cleans all generated files including the output results.

- **-cla, --clean-all**
  Cleans all training files to be used in the classifiers (CSVs).

- **-all, --run-all**
  Set of predefined parameters to use the program considering both reference-based options: FALCON-meta [39], and reference-free: XGB [254].

It makes use of Trimmomatic [234] receiving PE *reads* and metaSPAdes [97] in the Reference-based Reconstruction.

Finally, it limits the length and coverage values for the scaffolds in the filtering process to 100 and 30, respectively, also limiting the maximum and minimum thresholds for similarity in reference based analysis to a maximum of 70% and a minimum of 1%.

### Program Basic Features

Features for the definition of basic parameters when executing the program, such as definition of available threads, encryption and decryption of files, generation of synthetic sequences, among others.

- **-t, --threads *THREADS***
  Number of threads (*THREADS*) inserted by the user to be used in the program.

- **-dec, --decrypt**
  Option to decrypt all files in */Data_Security/Decrypted_Data*.

- **-enc, --encrypt**
  Option to encrypt all files in */Data_Security/Encrypted_data*.

- **-tmm, --set-threshold-max-min *MAX MIN***
  Define maximum (*MAX*) and minimum (*MIN*) percentage of thresholds for similarity in reference based analysis.

- **-dlc, --set-len-cov *LEN COV***
  Define the length (*LEN*) and coverage (*COV*) values for the scaffolds filtering process.

- **-synt, --synthetic *FILE1:FILE3***
  Option to generate a synthetic sequence using three reference files (for testing purposes).

- **-gad, --gen-adapters**
  Generate a FASTA file with adapters for trimming phase.

- **-orf, --orf-finder**
  Perform DNA sequence translation for amino acids using the ORFfinder [239] tool, finding all open reading frames (ORF) and removing stop codons.

- **-orfd, --orf-dataset *TOOL DOMAIN***
  Similar to the previously option, converts nucleotide sequences presented in the NCBI databases into protein sequences, allowing the user to select the tool to use (*TOOL*), between ORFfinder [239] and OrfM [240], and defining the domain to be converted through the *DOMAIN* parameter.

- **-efetch, --efetch-fasta *ID FOLDER***
  Makes use of the entrez efetch [236] tool to download a nucleotide using an Nucleotide Identifier (*ID*) and selecting the destination storage in the *FOLDER* parameter, that could be the *RefBased* or *RefFree* folder.

### Build Databases Features

Options that enables the download and build of NCBI databases locally.

- **-bviral, --build-ref-virus**
  Option to build a reference database for virus from NCBI.
- **-bbact, --build-ref-bacteria**
  Option to build a reference database for bacterias from NCBI.
- **-barch, --build-ref-archaea**
  Option to build a reference database for archaeas from NCBI.
- **-bprot, --build-ref-protozoa**
  Option to build a reference database for protozoas from NCBI.
- **-bfung, --build-ref-fungi**
  Option to build a reference database for fungis from NCBI.
- **-bplan, --build-ref-plant**
  Option to build a reference database for plants from NCBI.
- **-bmito, --build-ref-mitochondrial**
  Option to build a reference database for mitochondrial genomes from NCBI.
- **-bplas, --build-ref-plastid**
  Option to build a reference database for plastids from NCBI.

### CSV Generation Features

Generation of CSV files for training (and testing) the classifier models.

- **-ncd, --nc-dna-csv *DOMAIN***
  Compresses the sequences of a selected *DOMAIN* and generates a CSV file for that DNA NCBI dataset.
- **-nca, --nc-aa-csv *DOMAIN***
  Compresses the sequences of a selected *DOMAIN* and generates a CSV file for that AA NCBI dataset.
- **-gc, --gc-content-csv *DOMAIN***
  Analyses the percentage of GC-Content in each sequence of the chosen NCBI database *DOMAIN*.
- **-lenseq, --len-dna-aa-csv *DOMAIN***
  Analyses the lengths of DNA and AA sequences in the chosen NCBI database *DOMAIN*.
- **-train-test, --train-test-dataset-csv *TRAIN_PARTITION***
  Option to divide the dataset into a train and test dataset for testing purposes, setting the *TRAIN_PARTITION* parameter as a value between 0 and 1 for the definition of the train partition percentage.
- **-sdataset, --small-dataset-csv *MAX_SAMPLES***
  Option to create a reduced dataset with a maximum number of samples (*MAX_SAMPLES*) for each domain.

### Reference-Free Reconstruction Features

Features that enables the performance of trimming and/or *de-novo* genome assembly.

- **-trim, --filter *TOOL MODE***
  Application of trimming operations in order to filter the *reads* present in the *Input_Data/ReferenceBased/* folder using a specific tool defined by the *TOOL* parameter, which can be the Trimmomatic [234] (TT) or the FASTP [235] (FP), and selecting the type of end *reads* in the MODE as being PE or SE.

- **-rda, --run-de-novo**
  Application of a *De-Novo* sequence assembly tool, the metaSPAdes [97], to sequences arriving from the trimming stage.

### Reference-Based Classification Features

Options that allows the reference-based classification.

- **-rfa, --run-falcon *MODE DOMAIN***
  Applies a reference-based classification tool, FALCON-meta [39], to sequences from the reference-free reconstruction stage, selecting the mode of analysis of the previously generated scaffold nodes through the *MODE* parameter (that could use the complete scaffold: SO, or split the scaffold into different smaller nodes: RM). Finally, it enables the user to select the domain to analyze through the *DOMAIN* parameter.

- **-rbr, --run-blastn-remote**
  Applies a reference-based classification tool, similar to the previous option, but using instead remote Blastn [238] for remote access to the NCBI databases.

### Reference-Free Classification Features

Options that allows the use of machine learning classifiers in order to perform reference-free classification.

- **-gnb, --run-gaussian-naive-bayes-classifier *NUM_DOMAINS PREDICTORS***
  Applies a reference-free classifier: Gaussian Naive Bayes (GNB) Classifier, to the sequence(s) presented in the *Input_Data/ReferenceFree* folder.

  It allows the user to select the number of domains present for classification through the parameter *NUM_DOMAINS* (default: 8) and the desired predictors to be used in the classification through the parameter *PREDICTORS* following a binary logic described below:

  $'1111' -> $ Applies all the predictors;

  $'0001' -> $ Applies only the DNA compression predictor;

  $'0010' -> $ Applies only the AA compression predictor;

  $'0011' -> $ Applies only the GC-Content predictor;

  $'0100' -> $ Applies only the DNA length predictor;

  $'0101' -> $ Applies only the AA length predictor;

$'0110' - >$ Applies both the DNA and AA compression predictors;

$'0111' - >$ Applies the DNA and AA compression, and the GC-Content predictors;

$'1000' - >$ Applies the DNA and AA compression, the GC-Content, and the DNA length predictors;

$'1001' - >$ Applies the DNA and AA compression, the GC-Content, and the DNA and AA length predictors;

$'1010' - >$ Applies the DNA and AA compression, together with the DNA and AA length predictors;

- **-knn, --run-k-nearest-neighbor-classifier *K***
  Applies a reference-free classifier: K-Nearest Neighbor (KNN) Classifier, to the sequence(s) presented in the *Input_Data/ReferenceFree* folder, enabling the user to chose the *K* neighbors to be considered in the prediction.

- **-xgb, --run-xgboost**
  Applies a reference-free classifier: eXtreme Gradient Boosting (XGB) Classifier, to the sequence(s) presented in the *Input_Data/ReferenceFree* folder.

### Test Classifier Performance Features

Set of options that allows the retrieval of accuracy levels concerning the predictive quality of the classifiers.

- **-testKNN *MODE***
  Testing mode developed for the KNN classifier where the *MODE* parameter allows the toggles between using a train and test database (*--test*) and using a Cross-Validation technique, defining which domain to test (i.e. *--viral*).

- **-testXGB *MODE***
  Testing mode developed for the XGB classifier where the *MODE* parameter allows the toggles between using a train and test database (*--test*) and using a Cross-Validation technique, defining which domain to test (i.e. *--viral*).

- **-testGNB *PERCENTAGE***
  Testing mode developed for the GNB classifier that allows the use of a train and test database where the *PERCENTAGE* represents the percentage of the dataset reserved for the training section.

- **-testGNB-CV, --testGNB-CrossV *DOMAIN***
  Testing mode developed for the GNB classifier that allows the use of a Cross-Validation technique where the *DOMAIN* represents the domain that is going to be tested.

- **-aKNN, --accuracy-KNN *AC-MODE T-MODE***
  Analyse the accuracy of the KNN classifier when testing it against a known dataset, where the *AC-MODE* toggles between a simple accuracy mean (*Accuracy*) and a weighted F1-Score (*F1Score*), and the *T-MODE* toggles between the Cross-Validation Method (*CV*) and the Train-Test Database (*Test*).

  There is a need to run the option ***-testKNN MODE*** first.

- **-aXGB, --accuracy-XGB *AC-MODE T-MODE***
  Analyse the accuracy of the XGB classifier when testing it against a known dataset, where the *AC-MODE* toggles between a simple accuracy mean (*Accuracy*) and a weighted F1-Score (*F1Score*), and the *T-MODE* toggles between the Cross-Validation Method (*CV*) and the Train-Test Database (*Test*).

  There is a need to run the option ***-testXGB MODE*** first.

- **-aGNB, --accuracy-GNB *AC-MODE T-MODE TRAIN-PERCENTAGE***
  Analyse the accuracy of the GNB classifier when testing it against a known dataset, where the *AC-MODE* toggles between a simple accuracy mean (*Accuracy*) and a weighted F1-Score (*F1Score*), and the *T-MODE* toggles between the Cross-Validation Method (*CV*) and the Train-Test Database (*Test*).

  An extra *PERCENTAGE* parameter is introduced in this option to specify the percentage of the dataset used for training purposes (in case of Cross-Validation the parameter should be set to '0').

