Extraction of discriminative regions over genomic sequences

Felipe Bueno de Souza

Dissertation presented to the School of Technology and Management of Bragança to obtain the master's degree in Informatics within the scope of the double degree program with the Federal University of Technology – Paraná.

Supervisors:
Fabrício Martins Lopes (UTFPR)
José Rufino (ESTiG/IPB)
M. Alice Pinto (ESA/IPB)

Bragança
November 2025

Extraction of discriminative regions over genomic sequences

Felipe Bueno de Souza

Dissertation presented to the School of Technology and Management of Bragança to obtain the master's degree in Informatics within the scope of the double degree program with the Federal University of Technology – Paraná.

Supervisors:
Fabrício Martins Lopes (UTFPR)
José Rufino (ESTiG/IPB)
M. Alice Pinto (ESA/IPB)

Bragança
November 2025

Acknowledgment

First, I must thank my parents for all the support and opportunities they provided me to reach this point with this personal level of mastery. Without them encouraging me to pursue what I always wanted, I would not have succeeded. I thank them for all the education and advice they have given me. I have to thank my girlfriend Beatriz, who stayed with me and supported this life dream of mine, and now I hope to give back all the support I received from her in this new stage of our lives.

To all my advisors: Fabricio Martins Lopes, José Rufino, Maria Alice, Carlos Balsa, Dora Henriques, and last but not least, Matheus Pimenta. Thank you for guiding me with such mastery and seeing potential in me that I had not yet perceived, believing in me from the beginning. You contributed to my growth both academically and socially, with invaluable contributions to something I greatly value in my scholarly journey: interdisciplinarity, something that brings a solid contribution for science.

A huge thank you to my cousin Giovana, who inspired me to follow her academic path, which she pursues with such excellence, and to her partner Lucas. They both gave me all the family support that built my foundation in Bragança. And also to all the other friends I made here, where I do not need to specify by name, as they know who they are.

I also would like to acknowledge my friends who began this journey with me: Messias, Gregory, Lucas, Giovanni, Gustavo, Jonathan and Henrique. I also thank those I left in Brasil who always supported me in being who I am.

Abstract

As computing technologies continue to evolve, new generations of processors have achieved increased levels of computational power and efficiency. This progress enables the execution of tasks that, in the past, required high-end computers and can now be performed on personal systems, allowing many scientific fields to benefit from this progress, including biology.

Along with this computational progress, the advancement of DNA sequencing technology is responsible for the exponential growth in the volume and complexity of available genomic data. This scenario requires methods that can efficiently handle and analyze such data in a scalable and interpretable manner, addressing the high volume and inherent complexity of biological sequences. In this context, this work proposes a novel methodology – GREAC (Genomic Region Extraction and Classifier) – for extracting discriminative regions from genomic sequences, reducing data dimensionality, identifying biologically relevant patterns, and variant classification.

The proposed methodology is grounded in digital signal processing principles, such as filters and sequences transformation, employing k-mers as the primary source of information to filter and identify informative genomic regions. The relative frequency values of these regions are then measured to construct standardized signals across different variants. Each reference signal represents the characteristic behavior of a variant, enabling the identification of genomic patterns that allow their classification through statistical divergence measures, distance metrics, and supervised classifiers such as XGBoost.

GREAC was implemented in the Julia programming language and is public domain open-source software, emphasizing efficiency, transparency, and scientific reproducibility. The implementation enables execution on personal computers, thereby promoting accessibility and encouraging contributions from the scientific community for further improvements. GREAC

represents thus a significant contribution to the fields of bioinformatics and computational genomics, presenting a novel methodology for pattern recognition in genomic sequences.

Resumo

À medida que as tecnologias de computação continuam evoluindo, novas gerações de processadores vêm alcançando níveis cada vez maiores de poder e eficiência computacional. Esse progresso possibilita a execução de tarefas que, no passado, exigiam computadores de alto desempenho e que agora podem ser realizadas em sistemas pessoais, permitindo que diversos campos científicos se beneficiem desses avanços, incluindo a biologia.

Juntamente com esse progresso computacional, o avanço das tecnologias de sequenciamento de DNA é responsável pelo crescimento exponencial no volume e na complexidade dos dados genômicos disponíveis. Esse cenário exige métodos capazes de lidar e analisar esses dados de forma eficiente, escalável e interpretável, enfrentando tanto o grande volume quanto a complexidade inerente das sequências biológicas. Nesse contexto, este trabalho propõe uma nova metodologia — GREAC (Genomic Region Extraction and Classifier) — para extração de regiões discriminativas em sequências genômicas, visando à redução da dimensionalidade dos dados (reduzindo o comprimento final das sequências), à identificação de padrões biologicamente relevantes e à classificação de variantes.

A metodologia proposta baseia-se em princípios de processamento digital de sinais, como filtros e transformação de sequências, empregando k-mers como principal fonte de informação para filtrar e identificar regiões genômicas informativas. Os valores de frequência relativa dessas regiões são então medidos para construir sinais padronizados entre diferentes variantes. Cada sinal de referência representa o comportamento característico de uma variante, permitindo a identificação de padrões genômicos que possibilitam sua classificação por meio de medidas de divergência estatística, métricas de distância e classificadores supervisionados, como o XGBoost.

O GREAC foi implementado na linguagem de programação Julia e disponibilizado como software de código aberto em domínio público, destacando eficiência, transparência e reprodutibilidade científica. A implementação permite sua execução em computadores pessoais, promovendo acessibilidade e incentivando contribuições da comunidade científica para aprimoramentos futuros. Dessa forma, o GREAC representa uma contribuição significativa para os campos da bioinformática e da genômica computacional apresentando uma nova metodologia para reconhecimento de padrões genômicos.

Contents

List of Figures

Acronyms

BLAST Basic Local Alignment Search Tool.

CASTOR-KRFE Classification And Segmentation Tool for Optical Recognition - K-mer Recursive Feature Elimination.

CLI Command-line Interface.

DFT Discrete Fourier Transform.

DNA Deoxyribonucleic Acid.

EIIP Electron-Ion Interaction Potention.

ESTiG Escola Superior de Tecnologia e Gestão.

FFT Fast Fourier Transform.

GRAMEP Genome vaRiation Analysis from the Maximum Entropy Principle.

GREAC Genomic Region Extraction and Classifier.

IPB Instituto Politécnico de Bragança.

KEVOLVE K-mer Evolution-based Viral Genome Classifier.

lcRNA Long non-coding RNA.

mRNA Messenger RNA.

PCA Principal Component Analysis.

RFE Recursive Feature Elimination.

RNA Ribonucleic Acid.

RRM Resonant Recognition Model.

SNP Single Nucleotide Polymorphism.

SNPs Single Nucleotide Polymorphisms.

SVM Support Vector Machines.

List of Tables

Chapter 1

Introduction

The analysis of genomic sequences stands as a cornerstone in modern biology, offering profound insights into the diversity, evolution, and fundamental functions of species. With the relentless progress in high-throughput sequencing technologies, an unprecedented volume of genomic data is being generated. This data deluge, while offering immense opportunities for research into genetic variants, disease associations, and intricate life mechanisms, simultaneously introduces significant computational challenges.

The inherent high dimensionality of biological sequences, such as DNA or protein chains, which can span millions of elements, often leads to a feature space with an extremely large number of dimensions. This phenomenon, widely known as the "curse of dimensionality", can severely impede the performance of classification algorithms by scaling computational complexity, demanding extensive storage, and increasing the risk of overfitting, where models inadvertently learn noise rather than the true underlying biological signals. Consequently, there is a critical and pressing need for the development of efficient, scalable, interpretable, and data-driven computational methods that can effectively navigate and identify meaningful knowledge from these vast and complex datasets.

Historically, biological sequence analysis has been based on alignment-based methods. These approaches, often employing techniques such as dynamic programming, compare sequences by identifying homologous regions. Although powerful and capable of revealing subtle evolutionary relationships, their application to large-scale genomic datasets is

often unfeasible due to their high computational cost, which scales poorly with the increasing size and number of sequences. This computational bottleneck has spurred the development of various alignment-free methods. These innovative approaches combine techniques from data mining, signal processing, and machine learning to transform sequences into numerical vectors, often based on k-mer frequencies. This conversion enables rapid comparison and classification, bypassing limitations of traditional alignment-based methodologies and providing a more scalable solution for analyzing vast genomic datasets.

Within this evolving landscape, supervised machine learning approaches have emerged as reliable and efficient methodologies for identifying discriminative genomic features and enabling accurate taxonomic classification, particularly within highly diverse viral populations. However, many existing solutions remain proprietary, computationally expensive, or require specialized infrastructure that limits their accessibility to the broader scientific community. Recognizing this gap, the present research developed an open-source software framework – GREAC – that democratizes access to advanced genomic analysis capabilities while maintaining scientific rigor and computational efficiency.

GREAC aims to enhance genomic surveillance efforts by representing a reproducible, scalable software solution that is openly adaptable to community-driven modifications, enabling researchers to tailor the approach to their specific research questions and datasets. By prioritizing computational efficiency alongside biological interpretability, it ensures the transparent and explainable nature of classification decisions, a critical feature for both research validation and public health applications, where understanding the basis of genomic classifications is paramount. The emphasis on open science principles facilitates collaborative development, peer review, and continuous improvement of the tool.

1.1 Motivations and Objectives

The interest in developing this study was an opportunity to apply my computational knowledge to develop a methodology that addresses a biological problem. This motivation was significantly amplified when the research demonstrated broad applicability across

various organisms, presenting substantial potential for biotechnological contributions.

Developing this research as open-source code also aligns with my personal goals, which involve putting effort into developing genomic analysis tools that are not only freely available but also computationally accessible to researchers with limited resources. This approach recognizes that true democratization requires not just open licensing but also technical accessibility, ensuring that the benefits of advanced computational methods are not concentrated among well-resourced institutions.

The main objective of this work was thus to develop a methodology capable of extracting genomic patterns through discriminative regions that can be used for the analysis and classification between variants and subspecies of diverse organisms as an open-source initiative, using discriminative sets of k-mers from the GRAMEP tool, providing a solid and transparent information source foundation for extracting such regions.

Complementing the objectives with the extraction of these regions falls under the direct objective of data dimensionality reduction, where, within the biological domain, a large volume of genomic sequences consumes significant storage and thus requires substantial memory for processing. By extracting the discriminative regions of a genome, one can not only provide insights about the organisms but also achieve better processing efficiency for this information, contributing to low resource usage.

1.2 Thesis Structure

This thesis is organized into four main chapters in addition to the introduction.

In chapter 2, a comprehensive literature review of the concepts necessary to obtain a deep understanding of the methodological approaches employed in this work is presented. This chapter describes the biological concepts that underpin the problem resolution addressed in the new methodology, the mathematical concepts utilized in the tools employed for data extraction and investigation, and the algorithms used in developing the methodology. It concludes with a description of similar works used for comparative analysis.

Chapter 3 presents the proposed methodology, describing its principles, workflow, and

the supporting data used during development.

Next, Chapter 4 details the development process, beginning with an overview of the complete software architecture and its distribution, followed by the functionalities implemented, and concluding with the algorithmic implementations, parameter selection, and metric validation of the proposed method.

Chapter 5 presents and analyzes the results obtained from the experimental validation of the methodology, including its application to viral genomic datasets and experiments conducted on *Apis mellifera* honeybee data.

Finally, Chapter 6 concludes the work by highlighting the main contributions and discussing the results achieved, the insights that come from them, and future research directions that could be pursued.

1.3 Publications

The research conducted during this thesis has yielded several scientific contributions that have been accepted, published, and presented at international conferences:

1. Bueno de Souza, F. et al. (2025). **Resonant Recognition Model as a Preprocessing Technique for RNA Classification.** In: Guarda, T., Portela, F., Augusto, M.F. (eds) *Advanced Research in Technologies, Information, Innovation and Sustainability.* ARTIIS 2024. Communications in Computer and Information Science, vol. 2348. Springer, Cham. `https://doi.org/10.1007/978-3-031-83435-6_1`

2. Bueno de Souza, F. et al. (2025). **An Efficient Feature Extraction Method for Identifying Signatures of Viral Genomic Variants.** In: *Proceedings of the ARTIIS 2025 International Conference.* (Accepted and presented). To appear in Springer Communications in Computer and Information Science (CCIS).

A more in-depth article (**GREAC: An Open-source software for Genome Region Extraction and Classification**) is also under preparation for submission to a

peer-reviewed journal in the field of Bioinformatics or Computational Genomics.

Chapter 2

Background

This chapter presents concepts from biology, mathematics, and computer science that are fundamental to this dissertation, along with contributions from previous work by the scientific community on biological analysis and genome classification.

2.1 Molecular Biology

Molecular biology focuses on advancing the understanding of cellular processes, the molecular mechanisms underlying genetic diseases, and other possible diseases and mutations that gene variation may trigger.

DNA (deoxyribonucleic acid) is a nucleic acid that carries the genetic information of all living organisms. DNA provides instructions pertaining to cellular functions, their role in the organism, and defines hereditary characteristics. A set of DNA molecules (including coding and non-coding regions) is known as a *genome*, which can be described as a finite sequence of elements from a set of four nitrogenous bases, known as the nucleotides: adenine (A), cytosine (C), thymine (T), and guanine (G).

Individuals of the same species have similar genomes, differentiated by Single Nucleotide Polymorphisms (SNPs) – nucleotide changes at specific positions in a sequence that create different morphological phenotypes (e.g., wing shape, proboscis length in the honeybee, etc.) in individuals of the same species, classifying them into subspecies.

Sometimes, a single SNP, discovered through various analyses against a reference genome, does not represent a significant change in the genome. However, a region in the sequence that contains one or several SNPs can carry information that differentiates a subspecies or defines unique characteristics of a set of individuals with these alterations.

2.1.1 RNA transcript from DNA

RNA (ribonucleic acid) is the other type of nucleic acid found in the nucleus of the cells. It is chemically similar to DNA, with the difference that thymine (T) is replaced by uracil (U). Messenger RNA sequences define which DNA genes will encode proteins in the organism. RNA sequences can also be used for gene comparison and classification of organisms of the same species or different species (showing similar functional proteins) [1].

When the DNA genome is available, it is possible to transcribe it into an RNA sequence, a process of paramount importance in the gene expression process [1]. Through transcription, RNA molecules are synthesized from information contained in DNA, generating the *transcriptome* as the final result.

2.1.2 K-mers

Various approaches can be employed to analyze genome sequences. The analysis may focus on the comparison of SNPs, which can result in high variability rates, reducing analysis efficiency, as not all variations may be exclusive and informative.

Alternatively, the analysis may compare sequences of k consecutive nucleotides, known as k-mers, which reduces the variety rate, either due to the decreased probability of a k-mer occurring at a specific sequence position, or due to the reduced probability of a k-mer being present in the sequence. Moreover, the longer the k-mer, the lower the variability value, causing exclusive and informative SNPs to be excluded from the analysis. Therefore, it is necessary to choose a size k that is suitable for performing this kind of analysis.

In this work, k-mers serve as a source of information for identifying patterns and classifying variants and sequences. But their versatility allows for other usage. For instance,

they may also be used for the mounting of genomic sequences and other applications that involve sequential patterns, such as the detection of sequencing errors and database information retrieval [2]–[8].

2.1.3 Genome Introgression

The *Apis mellifera* honeybee has been experiencing significant genetic pressure due to cross-breeding between subspecies, resulting in fertile hybrids capable of reproduction, which makes the species particularly susceptible to introgression.

Unlike other reproductive processes that preserve similar genetic patterns, introgression directly incorporates other subspecies DNA [9], leading populations to acquire genetic characteristics from other groups while simultaneously losing their own distinctive traits.

Introgression has direct implications for the genetic integrity, preservation, and conservation of the *Apis mellifera*. Human-mediated gene flow causes introgressive hybridization, promoting the local extinction of its subspecies by reducing the frequency of locally adapted genomic complexes. This process results in increased artificial genetic diversity, while causing the loss of subspecies-specific adaptations that can significantly affect individual survival [10], [11].

Introgression also makes it more difficult the task of genome classification, requiring more robust methods for subspecies discrimination, like the one developed in this work.

2.2 Maximum Entropy Principle

The concept of *entropy* was first introduced in thermodynamics as a measure of uncertainty and dispersion, calculating the probabilistic arrangement of atoms and molecules [12]. Later, it was adopted in Information Theory as a metric for the amount of disorder or information (equivalent, under the theory) of a given data distribution. Here, the entropy value is defined by Equation 2.1, which calculates the average degree of uncertainty or information associated with all possible values of a random variable [12], [13]:

$$H(X) = -\sum_{k=1}^{K} p(x_k)\, log\, p(x_k), \tag{2.1}$$

where K is the number of possible values of the random variable X, represented by the set $\{x_1, x_2, x_3, ..., x_K\}$, and $p(x_k)$ is the specific probability of a particular value x_k.

The Maximum Entropy Principle (MEP) states that the probability distribution that best represents the current knowledge about a system is the one with the highest entropy [14], [15]. Therefore, this principle can guide the choice from among all possible probability distributions of a system, allowing one to make inferences without assuming any additional information (that is, solely based on the fact that a specific probability distribution is the one that maximizes the amount of information).

2.2.1 Application to Genome Sequence Analysis

The application of the MEP principle to genome sequence analysis was introduced in GRAMEP [16], an alignment-free approach for the precise identification of SNPs and k-mers within biological sequences.

To understand how the MEP principle is applied in this context, consider a generic scenario with a discrete probability distribution comprising n elements. One can divide this distribution into two complementary classes, A and B, the first with s elements and the second with $n - s$ elements, such that $P_A = \sum_{i=1}^{s} p_i$, $P_B = \sum_{i=s+1}^{n} p_i$, and $P(A) + P(B) = 1$. Then, the entropy of each class is given by:

$$H(A) = -\sum_{i=1}^{s} \frac{p_i}{P_A}\, log\, \frac{p_i}{P_A}, \tag{2.2}$$

$$H(B) = -\sum_{i=s+1}^{n} \frac{p_i}{P_B}\, log\, \frac{p_i}{P_B}. \tag{2.3}$$

Under the MEP, one would find the values of s that maximize $H(A) + H(B)$, that is:

$$ME = \arg \max_{s=1,2,...,n} \{H(A) + H(B)\}, \tag{2.4}$$

where ME is the quantitative value of the maximum entropy.

To analyze DNA or RNA sequences using the MEP, it is first necessary to calculate the frequency of occurrence of the different k-mers in the sequences. By defining a k-mer length (k) and step (distance or stride between consecutive k-mers), a histogram of the frequencies of all k-mers is generated, after traversing all the sequences (see Figure 2.1).

Figure 2.1: Iterating over sequences, with k-mer length 6 and step 1 (example).

Sorting the histogram in descending order, and hypothetically considering the existence of two classes or distributions in the histogram (informative k-mers and non-informative k-mers), the histogram can be divided into two distributions, A and B, calculating the respective entropies. This is demonstrated in Figure 2.2, where the s distributions are mounted aggregating in each $H(A)$ a k-mer frequency value that was part of the previous distribution $H(B)$, to compare the sum value ME.

Creating a distribution of sums $H(A)+H(B)$, the highest value indicates the position where the k-mer that will be used as the cutoff threshold to select the most informative k-mers is located, excluding the possible noise and biases that the sequences may have.

2.3 Hashing Algorithms

This section presents two hashing algorithms employed during the implementation of the method proposed in this work. Since the methodology was designed to handle genomic sequences of varying lengths and potentially large datasets, computational efficiency became

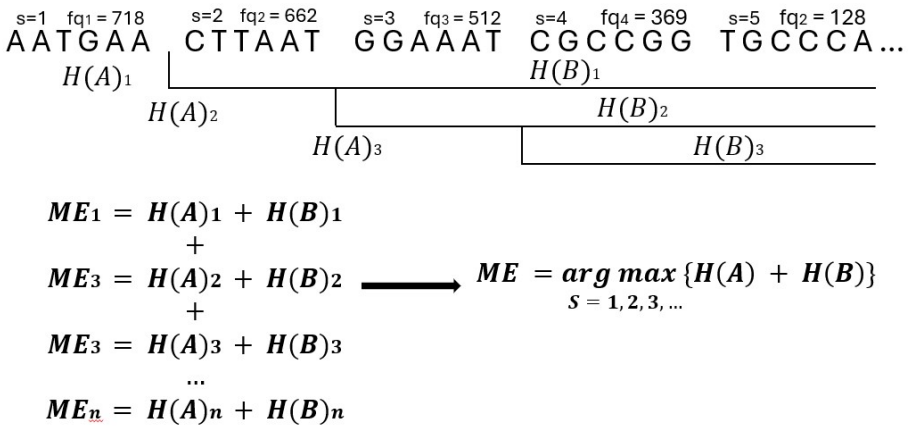


Figure 2.2: MEP principle applied to the k-mer frequency distribution (example).

a critical requirement for the core processing functions. Following feasibility studies, two hashing algorithms with optimal time complexity were selected: one for sliding window searches and another for hash-based comparisons and correlation storage and retrieval, the latter based on MinHash as implemented in the Mash framework [17].

2.3.1 Rolling Hash Algorithm

The Rolling Hash algorithm enables efficient computation of hash values for fixed-length substrings by using a sliding-window approach. Rather than recalculating the hash from scratch for each position, the algorithm incrementally updates the hash value based on the previous computation, significantly reducing the computational complexity from $O(s \cdot k)$ to $O(s + k)$ for pattern matching, where s is the length of the sequences and k is the length of the k-mer pattern [18].

The algorithm can be conceptualized as viewing a sequence through a moving window, where the hash value is continuously adjusted as the window slides along the input. This approach is particularly valuable in bioinformatics applications, such as k-mer analysis, where numerous overlapping substrings must be processed efficiently [19].

Mathematically, a Rolling Hash treats each substring as a number in a fixed-base. For a given base-256 system using ASCII characters, the hash value is calculated as:

$$Hash('ACT') = A \cdot 256^2 + C \cdot 256^1 + T \tag{2.5}$$

Noting that A, C, T are $ASCII$ code characters, sliding the window from "ACT" to "CTG" does not require an entire recalculation (see Equation 2.6): the new hash value can be calculated from the previous one by removing the value of the leftmost character, shifting the remaining characters, and adding the new character (see Equation 2.7).

$$Hash('CTG') = C \cdot 256^2 + T \cdot 256^1 + G \tag{2.6}$$

$$Hash('CTG') = (Hash("ACT") - A \cdot 256^2) \cdot 256 + G \tag{2.7}$$

2.3.2 MinHash and Mash Algorithms

MinHash is a probabilistic hashing algorithm designed to rapidly estimate the similarity between high-dimensional datasets by reducing them through sketch representations. This approach efficiently captures informal notions of "approximately equal" and "approximately contained" relationships between sets [17], [20].

MinHash is based on the generation of sketches, a compact data structure that stores reduced sets of data, where, for a given sketch size s, MinHash retains the s smallest hash values (bottom k sketch) from all k-mers in a sequence. The comparison of similarity between sketches is based on the resemblance measure, defined as the intersection size divided by the union size, which produces the Jaccard index (Equation 2.8) [17].

