Deep Unsupervised Learning for Biodiversity Analyses

Representation learning and clustering of bacterial, mitochondrial, and barcode DNA sequences

by

Pablo Millan Arias

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor:	Lila Kari Professor, School of Computer Science University of Waterloo
Faculty Member(s):	Bin Ma Professor, School of Computer Science University of Waterloo
	Yaoliang Yu Associate Professor, School of Computer Science University of Waterloo
Internal-External Examiner	Andrew Doxey Associate Professor, Dept. of Biology University of Waterloo
External Examiner:	Dan Tulpan Associate Professor, Dept. of Animal Biosciences University of Guelph

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The main contribution of this thesis consists of the following research articles:

- "DeLUCS: Deep Unsupervised Clustering of DNA Sequences," published in *PloS ONE*. The significant individual contributions are listed below.

  - **Fatemeh Alipour**: Conceptualization, data curation, formal analysis, design of computational experiments, methodology, software implementation, validation, visualization, writing of initial manuscript, review, and editing.
  - **Kathleen A. Hill**: Conceptualization, funding acquisition, design of computational experiments, computational resources, review, and editing.
  - **Lila Kari**: Conceptualization, formal analysis, funding acquisition, design of computational experiments, methodology, project administration, computational resources, review, and editing.
  - **Pablo A. Millán Arias**: Conceptualization, data curation, formal analysis, investigation, methodology, software implementation, validation, visualization, writing of initial manuscript, review, and editing.

    PMA and FA contributed equally to this work.

- "*i*DeLUCS: A deep learning interactive tool for alignment-free clustering of DNA sequences," published in *Bioinformatics*. The significant individual contributions are listed below.

  - **Kathleen A. Hill**: Funding acquisition, computational resources, review, and editing.
  - **Lila Kari**: Conceptualization, Formal analysis, funding acquisition, design of computational experiments, project administration, computational resources, writing of initial manuscript, review, and editing.
  - **Pablo A. Millán Arias**: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing of initial manuscript, review, and editing.

- "Environment and taxonomy shape the genomic signature of prokaryotic extremophiles," published in *Scientific Reports*. The significant individual contributions are listed below.

- **Joseph Butler**: Data collection, Data curation, writing of initial draft, biological interpretation of the results.
- **Gurjit Randhawa**: Software implementation, writing of initial draft.
- **Kathleen A. Hill**: Design of computational experiments, funding acquisition, computational resources, biological interpretation of the results, review, and editing.
- **Lila Kari**: Design of computational experiments, funding acquisition, project administration, computational resources, manuscript writing, review, and editing.
- **Maximillimian Solstysiak** : Design of computational experiments, data collection, review, and editing.
- **Pablo A. Millán Arias**: Design of computational experiments, data collection, software implementation, manuscript writing, review, and editing.

  PMA and JB contributed equally to this work.

- "BarcodeBERT: Transformers for Biodiversity Analysis", presented at the 4^{th} *Workshop of Self-Supervised Learning: Theory and Practice* at the NeurIPS 2023 Conference. The significant individual contributions are listed below.
  - **Angel X. Chang**: Design of computational experiments, funding acquisition, computational resources, review, and editing.
  - **Austin Wang**: Software implementation of Bayesian zero-shot learning evaluation methodology, manuscript writing, review, and editing.
  - **Dirk Steinke**: Manuscript writing, review, and editing.
  - **Graham Taylor**: Design of computational experiments, funding acquisition, project administration, computational resources, manuscript writing, review, and editing.
  - **Iuliia Zarubiieva**: Project management and editing.
  - **Joakim Bruslund Haurum**: Conceptualization, manuscript review, and editing.
  - **Lila Kari**: Funding acquisition, computational resources, review, and editing.
  - **Monireh Safari**: Data collection, software implementation of BarcodeBERT, review, and editing.
  - **Niousha Sadjadi**: Software implementation of BarcodeBERT, review, and editing.

– **Pablo A. Millán Arias**: Design of computational experiments, data collection, software implementation of BarcodeBERT, manuscript writing, review, and editing.
– **Scott C. Lowe**: Design of computational experiments, conceptualization, manuscript review, and editing.
– **ZeMing Gong**: Software implementation of Bayesian zero-shot learning evaluation methodology, manuscript writing, review, and editing.

PMA, MS, and NS contributed equally to this work.

Abstract

Amid the recent surge in next-generation sequencing technologies, alignment-free algorithms stand out as a promising alternative to traditional alignment-based methods in phylogenetic analyses. Specifically, the use of genomic signatures has enabled the success of supervised machine learning-based alignment-free methods in taxonomic classification. Motivated by this success, this dissertation investigates the potential of unsupervised learning-based alignment-free algorithms in genomic signature categorization. We conclude that meaningful information can be learned without reliance on labels, suggesting that supervision can be effectively eliminated from the learning process.