  There is a need to run the option ***-testGNB PERCENTAGE*** or ***-testGNB-CV DOMAIN*** first.

### 3.4.2 Running in Docker Container

The Docker Container is used to allow the full replication of the experiments. In order to run the program using a Docker container, the following steps will need to be performed:

```
git clone https://github.com/cobilab/RFSC
cd RFSC
docker-compose build
docker-compose up -d && docker exec -it rfsc bash && docker-compose down
```

### 3.4.3 Install Program and Dependencies Locally

In order to install this tool, the following steps will need to be performed:

```
git clone https://github.com/cobilab/RFSC
cd RFSC
./RFSC.sh --install
```

### 3.4.4 Re-building NCBI Reference Databases

If there is interest in re-building the NCBI reference databases, the RFSC can once again be used as follows:

```
./RFSC.sh --build-ref-virus --build-ref-bacteria --build-ref-archaea --build-ref-protozoa --build-ref-fungi --build-ref-plant --build-ref-mitochondrial --build-ref-plastid
```

### 3.4.5 Running Examples

In order to help to understand the interaction with the program, some examples of how to use it will be presented.

### Reference-Free Reconstruction of Synthetic Sequences

Some genomes will be retrieved from the NCBI repository using the entrez efetch [236] tool in order to generate a synthetic sequence, subsequently proceeding to a Reference-Free Reconstruction of the same. These steps are shown in the set of commands below.

```
./RFSC.sh --clean y
./RFSC.sh --threads 8 --gen-adapters
./RFSC.sh --efetch-fasta 155971 Input_Data/EntrezGenomes
./RFSC.sh --efetch-fasta EF491856.1 Input_Data/EntrezGenomes
./RFSC.sh --efetch-fasta MT682520 Input_Data/EntrezGenomes
./RFSC.sh -synt Input_Data/EntrezGenomes/155971.fna Input_Data/EntrezGenomes/EF491856.1.fna Input_Data/EntrezGenomes/MT682520.fna
./RFSC.sh -trim TT PE --run-de-novo
```

### Reference-Based Classification

If the reference databases have already been built, the Reference-Free Reconstruction stage is finished and will be needed to carry out a Reference-Based Classification, FALCON-meta can be used for that purpose.

```
./RFSC.sh --threads 8 --set-len-cov 100 3 --set-threshold-max-min 70 1 --run-falcon SO Viral
```

### Reference-Free Classification

Finally, if there is a requirement to classify a genome using only a Reference-Free Classification method, the XGBoost method can be used.

Below are the commands that exemplify how to download a viral genome (GeneID: 155971 that corresponds to a *B19* genome, also known as *Parvovirus*) from the NCBI repository using the entrez efetch [236] tool, and submitting it to a Reference-Based Classifier, in this case the Gaussian Naive bayes, K-Nearest Neighbor, and XGBoost classifier.

```
./RFSC.sh --threads 8 --efetch-fasta 155971 RefFree  # Download a viral genome
./RFSC.sh --run-gaussian-naive-bayes-classifier 1111 # Gaussian Naive Bayes
./RFSC.sh --run-k-nearest-neighbor-classifier 2      # K-nearest neighbor
./RFSC.sh --run-xgboost                              # XGBoost
```

## 3.4.6 Availability of source code and requirements

- Project name: RFSC
- Project home page: `http://github.com/cobilab/RFSC`
- RRID: SCR_021724
- biotools: rfsc
- Operating system(s): Platform independent
- Programming language: bash and python
- Other requirements: Conda
- License: GNU GPL

### 3.4.7 Software and Hardware recommendations

Laptop computer running Linux Ubuntu (for example, 18.04 LTS or higher) with GCC (`https://gcc.gnu.org`), Conda (`https://docs.conda.io`) and CMake (`https://cmake.org`) installed. The hardware must contain at least 8 GB of RAM, and a 800 GB disk. In turn, if the database is not re-built, it is only needed near 10 GB of space. There is, however, a substantial space disk increase need for applications where multiple and curated databases are merged for higher diversity. Therefore, this option requires a disk space according to the size of the databases in use, and the sequencing reads space.

# Chapter 4

# Benchmark

## 4.1 Methodology

As a way to test and validate the classification methodology developed, there was a need to adopt robust test measures that would enable the most realistic results to be obtained for the analysis. For this purpose, two testing methods were implemented for all machine learning models implemented, namely the K-Fold Cross-Validation [255] and the Train-Test Split [256] Database.

### K-Fold Cross-Validation

K-Fold Cross-Validation, one variation of Cross-Validation [255], was one of the methods chosen to validate the implemented models. This resampling procedure was applied using a K=5, thus giving rise to a 5-Fold Cross-Validation.

This methodology was applied to all implemented models following the guidelines presented below, although with some differences described next. In Gaussian Naive Bayes, a 5-Fold Cross-Validation was applied to each domain, individually, presented in the NCBI dataset (built locally). In contrast, in the other two classification methods (KNN and XG-Boost), the NCBI dataset mentioned above was divided into Training-Testing datasets using a 75%-25% rule, where the 5-Fold Cross-Validation was applied to the training dataset.

### Train-Test Split

Another of the techniques used for evaluating the performance of models was the Test-Train Split Dataset. In this method, the databases of all domains are concatenated and submitted to a split process that, in a pseudo-random way, selects 75% of the dataset for training purposes and 25% for testing purposes. This method seeks to simulate a more realistic test environment that would allow to obtain results as similar as possible to a real use case.

Passing the testing phase, it is necessary to evaluate the quality of the data in order to analyze the accuracy of the implemented models. For this purpose, two analysis methods are used in order to calculate the percentage of success of each model, these being the Accuracy Metric and the Weighted F1-Score.

### Accuracy Metric

One of the selected methods was the Accuracy, however in this method there is no distinction between True Positives (TP) and False Negatives (FN), thus making a measure of all the correctly identified cases.

The Accuracy can then be calculated according to

$$\mathcal{A}ccuracy = \frac{TP + TN}{(TP + FP + TN + FN)}. \tag{4.1}$$

In Eq. 4.1 is implemented a ratio between all correctly labeled sequences and all the sequences, with TP, FP, TN and FN, corresponding to the True Positive, False Positive, True Negatives and False Negative, respectively.

This method, even though it is quite reliable, does not differentiate imbalanced classes, giving equal importance to all classes.

### Weighted F1-Score

As a way to solve the problem of imbalanced classes, a second evaluation method was selected, the Weighted F1-Score.

This method is based on the F1-Score which can be calculated according to

$$\mathcal{F}1Score = (\frac{(\frac{TP}{TP+FN})^{-1} + (\frac{TP}{TP+FP}))^{-1}}{2})^{-1}. \tag{4.2}$$

In Eq. 4.2 is implemented a harmonic mean of Precision and Recall, with TP, FP and FN, corresponding respectively to the True Positive, False Positive and False Negative. With this method it is possible to obtain more conservative data in relation to Accuracy Metric.

The use of an average *Weighted* parameter allows to calculate metrics for each domain in order to determine the number of true instances for each domain, specifically

$$\mathcal{F}1Score(Weighted) = (\frac{\beta^2}{1+\beta^2}(\frac{TP}{TP+FN})^{-1} + \frac{1}{1+\beta^2}(\frac{TP}{TP+FP})^{-1})^{-1}. \tag{4.3}$$

In Eq. 4.3 a variable $\beta$ appears, representing the increasing of sensitivity when compared to specificity. Still in this equation, the variables TP, FP, and FN, correspond respectively to the True Positive, False Positive, and False Negative.

## 4.2 Results

This section describes the results obtained regarding the accuracy and F1-Score metrics for the classification task using the RFSC. Two types of datasets have been used for benchmarking, namely a synthetic dataset and a natural dataset (mentioned as real). The synthetic dataset has been constructed to simulate the characteristics of progressive mutations for analysis of the classification accuracy deviation.

### 4.2.1 Synthetic Data

To test the RFSC pipeline's capability to deal with pseudo-random mutations (uniform distribution) in reconstruction followed by the classification, we used a set of 140 randomly selected natural genomic sequences and applied different levels of Single Nucleotide Polymorphisms (SNP).

The randomly-selected selected sequences were divided into different groups, specifically:

- 20 Viral sequences;
- 20 Bacterial sequences;
- 20 Archaea sequences;
- 20 Fungi sequences;
- 10 Plant sequences;
- 10 Protozoa sequences;
- 20 Mitochondria sequences;
- 20 Plastid sequences.

This number of sequences was selected to allow a better balance between classes of different domains and prevent the results from becoming as less biased as possible.

Using GTO [15], different levels of mutations (SNP) were applied to the sequences, ranging from 0% to 10%. Then, we used the output of each mutated sequence for recreating the process of sequencing using the ART tool [257]. The output of the ART was FASTQ *reads* containing the applied mutations. Finally, the FASTQ *reads* of each mutated sequence were used as input data to the RFSC pipeline.

The percentage of successful predictions was subsequently analyzed using both the Accuracy metric and the Weighted F1-Score after the automatic reconstruction and classification by the RFSC pipeline, as shown in Table 1.