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.8}$$

The algorithm efficiently approximates similarity by addressing the computational challenge of comparing high-dimensional spaces, estimating the Jaccard index by comparing sketch elements. The shared fraction among s hashes provides an estimate of both the resemblance and containment relationships, where the containment measure, shown

in Equation 2.9, is an impartial estimate of the Jaccard index measured by these sketches.

$$C(A, B) = \frac{|A \cap B|}{|A|} \tag{2.9}$$

This sketch-based approach reduces computational complexity from $O(|A \cup B|)$ to $O(s)$, where s is the sketch size, enabling rapid similarity assessments across large genomic datasets while maintaining statistical precision proportional to the sketch size.

Mash is a methodology that increments MinHash, a technique for dimensionality reduction from Hash sketches, including pairwise mutation and a significant P-value test to efficiently cluster, search, and estimate similarity between large datasets [17].

2.4 Data Modeling Approaches

This section presents the fundamental data modeling concepts underlying the methodology developed in this work, which derives patterns from the input data. The data-driven approach and the parametric versus non-parametric distinction are introduced to establish their relevance for how the GREAC classifier works using internal data information (the k-mers), without imposing restrictive assumptions about the underlying data structure.

2.4.1 Data-driven Approach

A data-driven modeling approach maintains data characteristics throughout the pattern extraction process. By grounding the model in empirical analysis of intrinsic properties of the study case (such as sequence variability and k-mer frequency distributions), this approach avoids *a priori* assumptions about underlying data structures and relies solely on observed properties [21]. For this work, this concept appears aligned with two other types of machine learning modeling, which can be classified as parametric and non-parametric.

2.4.2 Parametric and Non-parametric Models

Parametric models, departing from purely data-driven approaches, impose predefined distributions and create models with fixed numbers of parameters, instead of directly inferring patterns from observed data. In contrast, non-parametric models do not assume predefined distributions, with their parameter count derived from dataset analysis. This modeling approach is capable of quantifying divergence in variant sequences, allowing robust detection of discriminative genomic regions.

2.5 Related Work

This section reviews previous works related to the main fields of this study (genomic sequence classification, discriminative k-mer extraction, and identification of evolutionary patterns), listing foundational bioinformatics tools and classical statistical methods.

2.5.1 BLAST

Basic Local Alignment Search Tool (BLAST) [22] is the reference tool of alignment-based methods, remaining one of the most widely used tools in bioinformatics for sequence similarity searching. It employs heuristic algorithms to find local alignments between sequences, making it computationally efficient for large-scale database searches. BLAST uses a seed-and-extend approach, first identifying short matches (seeds) and then extending them to find longer alignments. While BLAST is highly effective for identifying homologous sequences and functional domains, its computational complexity increases significantly with database size and sequence length, limiting its scalability for whole-genome comparisons and high-throughput analysis.

2.5.2 Fixation Index and PCA

Within alignment-based methodologies, one of the most widely used methods for SNP selection remains the estimation of population differentiation using the Fixation Index,

F_{ST}, which measures allele frequency differences between two populations [23]. This methodology has an inherent limitation as it only permits direct comparison between two populations at a time. Another popular approach is Principal Component Analysis (PCA)[24]. These two methods are commonly used to identify discriminative information, SNPs and genetic variation that differentiates populations in genotypic datasets [25].

2.5.3 GRAMEP

Genome vaRiation Analysis from the Maximum Entropy Principle (GRAMEP) [16] is an alignment-free approach that adopts the principle of maximum entropy for the identification of SNPs and exclusive k-mers as previously mentioned. GRAMEP offers advanced functionalities, including the detection and extraction of variant-specific k-mer sets, from mutation analysis compared to reference sequences, that are the base information source used in this study for the extraction of discriminative regions and pattern recognition.

2.5.4 CASTOR-KRFE

Classification And Segmentation Tool for Optical Recognition - K-mer Recursive Feature Elimination (CASTOR-KRFE) [3] is an alignment-free method to identify a set of k-mers to discriminate between groups of genomic sequences. The core of CASTOR-KRFE is based on feature elimination using Support Vector Machines (SVM), which is a machine learning feature selection method. CASTOR-KRFE provides results in a wide range of viruses and shows better performance on complex virus data. The method is particularly effective for viral genome classification, using recursive feature elimination to identify the minimal set of k-mers that best discriminate between different viral species or strains.

2.5.5 KEVOLVE

K-mer Evolution-based Viral Genome Classifier (KEVOLVE) [2] is a machine learning-based approach that efficiently identifies variant-specific genomic signatures among viral sequences. These signatures are defined as pervasive motifs in the viral genome that

allow discrimination between species or variants. The method uses genetic algorithms with machine learning to identify distinctive genomic signatures. KEVOLVE represents an innovative application of evolutionary optimization algorithms to biological sequence analysis, where genetic algorithms evolve populations of k-mer sets to maximize their discriminative power for viral classification tasks.

2.5.6 RRM

Resonant Recognition Model (RRM) is a digital signal processing method that is applied mainly to protein sequences [26]. RRM operates by transforming amino acid sequences into numerical series using Electron-Ion Interaction Potention (EIIP) values. The EIIP values are numerical representations of nucleotides (A, T, G, C, U) and amino-acids. These values are derived from the average energy of the valence electrons in the nucleotide (the values are displayed in the appendix C: Table C.1 has the values for the nucleotides, and Table C.2 has the values for the amino-acids).

These numerical series are then converted into the frequency domain using techniques such as the Fast Fourier Transform (FFT) to analyze their frequency spectra. The core idea is to identify common frequencies through cross-spectrum functions that are believed to correspond to specific biological functionalities. To identify common frequencies among multiple sequences with similar biological functions, RRM uses a cross-spectrum function (Equation 2.10). For two frequency spectra, this function multiplies the DFT coefficient of one series by the conjugate complex of the DFT coefficients of another series, thereby amplifying common frequencies and highlighting prominent peaks.

$$S_n = X_n Y_n^* \quad n = 1, 2, 3, ..., N/2 \tag{2.10}$$

However, the RRM approach presents certain limitations: when applied to the direct identification of specific positions within DNA or RNA sequences, the frequencies extracted by RRM, while indicative of biological functions, do not directly map to precise positions in DNA sequences, and more critically, the RNA genetic code is non-bijective,

meaning multiple codon triplets can code for the same amino acid, making accurate reverse-translation from protein sequences back to nucleotide sequences infeasible without relying on alignment with the original sequence. These limitations are widely discussed in Subsection 3.3.1 as the justification to use k-mers and why GREAC was developed.

Chapter 3

Materials and Methods

This chapter describes the Genomic Region Extraction and Classifier (GREAC) methodology proposed in this work. It starts by introducing the datasets used to support the initial development of the method. It then lays out and explains its main steps (genomic region extraction, k-mer analysis, dimensionality reduction, and classification).

3.1 Support Datasets

Developing a new data analysis methodology requires careful selection of the datasets used to validate the methodology throughout its various design and development stages. GREAC's development was conducted in parallel with its validation using viral genomes. This was grounded on the fact that viruses exhibit the highest known mutation rates and recombination frequencies among biological organisms, presenting a computational challenge where the detection of generalist patterns is particularly complex [27], [28].

The collection of datasets used in this work is shown in Table 3.1. These pertain to five epidemiologically significant viral pathogens: SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2), DENV (dengue virus), HBV (hepatitis B virus), Monkeypox (Orthopoxvirus monkeypox), and HIV-1 (human immunodeficiency virus type 1).

This diverse collection enables a comprehensive evaluation of methodologies across different types, sizes, and evolutionary patterns of the viral genome, providing a robust

Table 3.1: Viral Datasets Overview

Dataset (Source)	Variant	Average Length (σ)	Sequence Amount
SARS-CoV-2 (KEVOLVE)	Alpha	29,838 (55.72)	175,212
	Beta	29,810 (56.96)	695
	Delta	29,801 (61.85)	9,408
	Epsilon	29,834 (63.59)	14,674
	Eta	29,741 (70.91)	716
	Gamma	29,810 (63.19)	8,129
	Iota	29,802 (56.11)	19,274
	Kappa	29,841 (52.04)	127
	Lambda	29,793 (55.19)	428
	Omicron	29,762 (65.99)	106,293
	Total		334,956
HIV (CASTOR-KRFE)	HIV1_A	8,878 (438.02)	133
	HIV1_B	9,356 (439.44)	130
	HIV1_C	9,148 (349.10)	130
	HIV1_D	8,966 (381.55)	89
	HIV1_F	8,953 (361.35)	91
	HIV1_G	9,041 (363.94)	94
	Total		667
DENV (GRAMEP)	Type 1	10,585 (196.02)	2,571
	Type 2	10,647 (137.46)	1,756
	Type 3	10,647 (137.46)	1,272
	Type 4	10,422 (266.50)	462
	Total		6,061
HBV	A	3,220 (5.14)	818
	B	3,215 (2.48)	1,790
	C	3,213 (6.24)	2,724
	D	3,182 (3.32)	1,129
	E	3,211 (3.88)	298
	F	3,214 (3.84)	187
	Total		6,946
MONKEYPOX	I	196,058 (1537.40)	819
	II	196,952 (1482.09)	2,727
	Total		3,546

framework for assessing the generalizability and performance of our computational approaches across various viral families and genomic characteristics.

The selected datasets were obtained from previous studies and from public repositories, allowing direct comparative evaluation across different methodologies. The SARS-CoV-2 dataset corresponds to the same collection used in KEVOLVE [2]; the HIV dataset was obtained from CASTOR-KRFE [3]; the DENV dataset was derived from GRAMEP [16]; the HBV dataset was obtained from HBVdb [29]; and the Monkeypox sequences comprise all complete genomes from both lineages retrieved from the NCBI Virus database. Such a variety of datasets helps ensure methodological consistency and allows fair comparison of classification performance across established benchmarks in viral genomics research.

For exclusive k-mer extraction using GRAMEP and MinHash sketch comparison construction on the XGBoost feature vector, the following reference sequences from the National Center for Biotechnology Information (NCBI) were employed: a) for SARS-CoV-2 analysis, the Wuhan reference genome with identification NC_045512.2; b) for Monkeypox, the genome assembly ViralProj15142 with identification GCF_000857045.1; c) for Dengue (DENV), the reference sequence with identification NC_001477.1; and for d) Hepatitis B (HBV), the reference genome NC_003977.2.

3.2 GREAC workflow

The GREAC methodology workflow comprises three main steps, as illustrated in Figure 3.1. The rationale for each of these steps is described in the next sections.

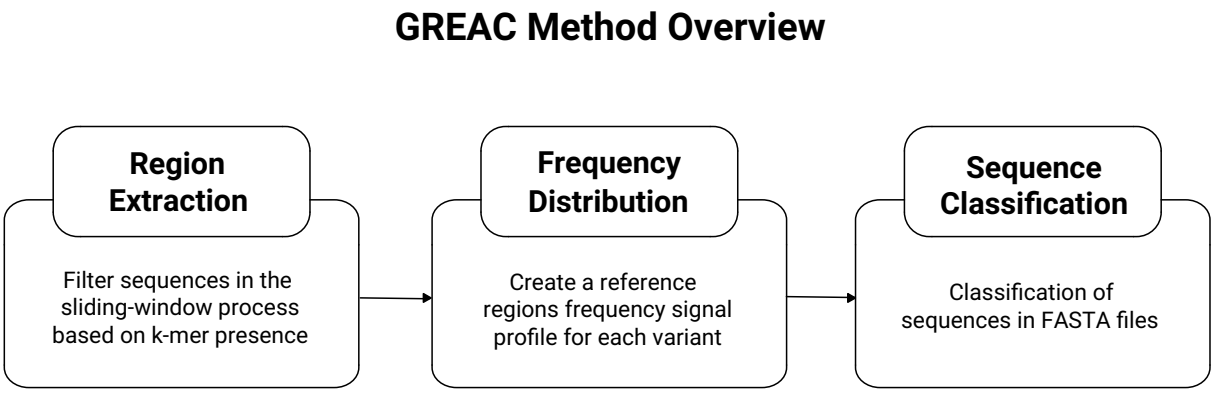


Figure 3.1: GREAC method overview

The implementation includes one extra preliminary step, based on a different tool. Details on this step, as well as the format of the inputs and outputs between successive stages, are provided in Chapter 4.

3.3 Region extraction based on k-mer appearances

Identifying discriminative regions within genomic sequences requires establishing variant-specific information sources that can distinguish between different variants. K-mer analysis represents a well-established genomic discrimination approach that employs word-based methodologies, similar to single-nucleotide polymorphisms (SNPs) but with substantially larger information, providing enhanced discriminative power for sequence classification. To identify these discriminative k-mers, it is employed GRAMEP [16], an alignment-free open-source tool that applies Shannon entropy and max entropy principles to extract the exclusive k-mer sets for each variant sequences of the dataset.

The region identification is performed individually for each set of variant sequences of the dataset, acting as a filter to find k-mer occurrences over the genome. Algorithmically, the implementation (see section 4.4) follows a sliding-window approach to enhance the performance. This approach is well-established in related previous studies and serves as a standard algorithm for data-driven genomic analyses [16], [30]–[32].

The window size calculation is formalized in Equation 3.1, where $|S|$ represents the vector containing all the lengths of the sequences, and P denotes the selected percentage value, thus determining the window size based on the shortest sequence length.

$$W(|S|, P) = argmin(|S|) \cdot P \tag{3.1}$$

During the sliding-window process, each window containing at least one of the k-mers from the variant-specific set increments a positional counter. This counter, after processing all the training samples and traversing the entire genomes, generates a frequency

histogram, where peaks indicate regions with high concentrations of discriminative k-mers. Figure 3.2 shows the result of this step applied to the viral datasets selected for this work (recall section 3.1).

The selected regions of the histograms have a threshold parameter, representing the minimum percentage of appearance among all the training samples. Applying that threshold (see Figure 3.3) refines the set of windows (based on their positions) to be further considered in the next steps of the methodology. With the selected windows, it is possible to retrieve all nucleotide positions and extract variant-specific regions.

Next, it is necessary to combine all variant-specific regions into a unified positional conjunction. This process enables a suitable comparative analysis across the variants, ensuring consistent regional representation while preserving the discriminative characteristics identified through k-mer frequency analysis.

An example of the effectiveness of dimensionality reduction using discriminatory regions for the four analyzed organisms is presented in Figure 3.4. In organisms such as SARS-CoV-2 and Monkeypox, the reduction in dimensionality is evident through the selection of specific genomic regions. When analyzing the origin of these extracted regions, it becomes clear that they predominantly belong to the coding sequences of structural proteins of the SARS-CoV-2 virus, as exemplified by the method [33], which justifies the interpretative power associated with the method.

3.3.1 Using K-mers as information over EIIP values

The selection of k-mers as the primary information source for region extraction and subsequent pattern discovery was not arbitrary, as explained next.

Previous studies have demonstrated the effectiveness of numerical representations for genomic sequences, particularly through the use of Electron-Ion Interaction Potential (EIIP) values, where each amino acid is assigned a specific numerical value. This approach extends to the translation from DNA to RNA and subsequently to the primary protein sequence, with each one receiving its corresponding EIIP value.

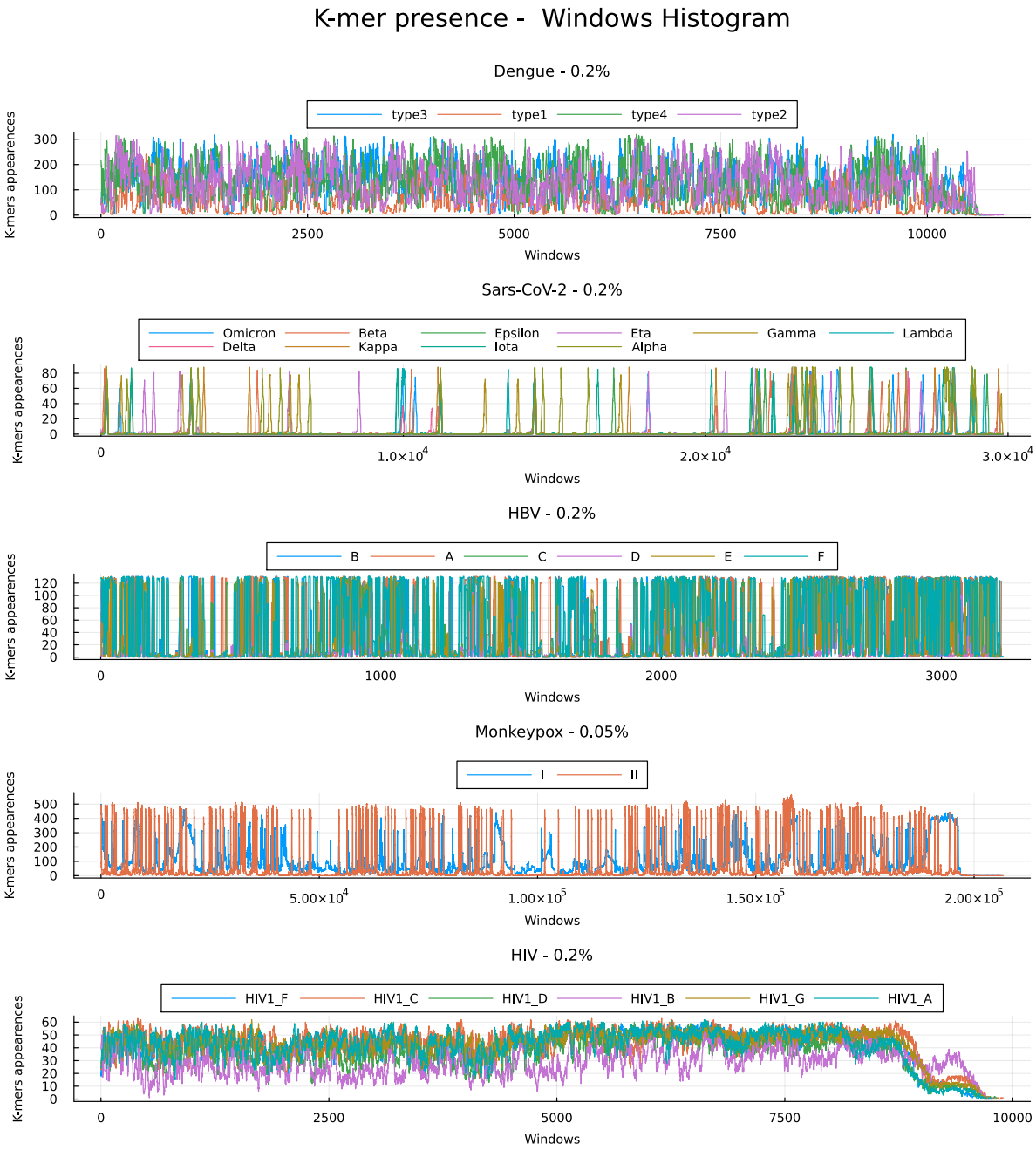


Figure 3.2: Histograms counting the presence of an exclusive k-mer in each region among all the training samples. Each chart corresponds to a different virus histogram, where the x-axis represents the windows and the y-axis the count of how many of the training samples have the presence of one of the k-mers in that region.

Using numerical series to represent amino acid sequences enables the application of digital signal processing methods, such as the Resonant Recognition Model (RRM) [26].

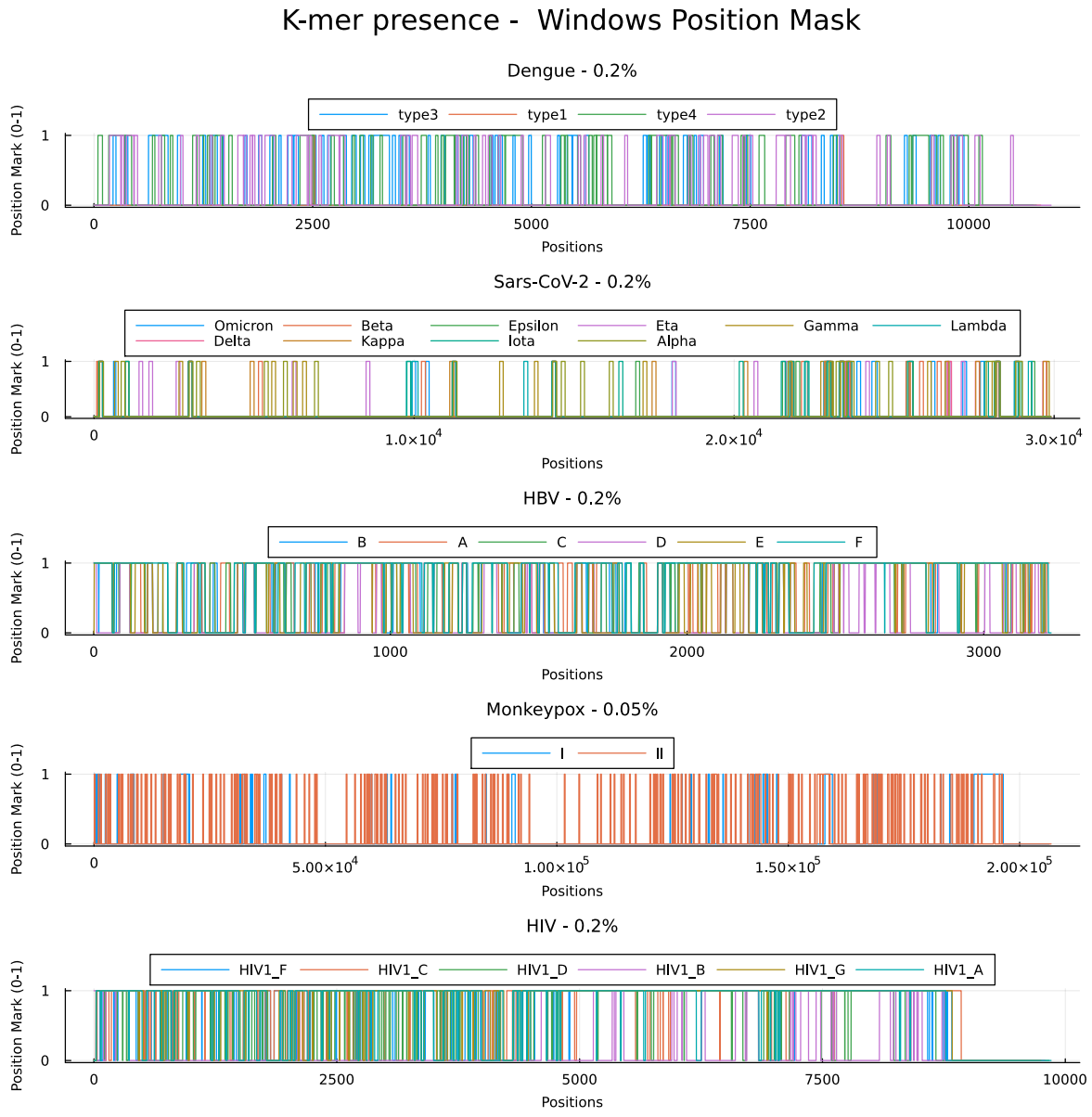
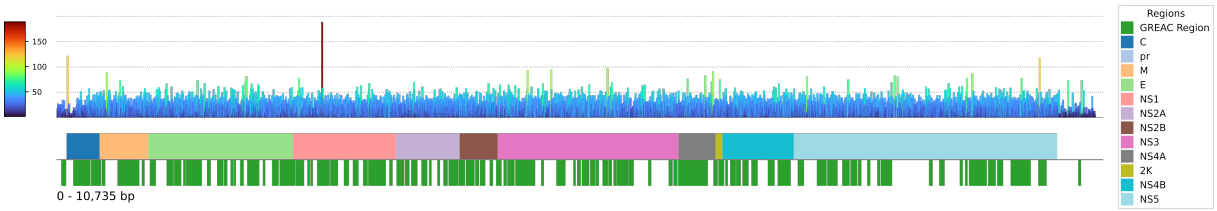
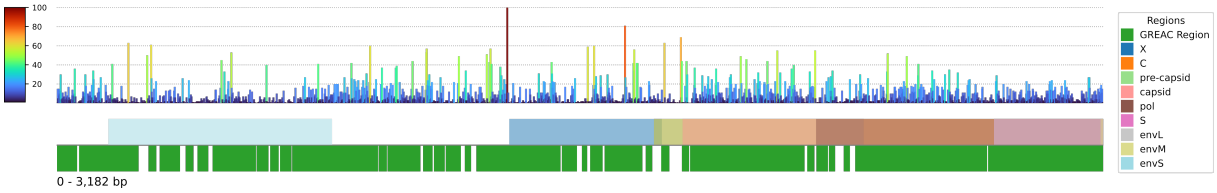


Figure 3.3: Extracted regions' positions in each variant by histogram. Each chart shows, respectively, the position mask for each virus, where the x-axis represents the sequence position and the y-axis represents whether the position is part of the region.