Table 1: Accuracy and F1-score results obtained for the classification of 140 synthetic sequences and the respective instances with the mutation rates using Gaussian Naive Bayes (GNB), k-Nearest Neighbors (KNN) and eXtreme Gradient Boosting (Xgboost) classifiers. These results were obtained for synthetic sequences after mutations were applied in the sequences, ranging from 0% to 10%.

| **Results: Synthetic data** | | | | | | |
|---|---|---|---|---|---|---|
| **Mutation** | **GNB** | | **KNN** | | **Xgboost** | |
| **(%)** | **Accuracy** | **F1Score** | **Accuracy** | **F1Score** | **Accuracy** | **F1Score** |
| **0** | 29.290 | 22.097 | 80.000 | 78.773 | 90.710 | 90.001 |
| **1** | 29.290 | 22.328 | 80.710 | 79.902 | 87.860 | 87.224 |
| **2** | 17.140 | 10.340 | 70.000 | 69.802 | 87.140 | 86.531 |
| **4** | 15.710 | 9.443 | 55.000 | 51.110 | 69.290 | 65.755 |
| **6** | 15.000 | 8.914 | 47.140 | 44.306 | 65.000 | 60.958 |
| **8** | 14.290 | 7.420 | 50.000 | 48.163 | 63.570 | 59.347 |
| **10** | 15.000 | 7.798 | 50.710 | 50.030 | 61.430 | 57.645 |

As reported in Table 1, for all mutation rates, the best results were obtained using the XGBoost classifier. Specifically, for 0% mutation, the accuracy obtained was 90.710% and the F1-score of 90.001%, while for 10% mutation, the accuracy was 61.430% and the F1-score of 57.645%. These results improve the second-best classifier (KNN) on average 10.3%, and 12.18% accuracy and F1-score, respectively. This improvement shows that XGBoost is the most suitable classifier since it performs better than other classifiers even when a high mutation rate is applied to the generated data.

On the other hand, the robustness of the results shows that the predictors are resistant to significant mutations. Notice that these results are the outcome of automatic classification after automatic genome reconstruction using a balanced dataset while containing simulated levels of mutations. Moreover, genomic sequences often suffer errors in the sequencing and assembly process. The fact that these predictors are robust to noise (in this event caused by inducing mutations on the sequence) demonstrates that they are highly suitable for classifying genomic sequences.

The simulated mutations have a uniform distribution, changing the normalized compression values to higher complexity and approximating the GC-content to a uniform distribution. Notwithstanding, the RFSC pipeline can adapt and still perform with high accuracy in the highest mutations levels. In a real scenario, it is not common to have this dramatic change in the distribution complexity. Nevertheless, it was used the most complex distribution as random mutations to understand this type of limit and adaptability of RFSC. Therefore, in a real scenario, it should be expected equal or higher adaptability since the distribution contains, in the worst case, this maximum complexity for these levels of mutations.

In order to apply a certain percentage of mutation in a set of genomes, it is first necessary to select those that will participate in this process, place them in a directory together with the code provided below and execute the same, introducing the desired mutation percentage as an argument (i.e. 0.01).

```bash
#!/bin/bash

MUTATION_PERC=$1;   # ex. 0.01
PERC=$(echo "$MUTATION_PERC * 100" | bc)

for file in *.gz
do
    genome=${file%".gz"}
    echo "Starting analysing the file: $genome"

    gunzip $file

    gto_fasta_mutate -e $MUTATION_PERC -a < $genome > Temporary.fna
    rm $genome
    mv Temporary.fna $genome
    gzip $genome

    echo "$genome has suffered a $PERC mutation!"
done
```

After this procedure, the mutated genomes must be moved in the *RFSC/Input_Data/ReferenceFree* folder, and then the script must be executed using the desired prediction mechanisms.

### 4.2.2 Real Data

Identifying and classifying unknown DNA sequences from metagenomic samples is a complex challenge to benchmark, mainly because a high quantity of unknown genomes very rarely exists, where most of the samples contain highly similar to similar DNA sequences. Specifically, the currently reported and verified genomes number that is new (or significantly dissimilar) is minimal, which difficults the process of achieving an accuracy ratio of a methodology using a fair number of predictions (at least more than one million).

Therefore, to provide a fair classification benchmark using only the reference-free approach, it was assumed that each genome from the database (described previously at Table 2) is new and has already been assembled. For providing the accuracy metrics, it was used the accuracy and F1-score classification described previously and present the results in Table 2.

Table 2: Accuracy and F1-score classification results of raw data (obtained from the NCBI database) using a random classification process ($p_{Random}$), Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN), and eXtreme Gradient Boosting (Xgboost) classifiers.

| **Real Data** | | | | |
|---|---|---|---|---|
| **Classification** | $p_{Random}$ | **GNB** | **KNN** | **Xgboost** |
| **Accuracy (%)** | 12.500 | 71.560 | 86.249 | **96.970** |
| **F1Score (%)** | - | 46.119 | 86.210 | **96.960** |

As shown in the results obtained for the classification of real data, the XGBoost classifier obtained the best classification. Specifically, the XGBoost classifier achieved an accuracy of 96.97% and a weighted F1-score of 96.96%, incrementing 10% relatively to the second-best classifier KNN.

Furthermore, since the classification task was performed by utilizing the natural data directly, part of the inaccuracy of 3.04% could be explained by possible errors in the assembly process of the original sequence or eventual sub-sequence contamination of parts of the genomes. Moreover, several genomes were reconstructed many years ago, using older methods that have been improved over the years. Since some of the sequences are singular, they are considered references. Therefore, by running the entire pipeline, including the reconstruction from the reads (if they were all available), better results may be achieved.

Overall, these results show the potential of this pipeline to accurately perform the discovery and classification of unknown DNA sequences in metagenomic samples, specifically in the most complex areas.

# Chapter 5

# Conclusions

This thesis describes a computational pipeline for efficient reconstruction and accurate classification of unknown DNA sequences in metagenomic samples. A fully automatic and flexible pipeline was developed, additionally allowing secure storage for sensitive data.

This pipeline (RFSC) combines reference-free approaches with reference-based approaches, both using alignments and alignment-free methods. Moreover, both DNA and protein sequence levels are used, where the latter is automatically predicted and extracted.

The features-based classification (reference-free classification) classifies DNA sequences without resorting directly to the reference genomes, but instead to features that the biological sequences share. The extraction of features is provided by five predictors, namely the normalized compression and normalized lengths (both for DNA and amino acids sequences) and the GC-content of DNA sequences.

Considering all the benchmark results described in this manuscript, it can be concluded that using a set of predictors (which individually show themselves to be insufficient) and an efficient classifier, it is possible to make classifications with a high degree of accuracy. Specifically, two primary benchmarks were performed: reconstruction and classification of raw data with increasing degrees of synthetic mutations and classification of reference biological data obtained from the NCBI database.

The XGBoost classifier achieved the best performance in both classification tasks, obtaining on average a 10% improvement over the second-best classifier. Furthermore, this improvement was maintained even when a higher mutation rate was applied to the synthetic sequences. Genomic sequences often suffer errors in the sequencing and assembly process. The fact that these predictors are robust to noise (in this case caused by inducing mutations on the sequence and by eventual reconstruction imprecision) proves that they are suitable for classifying genomic sequences in the most complex scenarios.

Regarding the classification of real data, it was obtained an accuracy of 96.97% and a weighted F1-score of 96.96%. As far as it is known, this high accuracy for unknown DNA sequences using a reference-free approach has not been reported in the state-of-the-art. Moreover, it is believed that these results are inspiring since they clearly show the capability of this method to perform the discovery and classification of unknown DNA sequences in metagenomic samples. This pipeline can now be used in the most challenging natures, namely in clinical, archaeogenomics, or exobiology areas.

## 5.1 Future Work

Although this thesis has already supported the theory that it is possible to classify organisms without resorting exclusively to reference-based mechanisms, and the results obtained largely embrace this assertion, there is always room for improvement and evolution.

Following this idea, the next steps to be taken would be the enrichment of the training datasets given to the classifiers in the learning phase in order to further mitigate the possibility of prediction error.

Another addition that would bring added value to the project would be the introduction of support for new domains and intra-domain in the prediction analysis.

Lastly, the study and possible introduction of new predictors in the ensemble, such as specific group features, minimal absent words, nucleotide and/or aminoacid distribution, chemical distributions properties, normalized similar-gene count and normalized BDM [258], could also translate into an even better quality of the performed predictions.

# Bibliography

[1] Christopher Quince, Alan W Walker, Jared T Simpson, Nicholas J Loman, and Nicola Segata. Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology*, 35:833–844, 2017.

[2] Diogo Pratas, Morteza Hosseini, Jorge M Silva, and Armando J Pinho. A reference-free lossless compression algorithm for DNA sequences using a competitive prediction of two classes of weighted models. *Entropy*, 21(11), 2019.

[3] Diogo Pratas and Armando J Pinho. On the Approximation of the Kolmogorov Complexity for DNA Sequences. *Springer, Cham*, 2017.

[4] Milton Silva, Diogo Pratas, and Armando J Pinho. Efficient DNA sequence compression with neural networks. *GigaScience*, 9(11), 2020.

[5] Morteza Hosseini, Diogo Pratas, and Armando J Pinho. AC: A compression tool for amino acid sequences. *Interdisciplinary Sciences: Computational Life Sciences*, 11(1):68–76, 2019.

[6] Rajeev Raizada and Yune Lee. Smoothness without smoothing: Why gaussian naive bayes is not naive for multi-subject searchlight studies. *PloS one*, 8:e69566, 2013.

[7] Weilun Wang, Goutam Chakraborty, and Basabi Chakraborty. Predicting the risk of chronic kidney disease (ckd) using machine learning algorithm. *Applied Sciences*, 11:202, 2020.

[8] Leslie G Biesecker, Wylie Burke, Isaac Kohane, Sharon E Plon, and Ron Zimmern. Next-generation sequencing in the clinic: are we ready? *Nat Rev Genet*, 13(11):818–824, 2012.

[9] Charles Y Chiu and Steven A Miller. Clinical metagenomics. *Nature Reviews Genetics*, 20(6):341–355, 2019.

[10] Jarrad T Hampton-Marcell, Jose V Lopez, and Jack A Gilbert. The human microbiome: an emerging tool in forensics, 2017.

[11] Antonio Amorim, Filipe Pereira, Cíntia Alves, and Oscar García. Species assignment in forensics and the challenge of hybrids. *Forensic Science International: Genetics*, 48:102333, 2020.

[12] Emiley A Eloe-Fadrosh, David Paez-Espino, Jessica Jarett, Peter F Dunfield, Brian P Hedlund, Anne E Dekas, Stephen E Grasby, Allyson L Brady, Hailiang Dong, Brandon R

Briggs, et al. Global metagenomic survey reveals a new bacterial candidate phylum in geothermal springs. *Nature communications*, 7(1):1–10, 2016.

[13] Cristian Del Fabbro, Simone Scalabrin, Michele Morgante, and Federico M Giorgi. An extensive evaluation of read trimming effects on illumina ngs data analysis. *PLoS ONE*, 8(12), 2013.

[14] Elaine R Mardis. DNA sequencing technologies: 2006-2016. *Nat Protoc*, 12(2):213–218, 2017.

[15] João R Almeida, Armando J Pinho, José L Oliveira, Olga Fajarda, and Diogo Pratas. GTO: A toolkit to unify pipelines in genomic and proteomic research. *SoftwareX*, 12, 2020.

[16] Torsten Thomas, Jack Gilbert, and Folker Meyer. Metagenomics-a guide from sampling to data analysis. *Microbial informatics and experimentation*, 2(1):1–12, 2012.

[17] Irina Abnizova, Steven Leonard, Tom Skelly, Andy Brown, David Jackson, Marina Gourtovaia, Guoying Qi, Rene Te Boekhorst, Nadeem Faruque, Kevin Lewis, and Tony Cox. Analysis of context-dependent errors for illumina sequencing. *J Bioinform Comput Biol*, 10(2), 2012.

[18] Rene Te Boekhorst, F M Naumenko, N G Orlova, Elvira Rasimovna Galieva, A M Spitsina, Irina Chadaeva, Yuriy Orlov, and Irina Abnizova. Computational problems of analysis of short next generation sequencing reads. *Vavilov Journal of Genetics and Breeding*, 20(6):746–755, 2016.

[19] Vitor C Piro, Temesgen H Dadi, Enrico Seiler, Knut Reinert, and Bernhard Y Renard. ganon: precise metagenomics classification against large and up-to-date sets of reference sequences. *Bioinformatics*, 36:i12–i20, 2020.

[20] Shifu Chen, Changshou He, Yingqiang Li, Zhicheng Li, and E Melancon III Charles. A computational toolset for rapid identification of sars-cov-2, other viruses, and microorganisms from sequencing data. *Briefings in Bioinformatics*, 22(2):924–935, 2021.

[21] Atlas Khan, Qian Liu, Xuelian Chen, Andres Stucky, Parish P Sedghizadeh, Daniel Adelpour, Xi Zhang, Kai Wang, and Jiang F Zhong. Detection of human papillomavirus in cases of head and neck squamous cell carcinoma by rna-seq and virtect. *Molecular Oncology*, (13):829–839, 2018.

[22] Xun Chen, Jason Kost, Arvis Sulovari, Nathalie Wong, Winnie Liang, Jian Cao, and Dawei Li. A virome-wide clonal integration analysis platform for discovering cancer viral etiology. *Genome Research*, 2019.

[23] Brett E Pickett, Eva L Sadat, Yun Zhang, Jyothi M Noronha, R Burke Squires, Victoria Hunt, Mengya Liu, Sanjeev Kumar, Sam Zaremba, Zhiping Gu, et al. ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic acids research*, 40(D1):D593–D598, 2012.

[24] Michael Vilsker, Yumma Moosa, Sam Nooij, Vagner Fonseca, Yoika Ghysens, Korneel Dumon, Raf Pauwels, Luiz Carlos Alcantara, Ewout Vanden Eynden, Anne-Mieke Vandamme, Koen Deforche, and Tulio de Oliveira. Genome detective: an automated system for virus identification from high-throughput sequencing data. 35(5):871–873, 2019.

[25] Folker Meyer, Daniel Paarmann, Mark D'Souza, Robert Olson, Elizabeth M Glass, Michael Kubal, Tobias Paczian, Alex Rodriguez, Rick Stevens, Andreas Wilke, et al. The metagenomics RAST server–a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC bioinformatics*, 9(1):1–8, 2008.

[26] Daniel H Huson, Alexander F Auch, Ji Qi, and Stephan C Schuster. MEGAN analysis of metagenomic data. *Genome research*, 17(3):377–386, 2007.

[27] Stuart M Brown, Hao Chen, Yuhan Hao, Bobby P Laungani, Thahmina A Ali, Changsu Dong, Carlos Lijeron, Baekdoo Kim, Claudia Wultsch, Zhiheng Pei, et al. MGS-Fast: metagenomic shotgun data fast annotation using microbial gene catalogs. *GigaScience*, 8(4):giz020, 2019.

[28] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature methods*, 12(10):902–903, 2015.

[29] Florian P Breitwieser, Jennifer Lu, and Steven L Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, 20(4):1–15, 2017.

[30] Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and David L Wheeler. Genbank. *Nucleic Acids Res.*, 33:D34–D38, 2005.

[31] Daniel H Haft, Michael DiCuccio, Azat Badretdin, Vyacheslav Brover, Vyacheslav Chetvernin, Kathleen O'Neill, Wenjun Li, Farideh Chitsaz, Myra K Derbyshire, Noreen R Gonzales, and et al. Refseq: an update on prokaryotic genome annotation and curation.

[32] Asami Fukuda, Yuichi Kodama, Jun Mashima, Takatomo Fujisawa, and Osamu Ogasawara. DDBJ update: streamlining submission and access of human data. *Nucleic Acids Research*, 49(D1):D71–D75, 2021.

[33] Masanori Arita, Ilene Karsch-Mizrachi, and Guy Cochrane. The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 49(D1):D121–D124, 2021.

[34] Hans-Werner Mewes, Dmitrij Frishman, Ulrich Güldener, Gertrud Mannhaupt, Klaus Mayer, Martin Mokrejs, Burkhard Morgenstern, Martin Münsterkötter, Stephen Rudd, and B Weil. MIPS: a database for genomes and protein sequences. *Nucleic acids research*, 30(1):31–34, 2002.

[35] Andreas Wilke, Jared Bischof, Wolfgang Gerlach, Elizabeth Glass, Travis Harrison, Kevin P Keegan, Tobias Paczian, William L Trimble, Saurabh Bagchi, Ananth Grama, et al. The MG-RAST metagenomics database and portal in 2015. *Nucleic acids research*, 44(D1):D590–D594, 2016.

[36] Scott McGinnis and Thomas L Madden. Blast: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res*, 32:W20–W25, 2004.

[37] Derrick E Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with Kraken 2. *Genome biology*, 20(1):1–13, 2019.

[38] Daehwan Kim, Li Song, Florian P Breitwieser, and Steven L Salzberg. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome research*, 26(12):1721–1729, 2016.

[39] Diogo Pratas, M Hosseini, G Grilo, A J Pinho, R M Silva, T Caetano, J Carneiro, and F Pereira. Metagenomic composition analysis of an ancient sequenced polar bear jawbone from svalbard. *Genes*, 9(9):445, 2018.

[40] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with kaiju. *Nature Communications*, 7, 2016.

[41] Tracey Allen K Freitas, Po-E Li, Matthew B Scholz, and Patrick SG Chain. Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. *Nucleic acids research*, 43(10):e69–e69, 2015.

[42] H Ye Simon, Katherine J Siddle, Daniel J Park, and Pardis C Sabeti. Benchmarking metagenomics tools for taxonomic classification. *Cell*, 178(4):779–794, 2019.

[43] Migun Shakya, Chien-Chi Lo, and Patrick SG Chain. Advances and challenges in metatranscriptomic analysis. *Frontiers in genetics*, 10:904, 2019.

[44] Feng Gao and Chun-Ting Zhang. Gc-profile: a web-based tool for visualizing and analyzing the variation of gc content in genomic sequences. *Nucleic Acids Research*, 34:W686–W691, 2006.

[45] Allison Piovesan, Maria Chiara Pelleri, Francesca Antonaros, Pierluigi Strippoli, Maria Caracausi, and Lorenza Vitale. On the length, weight and gc content of the human genome. *BMC Research Notes*, 12(106), 2019.

[46] Mohammad Shahnaz, Mamta Chowdhary, Asha Rani, and Jyoti Parkash. Bioinformatics: an overview for cancer research. *Journal of Drug Delivery and Therapeutics*, 6(4):69–72, 2016.

[47] Ardeshir Bayat. Bioinformatics. *BMJ*, 324(7344):1018–1022, 2002.

[48] Antti Sajantila. Editors' Pick: Contamination has always been the issue!, 2014.

[49] Hongan Long, Way Sung, Sibel Kucukyildirim, Emily Williams, Samuel F Miller, Wanfeng Guo, Caitlyn Patterson, Colin Gregory, Chloe Strauss, Casey Stone, et al. Evolutionary determinants of genome-wide nucleotide composition. *Nature ecology & evolution*, page 1, 2018.

[50] F Sanger, S Nicklen, and A R Coulson. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, 74(12):5463–5467, 1997.

[51] Patrick Forterre, Jonathan Filee, Hannu Myllykallio, and Lluís Pouplana. *Origin and Evolution of DNA and DNA Replication Machineries*, pages 145–168. 2007.

[52] Stryer L Berg JM, Tymoczko JL. Biochemistry. 5th edition, 2002.

[53] Nar Singh Chauhan. Chapter 10 - metagenome analysis and interpretation. In Gauri Misra, editor, *Data Processing Handbook for Complex Biological Data Sources*, pages 139–160. Academic Press, 2019.

[54] Christian Riesenfeld, Patrick Schloss, and Jo Handelsman. Metagenomics: genomic analysis of microbial communities. *Annual review of genetics*, 38:525–52, 2004.

[55] R Staden. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Res*, 6(7):2601–2610, 1979.

[56] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, and et al. The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001.

[57] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, and et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[58] Hannelore Daniel. Genomics and proteomics: Importance for the future of nutrition research. *The British journal of nutrition*, 87 Suppl 2:S305–11, 2002.

[59] Sara Goodwin, John Mcpherson, and W Mccombie. Coming of age: Ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17:333–351, 2016.