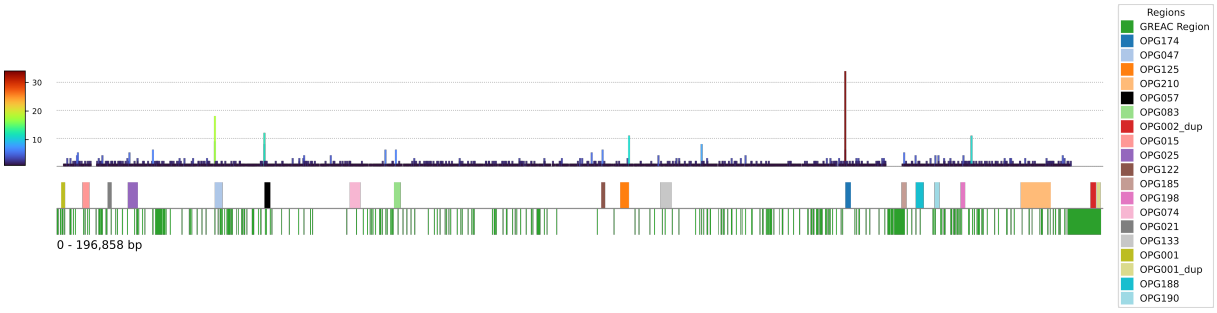
This method involves converting primary protein sequences into EIIP numerical series and subsequently transforming them to the frequency domain via the Fast Fourier Transformation (FFT). The multiplication of these signals (*cross-spectrum* function) allows for the extraction of the most discriminative information regarding their biological functions.



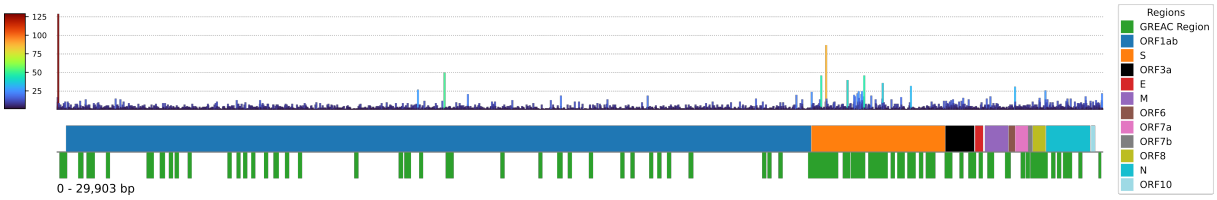
(a) DENV



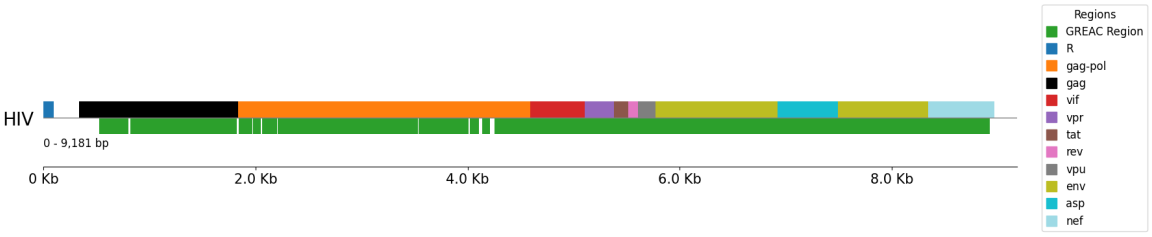
(b) HBV



(c) MPOX



(d) SARS-CoV2



(e) HIV

Figure 3.4: Genomic annotations and regions identified as important for different viruses. Each subfigure represents a viral genome: 3.4a) DENV (Dengue virus), 3.4b) HBV (Hepatitis B virus), 3.4c) MPOX (Monkeypox), 3.4d) SARS-CoV-2, and 3.4e) HIV (Human Immunodeficiency Virus). Known genomic annotations are shown in the upper tracks of each subfigure. Regions highlighted as "important" are shown in the lower tracks, indicating overlap with previously annotated functional elements and revealing potentially relevant regions.

Previous investigations employing the RRM methodology have yielded promising results when applied to genomic data from diverse organisms. Specifically, studies utilizing DNA reads from PLEK [34] and CPC2 [35] tool datasets have successfully classified non-coding RNA (ncRNA) and messenger RNA (mRNA) sequences [36].

The transition from EIIP to k-mer representation was motivated by two fundamental limitations of the RRM approach when applied to direct sequence position identification:

1. The frequencies extracted through RRM that represent biological functions do not correspond to direct positions in DNA sequences, as these frequencies only indicate the possibility of applying filters that would transform protein sequence signals into sinusoidal waves.

2. More critically, there is no feasible method for reverse-translating protein sequences back to nucleotide sequences, since the genetic code represents a non-bijective function where each codon triplet codes for a specific amino acid, but multiple codons can code for the same amino acid. Any solution would require alignment with the original sequence, which contradicts the principle of alignment-free methodology and would introduce computational complexity far beyond the scope of this study.

Therefore, k-mers provide a more direct and tractable approach to positional analysis while maintaining the numerical processing advantages of the established methodologies.

3.4 K-mer frequency distribution reference

With the region's positions extracted, frequency values are aggregated based on the *relative frequency*, which is used to calculate the number of times the event occurs in a given set of possible occurrences [37], this set comprising all the variants exclusive k-mers.

The relative frequency of the regions is measured using Equation 3.2, considering all variant training sequences, where K is the k-mer set, $|K|$ denotes the total number of k-mers, and N is the number of samples. The numerator is the sum of $count(K)$ across

the samples, which counts the number of k-mers that appear in each region. This way, a normalized frequency is produced for each wr region.

$$F_{wr} = \frac{1}{N \cdot |K|} \sum_{i=0}^{N} count(K)_i \qquad wr = 1, 2, 3, ..., R \tag{3.2}$$

Having all relative frequencies of the regions for each variant, a reference signal profile is mounted, representing variant-specific frequency patterns and showing the organism behavior, as illustrated in Figure 3.5.

At this stage of the GREAC methodology, a data-driven concept is enforced, using the regions extracted from the sliding-window with specific k-mers, from all data sequences to avoid *a priori* assumptions about the data [21].

3.5 Sequence classification approaches

Using the k-mer frequencies reference signals, in addition to the distances between these signals, it is possible to create distribution data around those reference distance values.

Distributions and signals are valuable data for explaining and classifying behavior, and different digital signal processing methods focus on addressing different problems with them. Distances and divergence equations appear as possible approaches, where these metrics enable quantification and describe the closeness between probability (frequency) distributions [38], [39]. Next, the various approaches adopted in GREAC are discussed.

3.5.1 Distances Metrics

Three distance metrics were evaluated and used as feature possibilities for classification: i) the Euclidean distance, the common standard distance (Equation 3.3); ii) the Manhattan distance, with a more direct comparison (Equation 3.4); and iii) the Mahalanobis distance (Equation 3.5), for a more robust and refined metric, focusing on calculating the difference between two distinct groups (here two value vectors), using the inverse covariance matrix (Σ^{-1}) to encode the relationship between different characteristics and scale values [40].

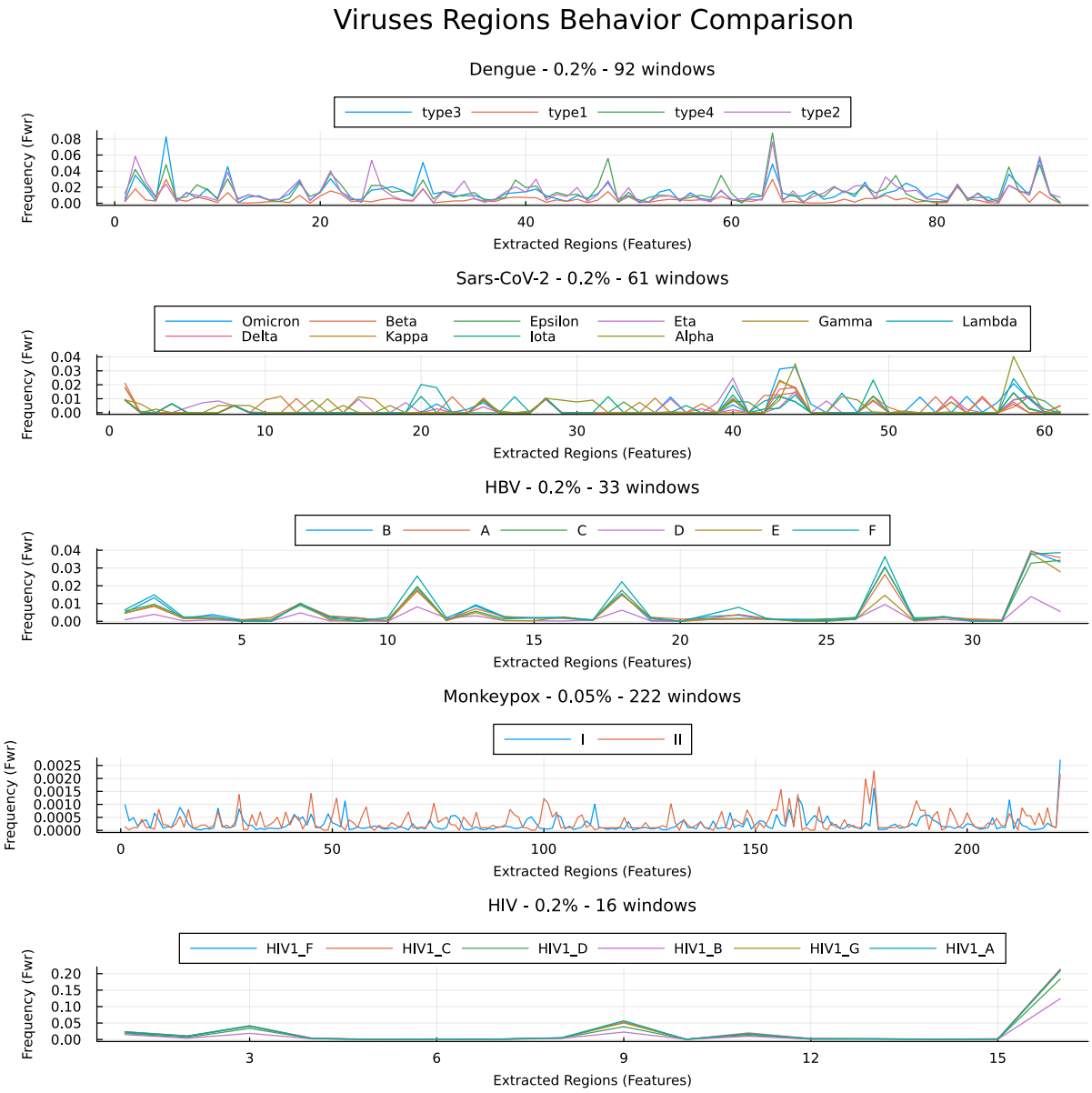


Figure 3.5: Reference behavior signals created using Equation 3.2. Each chart shows the behavior of each virus variant, where the x-axis represents the extracted regions and the y-axis represents the relative frequency of k-mers.

$$D(Q,P) = \sqrt{\sum (Q_i - P_i)^2} \tag{3.3}$$

$$D(Q,P) = \sum |(Q_i - P_i)| \tag{3.4}$$

$$D(Q,P) = \sqrt{(Q-P)^T \cdot \Sigma^{-1}(Q-P)} \tag{3.5}$$

3.5.2 Kullback–Leibler Divergence

The Kullback-Leibler divergence, given by Equation 3.6, is a metric capable of measuring the divergence between two probability distributions [41].

$$D_{KL}(P||Q) = \sum P_i \log(\frac{P_i}{Q_i}) \tag{3.6}$$

The frequencies of the regions are probabilistic values, given by Q, where the feature value is the number of k-mers that appear, given the total k-mer set (the frequency values of each region extracted in the input sequence for classification).

The P distribution, our reference signal, can be interpreted as the *"true probability distribution"* (frequency value amount for all samples shown in Equation 3.2).

3.5.3 Gaussian Membership Attribution

For multiple classes problems, the fuzzy sets approach uses membership functions to assign degrees of belonging based on distance values. Membership degrees indicate how closely the new input values correspond to each class [42].

The Gaussian Membership function $\mu(d)$, calculated using Equation 3.7, is used to determine whether that sequence belongs to a particular class [43], where d represents the calculated distance between the input sequence and the variant reference distribution, and μ and σ are the mean and standard deviation distances within the variant group.

$$\mu(d) = \exp\left(-\frac{(d-\mu)^2}{2\sigma^2}\right) \tag{3.7}$$

The equation outputs values between 0 and 1 for each variant reference signal, making it possible for classification by applying the *argmax* method to select the variant with the maximum degree of membership or to be used as a feature in classifiers such as XGBoost.

3.5.4 XGBoost Classifier

The XGBoost algorithm is a scalable machine learning system for tree boosting [44]. In GREAC, it improves classification accuracy by analyzing more complex genomes with high variation, allowing a more comprehensive understanding of the components that comprise the signals, while maintaining the performance and explainability of the methodology.

The training features incorporate the measured frequency signals from the sequences, the previously calculated distance and membership values between comparative reference signals, and a new feature developed to capture variations between windows. This feature is a new vector calculated using Equation 3.8, where $diverg[n]$ represents the divergence signal between the relative frequency of the k-mer set of the current window and the consecutive window of the signal x.

$$diverg[n] = x[n+1] - x[n] \tag{3.8}$$

The aggregation of this signal in the model's feature vector enhances the classificatory power, allowing for the identification of internal variation between signal windows as a behavioral pattern across frequency values.

Another complement is the measure of the Jaccard similarity for each extracted region against the reference sequence using the MinHash algorithm (recall Section 2.3.2). The integration is further enhanced by complementing the Jaccard similarity scores with Mash (recall also Section 2.3.2) distance metrics and their corresponding P-values, which provide statistical significance measures for each similarity assessment.

The complete set of these features (Jaccard indices, Mash distances, and P-values), are concatenated into the XGBoost training vector, creating a comprehensive representation that captures both structural similarity patterns and their statistical reliability.

For model training, default parameters of the XGBoost implementation are used, setting a multiclass classification with the *softprob* activation function. This configuration produces as output the probabilities of belonging to each class, allowing a more granular analysis of the uncertainty associated with each prediction.

The classification process is executed for each variant available in the dataset, since one of the main characteristics is the distance between the F_{wr} signal corresponding to each class, the Gaussian membership, and the window frequency signal of the input data to be classified. For each individual classification, the probability assigned by the classifier is extracted. Subsequently, the *argmax* function is applied to all probabilities obtained to determine the final class with the highest confidence.

This process involves assembling the set of probabilities from *softprob* classifications into a probability matrix $\mathbf{P}$ $n \times n$, where n is the number of classes.

$$\mathbf{P} = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{pmatrix} \quad \mathbf{d} = \begin{pmatrix} P_{11} \\ P_{22} \\ \vdots \\ P_{nn} \end{pmatrix} \tag{3.9}$$

In matrix $\mathbf{P}$ (Equation 3.9), each row i represents the probabilities when the model iterates over class j, and P_{ij} is the probability that the sample belongs to class j when iterating over class i (diagonal element), where $\sum_{j=1}^{n} P_{ij} = 1$ for each row i. Thus, one can extract the diagonal vector $\mathbf{d}$ and finalize the classification through **argmax** of the diagonal vector where $\hat{y}$ is the predicted class, as shown in Equation 3.10.

$$\hat{y} = \underset{i \in \{1,2,\ldots,n\}}{\arg\max} \; P_{ii} \tag{3.10}$$

3.5.5 Metric Performance Evaluation

Robust performance evaluation is crucial for validating the reliability and generalizability of machine learning classification models, particularly in genomic classification tasks, where class imbalance and biological variability are prevalent. The results from this phase reflect the expected behavior in practical applications in real-world environments.

Widely recognized evaluation metrics were adopted, the main one being the *confusion matrix*, which was used throughout this work to present the results in a reliable way.

Other metrics, such as *precision*, *recall*, and *F1-Score*, defined in Equations 3.11–3.13, respectively, are provided in the tables in the Appendices for a more detailed analysis.

$$Precision = \frac{1}{N}\sum_{i=0}^{N}\frac{TP_i}{TP_i + FP_i} \tag{3.11}$$

$$Recall = \frac{1}{N}\sum_{i=0}^{N}\frac{TP_i}{TP_i + FN_i} \tag{3.12}$$

$$F1 - score = \frac{1}{N}\sum_{i=0}^{N} 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i} \tag{3.13}$$

For multiclass classification, these metrics were computed as unweighted averages across all classes to mitigate bias toward prevalent variants.

Chapter 4

Implementation

This chapter focuses on the most relevant technical aspects of the development and implementation of the GREAC methodology presented in the previous chapter.

4.1 Overview

The GREAC software is fully coded in the Julia Language [45], a multi-threaded, high-performance, dynamically typed and compiled programming language, optimized for scientific computing. It is distributed as an open-source tool, making its source code available to run through the Julia interactive command-line REPL (read-eval-print loop) built into the Julia executable, providing flexibility for various deployment scenarios.

The GREAC repository, available in GitHub (see Figure 4.1), includes extensive documentation covering proper usage procedures, Jupyter notebooks demonstrating parameter analysis during feature extraction and modeling processes, and shell scripts providing practical examples for source code execution and toy models. It ensures accessibility for users of different experience levels and facilitates reproducible research workflows.

The core GREAC program is designed for Command-line Interface (CLI) execution, with the documentation providing example shell scripts to guide users through the executions and help customize the tool according to their specific needs.

Recognizing that the primary end-users are biologists researchers who may not have

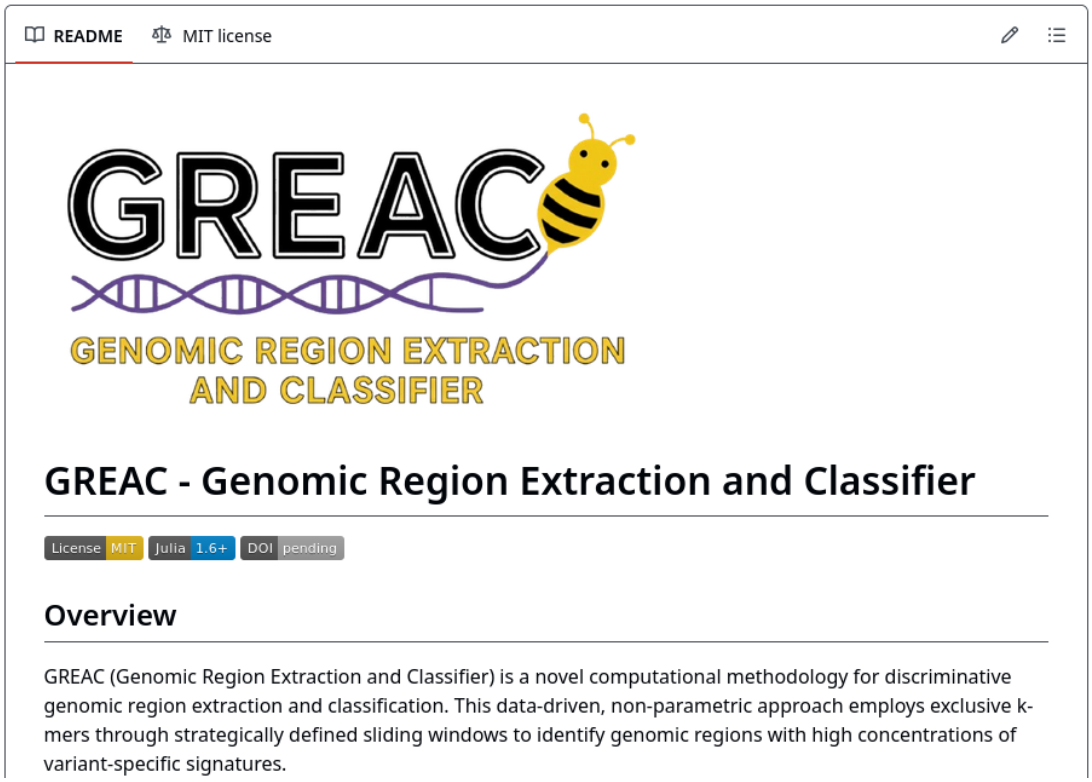


Figure 4.1: README file introduction of GREAC on GitHub.

extensive computational backgrounds, it was also developed a user-friendly web interface, based on the *Streamlit* Python framework (Figure 4.2).

This interface streamlines the workflow by offering an intuitive graphical environment where users can easily input the necessary parameters. The interface then executes the underlying shell scripts with GREAC functionalities as sub-processes, without requiring users to interact directly with the command line.

4.2 Processing Steps

Using GREAC involves four processing steps, of which three belong to the GREAC implementation (Region Extraction, Frequency Distribution, Sequence Classification) and one is based on the execution of the external GRAMEP tool (K-mer Set Extraction).