[60] Johanna Craig. Complex diseases: Research and applications. *Nature Education*, 1(1), 2008.

[61] Karen Bulaklak and Charles A Gersbach. The once and future gene therapy. *Nature Communications*, 11(5820), 2020.

[62] Hongyi Li, Yang Yang, Weiqi Hong, Mengyuan Huang, Min Wu, and Xia Zhao. Applications of genome editing technology in the targeted therapy of human diseases: mechanisms, advances and prospects. *Signal Transduction and Targeted Therapy*, 5(1), 2020.

[63] Bilal Aslam, Madiha Basit, Muhammad Atif Nisar, Mohsin Khurshid, and Muhammad Hidayat Rasool. Proteomics: Technologies and their applications. *J Chromatogr Sci*, 55(2):182–196, 2017.

[64] Jill Adams. The proteome: Discovering the structure and function of proteins. *Nature Education*, 1(3), 2008.

[65] Allison C Galassie and Andrew J Link. Proteomic contributions to our understanding of vaccine and immune responses. *Proteomics Clin Appl*, 9(0):972–989, 2016.

[66] Jing Tang, Yunxia Wang, Yongchao Luo, Jianbo Fu, Yang Zhang, Yi Li, Ziyu Xiao, Yan Lou, Yunqing Qiu, and Feng Zhu. Computational advances of tumor marker selection and sample classification in cancer proteomics. *Computational and Structural Biotechnology Journal*, 18:2012–2025, 2020.

[67] Jagath C Rajapakse, Kai-Bo Duan, and Wee Kiang Yeo. Proteomic cancer classification with mass spectrometry data. *Am J Pharmacogenomics*, 5(5):281–92, 2005.

[68] S A Merrill and A M Mazza. Reaping the benefits of genomic and proteomic research: Intellectual property rights, innovation, and public health. *National Academies Press (US)*, 2006.

[69] Zoonomia Consortium. A comparative genomics multitool for scientific discovery and conservation. *Nature*, 587:240–245, 2020.

[70] Denis Faure and Dominique Joly. Introduction. In Denis Faure and Dominique Joly, editors, *Insight on Environmental Genomics*, pages xv–xix. Elsevier, 2016.

[71] Warren C Lathe, Jennifer M Williams, Mary E Mangan, and Donna Karolchik. Genomic data resources: Challenges and promises. *Nature Education*, 1(3):2, 2008.

[72] Wendy Baker, Alexandra van den Broek, Evelyn Camon, Pascal Hingamp, Peter Sterk, Guenter Stoesser, and Mary Ann Tuli. The embl nucleotide sequence database. *Nucleic Acids Res.*, 28(1):19–23, 2000.

[73] National Research Council (US). Space science in the twenty-first century: Imperatives for the decades 1995 to 2015: Life sciences. *National Academies Press (US)*, 1988.

[74] Michael D Lee, Aubrie O'Rourke, Hernan Lorenzi, Brad M Bebout, Chris L Dupont, and R Craig Everroad. Reference-guided metagenomics reveals genome-level evidence of potential microbial transmission from the ISS environment to an astronaut's microbiome. *Iscience*, 24(2):102114, 2021.

[75] Aristóteles Góes-Neto, Olga Kukharenko, Iryna Orlovska, Olga Podolich, Madangchanok Imchen, Ranjith Kumavath, Rodrigo Bentes Kato, Daniel Santana De Carvalho, Sandeep Tiwari, Bertram Brenig, et al. Shotgun metagenomic analysis of kombucha mutualistic community exposed to Mars-like environment outside the International Space Station. *Environmental Microbiology*, 2021.

[76] Michael P Callahan, Karen E Smith, H James Cleaves, Josef Ruzicka, Jennifer C Stern, Daniel P Glavin, Christopher H House, and Jason P Dworkin. Carbonaceous meteorites contain a wide range of extraterrestrial nucleobases. *Proceedings of the National Academy of Sciences*, 108(34):13995–13998, 2011.

[77] Aaron S Burton, Jennifer C Stern, Jamie E Elsila, Daniel P Glavin, and Jason P Dworkin. Understanding prebiotic chemistry through the analysis of extraterrestrial amino acids and nucleobases in meteorites. *Chem. Soc. Rev.*, 41:5459–5472, 2012.

[78] Yoshihiro Furukawa, Yoshito Chikaraishi, Naohiko Ohkouchi, Nanako O Ogawa, Daniel P Glavin, Jason P Dworkin, Chiaki Abe, and Tomoki Nakamura. Extraterrestrial ribose and other sugars in primitive meteorites. *Proceedings of the National Academy of Sciences*, 116(49):24440–24445, 2019.

[79] Zita Martins, Oliver Botta, Marilyn L Fogel, Mark A Sephton, Daniel P Glavin, Jonathan S Watson, Jason P Dworkin, Alan W Schwartz, and Pascale Ehrenfreund. Extraterrestrial nucleobases in the Murchison meteorite. *Earth and planetary science Letters*, 270(1-2):130–136, 2008.

[80] S Onofri, L Selbmann, L Zucconi, and S Pagano. Antarctic microfungi as models for exobiology. *Planetary and Space Science*, 52(1-3):229–237, 2004.

[81] Don A Cowan, Jean-Baptiste Ramond, Thulani P Makhalanyane, and Pieter De Maayer. Metagenomics of extreme environments. *Current opinion in microbiology*, 25:97–102, 2015.

[82] Yoav Mathov and Liran Carmel. The revolution of ancient DNA—what does genetics tell us about the past? *Front. Young Minds*, 7(24), 2019.

[83] David Gokhman, Eitan Lavi, Kay Prüfer, Mario F Fraga, José A Riancho, Janet Kelso, Svante Pääbo, Eran Meshorer, and Liran Carmel. Reconstructing the DNA methylation maps of the neandertal and the denisovan. *Science*, 344(6183):523–527, 2014.

[84] Rebekah L Rogers and Montgomery Slatkin. Excess of genomic defects in a woolly mammoth on wrangel island. *PLOS Genetics*, 13(3):1–16, 03 2017.

[85] Erika Rosengren, Arina Acatrinei, Nicolae Cruceru, Marianne Dehasque, Aritina Haliuc, Edana Lord, Cristina I Mircea, Ioana Rusu, Emilio Mármol-Sánchez, Beatrice S Kelemen, and Ioana N Meleg. Ancient faunal history revealed by interdisciplinary biomolecular approaches. *Diversity*, 13(8), 2021.

[86] Laurent Eyers, Isabelle George, Luc Schuler, Ben Stenuit, Spiros Agathos, and Said Fantroussi. Environmental genomics: exploring the unmined richness of microbes to degrade xenobiotics. *Applied Microbiology and Biotechnology*, 66:123–30, 2005.

[87] Helmholtz Association of German Research Centres. Bacteria in extremely hostile environments: New protein discovered that repairs DNA under extreme conditions. *ScienceDaily*, 2008.

[88] Andreia Cruz, Tânia Caetano, Satoru Suzuki, and Sónia Mendo. Aeromonas veronii, a tributyltin (tbt)-degrading bacterium isolated from an estuarine environment, ria de aveiro in portugal. *Marine Environmental Research*, 64(5):639–650, 2007.

[89] Diana Dias, Rita T Torres, Göran Kronvall, Carlos Fonseca, Sónia Mendo, and Tânia Caetano. Assessment of antibiotic resistance of escherichia coli isolates and screening of salmonella spp. in wild ungulates from portugal. *Research in Microbiology*, 166(7):584–593, 2015.

[90] Ivan Sovic, Karolj Skala, and Mile Sikic. Approaches to DNA de novo assembly. In *MIPRO*, volume 264, pages 351–359, 2013.

[91] Diogo Pratas, Mari Toppinen, Lari Pyöriä, Klaus Hedman, Antti Sajantila, and Maria Perdomo. A hybrid pipeline for reconstruction and analysis of viral genomes at multi-organ level. *GigaScience*, 9:1–11, 2020.

[92] Heng Li and Durbin Richard. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[93] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9:357–359, 2012.

[94] Paul Flicek and Ewan Birney. Sense from sequence reads: methods for alignment and assembly. *Nature Methods*, 6:S6–S12, 2009.

[95] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–9, 2009.

[96] Bob B Buchanan, Wilhelm Gruissem, and Russell L Jones. *Biochemistry and molecular biology of plants*. John wiley & sons, 2015.

[97] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A Pevzner. metaspades: a new versatile metagenomic assembler. *Genome Res*, 27(5):824–834, 2017.

[98] Daniel R Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Res.*, 18:821–829, 2008.

[99] Govinda M Kamath, Ilan Shomorony, Fei Xia, Thomas Courtade, and David N Tse. Hinge: Long-read assembly achieves optimal repeat resolution. *Genome Res.*, 2017.

[100] Chen-Shan Chin, David H Alexander, Patrick Marks, Aaron A Klammer, James Drake, Cheryl Heiner, Alicia Clum, Alex Copeland, John Huddleston, Evan E Eichler, Stephen W Turner, and Jonas Korlach. Nonhybrid, finished microbial genome assemblies from long-read smrt sequencing data. *Nat Methods*, 10:563–569, 2013.

[101] Tim Burland. DNASTAR's lasergene sequence analysis software. *Methods in molecular biology (Clifton, N.J.)*, 132:71–91, 2000.

[102] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein, Steven J M Jones, and İnanç Birol. Abyss: A parallel assembler for short read sequence data. *Genome Res.*, 19(6):1117–1123, 2009.

[103] Michael Alonge, Sebastian Soyk, Srividya Ramakrishnan, Xingang Wang, Sara Goodwin, Fritz J Sedlazeck, Zachary B Lippman, and Michael C Schatz. Ragoo: fast and accurate reference-guided scaffolding of draft genomes. *Genome Biology*, 20(224), 2019.

[104] Mikhail Kolmogorov, Brian Raney, Benedict Paten, and Son Pham. Ragout—a reference-assisted assembly tool for bacterial genomes. *Bioinformatics*, 30(12):i302–i309, 2014.

[105] Krisztian Buza, Bartek Wilczynski, and Norbert Dojer. Record: Reference-assisted genome assembly for closely related genomes. *International Journal of Genomics*, 2015.

[106] Bastien Chevreux. *MIRA: An Automated Genome and EST Assembler*. PhD thesis, 2005.

[107] Sergey Nurk, Brian Walenz, Arang Rhie, Mitchell Vollger, Glennis Logsdon, Robert Grothe, Karen Miga, Evan Eichler, Adam Phillippy, and Sergey Koren. Hicanu: Accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Research*, 30:gr.263566.120, 2020.

[108] Haoyu Cheng, Gregory Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature Methods*, 18:170–175, 2021.

[109] C S Chin, P Peluso, F J Sedlazeck, and et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat Methods*, 13(12):1050–1054, 2016.

[110] Fanli Zeng, Jinping Zang, Suhua Zhang, Zhimin Hao, Jin-gao Dong, and Yibin Lin. Afeap cloning: A precise and efficient method for large DNA sequence assembly. *BMC Biotechnology*, 17, 2017.

[111] Aleksey V Zimin, Guillaume Marçais, Daniela Puiu, Michael Roberts, Steven L Salzberg, and James A Yorke. The masurca genome assembler. *Bioinformatics*, 29(21):2669–2677, 2013.

[112] Florian P Breitwieser, Jennifer Lu, and Steven L Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, 20(4):1125–1136, 2017.

[113] Yu Peng, Henry Leung, Sm Yiu, and Francis Chin. Idba - a practical iterative de bruijn graph de novo assembler. In *Research in Computational Molecular Biology*, volume 6044, pages 426–440, 2010.

[114] Michael C Schatz, Adam M Phillippy, Daniel D Sommer, Arthur L Delcher, Daniela Puiu, Giuseppe Narzisi, Steven L Salzberg, and Mihai Pop. Hawkeye and AMOS: visualizing and assessing the quality of genome assemblies. *Briefings in Bioinformatics*, 14(2):213–224, 2011.

[115] Jaebum Kim, Denis Larkin, Qingle Cai, Asan, Yongfen Zhang, Ri-Li Ge, Loretta Auvil, Boris Capitanu, Guojie Zhang, Harris Lewin, and Jian Ma. Reference-assisted chromosome assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 110, 2013.

[116] Henry Qin, Qian Li, Jacqueline Speiser, Peter Kraft, and John K. Ousterhout. Arachne: Core-aware thread management. In *OSDI*, 2018.

[117] Xiangchao Gan, Oliver Stegle, Jonas Behr, Joshua G Steffen, Philipp Drewe, Katie L Hildebrand, Rune Lyngsoe, Sebastian J Schultheiss, Edward J Osborne, Vipin T Sreedharan, and et al. Multiple reference genomes and transcriptomes for arabidopsis thaliana.

[118] Ergude Bao, Tao Jiang, and Thomas Girke. Aligngraph: Algorithm for secondary de novo genome assembly guided by closely related references. *Bioinformatics (Oxford, England)*, 30:i319–i328, 2014.

[119] Eamonn J Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Da Wei, Sang-Hee Lee, and John C Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14:99–129, 2007.

[120] Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8:154–177, 2005.

[121] Morteza Hosseini, Diogo Pratas, and Armando J. Pinho. A survey on data compression methods for biological sequences. *Information*, 7(4), 2016.

[122] R Giancarlo, D Scaturro, and F Utro. Textual data compression in computational biology: Algorithmic techniques. *Computer Science Review*, 6(1):1–25, 2012.

[123] Sebastian Wandelt, Marc Bux, and Ulf Leser. Trends in genome compression. *Current Bioinformatics*, 9, 2014.

[124] Ahsan Habib, Mohammed J Islam, and Mohammad Rahman. A dictionary-based text compression technique using quaternary code. *Iran Journal of Computer Science*, 3, 2020.

[125] Minh Cao, Trevor Dix, Lloyd Allison, and Chris Mears. A simple statistical algorithm for biological sequence compression. In *In Proceedings of the Conference on Data Compression*, pages 43–52, 2007.

[126] Kirill Kryukov, Mahoko Ueda, So Nakagawa, and Tadashi Imanishi. Sequence compression benchmark (scb) database—a comprehensive evaluation of reference-free compressors for fasta-formatted sequences. *GigaScience*, 9, 2020.

[127] Kirill Kryukov, Mahoko Ueda, So Nakagawa, and Tadashi Imanishi. Nucleotide archival format (naf) enables efficient lossless reference-free compression of DNA sequences. *Bioinformatics (Oxford, England)*, 35, 2019.

[128] S Grumbach and F Tahi. Compression of DNA sequences. In *[Proceedings] DCC '93: Data Compression Conference*, pages 340–350, 1993.

[129] Stéphane Grumbach and Fariza Tahi. A new challenge for compression algorithms: Genetic sequences. *Inf. Process. Manag.*, 30:875–886, 1994.

[130] E Rivals, Jean-Paul Delahaye, Max Dauchet, and Olivier Delgrange. A guaranteed compression scheme for repetitive DNA sequences. In *Proc. of the Data Compression Conf., DCC-96*, page 453, 1996.

[131] David Loewenstern and Peter Yianilos. Significantly lower entropy estimates for natural DNA sequences. *Journal of computational biology : a journal of computational molecular cell biology*, 6:125–42, 1999.

[132] L Allison, T Edgoose, and Trevor Dix. Compression of strings with approximate repeats. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 6:8–16, 1998.

[133] Xin Chen, Sam Kwong, and Ming Li. A compression algorithm for DNA sequences and its applications in genome comparison. *Genome informatics. Workshop on Genome Informatics*, 10:51–61, 1999.

[134] A Apostolico and Stefano Lonardi. Compression of biological sequences by greedy offline textual substitution. In *Data Compression Conference Proceedings*, pages 143–152, 2000.

[135] Toshiko Matsumoto, Kunihiko Sadakane, and Hiroshi Imai. Biological sequence compression algorithms. *Genome informatics. Workshop on Genome Informatics*, 11:43–52, 2000.

[136] Xin Chen, Ming Li, Bin ma, and John Tromp. DNACompress: Fast and effective DNA sequence compression. *Bioinformatics (Oxford, England)*, 18:1696–8, 2003.

[137] I Tabus, G Korodi, and J Rissanen. DNA sequence compression using the normalized maximum likelihood model for discrete regression. In *Data Compression Conference, 2003. Proceedings. DCC 2003*, pages 253–262, 2003.

[138] Giovanni Manzini and Marcella Rastero. A simple and fast DNA compressor. *Softw., Pract. Exper.*, 34:1397–1411, 2004.

[139] E J T Lee and et al. DNAC: an efficient compression algorithm for DNA sequences. *National Taiwan University*, 1(1), 2004.

[140] Neva Durand and R E Ladner. Grammar-based compression of DNA sequences. 2004.

[141] Behshad Behzadi and Fabrice Le Fessant. DNA compression challenge revisited: A dynamic programming approach. In *Lecture Notes in Computer Science*, volume 3537, pages 190–200, 2005.

[142] Gergely Korodi and Ioan Tabus. An efficient normalized maximum likelihood algorithm for DNA sequence compression. *ACM Trans. Inf. Syst.*, 23:3–34, 2005.

[143] Gregory Vey. Differential direct coding: A compression algorithm for nucleotide sequence data. *Database : the journal of biological databases and curation*, 2009:bap013, 2009.

[144] Dr Mishra, Dr Aaggarwal, Edries Abdelhadi, and Prakash Srivastava. An efficient horizontal and vertical method for online DNA sequence compression. *International Journal of Computer Applications*, 3, 2010.

[145] P Rajeswari, Allam Apparo, and V Kumar. Genbit compress tool(GBC): A java-based tool to compress DNA sequences and compute compression ratio(bits/base) of genomes. *International Journal of Computer Science  Information Technology*, 2, 2010.

[146] Zexuan Zhu, Jiarui Zhou, Zhen Ji, and Yu-hui Shi. DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. *IEEE Trans. Evolutionary Computation*, 15:643–658, 2011.

[147] Armando J Pinho, Diogo Pratas, and Paulo J S G Ferreira. Bacteria DNA sequence compression using a mixture of finite-context models. In *IEEE Workshop on Statistical Signal Processing Proceedings*, pages 125–128, 2011.

[148] Armando J Pinho, Paulo J S G Ferreira, António Neves, and Carlos Bastos. On the representability of complete genomes by multiple competing finite-context (markov) models. *PloS one*, 6:e21588, 2011.

[149] Ashutosh Gupta. A novel approach for compressing DNA sequences using semi-statistical compressor. *International Journal of Computers and Applications*, 33:3, 2011.

[150] Tungadri Bose, Mohammed Haque, Anirban Dutta, and Sharmila Mande. BIND - an algorithm for loss-less compression of nucleotide sequence data. *Journal of biosciences*, 37:785–9, 2012.

[151] Subhankar Roy, Sunirmal Khatua, Sudipta Roy, and Samir Bandyopadhyay. An efficient biological sequence compression technique using LUT andrepeat in the sequence. *IOSR Journal of Computer Engineering (IOSRJCE)*, 2012.

[152] D. Satyanvesh, Kaliuday Balleda, Ajith Padyana, Pallav K. Baruah, and Sri Sathya Sai. Gencodex - a novel algorithm for compressing DNA sequences on multi-cores and gpus. In *Proc. IEEE, 19th International Conf. on High Performance Computing (HiPC), Pune, India. IEEE*, 2012.

[153] Muhammad Sardaraz, Muhammad Tahir, Ataul Aziz Ikram, and Hassan Bajwa. Seqcompress: An algorithm for biological sequence compression. *Genomics*, 104(4):225–228, 2014.