The latter is the 1st to be executed, producing Input Files that feed the Region

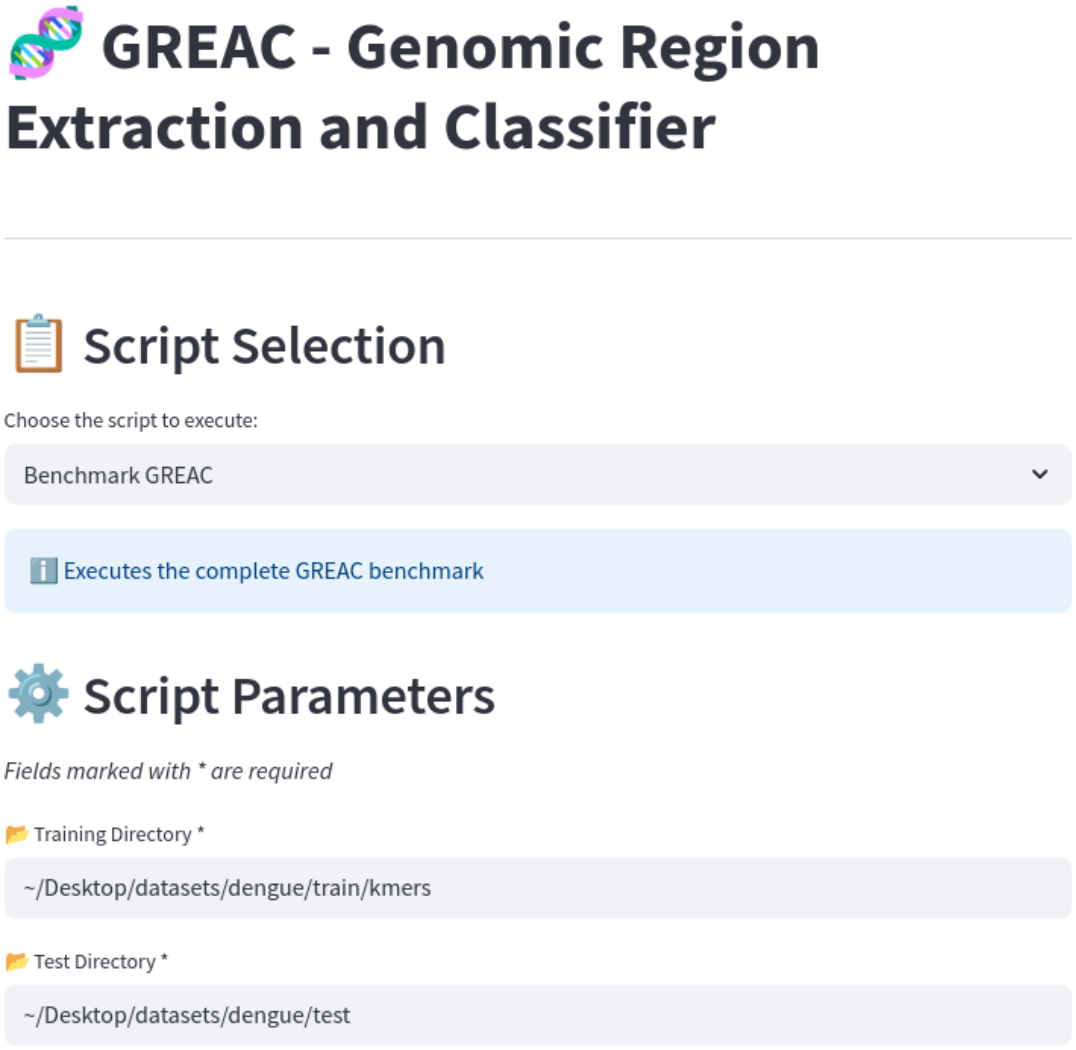


Figure 4.2: Web interface of GREAC built with Streamlit.

Extraction step; the final Outputs of GREAC are files produced as a result of the Sequence Classification final processing step. The GREAC steps are represented in Figure 4.3.

4.3 Main Functions

The GREAC implementation comprises five main functions, also shown in Figure 4.3:

1. ***extract-features***: executes the main methodology to extract the regions and generate reference signals for subsequent classification and genome behavior analysis;

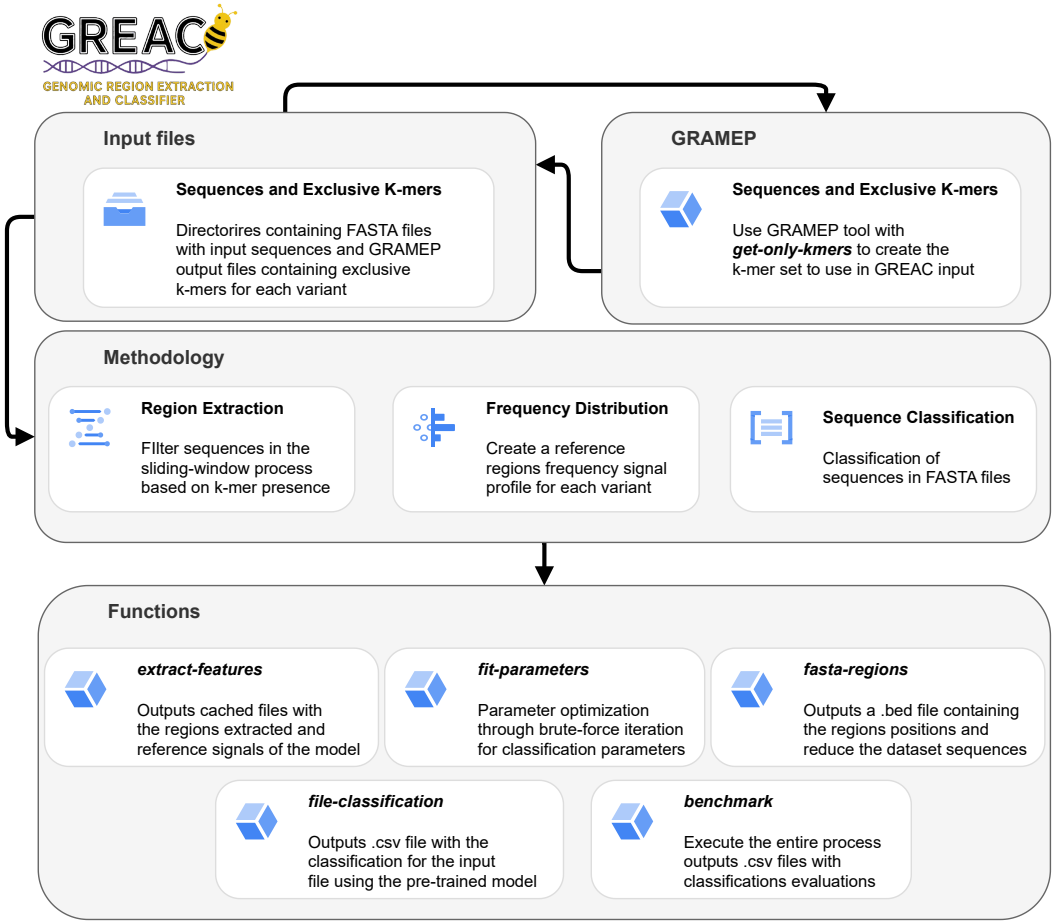


Figure 4.3: GREAC implementation workflow.

2. ***fit-parameters***: executes a comprehensive parameter optimization through brute-force iteration to identify optimal classification parameters, such as frequency threshold values for the regions consideration and window percent size;

3. ***benchmark***: trains the model; creates a data structure for classification, extracts the discriminative regions and creates the referencing signals (as in *extract-features*); also, evaluates performance on input test sequences – classifies the sequences and measures F1-score, precision, recall, and creates a confusion matrix in a `.csv` output file, and another `.csv` with per class probability that leads at the final classification for each sequence, along with a `.pdf` file having the execution summary with the classification metrics and the graphs with the respective frequency signals;

4. ***file-classification***: performs a singular classification for each sequence present in the input sequences file using the pre-trained model;

5. ***fasta-regions***: returns a `.bed` file with three columns delimited by organism name, start position and end position; and if the dataset input file was specified, it is applied the reduction for each variant class file, selecting only the extracted regions positions in each sequence and outputs a `*_extracted.fasta` file.

The program operates on structured data directories following the format specified in the documentation, as shown in Figure 4.4. During execution, GREAC generates model files and stores them in cached files for efficient reuse.

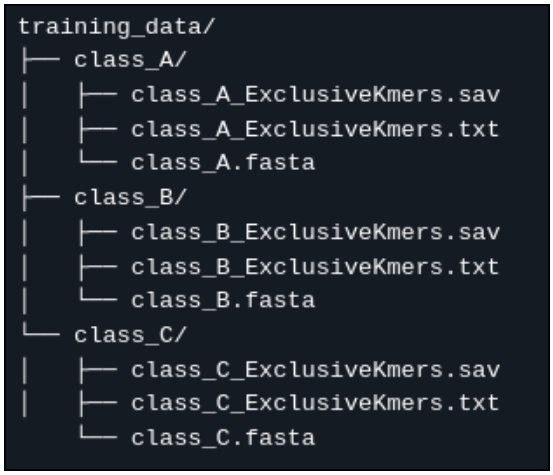


Figure 4.4: GREAC files directory structure. The exclusive k-mers files are the output from GRAMEP, and the *.fasta files have all the sequences from training/extraction.

4.4 Optimizations

The algorithm implementation focuses on performing computationally intensive operations as efficiently as possible and reducing code complexity. Its computational complexity is of order $\mathcal{O}(S \cdot n \cdot K)$, where S is the number of sequences, n is the number of marked positions, and K is the number of k-mers. To achieve this computational complexity, the implementation makes use of the Rolling-Hash algorithm to find regions containing k-mers, retrieving the positions in the sequence (see Listing 4.1).

```julia
function getOccursin_rolling_hash(sequence::String,
    kmer_hash_map::Dict{UInt64,Vector{String}}, k_len::Int )::Vector{Int}
    seq_len = length(sequence)
    positions = Int[]
    if seq_len < k_len
        return positions
    end
    base = UInt64(257)
    power = UInt64(1)
    for i in 1:(k_len-1)
        power *= base
    end
    # Calculate initial hash
    current_hash = UInt64(0)
    @inbounds for i in 1:k_len
        current_hash = current_hash * base + UInt64(sequence[i])
    end
    # Check first k-mer
    if haskey(kmer_hash_map, current_hash)
        kmer_candidate = SubString(sequence, 1, k_len)
        if String(kmer_candidate) in kmer_hash_map[current_hash]
            push!(positions, 1)
        end
    end
    # Roll through sequence
    @inbounds for i in (k_len+1):seq_len
        # Rolling hash: remove leftmost char, add rightmost char
        current_hash = current_hash - UInt64(sequence[i-k_len]) * power
        current_hash = current_hash * base + UInt64(sequence[i])
        # Check if hash matches any k-mer
        if haskey(kmer_hash_map, current_hash)
            kmer_candidate = SubString(sequence, i - k_len + 1, i)
            if String(kmer_candidate) in kmer_hash_map[current_hash]
                push!(positions, i - k_len + 1)
            end
        end
    end
    return positions
end
```

Listing 4.1: Rolling Hash to find K-mer occurence

Concerning memory consumption, loading sequences into string vectors introduced high memory overhead and subsequent processing costs due to their data structure characteristics. To address this challenge, all sequences were loaded as CodeUnits (see Listing 4.2), which are vectors of UInt8 (unsigned 8-bit integers), significantly reducing memory consumption during processing. This approach ensures that, for each sequence that requires region frequency calculations, memory consumption is highly optimized, making GREAC suitable for systems with modest memory resources.

```
function loadCodeUnitsSequences(file::String)::Vector{Base.CodeUnits}
    # FASTX is the library to handle FASTA files
    sequences = Vector{Base.CodeUnits}()
    for record in open(FASTAReader, file)
        push!(sequences, codeunits(sequence(String, record)))
    end
    return sequences
end
```

Listing 4.2: Sequences Loading

Combined with multi-threading capabilities, the process is executed in parallel with minimal memory usage and computational complexity, as shown in the Listing 4.3.

```
function sequence_kmer_distribution_optimized(
    regions::Vector{Tuple{Int,Int}},
    seq::Base.CodeUnits,
    kmerset::Vector{String})::Vector{UInt64}

    kmer_set = Set(codeunits(kmer) for kmer in kmerset)
    kmer_length = length(kmerset[1])
    kmer_distribution::Vector{UInt64} = zeros(UInt64, length(regions))
    seq_len = length(seq)
    @inbounds for (region_idx, (init_pos, end_pos)) in enumerate(regions)
        # Adjust end_pos if necessary
        actual_end = min(end_pos, seq_len)
        # Extract all possible k-mers in only one iteraction
        region_kmers = Set{Vector{UInt8}}()
        for i in init_pos:(actual_end-kmer_length+1)
            kmer_candidate = @view seq[i:(i+kmer_length-1)]
            kmer_vector = Vector{UInt8}(kmer_candidate)

```

```
            if kmer_vector in kmer_set
                push!(region_kmers, kmer_vector)
            end
        end
        kmer_distribution[region_idx] = length(region_kmers)
    end
    return kmer_distribution
end
```

Listing 4.3: Measure the K-mer frequencies for each region

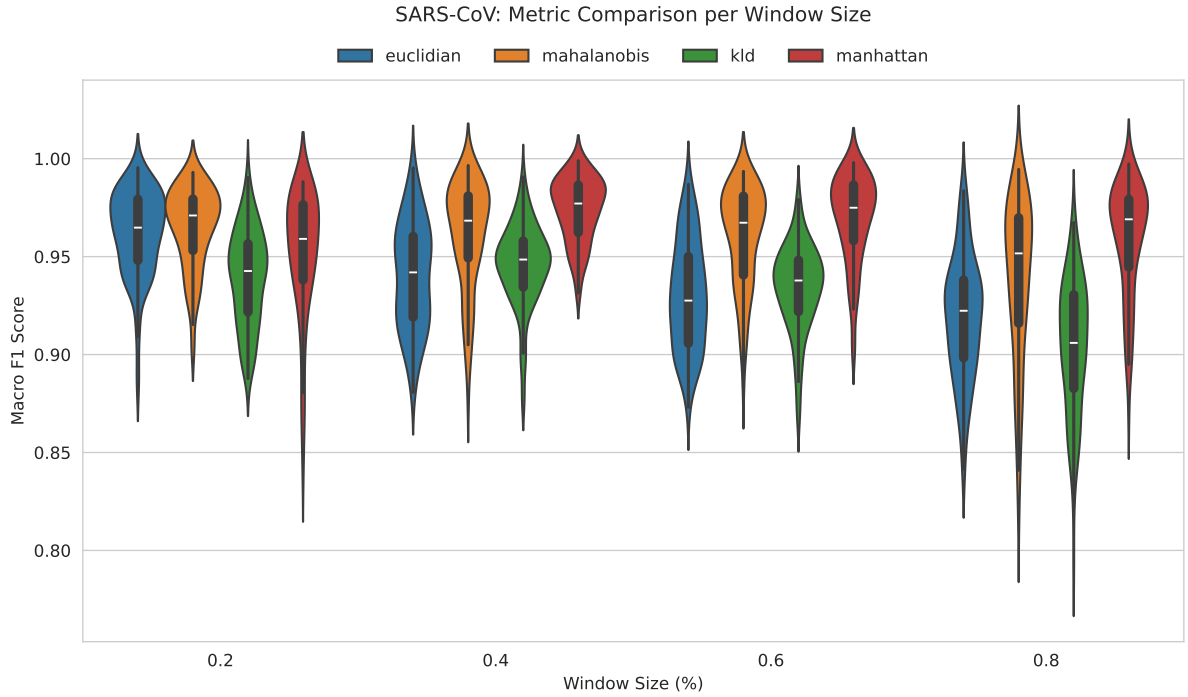
4.5 Choice of Classification Metrics

During GREAC development, a critical decision point arose regarding which metric to use to yield reliable classification results with the XGBoost classifier. Consequently, comprehensive benchmarking experiments were conducted, changing window sizes and comparing results between SARS-CoV and DENV variants, two organisms with distinct behavioral patterns and considerable differences in sequence length.

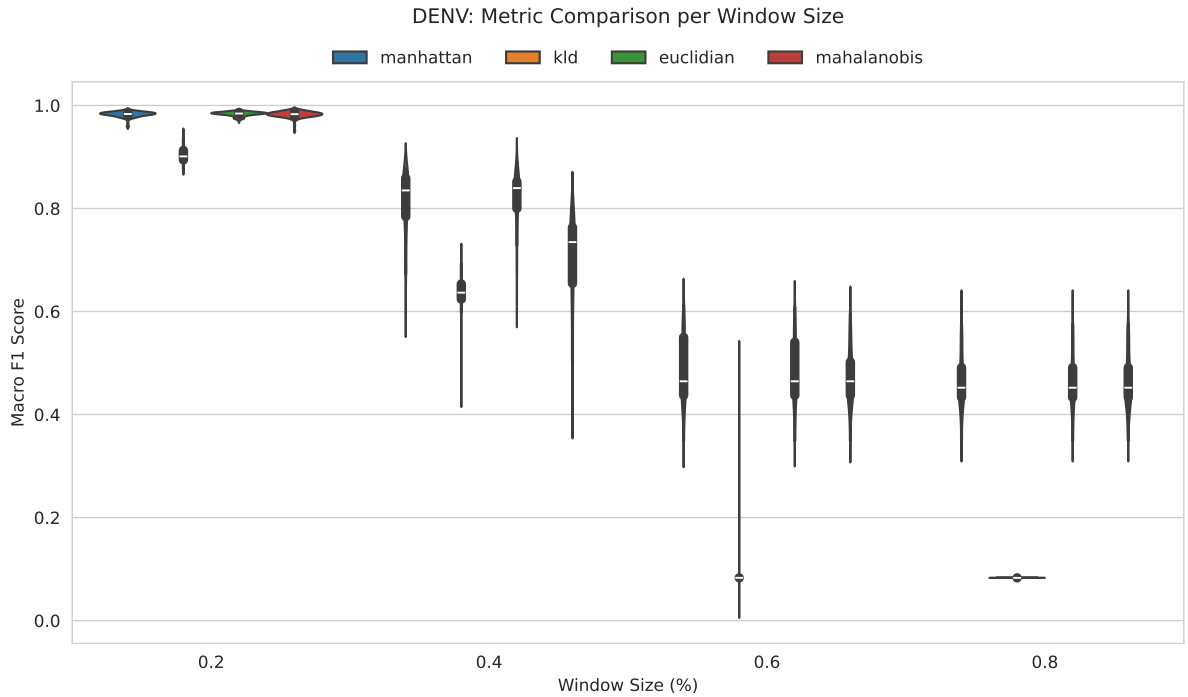
Classification methodologies based on distance metrics and divergence measures exhibit different performance across different viral datasets. Observing Figure 4.5, when applied to SARS-CoV data, the Kullback-Leibler (KL) divergence metric outperforms traditional distance-based metrics, while the Manhattan distance metric yields outstanding results for Dengue virus (DENV) data.

A critical factor is the disparity in sequence length: Dengue virus (DENV) is substantially shorter than SARS-CoV. This difference impacts feature extraction, as shorter sequences produce lower-resolution statistical frequency distributions. Reduced resolution affects the discriminative power of metrics sensitive to distributional details, such as KL divergence [41], while favoring simpler distance measures such as Manhattan distance.

The reference distribution (a statistical representation of sequence features) requires sufficient data points to capture meaningful distinctions. Consequently, KL divergence benefits from increased resolution (e.g., longer sequences or optimized window sizes),



(a) Performance comparison between metrics on SARS-CoV-2 data



(b) Performance comparison between metrics on DENV data

Figure 4.5: Metric comparison to evaluate different distance/divergence metrics across different window sizes.

whereas Manhattan distance proves more robust to sparse or low-resolution feature spaces.

More detailed analytical results are available in Appendix B.2, where Table B.5 presents the classification result metrics for DENV, Table B.6 presents the classification result metrics for SARS-CoV-2 and Table B.7 shows the relationship between the percentage window size and the resulting region resolution compared to the number of k-mers used.

Based on these findings, it was concluded that the Manhattan distance provides greater flexibility in discriminating signal variations, and smaller window sizes enabled a more intrinsic analysis of genomes, allowing for higher resolution of regional points and increasing the classification power of the methodology.

The definition of Manhattan distance as the default metric does not exclude the possibility of using different ones; alternative distance measures remain available within the implementation, allowing users to select, evaluate, and optimize the most appropriate metric for their specific applications.

4.6 Parameters Selection

Given the data imbalance of the datasets, it was necessary to apply a balancing approach to ensure a fair evaluation. To achieve this, the open-source tool ***FastaSplitter*** was used to perform a split between training and test data in a ratio of 70%-30% based on the variant with the fewest sequences, equalizing the training sample for all classes (see Table 4.1), and testing with complements. These sequences are randomly selected to avoid bias.

In this way, it is possible to evaluate the method's generalization capacity. Given the low number of samples for the classes with the fewest sequences, the training step is performed using a low-volume dataset (as shown in Table 4.1). With a balanced selection of training data, the GRAMEP *get-only-kmers* function is employed to perform variant-exclusive k-mer extraction and return the k-mers set from the training samples for use in region extraction and in the mounting of the reference distributions.

GREAC's data-driven methodology requires prior behavioral analysis to select optimal parameters for each dataset. Given biological differences between organisms, such as

sequence lengths and variation rate, adjustments are necessary in k-mer size, window percentage, and appearance threshold to achieve optimal performance.

Defining the value of K in alignment-free analysis remains challenging due to divergent sequence lengths. Depending on the use of the information and the final purpose, the range of values can change the analysis context, as evidenced by research using similar k-mer values (from $K = 10$ to $K = 20$) when searching for unique information sets [46].

Higher K values generate larger information sets, which may contradict the research proposal of identifying specific genomic regions, as extensive k-mer sets increase the complexity of the analysis. Therefore, the determination of K values was performed by seeking the smallest K value that could generate exclusive information sets and results when executing the GRAMEP method. Similar alignment-free analysis methods have employed reduced-size k-mers in variant analysis [47], [48].

For the remaining parameters, an exhaustive brute-force approach was employed, systematically testing window size percentages between ranges depending on the sequence length. This allowed to identify window sizes that maintain proportional relationships with k-mer sizes, in which longer genome sequences use smaller percentages, while shorter sequences employ larger percentages to achieve optimal feature extraction performance.

These parameter ranges were determined based on preliminary analyses, in which a window size range of 0.1% to 0.3% was employed for organisms with shorter genomes, such as HBV, HIV, DENV, and SARS-CoV-2, with increments of 0.05%. For Monkeypox, which comprises significantly longer sequences, a more refined range of 0.01% to 0.05% was used. Simultaneously, the thresholds for the appearance of k-mer in windows among the sequences were evaluated in a range of 50% to 95% with incremental steps of 5%.

This process resembles Recursive Feature Elimination (RFE) but focuses on identifying optimal methodological parameters rather than feature selection for classification models. The selected parameters for each organism are presented in Table 4.1.

Table 4.1: Parameters

Dataset	K	**Window Size (%)**	**Threshold (%)**	**Training Samples per Class**
SARS-CoV-2	9	0.2	50	89
Monkeypox	9	0.05	60	574
HBV	7	0.2	55	131
DENV	6	0.2	75	324
HIV	7	0.15	70	63

Chapter 5

Validation and Experiments

The viral datasets used were introduced in Section 3.1, and their use in developing the methodology was also highlighted in Chapter 3. Here, in Section 5.1, are presented the results obtained by applying the GREAC method to viral datasets, validating the method by comparing results across different organisms and against other methods.

Subsequently, in Section 5.2, an application to the honeybee *Apis mellifera* to detect genetic diversity is presented, validating the discriminative performance for differentiation between distinct lineages and subspecies. This also demonstrates the method's versatility when applied to organisms with larger genomes, such as honeybees.