[154] Diogo Pratas and Armando J Pinho. Exploring deep markov models in genomic data compression using sequence pre-analysis. In *European Signal Processing Conference*, 2014.

[155] M Chen, J J Shao, and X M Jia. Genome sequence compression based on optimized context weighting. *Genetics and Molecular Research*, 16, 2017.

[156] D Mansouri and X Yuan. One-bit DNA compression algorithm. in proceedings of the international conferenceon neural information processing. In *Springer: Siam reap*, pages 378–386, 2018.

[157] Diogo Pratas, Morteza Hosseini, Jorge M Silva, and Armando J Pinho. A reference-free lossless compression algorithm for DNA sequences using a competitive prediction of two classes of weighted models. *Entropy*, 21(11), 2019.

[158] Diogo Pratas, Armando J Pinho, and Paulo J S G Ferreira. Efficient compression of genomic sequences. In *2016 Data Compression Conference (DCC)*, pages 231–240, 2016.

[159] Diogo Pratas, Morteza Hosseini, and Armando Pinho. *GeCo2: An Optimized Tool for Lossless Compression and Analysis of DNA Sequences*, pages 137–145. 2020.

[160] Carl Kingsford and Rob Patro. Reference-based compression of short-read sequences using path encoding. *Bioinformatics*, 31(12):1920–1928, 2015.

[161] Bobbie Chern, Idoia Ochoa, Alexandros Manolakos, Albert No, Kartik Venkat, and Tsachy Weissman. Reference based genome compression. *IEEE Inf Theory Workshop, ITW*, 2012.

[162] Scott Christley, Yiming Lu, Chen Li, and Xiaohui Xie. Human genomes as email attachments. *Bioinformatics*, 25(2):274–5, 2009.

[163] Shanika Kuruppu, Simon J Puglisi, and Justin Zobel. Relative lempel-ziv compression of genomes for large-scale storage and retrieval. pages 201–206. Springer Berlin Heidelberg, 2010.

[164] Congmao Wang and Dabing Zhang. A novel compression tool for efficient storage of genome resequencing data. *Nucleic Acids Res*, 39(7):e45, 2011.

[165] Armando J Pinho, Diogo Pratas, and Sara P Garcia. GReEn: a tool for efficient compression of genome resequencing data. *Nucleic Acids Research*, 40(4):e27–e27, 2011.

[166] Sebastian Deorowicz and Szymon Grabowski. Robust relative compression of genomes with random access. *Bioinformatics*, 27(21):2979–2986, 2011.

[167] Shanika Kuruppu, Bryan Beresford-Smith, Thomas Conway, and Justin Zobel. Iterative dictionary construction for compression of large DNA data sets. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 9, 2011.

[168] Sebastian Wandelt and Ulf Leser. Fresco: Referential compression of highly similar sequences. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 10:1275–88, 2013.

[169] Idoia Ochoa, Mikel Hernaez, and Tsachy Weissman. idocomp: A compression scheme for assembled genomes. *Bioinformatics (Oxford, England)*, 31, 2014.

[170] Diogo Pratas. *Compression and analysis of genomic sequences.* PhD thesis, 2016.

[171] Nima Nikvand and Zhou Wang. Image distortion analysis based on normalized perceptual information distance. *Signal, Image and Video Processing*, 7(3):403–410, 2013.

[172] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul MB Vitányi. The similarity metric. *IEEE transactions on Information Theory*, 50(12):3250–3264, 2004.

[173] Minh Cao, Trevor Dix, Lloyd Allison, and Chris Mears. A simple statistical algorithm for biological sequence compression. In *In Proceedings of the Conference on Data Compression*, pages 43–52, 2007.

[174] Pinghao Li, Shuang Wang, Jihoon Kim, Hongkai Xiong, Lucila Ohno-Machado, and Xiaoqian Jiang. DNA-COMPACT: DNA COMpression based on a pattern-aware contextual modeling technique. *PloS one*, 8:e80377, 2013.

[175] Xiaojing Xie, Shuigeng Zhou, and Jihong Guan. Cogi: Towards compressing genomes as an image. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12:1, 2015.

[176] Manuel Cebrián, Manuel Alfonseca, and Alfonso De la Puente. Common pitfalls using the normalized compression distance: What to watch out for in a compressor. *Communications in Information and Systems*, 5, 2005.

[177] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

[178] Irina Rish. An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3, 2001.

[179] Pouria Kaviani and Sunita Dhotre. Short survey on naive bayes algorithm. *International Journal of Advance Research in Computer Science and Management*, 04, 2017.

[180] Carlos Bustamante, Leonardo Garrido, and Rogelio Soto. Comparing fuzzy naive bayes and gaussian naive bayes for decision making in robocup 3D. In *RoboCup*, pages 237–247, 2006.

[181] Padraig Cunningham and Sarah Delany. k-nearest neighbour classifiers. *Mult Classif Syst*, 54, 2007.

[182] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines: Theory and applications. In *Machine Learning and Its Applications*, volume 2049, pages 249–257, 2001.

[183] Alexander Sokolov, Ilya Pyatnitsky, and Sergei Alabugin. Research of classical machine learning methods and deep learning models effectiveness in detecting anomalies of industrial control system. In *2018 Global Smart Industry Conference (GloSIC)*, pages 1–6, 2018.

[184] Seema Sharma, Jitendra Agrawal, Shikha Agarwal, and Sanjeev Sharma. Machine learning techniques for data mining: A survey. In *2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, pages 1–6, 2013.

[185] T G Dietterich. Ensemble methods in machine learning. *Multiple Classifier Systems*, 1857:1–15, 2000.

[186] Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In *Neural Nets WIRN Vietri-2002, Series Lecture Notes in Computer Sciences*, volume 2486, pages 3–22, 2002.

[187] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[188] Maad M Mijwel. Artificial neural networks advantages and disadvantages. *Retrieved from LinkedIn https//www.linkedin.com/pulse/artificial-neuralnet Work*, 2018.

[189] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5–6):183–197, 1991.

[190] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[191] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997.

[192] Peter M Visscher, Matthew A Brown, Mark I McCarthy, and Jian Yang. Five years of gwas discovery. *Am. J. Hum. Genet.*, 90:7–24, 2012.

[193] Peter M Visscher, Naomi R Wray, Qian Zhang, Mark I McCarthy, Pamela Sklar, Matthew A Brown, and Jian Yang. 10 years of gwas discovery: Biology, function, and translation. *Am. J. Hum. Genet.*, 101:5–22, 2017.

[194] Melinda C Mills and Charles Rahal. A scientometric review of genome-wide association studies. *Commun Biol 2*, 9, 2019.

[195] Cynthia L Sears and Steven L Salzberg. Microbial Diagnostics for Cancer: A Step Forward but Not Prime Time Yet. *Cancer cell*, 37(5):625–627, 2020.

[196] Muin J Khoury, W Gregory Feero, Michele Reyes, Toby Citrin, Andrew Freedman, Debra Leonard, Wylie Burke, Ralph Coates, Robert T Croyle, Karen Edwards, and et al. The genomic applications in practice and prevention network. *Genet Med.*, 11(7):488–494, 2009.

[197] Soren K Thomsen and Anna L Gloyn. Human genetics as a model for target validation: finding new therapies for diabetes. *Diabetologia*, 60:960–970, 2017.

[198] James S Lawson, Brian Salmons, and Wendy K Glenn. Oncogenic viruses and breast cancer: mouse mammary tumor virus (MMTV), bovine leukemia virus (BLV), human papilloma virus (HPV), and epstein–barr virus (EBV). *Frontiers in oncology*, 8:1, 2018.

[199] Rauli Franssila and Klaus Hedman. Viral causes of arthritis. *Best practice & research Clinical rheumatology*, 20(6):1139–1157, 2006.

[200] Amr El-Sayed and Mohamed Kamel. Climatic changes and their role in emergence and re-emergence of diseases. *Environmental Science and Pollution Research*, 27(18):22336–22352, 2020.

[201] Verónica Mixão and Toni Gabaldón. Hybridization and emergence of virulence in opportunistic human yeast pathogens. *Yeast*, 35(1):5–20, 2018.

[202] Todd A Crowl, Thomas O Crist, Robert R Parmenter, Gary Belovsky, and Ariel E Lugo. The spread of invasive species and infectious disease as drivers of ecosystem change. *Frontiers in Ecology and the Environment*, 6(5):238–246, 2008.

[203] Maliya Alia Malek, Idir Bitam, Anthony Levasseur, Jérôme Terras, Jean Gaudart, Said Azza, Christophe Flaudrops, Catherine Robert, Didier Raoult, and Michel Drancourt. Yersinia pestis halotolerance illuminates plague reservoirs. *Scientific reports*, 7(1):1–10, 2017.

[204] Fan Wu, Su Zhao, Bin Yu, Yan-Mei Chen, Wen Wang, Zhi-Gang Song, Yi Hu, Zhao-Wu Tao, Jun-Hua Tian, Yuan-Yuan Pei, et al. A new coronavirus associated with human respiratory disease in China. *Nature*, 579(7798):265–269, 2020.

[205] Matthew I Hutchings, Andrew W Truman, and Barrie Wilkinson. Antibiotics: past, present and future. *Current opinion in microbiology*, 51:72–80, 2019.

[206] Mari Toppinen, Antti Sajantila, Diogo Pratas, Klaus Hedman, and Maria F Perdomo. The Human Bone Marrow Is Host to the DNAs of Several Viruses. *Frontiers in cellular and infection microbiology*, 11:329, 2021.

[207] Omer Gokcumen and Michael Frachetti. The impact of ancient genome studies in archaeology. *Annual Review of Anthropology*, 49:277–298, 2020.

[208] Stefanie Eisenmann, Eszter Bánffy, Peter van Dommelen, Kerstin P Hofmann, Joseph Maran, Iosif Lazaridis, Alissa Mittnik, Michael McCormick, Johannes Krause, David Reich, and Philipp W Stockhammer. Reconciling material cultures in archaeology with genetic data: The nomenclature of clusters emerging from archaeogenomic analysis. *Scientific Reports*, 8, 2018.