5.1 Viral Applications

To validate the proposed method, its classificatory power was tested through two complementary approaches. The first employs Gaussian membership function values to assign probabilistic belonging scores, providing a soft classification framework that captures uncertainty in boundary cases. The second approach implements a tree-based classifier that enhances discriminatory and classificatory power while preserving the explainability that the method inherently provides. This dual approach ensures robust performance across different classification scenarios while maintaining interpretability of the results.

Firstly, it will be analyzed the individual results for each organism, highlighting the

principal characteristics and distinctive features of each taxonomic group. Subsequently, it will be demonstrate the classification process employed using XGBoost, detailing how the training vectors were constructed and which specific features were utilized to achieve optimal classificatory and discriminatory performance. This comprehensive analysis will include feature importance rankings, cross-validation metrics, and interpretability measures to validate both the accuracy and biological relevance of the classification framework.

The methodology involves both unsupervised clustering through Gaussian membership functions, and supervised learning via gradient boosting trees, creating a hybrid approach that leverages the strengths of both paradigms while maintaining computational efficiency and biological interpretability.

5.1.1 HBV Classification

The classification results of the HBV (hepatitis B virus) dataset achieved consistent and high accuracies for all variants. Benefiting from its compact genome structure and low variability sequence length (representing $\approx 0.1\%$ of the genome), it shows accurate and consistent regional identification and discrimination when applied to new input sequences.

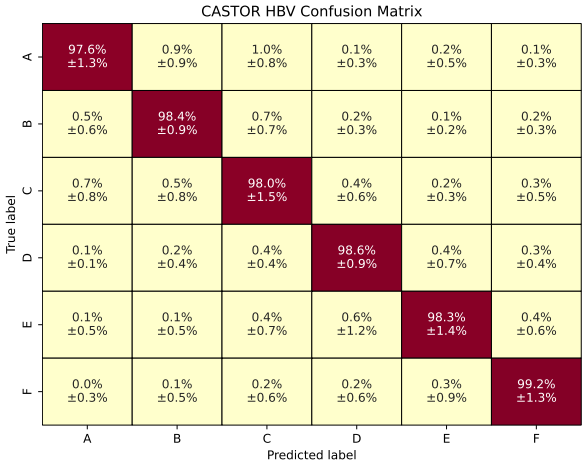
When comparing the confusion matrices (see Figure 5.1) with those from the other methods, GREAC outperform them: KEVOLVE shows low accuracies with high percentage values of standard deviation, while CASTOR-KRFE achieved similar high accuracies.

These results, and the ones for the other viruses, were obtained after 100 iterations.

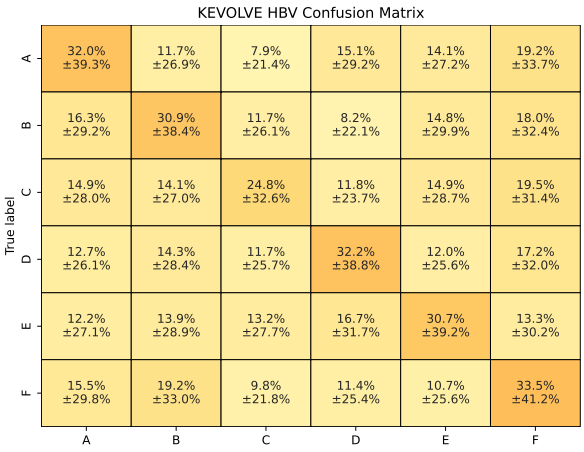
5.1.2 DENV Classification

The DENV (dengue) virus has a larger genome size, and the variation between the lengths is still low compared to the total length (representing $\approx 1.7\%$ of the genome), resulting in a consistent classification that achieves high values without high misclassification rates.

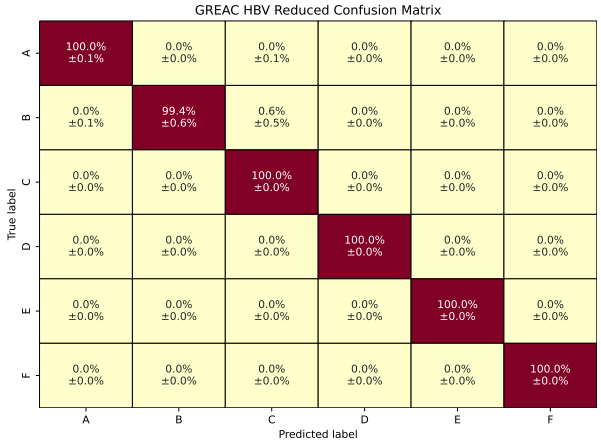
Analyzing and comparing the confusion matrices in Figure 5.2, it is possible to conclude that the high accuracy values align with the ones achieved by CASTOR-KRFE, while outperforming the ones from KEVOLVE.



(a) CASTOR-KRFE



(b) KEVOLVE



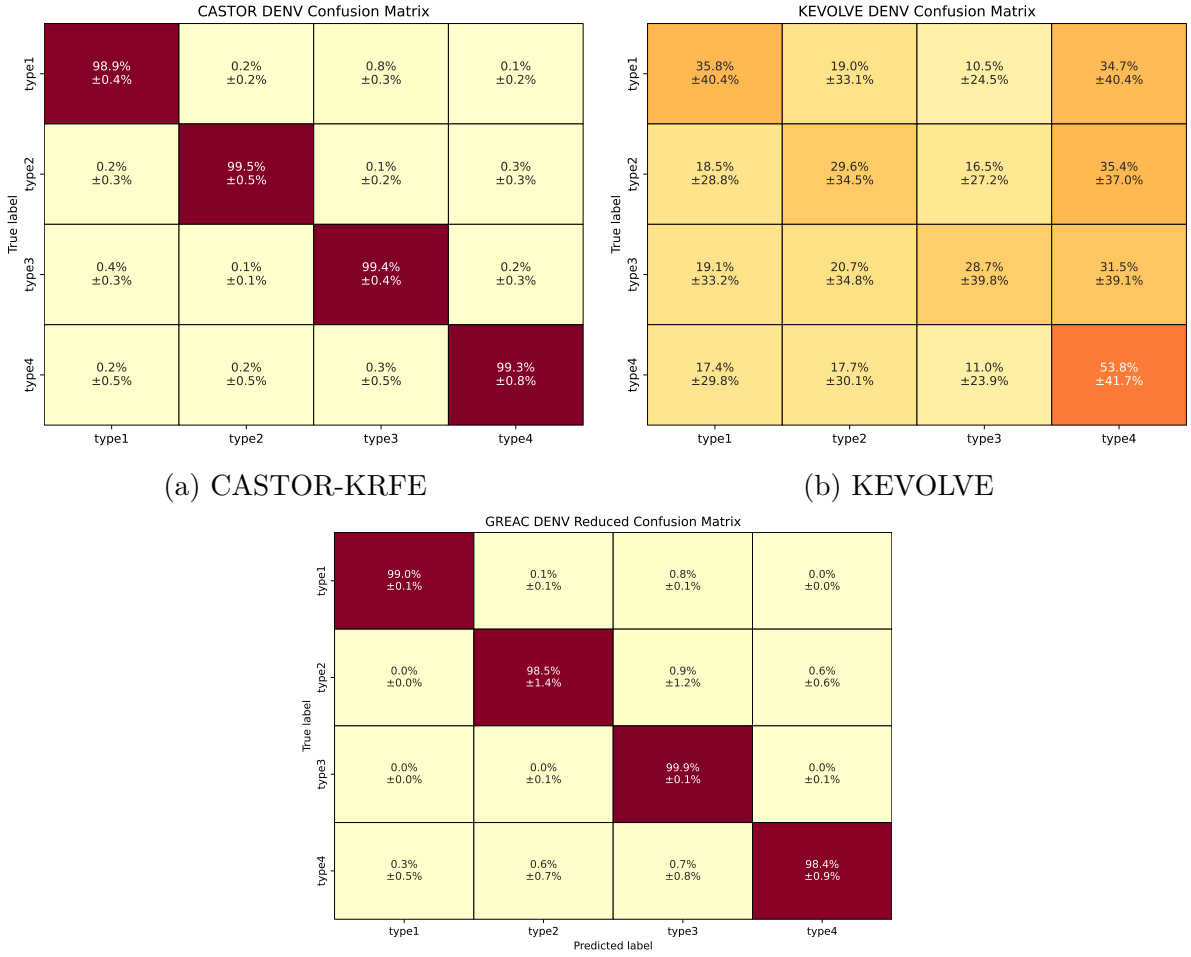
(c) GREAC

Figure 5.1: HBV confusion matrix classification: average accuracy and standard deviation.

5.1.3 SARS-CoV-2 Classification

The classification results achieved for the SARS-CoV-2 virus dataset demonstrates GREAC's versatility when applied to viruses, given the different in size compared to the others considered (being the most unbalanced dataset among them), but sharing with them the low variation in the lengths of the sequences (representing $\approx 0.2\%$ of the genome).

The classification results, shown in Figure 5.3, exhibit high consistency and generalization in all variants, given the disparity between training volumes versus evaluation



(a) CASTOR-KRFE (b) KEVOLVE

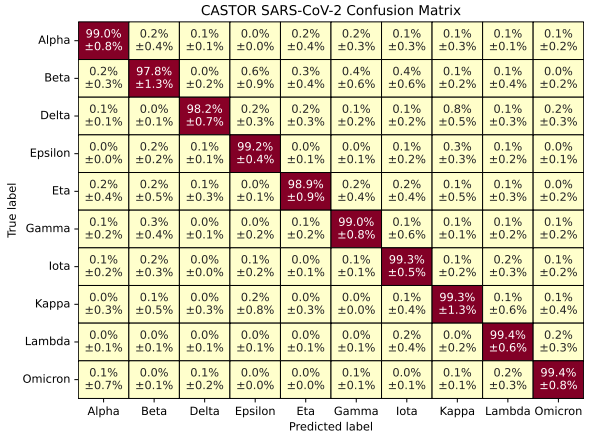
(c) GREAC

Figure 5.2: DENV confusion matrix classification: avg. accuracy and standard deviation.

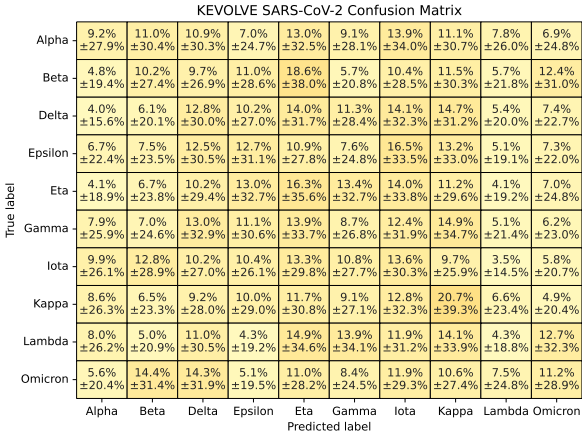
ones. Compared to CASTOR-KRFE, GREAC maintains consistency with similar accuracy values and outperforms KEVOLVE, which exhibits high misclassification rates.

5.1.4 Monkeypox Classification

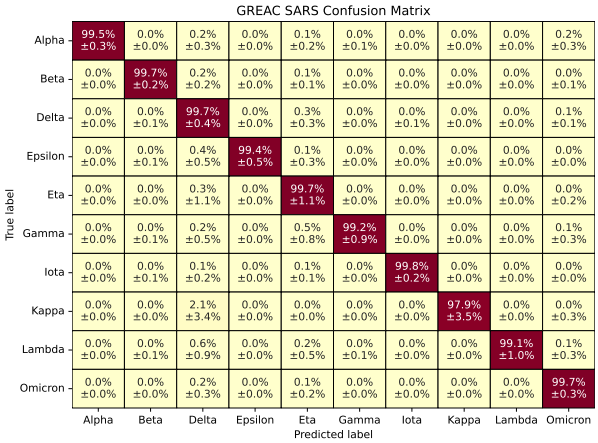
The Monkeypox dataset comprises the longest sequences among all tested datasets, highlighting the diversity in sequence lengths and demonstrating the versatile application of the methodology, which showcases its scalability in feature extraction across sequences of varying sizes. This characteristic offers valuable insights into the method's adaptability to diverse genomic contexts and sequence complexities.



(a) CASTOR-KRFE



(b) KEVOLVE

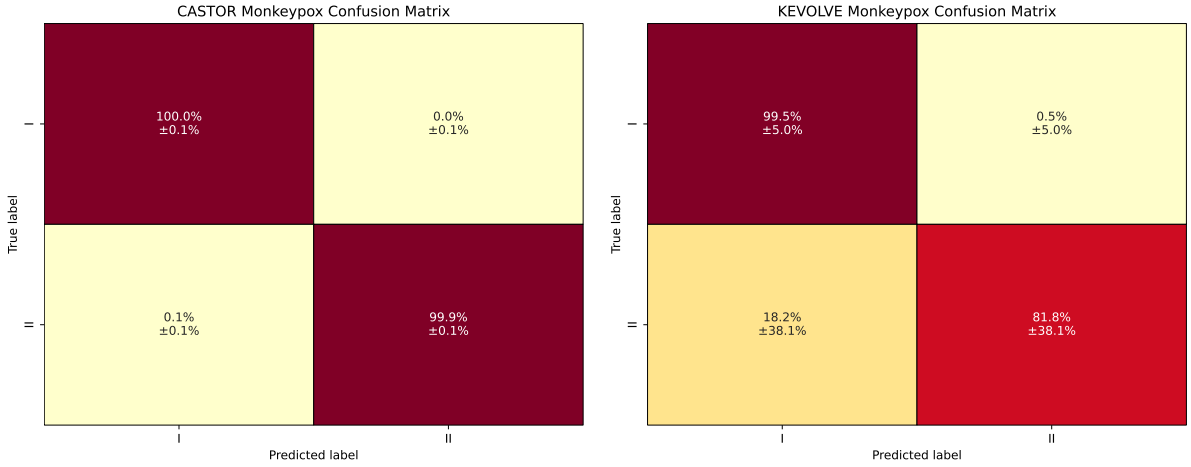


(c) GREAC

Figure 5.3: SARS-CoV-2 confusion matrix classif.: avg. accuracy and std. deviation.

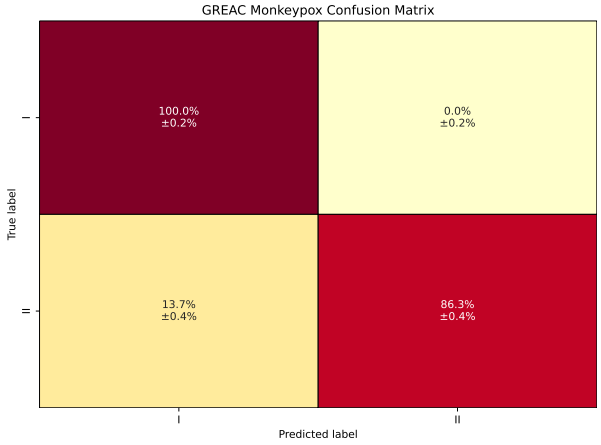
As the only virus whose dataset contains exclusively two classes, Monkeypox analysis results in a more straightforward binary classification assessment.

Upon examination of the confusion matrices presented in Figure 5.4, CASTOR-KRFE demonstrates superior performance compared to all other methods. The results from GREAC show similarity to KEVOLVE in terms of classification accuracy; however, they exhibit significantly higher consistency, as evidenced by a substantially lower standard deviation in the mean classification performance. This enhanced stability indicates an improved reliability and robustness of the GREAC methodology, particularly important for clinical and diagnostic applications where consistent performance is crucial.



(a) CASTOR-KRFE

(b) KEVOLVE



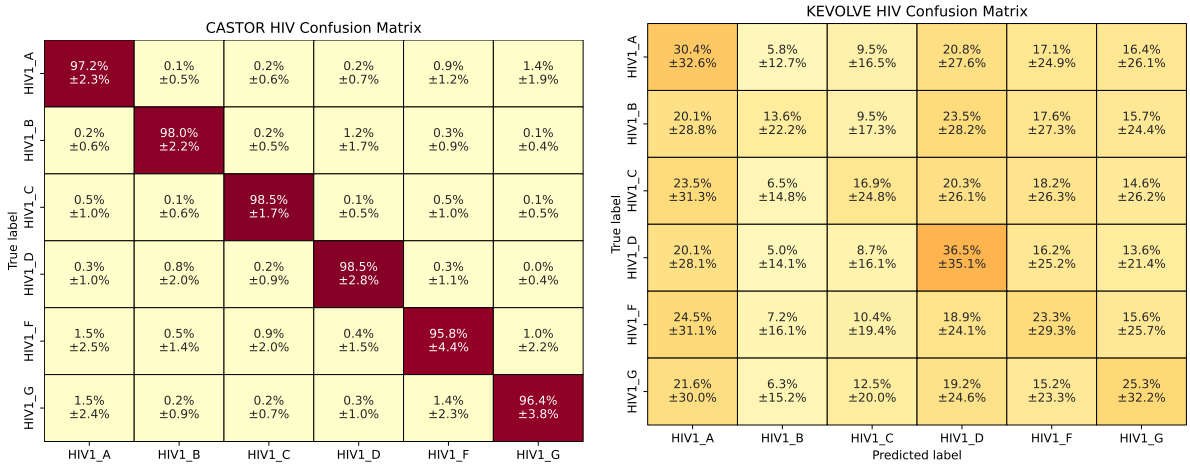
(c) GREAC

Figure 5.4: Monkeypox confusion matrix classif.: avg. accuracy and std. deviation.

5.1.5 HIV Classification

HIV classification presented the most challenging scenario due to the high variability in the length of the sequences (representing $\approx 4.3\%$ of the genome), which significantly affects the position-based methodology. The substantial standard deviation in sequence lengths disrupts the positional consistency required for accurate variant identification, resulting in reduced classification performance compared to other viral organisms.

Compared to the other methods, CASTOR-KRFE outperformed GREAC and KEVOLVE, consistently performing well in all classifications, as shown in Figure 5.5.



(a) CASTOR-KRFE

(b) KEVOLVE

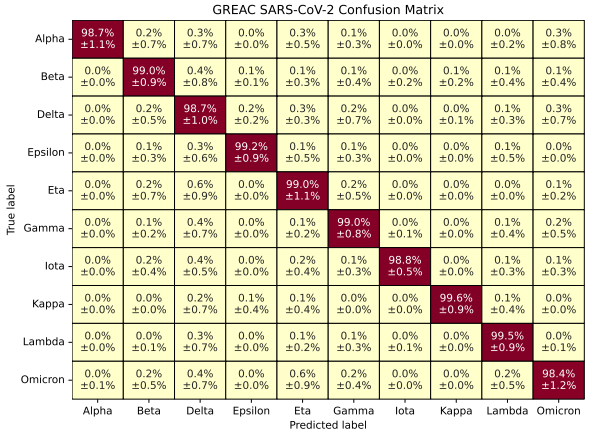
(c) GREAC

Figure 5.5: HIV confusion matrix classification: average accuracy and standard deviation.

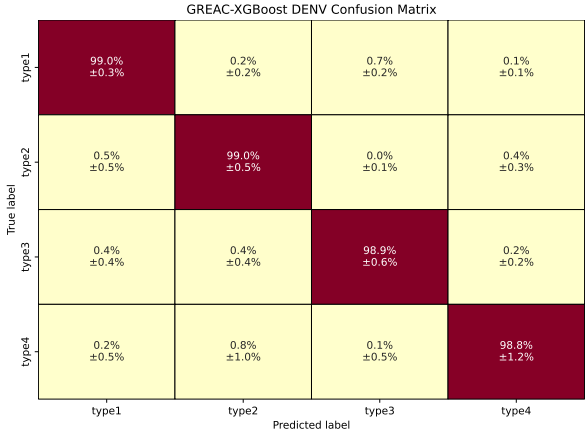
5.1.6 Classification Using XGBoost

To achieve improved performance in multi-class classification, features derived from genomic regions extracted by the GREAC method were utilized to construct input vectors for the XGBoost classifier. This approach complements the program with enhanced classification performance while maintaining explainability and transparency characteristics, as XGBoost is fundamentally based on decision tree ensembles.

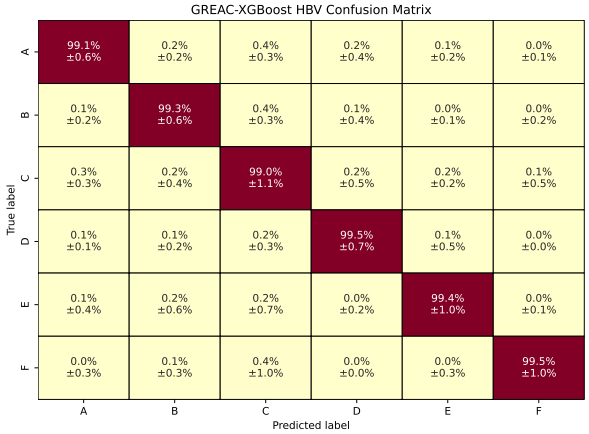
The experiment was conducted using the same parameters as the previous analysis; however, classification was performed by selecting the class with the highest probability



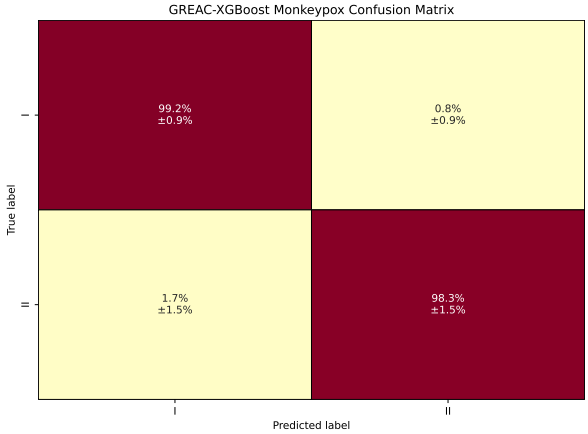
(a) SARS-CoV-2



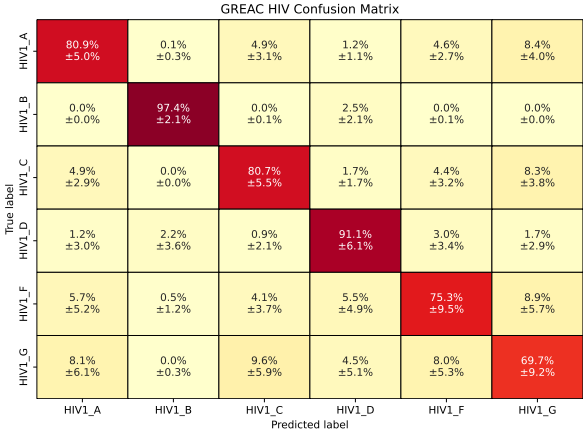
(b) DENV



(c) HBV



(d) Monkeypox



(e) HIV

Figure 5.6: Viruses XGBoost confusion matrix classif. average and standard deviation.

returned by the classifier. As shown in Figure 5.6, the objective of increasing classification performance using the features extracted from genomic regions identified by the GREAC method was achieved, resulting in high precision classifications.

5.1.7 Dimensionality Reduction

GREAC's region extraction methodology achieves substantial dimensionality reduction by focusing analysis on discriminative genomic regions rather than entire sequences. This reduction enhances the scalability of classification tasks by reducing computational resource requirements, lowering memory consumption, and increasing analysis speed. This improvement is shown by comparing the FASTA file sizes in Table 5.1.

Observing Table 5.2, we can conclude that there is a considerable reduction when applied to larger genomes, as in the case of SARS-CoV-2, where approximately a 2/3 reduction was achieved, and 72% for Monkeypox. Compared with other genomes that are already considered small in length, it was still possible to achieve considerable reductions: 400 nucleotides for HBV, representing approximately 12% of its length, and 2,300 nucleotides for DENV, representing approximately 20% of the total.

Table 5.2: Dimensionality reduction in base-parts (bp) results across datasets

SARS-CoV-2	**DENV**	**Monkeypox**	**HBV**	**HIV**
Median(σ)	**Median(σ)**	**Median(σ)**	**Median(σ)**	**Median(σ)**
10046.15 (472.57)	6564.89 (120.52)	54452.32 (1276.83)	2899.38 (3.71)	8763.38 (37.86)

To provide additional validation and demonstrate that the dimensionality reduction yields regions with discriminatory power for classification, we applied this reduction directly to the data by using the extracted regions to construct a dataset where sequences are reduced to comprise only these regions. The reduction was applied to the DENV, SARS-CoV-2, and Monkeypox datasets, which showed the most significant dimensionality reduction given their medium sequence length.

Table 5.1: FASTA files size reduction comparison

Variant File	**Original Size**	**Reduced Size**
Alpha	5.3 GB	1.7 GB
Beta	21.0 MB	6.9 MB
Delta	284.7 MB	93.2 MB
Epsilon	444.5 MB	146.3 MB
Eta	21.6 MB	7.1 MB
Gamma	246.0 MB	80.6 MB
Iota	583.2 MB	191.3 MB
Kappa	3.8 MB	1.3 MB
Lambda	12.9 MB	4.2 MB
Omicron	3.2 GB	1.1 GB
Total SARS-CoV-2	**10.1 GB**	**3.3 GB (32.6%)**
Type1	27.7 MB	17.1 MB
Type2	19.0 MB	11.7 MB
Type3	13.7 MB	8.5 MB
Type4	4.9 MB	3.1 MB
Total DENV	**65.3 MB**	**40.4 MB (61.8%)**
I	163.3 MB	41.7 MB
II	546.1 MB	149.8 MB
Total Monkeypox	**709.4 MB**	**191.5 MB (26.9%)**
A	2.7 MB	2.4 MB
B	5.9 MB	4.7 MB
C	8.9 MB	7.1 MB
D	3.7 MB	2.9 MB
E	0.9 MB	0.7 MB
F	0.6 MB	0.4 MB
Total HBV	**22.7 MB**	**18.2 MB (80.1%)**
HIV1_A	1.2 MB	0.9 MB
HIV1_B	1.2 MB	0.9 MB
HIV1_C	1.2 MB	0.9 MB
HIV1_D	0.8 MB	0.6 MB
HIV1_F	0.8 MB	0.6 MB
HIV1_G	0.8 MB	0.6 MB
Total HIV	**6 MB**	**4.5 MB (75%)**

As shown in the confusion matrices in Figure 5.7, the classification results remained high and similar to those obtained with the original dataset when applied in CASTOR-KRFE, demonstrating the efficiency of the data reduction approach while maintaining

equivalent classification performance. However, the results for KEVOLVE represent insignificant improvements, reinforcing that it does not exhibit precision loss, despite the presence of Monkeypox, where original inconsistent results were corrected, resulting in improved consistency accuracy.

5.1.8 Computational Performance

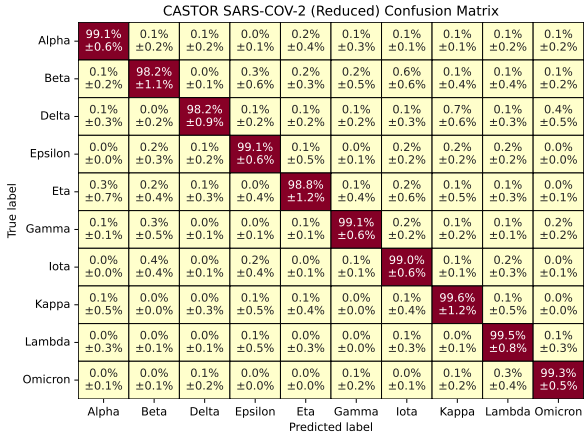
The performance of GREAC (implemented in Julia) was evaluated against KEVOLVE and CASTOR-KRFE (both implemented in Python). Additionally, the scalability of GREAC on a multicore system was also investigated.

Consumer-grade System

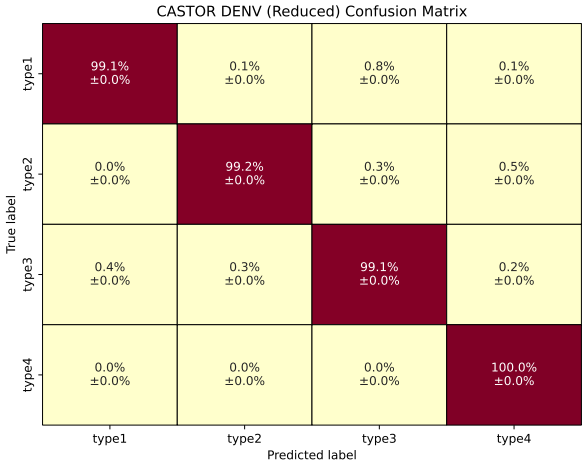
The first evaluation went on a consumer-grade laptop, with an Intel i7-1165G7 CPU (4 cores/8 threads, 4.70 GHz peak freq.) and 8 GB of RAM, running Linux Fedora 41.

To ensure a fair comparison across implementations developed in different programming languages, it was measured the real (wall-clock) execution time using the `time` command. Each benchmark was executed five times under identical conditions, and the average and standard deviation were computed. All benchmarks used the same amount of training data: the first dataset contained 89 sequence samples per SARS-CoV-2 variant, while the second consisted of 574 samples per Monkeypox variant.

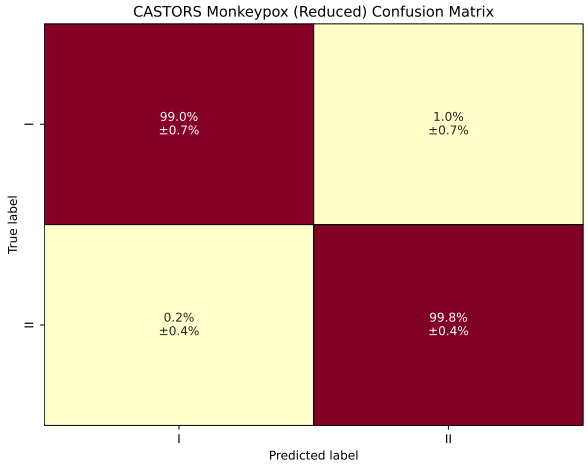
The performance comparison focused exclusively on two critical stages of the modeling process: **Feature Extraction** and **Model Fitting**. The classification stage was excluded to eliminate the influence of dataset-specific variability, allowing for a consistent assessment of computational efficiency. In this context, Model Fitting refers to the construction of reference behavior signals based on k-mer frequency distributions and the measurement of variant-specific divergences relative to those references. By restricting the comparison to these two stages, it was established a consistent analytical baseline that enables a fair evaluation of computational efficiency across all methods.



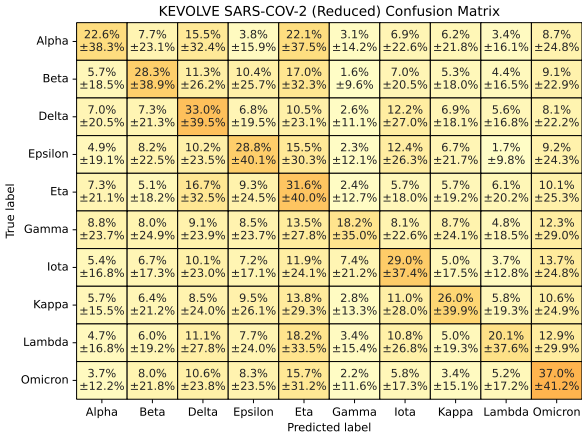
(a) CASTOR-KRFE on the SARS-CoV-2 reduced dataset.



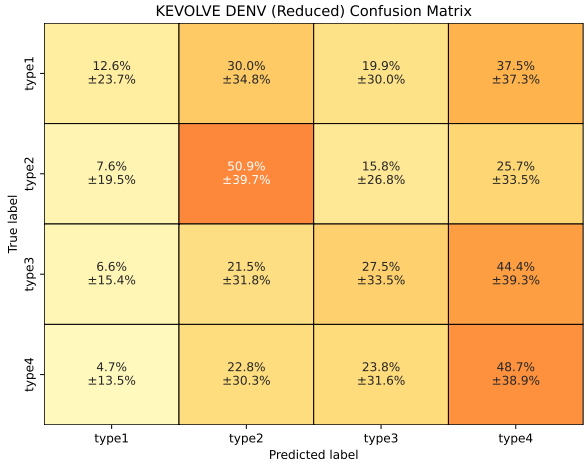
(b) CASTOR-KRFE on the DENV reduced dataset.



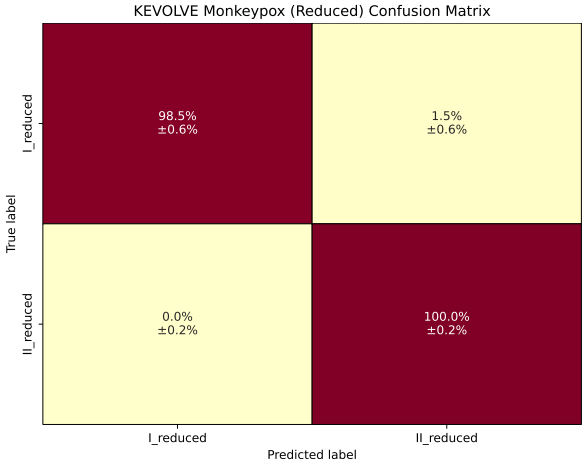
(c) CASTOR-KRFE results on Monkeypox reduced dataset.



(d) KEVOLVE on the SARS-CoV-2 reduced dataset.



(e) KEVOLVE on the DENV reduced dataset.



(f) KEVOLVE results on Monkeypox reduced dataset.

Figure 5.7: Confusion matrix classification average accuracy and standard deviation over 100 iterations using the reduced dataset produced only with the regions extracted by GREAC.

Table 5.3 has the results for the SARS-CoV-2 benchmark. GREAC clearly outperformed the other two approaches, achieving a performance gain of approximately one order of magnitude, with speedups of $\gtrapprox 9$ compared to both KEVOLVE and CASTOR-KRFE.

Table 5.3: Sars-Cov-2 Feature Extraction + Model Fitting Benchmark Results.

Methodology	**Execution Time** (s) *Average (Standard Deviation)*	**Speedup**
GREAC	19.3 (1.1)	—
KEVOLVE	176.9 (39.3)	9.2
CASTOR-KRFE	182.1 (1.3)	9.5

In contrast, when evaluating the Monkeypox dataset (Table 5.4), GREAC did not achieve a great advantage. Improvements were mainly observed for smaller window sizes, where the region selection process achieved greater reduction in dimensionality. However, for larger window sizes, GREAC exhibited slower performance, partially due to the substantial volume of training data processed.

Table 5.4: Monkeypox Feature Extraction + Model Fitting Benchmark Results.

Methodology	**Execution Time** (s) *Average (Standard Deviation)*	**Speedup**
GREAC (0.01%)	135.1 (5.2)	—
GREAC (0.05%)	725.1 (12.1)	5.36
KEVOLVE	90.2 (30.6)	—
CASTOR-KRFE	400.5 (2.6)	2.9

Server-grade System

Given the computational limitations observed for the Monkeypox dataset on the domestic machine, the performance and scalability of GREAC on a high-performance multi-core system was further investigated. This specific experiment was conducted on a KVM-based virtual machine, assigned with 64 vCores of an AMD EPYC 9554P processor (3.1 GHz peak frequency) and 128 GB of RAM, running Linux Ubuntu 24.04.3 LTS x86_64.

Table 5.5: Monkeypox Multi-core Evaluation.

Methodology	**Execution Time** (s) *Average (Standard Deviation)*	**Speedup**	**Efficiency**
1 core	2,422.4 (36.6)	—	—
6 cores	572.3 (6.6)	4.23	70.5%
8 cores	331.5 (4.9)	7.31	91.3%
16 cores	181 (2.7)	13.38	83.6%
32 cores	105 (0.7)	23.1	72.1%
64 cores	75.2 (0.2)	32.21	50.3%

As shown in Table 5.5, the results demonstrate that the current GREAC implementation scales efficiently (Efficieny = Speedup / Number of Cores), in this particular virtual environment, up to 32 virtual cores, with diminishing gains from there on, most probably due to the fact that virtual cores used are SMT-based (a technology similar to Intel's hypertheading, used in AMD processors).

A more detailed analysis on the inhibitors to further scalability (including those internal to the application and/or to the Julia runtime) is left for further work. However, the tests show that GREAC is already able to make good use of HPC-grade systems.

5.2 *Apis mellifera* Application

This section presents the results obtained from the application of GREAC for the analysis of genetic diversity in the honey bee species, *Apis mellifera*, beginning with an analysis and discussion of the method's capacity to identify discriminative regions and distinguish and classify different *A. mellifera* subspecies and lineages. Furthermore, the achieved dimensionality reduction is examined, which is particularly significant considering that the honeybee genome operates at a substantially larger scale than the viral datasets, and is composed of multiple chromosomes, making it suitable for applying GREAC.

Table 5.6: *Apis mellifera* lineages and sample amount

Lineages	**Samples**
M	94
C	39

Table 5.7: *Apis mellifera* subspecies and sample amount

Subspecie	**Samples**
A. m. anatoliaca	10
A. m. carnica	35
A. m. caucasia	8
A. m. iberiensis	68
A. m. jemenitica	16
A. m. meda	30
A. m. ruttneri	18
A. m. siciliana	57
A. m. cypria	11
A. m. intermissa	10
A. m. ligustica	70
A. m. mellifera	17
A. m. sahariensis	26
A. m. syriaca	21

5.2.1 *Apis Meliffera* Datasets

For the application of GREAC to the genetic diversity analysis of *Apis mellifera*, two datasets of this species were used.

The first dataset (see Table 5.6) has individuals from two distinct evolutionary lineages: lineage C, native to Southeastern Europe, and lineage M, native to Western and Northern Europe. This dataset was used to assess the ability of GREAC o distinguish between two different population groups with distinct geographical and evolutionary origins.

The second dataset is made of individuals already classified into subspecies of *Apis mellifera*, comprising a comprehensive collection of 14 subspecies, as laid out in Table 5.7. This dataset enabled a more granular analysis of the methodology's ability to discriminate between closely related taxonomic groups within the same species, providing a robust framework for evaluating classification accuracy at the subspecies level.

Table 5.8: Chromosomes dispositions and reduction achieved

Chromosome	**Size (bp)**	**Final Size Reduction (bp) Median (σ)**	**Windows Median (σ)**	**K-mer set Median (σ)**
LG1	29,893,408	129,121.90 (2,483.34)	2,255.90 (43.20)	1,100.60 (32.68)
LG2	15,549,267	34,315.58 (898.03)	1,205.46 (31.00)	672.00 (19.47)
LG3	13,234,341	48,877.88 (1,425.70)	2,071.33 (60.98)	1,287.83 (43.61)
LG4	12,718,334	40,309.10 (991.08)	1,875.80 (46.65)	1,151.10 (31.60)
LG5	14,363,272	50,693.30 (1,809.10)	1,988.90 (72.74)	1,207.10 (50.91)
LG6	18,472,937	51,741.70 (902.22)	1,518.80 (25.86)	847.80 (14.89)
LG7	13,219,345	56,101.50 (1,627.09)	2,406.80 (70.73)	1,486.00 (51.60)
LG8	13,546,544	42,829.50 (1,482.43)	1,826.30 (62.05)	1,121.80 (45.57)
LG9	11,120,453	36,273.80 (361.86)	1,885.60 (18.90)	1,115.50 (15.90)
LG10	12,965,953	34,803.90 (1,482.15)	1,627.40 (68.57)	1,015.50 (47.32)
LG11	14,726,556	45,251.20 (731.52)	1,771.40 (30.57)	1,047.20 (19.99)
LG12	11,902,654	34,107.30 (776.88)	1,758.00 (40.79)	1,076.90 (24.69)
LG13	10,288,499	34,043.10 (931.48)	1,947.00 (51.69)	1,256.00 (33.25)
LG14	10,253,655	37,266.80 (1,056.51)	2,148.50 (63.32)	1,371.30 (40.17)
LG15	10,167,229	24,420.00 (659.15)	1,421.80 (37.16)	888.80 (28.61)
LG16	7,207,165	190,436.06 (4860.59)	1,299.94 (33.88)	930.88 (26.99)

5.2.2 Analysis of Evolutionary Lineages

For the classification of the evolutionary lineages C and M of *Apis mellifera*, it was selected a k-mer size of $K = 9$, representing the minimum value to identify exclusive k-mers. Furthermore, it was applied an arbitrary window size of 0.0001% for chromosomes 1-15 and 0.001% for chromosome 16, because it was the shorter one, which would otherwise produce windows smaller than the k-mer size (7 bp versus 9 bp).

The reference genome used for the extraction of k-mers was obtained from NCBI, with the identification GCA_000002195.1_Amel_4.5. The honey bee reference genome is divided into 16 chromosomes, which are detailed in Table 5.8.

The table specifies the total length of each chromosome (or Linkage Group, LG) and the reduction achieved by applying GREAC's methodology, representing the quantity of base pairs (bp) used for the classification of each chromosome, along with the number of regions extracted (windows) and the total number of k-mers in the set.

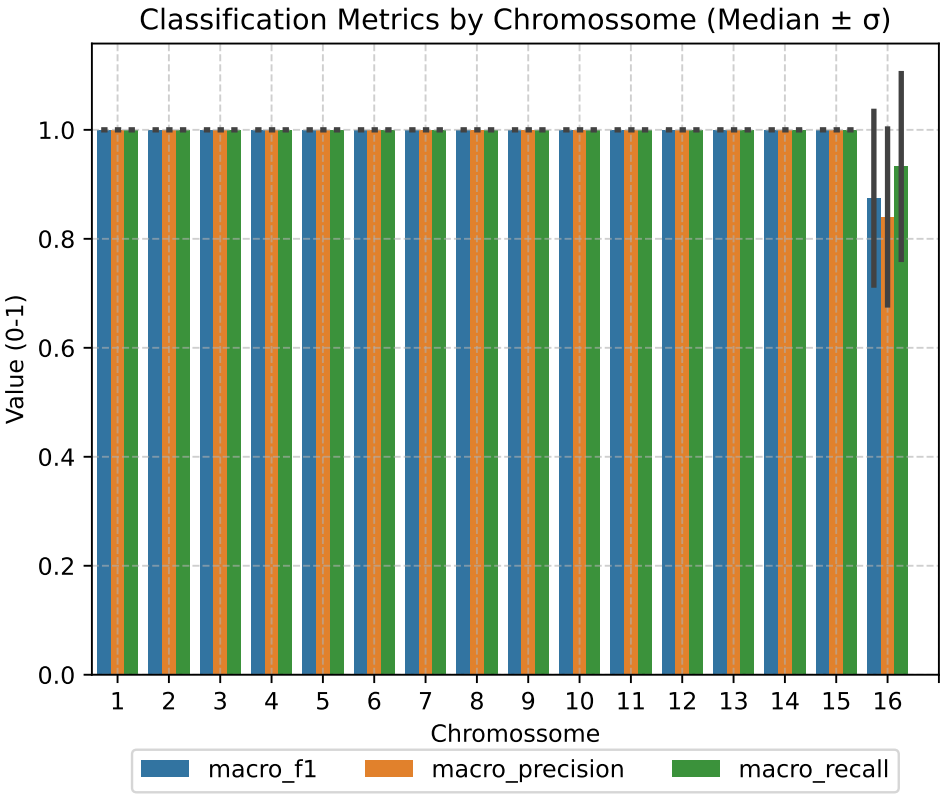


Figure 5.8: Classification metrics per chromosome when applied for classification between lineage C and M using Gaussian membership.

Lineage classification results using the fuzzy Gaussian membership function demonstrated excellent performance and stability across 10 iterations, as shown in Figure 5.8, despite chromosome 16, where the number of exclusive k-mers extracted for lineage C in most iterations never returned anything.

This outcome highlights the crucial importance of data selection, as the genomic sequences of the organisms from which k-mers are extracted directly impact the detection of lineage-specific exclusivity patterns. When exclusive k-mers for a particular lineage are not identified, the classification becomes unreliable, often resulting in uncertain classifications or complete classification failure. This occurs when zero-valued frequency signals measured during signal construction exclude these samples from the classification pipeline.

The reference region frequency signals generated from GREAC for classification and analysis between the two lineages are better detailed in Figure D.2 in the Appendix D.

5.2.3 Subspecies Analysis

Geographical isolation and adaptation to a wide range of contrasting environments have driven genetic divergence in *Apis mellifera* and the emergence of numerous subspecies, with approximately 31 currently recognized [11]. In this section, GREAC is applied to 14 subspecies for which genetically pure samples were available, enabling to evaluate the method's performance and analyze its output for these subspecies.

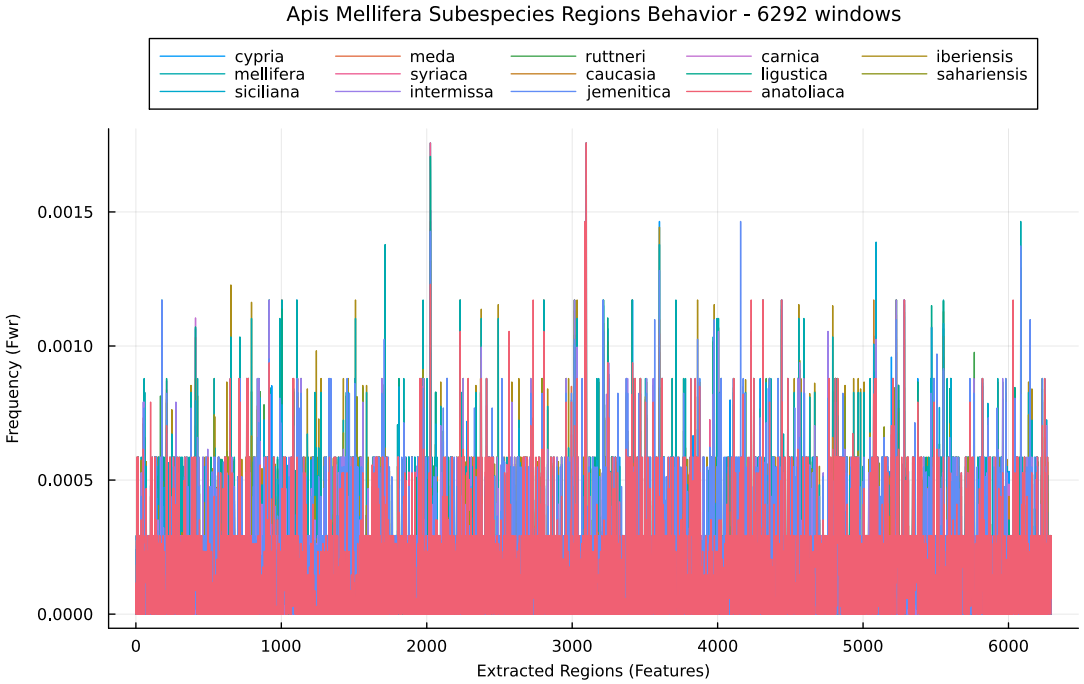
Identification of Discriminative Genomic Regions

Given the limited number of samples available per subspecies, this analysis focused on identifying discriminative genomic regions rather than on formal classification benchmarking. To stress-test the method under these data constraints, the analysis was concentrated on chromosome 1 (LG1), the longest chromosome in the *Apis mellifera* genome. Although the sample size precluded comprehensive classification validation, this application effectively demonstrates the method's primary strength: the identification and extraction of genomic regions that discriminate subspecies.