[209] Hendrik N Poinar, Carsten Schwarz, Ji Qi, Beth Shapiro, Ross DE MacPhee, Bernard Buigues, Alexei Tikhonov, Daniel H Huson, Lynn P Tomsho, Alexander Auch, et al. Metagenomics to paleogenomics: large-scale sequencing of mammoth DNA. *Science*, 311(5759):392–394, 2006.

[210] Jesse Dabney, Matthias Meyer, and Svante Pääbo. Ancient DNA damage. *Cold Spring Harbor perspectives in biology*, 5(7):a012567, 2013.

[211] Ludovic Orlando, Robin Allaby, Pontus Skoglund, Clio Der Sarkissian, Philipp W Stockhammer, María C Ávila-Arcos, Qiaomei Fu, Johannes Krause, Eske Willerslev, Anne C Stone, et al. Ancient DNA analysis. *Nature Reviews Methods Primers*, 1(1):1–26, 2021.

[212] J T Wang, S Rozen, B A Shapiro, D Shasha, Z Wang, and M Yin. New techniques for DNA sequence classification. *J Comput Biol*, 6(2):209–218, 1999.

[213] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison—a review. *Bioinformatics*, 19(4):513–523, 2003.

[214] Susana Vinga. Editorial: Alignment-free methods in computational biology. *Briefings in Bioinformatics*, 15(3):341–342, 2014.

[215] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, 18(186), 2017.

[216] Andrzej Zielezinski, Hani Z Girgis, Guillaume Bernard, Chris-Andre Leimeister, Kujin Tang, Thomas Dencker, Anna K Lau, Sophie Röhling, JaeJin Choi, Michael S Waterman, Matteo Comin, Sung-Hou Kim, Susana Vinga, Jonas S Almeida, Cheong Xin Chan, Benjamin T James, Fengzhu Sun, Burkhard Morgenstern, and Wojciech M Karlowski. Benchmarking of alignment-free sequence comparison methods. *bioRxiv*, 2019.

[217] Biswanath Chowdhury and Gautam Garai. A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics*, 109:419–431, 2017.

[218] Andrzej Zielezinski, Hani Z Girgis, Guillaume Bernard, Chris-Andre Leimeister, Kujin Tang, Thomas Dencker, Anna Katharina Lau, Sophie Röhling, Jae Jin Choi, Michael S Waterman, Matteo Comin, Sung-Hou Kim, Susana Vinga, Jonas S Almeida, Cheong Xin Chan, Benjamin T James, Fengzhu Sun, Burkhard Morgenstern, and Wojciech M Karlowski. Benchmarking of alignment-free sequence comparison methods. *Genome Biology*, 20(144), 2019.

[219] Fanny-Dhelia Pajuste, Lauris Kaplinski, Märt Möls, Tarmo Puurand, Maarja Lepamets, and Maido Remm. Fastgt: an alignment-free method for calling common snvs directly from raw sequencing reads. *Scientific Reports*, 7, 2017.

[220] Jason R Miller, Sergey Koren, and Granger Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010.

[221] C A Orengo, A D Michie, S Jones, D T Jones, M B Swindells, and J M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8), 1997.

[222] S Yooseph, G Sutton, DB Rusch, AL Halpern, SJ Williamson, K Remington, JA Eisen, KB Heidelberg, G Manning, W Li, and et al. The sorcerer II global ocean sampling expedition: Expanding the universe of protein families. *PLoS Biol*, 5(3), 2007.

[223] Shibu Yooseph, Weizhong Li, and Granger Sutton. Gene identification and protein classification in microbial metagenomic sequence data via incremental clustering. *BMC Bioinformatics*, 9(182), 2008.

[224] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3), 1990.

[225] E Gasteiger, E Jung, and A Bairoch. Swiss-prot: connecting biomolecular knowledge via a protein database. *Curr Issues Mol Biol*, 3(3):47–55, 2001.

[226] A Bateman, L Coin, R Durbin, RD Finn, V Hollich, S Griffiths-Jones, A Khanna, M Marshall, S Moxon, EL Sonnhammer, DJ Studholme, C Yeats, and SR Eddy. The pfam protein families database. *Nucleic Acids Res*, 32, 2004.

[227] E L Sonnhammer, S R Eddy, E Birney, A Bateman, and R Durbin. Pfam: multiple sequence alignments and hmm-profiles of protein domains. *Nucleic Acids Res*, 26(1):320–322, 1998.

[228] Brian D Ondov, Todd J Treangen, Páll Melsted, Adam B Mallonee, Nicholas H Bergman, Sergey Koren, and Adam M Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome biology*, 17(1):1–14, 2016.

[229] Angana Chakraborty, Burkhard Morgenstern, and Sanghamitra Bandyopadhyay. S-conLSH: Alignment-free gapped mapping of noisy long reads. *BMC bioinformatics*, 22(1):1–18, 2021.

[230] Junyi Li, Li Zhang, Huinian Li, Yuan Ping, Qingzhe Xu, Rongjie Wang, Renjie Tan, Zhen Wang, Bo Liu, and Yadong Wang. Integrated entropy-based approach for analyzing exons and introns in DNA sequences. *BMC Bioinformatics*, 20(283), 2019.

[231] Florent Lassalle, Séverine Périan, Thomas Bataillon, Xavier Nesme, Laurent Duret, and Vincent Daubin. Gc-content evolution in bacterial genomes: The biased gene conversion hypothesis expands. *PLoS Genet*, 11(2), 2015.

[232] Morteza Hosseini, Diogo Pratas, and Armando J Pinho. Cryfa: a secure encryption tool for genomic data. *Bioinformatics*, 35(1):146–148, 2019.

[233] Abukari Mohammed Yakubu and Yi-Ping Phoebe Chen. Ensuring privacy and security of genomic data and functionalities. *Brief Bioinform*, 21(2):511–526, 2020.

[234] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[235] Shifu Chen, Yanqing Zhou, Yaru Chen, and Jia Gu. fastp: an ultra-fast all-in-one fastq preprocessor. *Bioinformatics*, 34(17):i884–i890, 2018.

[236] Renata Geer and Eric Sayers. Entrez: Making use of its power. *Briefings in bioinformatics*, 4:179–84, 2003.

[237] Diogo Pratas, Morteza Hosseini, and Armando J Pinho. Cryfa: A tool to compact and encrypt fasta files. *Advances in Intelligent Systems and Computing*, 616, 2017.

[238] Zhijiao Zhang, Scott Schwartz, Lukas Wagner, and Webb Miller. Greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, 7(1-2):203–14, 2000.

[239] Irene T Rombel, Kathryn F Sykes, Simon Rayner, and Stephen Albert Johnston. Orf-finder: a vector for high-throughput gene identification. *Gene*, 282(Issues 1–2):33–41, 2002.

[240] Ben J Woodcroft, Joel A Boyd, and Gene W Tyson. Orfm: a fast open reading frame predictor for metagenomic data. *Bioinformatics*, 32(17):2702–3, 2016.

[241] Matthew D MacManes. On the optimal trimming of high-throughput mrna sequence data. *Frontiers in Genetics*, 5:13, 2014.

[242] Martin Kircher. Analysis of high-throughput ancient DNA sequencing data. *Methods in molecular biology (Clifton, N.J.)*, 840:197–228, 2012.

[243] Florian P Breitwieser, Jennifer Lu, and Steven L Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in bioinformatics*, 20(4):1125–1136, 2019.

[244] Jennifer Lu and Steven L Salzberg. Removing contaminants from databases of draft genomes. *PLoS computational biology*, 14(6):e1006277, 2018.

[245] Joel-E Kuon, Weihong Qi, Pascal Schläpfer, Matthias Hirsch-Hoffmann, Philipp Rogalla von Bieberstein, Andrea Patrignani, Lucy Poveda, Stefan Grob, Miyako Keller, Rie Shimizu-Inatsugi, et al. Haplotype-resolved genomes of geminivirus-resistant and geminivirus-susceptible African cassava cultivars. *BMC biology*, 17(1):1–15, 2019.

[246] Peter Rice, Ian Longden, and Alan Bleasby. Emboss: The european molecular biology open software suite. *Trends in Genetics*, 16(6):276–7, 2000.

[247] Elisabeth Gasteiger, Alexandre Gattiker, Christine Hoogland, Ivan Ivanyi, Ron D Appel, and Amos Bairoch. Expasy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res.*, 31(13):3784–3788, 2003.

[248] Monya Baker. De novo genome assembly: what every biologist should know. *Nature Methods*, 9:333–337, 2012.

[249] Morteza Hosseini, Diogo Pratas, Burkhard Morgenstern, and Armando J Pinho. Smash++: an alignment-free and memory-efficient tool to find genomic rearrangements. *GigaScience*, 9(5):giaa048, 2020.

[250] Andrei N Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7, 1965.

[251] Irina Rish et al. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[252] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. *Springer Berlin Heidelberg*, pages 986–996, 2003.

[253] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[254] Fei Cheng, Chunhua Yang, Can Zhou, Lijuan Lan, Hongqiu Zhu, and Yonggang Li. Simultaneous determination of metal ions in zinc sulfate solution using uv–vis spectrometry and spse-xgboost method. *Sensors*, 20:4936, 2020.

[255] Daniel Berrar. Cross-validation. *Academic Press*, 1:542–545, 2019.

[256] Ramesh Medar, Vijay Rajpurohit, and B Rashmi. Impact of training and testing data splits on accuracy of time series forecasting in machine learning. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pages 1–6, 2017.

[257] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[258] Hector Zenil, Santiago Hernández-Orozco, Narsis Aftab Kiani, Fernando Soler-Toscano, Antonio Rueda-Toicen, and Jesper Tegnér. A decomposition method for global evaluation of shannon entropy and local estimations of algorithmic complexity. *Entropy*, 20(8):605, 2018.