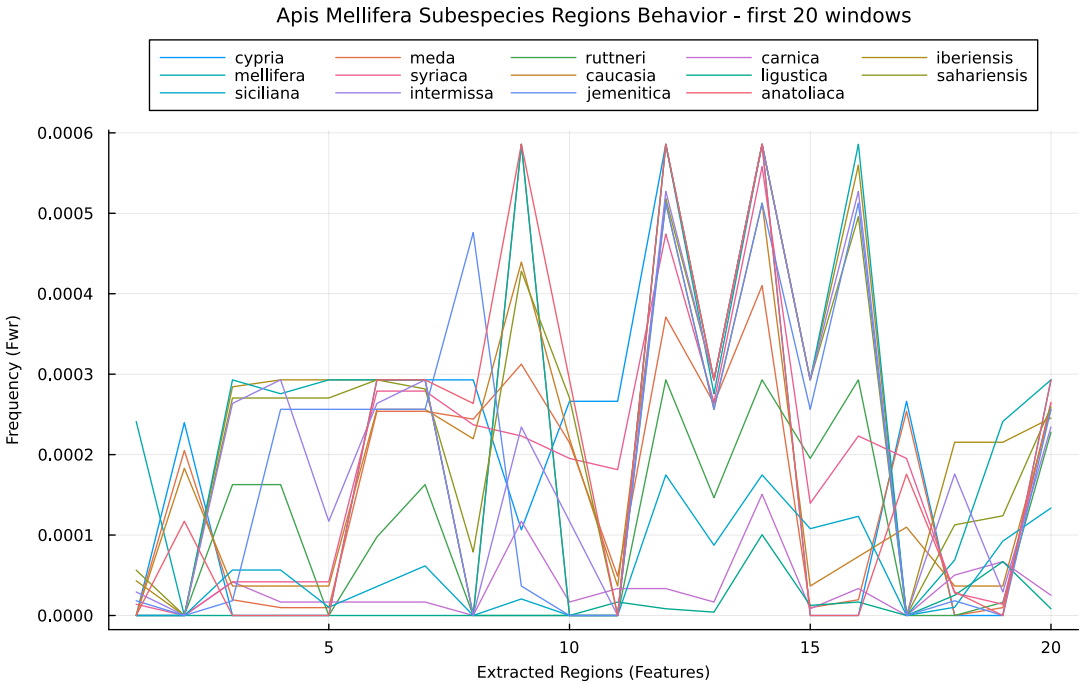
A key feature of GREAC is the ability to export the identified genomic regions in BED format (see Appendix D.1), facilitating subsequent laboratory analysis and validation by researchers. The GREAC-defined genomic positions can be visualized and cross-referenced with existing genomic annotations using tools such as NCBI Genome Data Viewer.

As shown in Figure 5.9, the extracted regions clearly reveal discriminative genomic signatures for each subspecies. This demonstrates GREAC's ability to identify biologically meaningful subspecies-specific patterns even from a limited number of samples.

A k-mer size of $K = 10$ was used for the subspecies analysis, an increase from $K = 9$ used in the lineage experiment. This adjustment was necessary due to the larger number of classes (i.e., subspecies). With $K = 9$, the resulting k-mer sets for several subspecies were extremely small, implying that they could be subsets of other sets. This generated highly similar frequency patterns, severely diminishing the discriminative power of the analysis. Increasing the k-mer size to K=10 expands the combinatorial space, enhancing



(a) Frequency Signal from chromosome 1 (LG1)



(b) Initial Frequency Signal from chromosome 1 (LG1)

Figure 5.9: Frequency distribution of extracted regions across *Apis mellifera* subspecies. Subfigure 5.9a shows the signal for all extracted regions across chromosome 1 (LG1), while Subfigure 5.9b presents a detailed view of the first 20 windows for closer analysis.

the probability of identifying unique k-mers for each subspecies. This, in turn, increases the available information and allows for more refined discrimination.

The significant dimensionality reduction achieved by GREAC is detailed in Table 5.9. The results demonstrate a substantial reduction in sequence length with different windowing parameters.

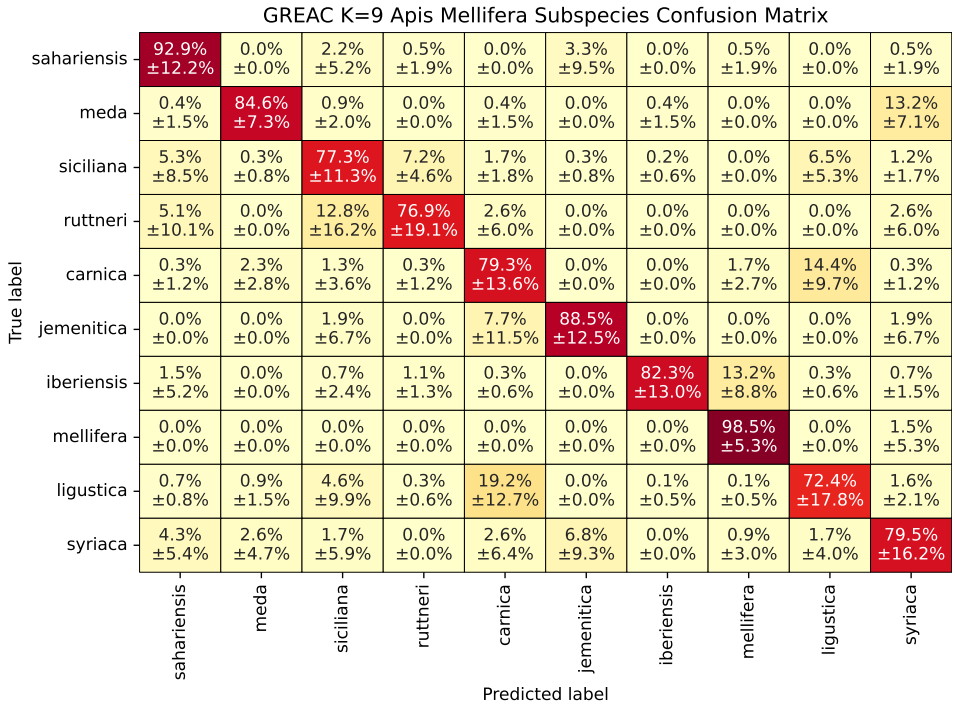
Table 5.9: Reduction Achieved

Window Size	**Size (bp)**	**Final Size Reduction (bp)**	**Windows**	**K-mer set**
0.0001 %	29,893,408	284,731	5,691	3,414
0.00005 %	29,893,408	137,036	6,403	3,414
0.00004 %	29,893,408	107,958	6,222	3,414

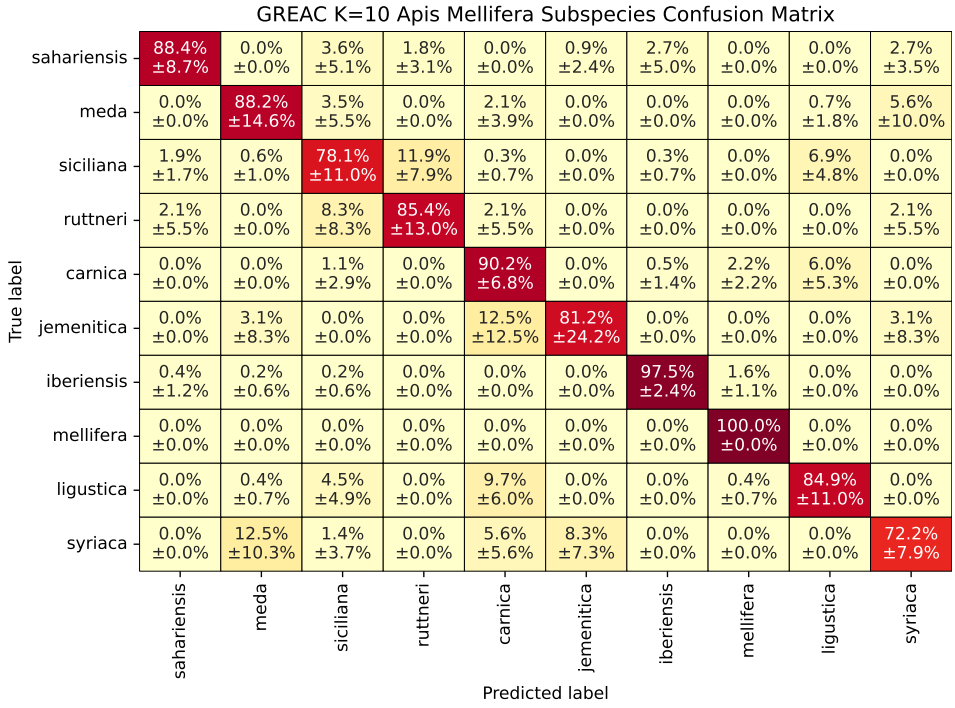
5.2.4 Subspecies Classification Experiment

To further evaluate GREAC's classification capabilities, a follow-up experiment was conducted on a curated subset of the data. Subspecies with insufficient samples for a meaningful training/validation split were excluded, resulting in a dataset of 10 subspecies: i) *A. m. carnica*, ii) *A. m. iberiensis*, iii) *A. m. jemenitica*, iv) *A. m. ligustica*, v) *A. m. meda*, vi) *A. m. mellifera*, vii) *A. m. ruttneri*, viii) *A. m. sahariensis*, ix) *A. m. siciliana* and x) *A. m. syriaca*. The smallest class contained 16 samples, and a balanced training set was created using 12 samples from each subspecies.

The analysis was performed with both $K = 9$ and $K = 10$ (see classification results in Figure 5.10), with window size percent values of 0.00005% for $K = 10$ given the higher amount size of the resulted k-mer set and 0.0001% for $K = 9$, corresponding to a physical window size of *14 bp* and *29 bp* respectively, which are very close to the k-mer sizes. Using $K = 9$ yielded a very limited information base, with a total k-mer set size of only 125 k-mers. In contrast, increasing the size to $K = 10$ produced a substantially larger and more informative set of 3,133 k-mers, providing a more robust foundation for classification.



(a) Using K=9



(b) Using K=10

Figure 5.10: Confusion matrix classification average accuracy and standard deviation over 10 iterations. Figure 5.10a shows the results using K=9 and Figure 5.10b K=10.

Chapter 6

Conclusion

Developing a novel methodology presents significant challenges, particularly when proposing relatively new algorithms rather than applying existing methodologies to previously untested datasets. This challenge becomes even more complex within the genomic domain, which is characterized by extreme divergences in information sources. Genomic datasets contain information about specific organisms that exhibit different behaviors, such as the fundamental differences between haploid and diploid organisms, where the quantity of DNA sets varies significantly due to differences in the chromosome complement.

Genetic datasets, such as DNA sequences, undergo constant mutation due to transcription processes and the transmission of genetic material to subsequent generations. These mutations can affect phenotypic and functional characteristics, and during the intergenerational transmission process, they can give rise to different variants or subspecies.

In cases of viral organisms compared to eukaryotic organisms, this mutation process tends to be more abrupt due to their high mutation rates, making it a challenging process for finding and extracting genetic patterns. Consequently, the application of viral datasets in the development represents rigorous validation compared to organisms such as honeybees, where the genome conservation rate is substantially higher.

The identification of regions capable of representing genomic divergence characteristics that can classify variants or subspecies represents a conceptually important advance for studies in this field. In this work, a novel approach that surpasses conventional methods

was proposed: an alignment-free methodology utilizing informative k-mer sets to identify these regions, thereby achieving dimensionality reduction for data applications.

The results achieved along with the analysis made in this report indicate that k-mers are suitable information sources for the development of viral genome analysis and classification methodologies, with potential for data reduction, and exhibit robust performance when applied in mathematical and machine-learning methods.

It is important to note, though, that the degree of dimensionality reduction achieved by the proposed method is not constant: it varies depending on the type of genome analyzed and the overall genetic length of the organism under study.

Another critical factor to consider is the optimization of the parameters, such as the selection of the window size. Smaller window sizes generally yield higher levels of dimensionality reduction; however, in some cases, slightly larger windows may provide deeper biological insights and more representative genomic patterns.

This type of analysis depends not only on the exhaustive computational process of parameter optimization but also on prior biological knowledge of the studied data. Integrating biological context into the optimization process can significantly improve its precision and guide the search toward more meaningful and interpretable results.

Observing HBV, which has a circular genome, the identified regions are distributed throughout the genome, with a high incidence in the genic regions [49]. Similarly, in the Dengue virus genome, a single-stranded RNA genome, importance windows are also found all over the genome, a feature that can be attributed to the presence of crucial RNA secondary structures and a single open reading frame that encodes a large polyprotein, subsequently cleaved into individual functional proteins [50], [51].

When comparing the regions extracted by GREAC with the high-frequency mutational events reported by Nextstrain [52], there is a clear overlap, providing valuable insights into biology. This demonstrates the effectiveness of dimensionality reduction in retaining the most informative genomic regions from a mutational perspective. In addition to simplifying the data, this approach helps pinpoint highly mutable regions, which may be linked to the evolution and adaptability of organisms [53].

These regions can reveal signals of speciation or population differentiation, helping us to understand the evolutionary mechanisms behind divergence. Further biological interpretation is possible by investigating whether such regions coincide with regulatory elements (e.g., enhancers, promoters), transcription factor binding motifs, or structural features such as splice sites. It is also possible to identify the dimensionality reduction potential of the tool as the size of the sequences increases; the reduction becomes more significant, thereby enhancing the tool's scalability.

The increasing volume of data from genetic sequencing presents considerable challenges, as working with complete genomes is computationally expensive. Genome lengths vary from smaller sequences in viruses to millions of base pairs in cases such as honeybees, where it was proposed an application in eukaryotic organisms for this study.

The implementation prioritizes extensive optimization for usability, allowing execution on consumer-grade computers due to low memory consumption and availability of precompiled executables. Open-source code ensures transparency and replicability, enabling the scientific community to improve and adapt GREAC for diverse applications.

To conclude, GREAC is a significant contribution to genomic research, offering broad flexibility in execution and demonstrating positive results in extracting characteristic regions from honeybee and viral genomes. The methodology successfully contributes to subspecies and lineage differentiation, as well as genetic diversity analysis, while reducing data dimensionality. The alignment-free approach represents a paradigm shift in comparative genomics, offering computational advantages while maintaining biological relevance. The k-mer-based approach provides a robust foundation for future developments in genomic pattern recognition and classification tasks across diverse taxonomic groups.

It is anticipated that GREAC will be used by researchers and that its development will evolve in tandem with the community, fostering collaborative improvements and adaptations to emerging genomic challenges.

Future work should focus on expanding the methodology to additional organism types and integrating machine learning approaches to enhance pattern recognition capabilities.

Bibliography

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Biologia Molecular da Celula - 5.ed.* Artmed Editora, 2009, ISBN: 9788536321707. [Online]. Available: https://books.google.pt/books?id=bGzbgGZ_A9UC.

[2] D. Lebatteux, H. Soudeyns, I. Boucoiran, S. Gantt, and A. B. Diallo, "Machine learning-based approach kevolve efficiently identifies sars-cov-2 variant-specific genomic signatures," *Plos one*, vol. 19, no. 1, e0296627, 2024.

[3] D. Lebatteux, A. M. Remita, and A. B. Diallo, "Toward an alignment-free method for feature extraction and accurate classification of viral sequences," *Journal of Computational Biology*, vol. 26, no. 6, pp. 519–535, 2019.

[4] D. Lebatteux, H. Soudeyns, I. Boucoiran, S. Gantt, and A. B. Diallo, "Kanalyzer: A method to identify variations of discriminative k-mers in genomic sequences," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 757–762. DOI: 10.1109/BIBM55620.2022.9995370.

[5] P. Pandey, F. Almodaresi, M. A. Bender, M. Ferdman, R. Johnson, and R. Patro, "Mantis: A fast, small, and exact large-scale sequence-search index," *Cell Systems*, vol. 7, no. 2, 201–207.e4, Aug. 2018, ISSN: 2405-4712. DOI: 10.1016/j.cels.2018.05.021. [Online]. Available: http://dx.doi.org/10.1016/j.cels.2018.05.021.

[6] R. S. Roy, D. Bhattacharya, and A. Schliep, "Turtle: Identifying frequent k -mers with cache-efficient algorithms," *Bioinformatics*, vol. 30, no. 14, pp. 1950–1957, Mar. 2014, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu132. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btu132.

[7] C. Marchet, M. Kerbiriou, and A. Limasset, "Blight: Efficient exact associative structure for k-mers," *Bioinformatics*, vol. 37, no. 18, A. Valencia, Ed., pp. 2858–2865, Apr. 2021, ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btab217. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btab217.

[8] P. E. C. Compeau, P. A. Pevzner, and G. Tesler, "How to apply de bruijn graphs to genome assembly," *Nature Biotechnology*, vol. 29, no. 11, pp. 987–991, Nov. 2011, ISSN: 1546-1696. DOI: 10.1038/nbt.2023. [Online]. Available: http://dx.doi.org/10.1038/nbt.2023.

[9] S. M. Aguillon, T. O. Dodge, G. A. Preising, and M. Schumer, "Introgression," *Current Biology*, vol. 32, no. 16, R865–R868, Aug. 2022, ISSN: 0960-9822. DOI: 10.1016/j.cub.2022.07.004. [Online]. Available: http://dx.doi.org/10.1016/j.cub.2022.07.004.

[10] M. A. Pinto, D. Henriques, J. Chávez-Galarza, *et al.*, "Genetic integrity of the dark european honey bee (apis mellifera mellifera) from protected populations: A genome-wide assessment using snps and mtdna sequence data," *Journal of Apicultural Research*, vol. 53, no. 2, pp. 269–278, Jan. 2014, ISSN: 2078-6913. DOI: 10.3896/ibra.1.53.2.08. [Online]. Available: http://dx.doi.org/10.3896/IBRA.1.53.2.08.

[11] D. Henriques, M. Parejo, A. Vignal, *et al.*, "Developing reduced <scp>snp</scp> assays from whole-genome sequence data to estimate introgression in an organism with complex genetic patterns, the iberian honeybee (apis mellifera iberiensis)," *Evolutionary Applications*, vol. 11, no. 8, pp. 1270–1282, Mar. 2018, ISSN: 1752-4571. DOI: 10.1111/eva.12623. [Online]. Available: http://dx.doi.org/10.1111/eva.12623.

[12] L. Boltzmann and B. McGuinness, *Theoretical Physics and Philosophical Problems: Selected Writings* (Vienna Circle Collection). Springer Netherlands, 2012, ISBN: 978-94-010-2091-6. [Online]. Available: https://books.google.com.br/books?id=cIC4BgAAQBAJ.

[13] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, ISSN: 00058580. DOI: 10.1002/j.1538-7305.1948.tb01338.x. [Online]. Available: https://ieeexplore.ieee.org/document/6773024 (visited on 07/04/2023).

[14] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, no. 4, pp. 620–630, May 15, 1957, ISSN: 0031-899X. DOI: 10.1103/PhysRev.106.620. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.106.620 (visited on 07/12/2023).

[15] S. Guiasu and A. Shenitzer, "The principle of maximum entropy," *The Mathematical Intelligencer*, vol. 7, no. 1, pp. 42–48, Mar. 1985, ISSN: 0343-6993. DOI: 10.1007/BF03023004. [Online]. Available: http://link.springer.com/10.1007/BF03023004 (visited on 07/13/2023).

[16] M. H. Pimenta-Zanon, A. Y. Kashiwabara, A. L. L. Vanzela, and F. M. Lopes, "Gramep: An alignment-free method based on the maximum entropy principle for identifying snps," *BMC Bioinformatics*, vol. 26, no. 1, Feb. 2025, ISSN: 1471-2105. DOI: 10.1186/s12859-025-06037-z. [Online]. Available: http://dx.doi.org/10.1186/s12859-025-06037-z.

[17] B. D. Ondov, T. J. Treangen, P. Melsted, *et al.*, "Mash: Fast genome and metagenome distance estimation using minhash," *Genome Biology*, vol. 17, no. 1, Jun. 2016, ISSN: 1474-760X. DOI: 10.1186/s13059-016-0997-x. [Online]. Available: http://dx.doi.org/10.1186/s13059-016-0997-x.

[18] R. M. Karp and M. O. Rabin, "Efficient randomized pattern-matching algorithms," *IBM Journal of Research and Development*, vol. 31, no. 2, pp. 249–260, 1987. DOI: 10.1147/rd.312.0249.

[19] M. Roberts, W. Hayes, B. R. Hunt, S. M. Mount, and J. A. Yorke, "Reducing storage requirements for biological sequence comparison," *Bioinformatics*, vol. 20, no. 18, pp. 3363–3369, Jul. 2004, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bth408. eprint: https://academic.oup.com/bioinformatics/article-pdf/20/

18/3363/48906547/bioinformatics_20_18_3363.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/bth408.

[20] A. Broder, "On the resemblance and containment of documents," in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, ser. SEQUEN-97, IEEE Comput. Soc, 1997. DOI: 10.1109/sequen.1997.666900. [Online]. Available: http://dx.doi.org/10.1109/SEQUEN.1997.666900.

[21] D. P. Solomatine and A. Ostfeld, "Data-driven modelling: Some past experiences and new approaches," *Journal of hydroinformatics*, vol. 10, no. 1, pp. 3–22, 2008.

[22] S. Altschul, "Gapped blast and psi-blast: A new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, Sep. 1997, ISSN: 1362-4962. DOI: 10.1093/nar/25.17.3389. [Online]. Available: http://dx.doi.org/10.1093/nar/25.17.3389.

[23] K. E. Holsinger and B. S. Weir, "Genetics in geographically structured populations: Defining, estimating and interpreting fst," *Nature Reviews Genetics*, vol. 10, no. 9, pp. 639–650, Sep. 2009, ISSN: 1471-0064. DOI: 10.1038/nrg2611. [Online]. Available: http://dx.doi.org/10.1038/nrg2611.

[24] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, Mar. 1993, ISSN: 0098-3004. DOI: 10.1016/0098-3004(93)90090-r. [Online]. Available: http://dx.doi.org/10.1016/0098-3004(93)90090-R.

[25] P. Paschou, E. Ziv, E. G. Burchard, *et al.*, "Pca-correlated snps for structure identification in worldwide human populations," *PLoS Genetics*, vol. 3, no. 9, D. B. Allison, Ed., e160, Sep. 2007, ISSN: 1553-7404. DOI: 10.1371/journal.pgen.0030160. [Online]. Available: http://dx.doi.org/10.1371/journal.pgen.0030160.

[26] I. Cosic, "Macromolecular bioactivity: Is it resonant interaction between macromolecules? - theory and applications," *IEEE transactions on bio-medical engineering*, vol. 41, pp. 1101–14, Jan. 1995. DOI: 10.1109/10.335859.

[27] S. Duffy, L. A. Shackelton, and E. C. Holmes, "Rates of evolutionary change in viruses: Patterns and determinants," *Nature Reviews Genetics*, vol. 9, no. 4, pp. 267–276, 2008.

[28] J. Á. Patiño-Galindo, I. Filip, and R. Rabadan, "Global patterns of recombination across human viruses," *Molecular Biology and Evolution*, vol. 38, no. 6, K. Crandall, Ed., pp. 2520–2531, Feb. 2021, ISSN: 1537-1719. DOI: 10.1093/molbev/msab046. [Online]. Available: http://dx.doi.org/10.1093/molbev/msab046.

[29] J. Hayer, F. Jadeau, G. Deleage, A. Kay, F. Zoulim, and C. Combet, "Hbvdb: A knowledge database for hepatitis b virus," *Nucleic Acids Research*, vol. 41, no. D1, pp. D566–D570, Nov. 2012, ISSN: 1362-4962. DOI: 10.1093/nar/gks1022. [Online]. Available: http://dx.doi.org/10.1093/nar/gks1022.

[30] G. A. Barros-Carvalho, M.-A. Van Sluys, and F. M. Lopes, "An efficient approach to explore and discriminate anomalous regions in bacterial genomes based on maximum entropy," *Journal of Computational Biology*, vol. 24, no. 11, pp. 1125–1133, 2017.

[31] M. M. Breve, M. H. Pimenta-Zanon, and F. M. Lopes, *Basinetentropy: An alignment-free method for classification of biological sequences through complex networks and entropy maximization*, 2022. arXiv: 2203.15635 [q-bio.MN].

[32] M. M. Breve and F. M. Lopes, "A simplified complex network-based approach to mrna and ncrna transcript classification," in *Advances in Bioinformatics and Computational Biology*, J. C. Setubal and W. M. Silva, Eds., Cham: Springer International Publishing, 2020, pp. 192–203, ISBN: 978-3-030-65775-8.

[33] P. Majumdar and S. Niyogi, "Sars-cov-2 mutations: The biological trackway towards viral fitness," *Epidemiology and Infection*, vol. 149, 2021, ISSN: 1469-4409. DOI: 10.1017/s0950268821001060. [Online]. Available: http://dx.doi.org/10.1017/S0950268821001060.

[34] A. Li, J. Zhang, and Z. Zhou, "Plek: A tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme," *BMC Bioinformatics*, vol. 15, no. 1, p. 311, 2014. DOI: 10.1186/1471-2105-15-311.

[35] Y.-J. Kang, D.-C. Yang, L. Kong, *et al.*, "CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features," *Nucleic Acids Research*, vol. 45, no. W1, W12–W16, May 2017, ISSN: 0305-1048. DOI: 10.1093/nar/gkx428.

[36] F. Bueno de Souza, M. H. Pimenta-Zanon, D. Henriques, *et al.*, "Resonant recognition model as a preprocessing technique for rna classification," in *Advanced Research in Technologies, Information, Innovation and Sustainability*, T. Guarda, F. Portela, and M. F. Augusto, Eds., Cham: Springer Nature Switzerland, 2025, pp. 3–17, ISBN: 978-3-031-83435-6.

[37] D. Kaplan, *Bayesian statistics for the social sciences*. Guilford Publications, 2023, pp. 5–5.

[38] H. Ghorbani, "Mahalanobis distance and its application for detecting multivariate outliers," vol. 34, no. 10, pp. 583–595, 2019. DOI: 0.22190/FUMI1903583G.

[39] R. O. Duda, P. E. Hart, *et al.*, *Pattern classification*. John Wiley & Sons, 2006.

[40] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.

[41] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951. DOI: 10.1214/aoms/1177729694.

[42] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[43] C. S. Reddy and R. KVSVN, "An improved fuzzy approach for cocomo's effort estimation using gaussian membership function.," *J. Softw.*, vol. 4, no. 5, pp. 452–459, 2009.

[44] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining, ser. KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794, ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[45] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[46] S. Kurtz, A. Narechania, J. C. Stein, and D. Ware, "A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes," *BMC genomics*, vol. 9, pp. 1–18, 2008.

[47] T.-J. Wu, Y.-H. Huang, and L.-A. Li, "Optimal word sizes for dissimilarity measures and estimation of the degree of dissimilarity between dna sequences," *Bioinformatics*, vol. 21, no. 22, pp. 4125–4132, 2005. DOI: 10.1093/bioinformatics/bti658.

[48] T.-J. Wu, J. P. Burke, and D. B. Davison, "A measure of dna sequence dissimilarity based on mahalanobis distance between frequencies of words," *Biometrics*, pp. 1431–1439, 1997. DOI: 10.2307/2533509.

[49] R. H. Miller, S. Kaneko, C. T. Chung, R. Girones, and R. H. Purcell, "Compact organization of the hepatitis b virus genome," *Hepatology*, vol. 9, no. 2, pp. 322–327, Feb. 1989, ISSN: 0270-9139. DOI: 10.1002/hep.1840090226. [Online]. Available: http://dx.doi.org/10.1002/hep.1840090226.

[50] E. C. Holmes, "The Evolutionary Biology of Dengue Virus," en, in *Novartis Foundation Symposia*, G. Bock and J. Goode, Eds., 1st ed., vol. 277, Wiley, Aug. 2006, pp. 177–192, ISBN: 978-0-470-01643-5 978-0-470-05800-8. DOI: 10.1002/0470058005.ch13. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/0470058005.ch13 (visited on 02/28/2024).

[51] E. Plummer, M. D. Buck, M. Sanchez, *et al.*, "Dengue virus evolution under a host-targeted antiviral," *Journal of Virology*, vol. 89, no. 10, K. Kirkegaard, Ed., pp. 5592–5601, May 2015, ISSN: 1098-5514. DOI: 10.1128/jvi.00028-15. [Online]. Available: http://dx.doi.org/10.1128/JVI.00028-15.

[52] I. Aksamentov, C. Roemer, E. Hodcroft, and R. Neher, "Nextclade: Clade assignment, mutation calling and quality control for viral genomes," *Journal of Open Source Software*, vol. 6, no. 67, p. 3773, Nov. 2021, ISSN: 2475-9066. DOI: 10.21105/joss.03773. [Online]. Available: http://dx.doi.org/10.21105/joss.03773.

[53] M. Barbieri, "What is code biology?" *Biosystems*, vol. 164, pp. 1–10, Feb. 2018, ISSN: 0303-2647. DOI: 10.1016/j.biosystems.2017.10.005. [Online]. Available: http://dx.doi.org/10.1016/j.biosystems.2017.10.005.

Appendix A

Codes

A.1 Region Search Code Listing

```
function _wndwExlcusiveKmersHistogram_bytes(
    exclusiveKmers::Vector{String},
    wndwSize::UInt64,
    sequences::Vector{String},
    histogramThreshold::Float16
)::Tuple{Vector{UInt16},BitArray}

    @assert all(<=(wndwSize), length.(exclusiveKmers)) "All k-mers must be <=
        window size"

    k_len::Int = length(exclusiveKmers[1])
    maxSeqLen = maximum(length, sequences)
    total_windows = maxSeqLen - wndwSize + 1

    kmer_hash_map = Dict{UInt64,Vector{String}}()

    for kmer in exclusiveKmers
        h = compute_hash(kmer)
        if !haskey(kmer_hash_map, h)
            kmer_hash_map[h] = String[]
        end
        push!(kmer_hash_map[h], kmer)
    end

```

```
    # FLoops is the multi - threading library used
    @floop for seq in sequences
        positions = getOccursin_rolling_hash(seq, kmer_hash_map, k_len)
        window_coverage = falses(total_windows)
        @inbounds for pos in positions
            start_window = max(1, pos - Int(wndwSize) + k_len)
            end_window = min(total_windows, pos)

            if start_window <= end_window
                window_coverage[start_window:end_window] .= true
            end
        end
        @reduce(histogram = zeros(UInt32, total_windows) .+
            UInt32.(window_coverage))
    end
    histogram_u16 = UInt16.(min.(histogram, typemax(UInt16)))
    threshold_count = UInt32(ceil(length(sequences) * histogramThreshold))
    threshold_mincount = UInt32(ceil(length(sequences) * 0.15))
    marked = falses(maxSeqLen)
    h_len = length(histogram)
    # hist = lowpass_filter(histogram_u16)
    @inbounds for i in eachindex(histogram)
        if (histogram[i] >= threshold_count)
            # end_pos = min(maxSeqLen, i + Int(wndwSize) - 1)
            # marked[i:end_pos] .= true
            init = max(1, i - 2)
            endi = min(h_len, i + 2)
            while (init <= endi)
                end_pos = min(maxSeqLen, init + Int(wndwSize) - 1)
                marked[init:end_pos] .= true
                init += 1
            end
        end
    end
    return histogram_u16, marked
end
```

Listing A.1: Region search and extraction in sequences

Appendix B

Viral Classification Results Metrics

B.1 Viral Experiments Results

Table B.1: SARS-CoV-2 Results over experiments

Experiment	**F1-Score (σ)**	**Precision (σ)**	**Recall (σ)**	**Final Length (σ)**
1	0.974 (0.018)	0.965 (0.023)	0.993 (0.004)	13265.67 (776.32)
2	0.949 (0.028)	0.945 (0.031)	0.982 (0.005)	10046.15 (472.57)

Table B.2: DENV Results over experiments

Experiment	**F1-Score (σ)**	**Precision (σ)**	**Recall (σ)**	**Final Length (σ)**
1	0.987 (0.003)	0.984 (0.005)	0.990 (0.003)	8167.36 (80.64)
2	0.982 (0.009)	0.976 (0.014)	0.989 (0.004)	6564.89(120.52)

Table B.3: HIV Results over experiments

Experiment	**F1-Score (σ)**	**Precision (σ)**	**Recall (σ)**	**Final Length (σ)**
1	0.556 (0.029)	0.558 (0.031)	0.570 (0.027)	8697.45 (43.79)
2	0.555 (0.027)	0.555 (0.028)	0.570 (0.027)	8763.62 (37.63)

Table B.4: HBV Results over experiments

Experiment	**F1-Score (σ)**	**Precision (σ)**	**Recall (σ)**	**Final Length (σ)**
1	0.998 (0.001)	0.999 (0.000)	0.998 (0.001)	2806.49 (20.30)
2	0.999 (0.000)	0.999 (0.000)	0.999 (0.001)	2899.38 (3.71)

B.2 Model comparisons

Table B.5: DENV Model Comparison Results

Model	**Precision** *mean*	**Precision** σ	**Recall** *mean*	**Recall** σ	**F1** *mean*	**F1** σ
KEVOLVE	0.9823	0.0062	0.9949	0.0018	0.9884	0.0034
GREAC - 0.2%						
Manhattan	0.9758	0.0088	0.9901	0.0029	0.9827	0.0054
KLD	0.8681	0.0161	0.9687	0.0050	0.9036	0.0144
Mahalanobis	0.9761	0.0096	0.9896	0.0027	0.9825	0.0059
Euclidian	0.9773	0.0071	0.9901	0.0026	0.9835	0.0042
GREAC - 0.4%						
Manhattan	0.7937	0.0592	0.8508	0.0514	0.8120	0.0623
KLD	0.6331	0.0419	0.7178	0.0547	0.6263	0.0493
Mahalanobis	0.6900	0.0712	0.7812	0.0870	0.7036	0.0838
Euclidian)	0.7979	0.0537	0.8605	0.0492	0.8177	0.0577
GREAC - 0.6%						
Manhattan	0.5061	0.0614	0.5522	0.0578	0.4848	0.0648
KLD	0.1340	0.1513	0.2786	0.0556	0.1362	0.0976
Mahalanobis	0.4939	0.0584	0.5435	0.0482	0.4730	0.0532
Euclidian	0.5044	0.0598	0.5497	0.0545	0.4830	0.0627
GREAC - 0.8%						
Manhattan	0.4927	0.0578	0.5268	0.0374	0.4630	0.0508
KLD	0.0497	0.0000	0.2500	0.0000	0.0829	0.0000
Mahalanobis	0.4927	0.0578	0.5268	0.0374	0.4630	0.0508
Euclidian	0.4927	0.0578	0.5268	0.0374	0.4630	0.0508

Table B.6: SARS-CoV Model Comparison Results

Model	**Precision** *mean*	**Precision** σ	**Recall** *mean*	**Recall** σ	**F1** *mean*	**F1** σ
KEVOLVE	0.1027	0.0815	0.1221	0.0844	0.0616	0.0638
GREAC - 0.2%						
Manhattan	0.9360	0.0380	0.9882	0.0054	0.9518	0.0319
KLD	0.9098	0.0312	0.9951	0.0024	0.9392	0.0240
Mahalanobis	0.9502	0.0283	0.9902	0.0037	0.9649	0.0205
Euclidian	0.9472	0.0300	0.9892	0.0034	0.9621	0.0217
GREAC - 0.4%						
Manhattan	0.9531	0.0294	0.9970	0.0014	0.9691	0.0223
KLD	0.9069	0.0230	0.9928	0.0057	0.9339	0.0216
Mahalanobis	0.9424	0.0321	0.9939	0.0036	0.9601	0.0253
Euclidian	0.9050	0.0308	0.9939	0.0024	0.9301	0.0274
GREAC - 0.6%						
Manhattan	0.9531	0.0294	0.9970	0.0014	0.9691	0.0223
KLD	0.9069	0.0230	0.9928	0.0057	0.9339	0.0216
Mahalanobis	0.9424	0.0321	0.9939	0.0036	0.9601	0.0253
Euclidian	0.9050	0.0308	0.9939	0.0024	0.9301	0.0274
GREAC - 0.8%						
Manhattan	0.9402	0.0357	0.9940	0.0050	0.9587	0.0286
KLD	0.8755	0.0333	0.9833	0.0117	0.9040	0.0337
Mahalanobis	0.9192	0.0430	0.9851	0.0138	0.9376	0.0409
Euclidian	0.8930	0.0343	0.9899	0.0074	0.9186	0.0310

Table B.7: Features across window sizes for DENV and SARS-COV datasets, showing the mean ± σ of k-mers and windows.

Dataset (Window Size)	**Num. of K-mers**		**Num. of Windows (features)**	
	Mean	σ	Mean	σ
DENV (0.002)	212.7400	1.8387	66.8900	2.7562
DENV (0.004)	212.7400	1.8387	5.1200	1.1615
DENV (0.006)	212.7400	1.8387	1.2400	0.4276
DENV (0.008)	212.7400	1.8387	1.0000	0.0000
SARS-CoV (0.002)	639.9700	41.4543	57.7125	3.2055
SARS-CoV (0.004)	634.3625	40.7321	35.9675	2.6832
SARS-CoV (0.006)	634.3050	40.6139	24.2400	1.9083
SARS-CoV (0.008)	634.3050	40.6139	17.6900	1.5844

Appendix C

Electron-Ion Interaction Potential

Table C.1: Electron-Ion Interaction Potential (EIIP) values for nucleotides [26].

Name	**Letter**	**EIIP Value**
Adenine	A	0.1260
Cytosine	C	0.1340
Guanine	G	0.0806
Thymine	T	0.1335
Uracil	U	0.0289

Table C.2: Electron-Ion Interaction Potential (EIIP) values for amino acids [26].

Name	**Amino Acid**	**Letter**	**EIIP Value**
Leucine	Leu	L	0.0000
Isoleucine	Ile	I	0.0000
Asparagine	Asn	N	0.0036
Glycine	Gly	G	0.0050
Valine	Val	V	0.0057
Glutamic Acid	Glu	E	0.0058
Proline	Pro	P	0.0198
Histidine	His	H	0.0242
Lysine	Lys	K	0.0371
Alanine	Ala	A	0.0373
Tyrosine	Tyr	Y	0.0516
Tryptophan	Trp	W	0.0548
Glutamine	Gln	Q	0.0761
Methionine	Met	M	0.0823
Serine	Ser	S	0.0829
Cysteine	Cys	C	0.0829
Thereonine	Thr	T	0.0941
Phenyalanine	Phe	F	0.0946
Arginine	Arg	R	0.0959
Aspartic Acid	Asp	D	0.1263

C.1 Table of SARS-COV SNPs positions

Frequency of mutated sequences per Lineage	Lineage	Genome position	Ref	ut seq
0.995240	BA.1	14408	C	T
0.991896	BA.1	10029	C	T
0.995566	BA.1	23403	A	G
0.990729	BA.1	23525	C	T
0.990348	BA.1	8393	G	A
0.991150	BA.1.1.7	10029	C	T
0.993117	BA.1.1.7	23403	A	G
0.994994	BA.2	14408	C	T
0.993588	BA.2	17410	C	T
0.992587	BA.2	10029	C	T
0.996946	BA.2	23403	A	G
0.995168	BA.2	23525	C	T
0.996308	BA.2	24424	A	T
0.996189	BA.2	24469	T	A
0.992156	BA.2	26060	C	T
0.990902	BA.2	2790	C	T
0.993259	P.1	14408	C	T
0.998197	P.1	23403	A	G
0.992730	P.1	23525	C	T
0.990741	P.1	25088	G	T
0.990569	P.1	26149	T	C
0.990669	P.1	28512	C	G
0.995120	P.1	5648	A	C
0.992134	BA.1.1.7	3037	C	T
0.994215	BA.1	3037	C	T
0.991406	BA.1	5386	T	G
0.990394	BA.1	13195	T	C
0.995811	BA.2	3037	C	T
0.994696	BA.2	12880	C	T
0.993514	BA.2	15714	C	T
0.994719	P.1	3037	C	T

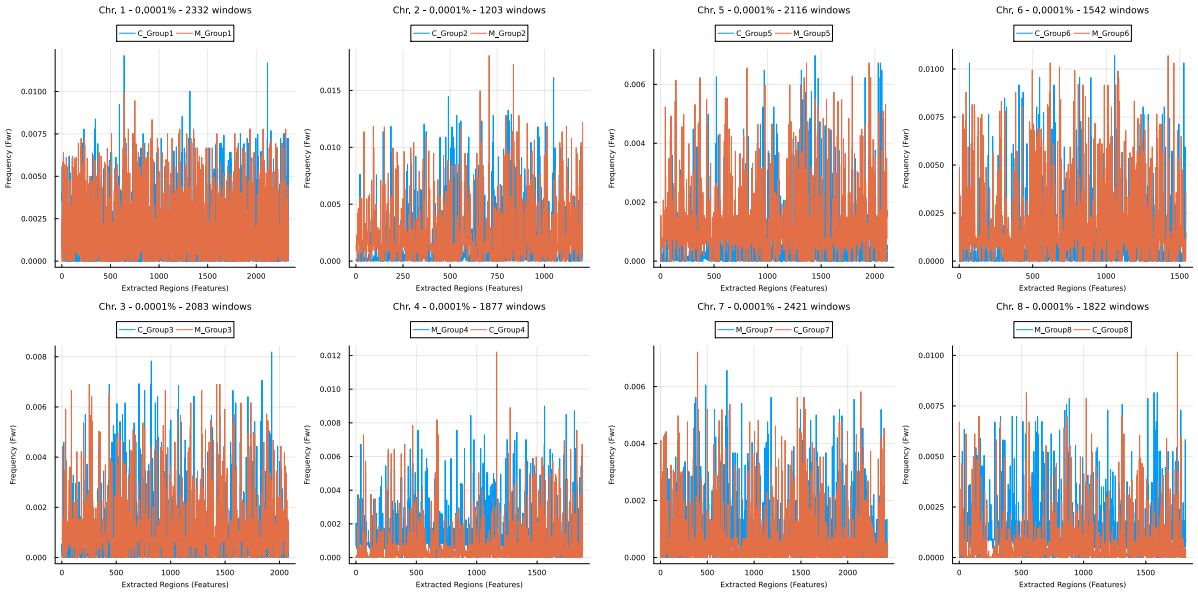
Appendix D

Honeybees Analysis

D.1 GREAC BED file Example

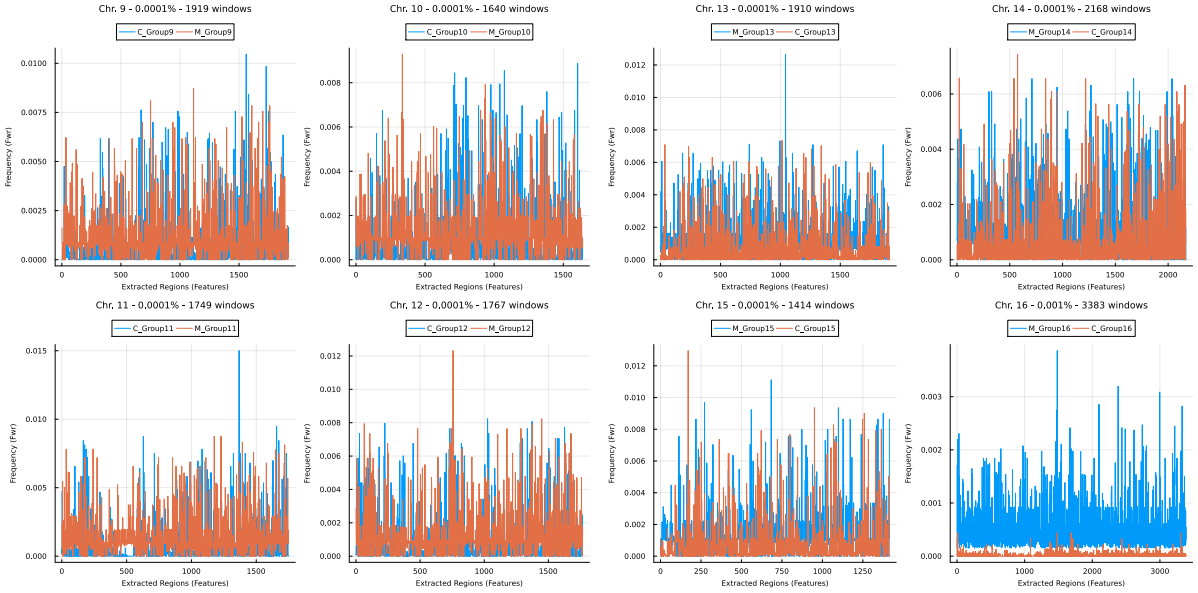
```
Group1  24534192    24534492
Group1  1452740     1453040
Group1  16796699    16796999
Group1  164802      165102
Group1  16795087    16795387
Group1  26443795    26444095
Group1  21434812    21435112
Group1  21085728    21086028
```

D.2 Lineages Chromosomes Frequency Behavior



(a) Signals from chromosome 1 to 4

(b) Signals from chromosome 5 to 8



(c) Signals from chromosome 9 to 12

(d) Signals from chromosome 13 to 16

Figure D.1: Reference behavior signals across all 16 chromosomes in lineages C and M analysis result.