First, we developed a **De**ep **L**earning-based **U**nsupervised **C**lustering method for DNA **S**equences, DeLUCS. It trains a discriminative neural network to identify meaningful taxonomic clusters without supervision. In this process, we designed and conducted several proof-of-concept experiments to validate the effectiveness of our methodology in various datasets. Building on the contrastive nature of DeLUCS, we enhance it through self-supervised representation learning. We introduce *i*DeLUCS and its applicability in non-parametric clustering of DNA sequences, matching the performance of alignment-based and alignment-assisted clustering algorithms. In addition, we successfully apply unsupervised learning to categorize the genomic signatures of microbial extremophiles. We provide quantitative evidence suggesting that microbial extremophile genomes may contain information beyond ancestry or taxonomy. The evidence provided by our computational experiments led to the biological insight that a pervasive environmental component exists in the genomic signature of extremophilic organisms and could potentially redefine the concept of genomic signature. Finally, we introduce BarcodeBERT, a transformer-based encoder optimized for DNA barcodes. Since barcodes are short DNA fragments that contain enough information for the taxonomic identification of an organism, our model learns this taxonomy information and generates expressive embeddings that enable efficient classification of barcodes of novel specimens. We evaluate the quality of these embeddings through several downstream tasks, such as supervised fine-tuning and linear probing for species classification of known species and nearest neighbours probing for genus classification of unknown species. Additionally, the learned embeddings proved effective in a zero-shot classification framework for images of insects, underscoring the model's utility in integrating genomic and visual data for species identification.

Our work attempts to connect the worlds of biodiversity and taxonomic identification with the world of deep unsupervised learning. Our findings reveal deep learning's untapped potential to capture taxonomic information, even without supervision. The methodologies presented in this dissertation can also be used to learn expressive DNA embeddings and test evolutionary hypotheses.

Acknowledgements

First and foremost, I want to thank my supervisor, Lila Kari. She has been the living example of a novel concept blending together a lab leader, a scientific advisor, and a personal counsellor, each of them with proportions that are fine-tuned for the success of her students. She is one of a kind, and without her guidance and encouragement, I would not have finished this thesis. I aspire that one day, I will give my students the same kind of mentorship I have received from her.

I would also like to thank Kathleen Hill, my biology mentor during my Ph.D. studies. Her advice and help with most of the biology material involved in this dissertation were fundamental for its completion. During my last year, I also had the privilege of being mentored by Graham Taylor through the BIOSCAN project. His guidance and encouragement were vital for completing the project presented in Chapter 6.

I thank the members of my Examining Committee, Bin Ma, Yaoliang Yu, Andrew Doxey, and Dan Tulpan, who graciously agreed to travel from Guelph to be physically present in my defence.

I want to thank all of my co-authors in the Kari Genomics Lab. My collaboration with Fatemeh Alipour was fundamental in developing the clustering work presented in Chapter 3. Our multiple research dinners made my first year at Waterloo more enjoyable. I thank Niousha Sadjadi and Monireh Safari for their technical contribution to the contents of Chapter 6. I also thank other members of the lab, especially Zihao Wang, for his friendship and for providing access to extra computational resources when I needed them. Lastly, I thank Shane Ding for his assistance with testing and releasing our software tools.

I thank all the members of the Hill lab at Western, especially my co-authors, Joseph Butler and Maximillian Soltysiak, who contributed to data curation and essential biological insight into the results obtained in Chapter 5. I also thank Gurjit Randhawa, who significantly helped me navigate the last year of my academic journey. I thank Daniel Olteanu and Connor Holmes for providing the users' perspectives as biologists during software testing.

I also thank all the members of the machine learning group in the BIOSCAN project, my co-authors, Angel Chang, ZeMing Gong, Austin Wang, Joakim Haurum, Iuliia Zarubiieva, and Dirk Steinke. I especially thank Scott Lowe, who has mentored me in the past few months and introduced me to the best practices in machine learning-oriented software development.

Thank you to my friends in Waterloo: Pang Sing, Robin Sheffler, Mike and Michelle Gropp, Mitch Tierney, Thomas and Darby Millman, Mike Brnjas, Issac Veldhuis, Elijah

Birley, the Velilla family, Daniel and Mariana Lozano, and all my friends at WMB church. I apologize to anyone who I have forgotten. They helped me adopt this foreign country as my new home.

Taking a step back, I want to thank Julián Quiroga, who taught my first course in machine learning, and Daniel Jaramillo, Germán Combariza, and Andrés Vargas at Javeriana University, who introduced me to research in this field. I will always be grateful for their support and encouragement when I was considering an academic career.

I want to thank my parents, Andres and Janeth, and my brother Gabriel for their prayers, love, and support despite the physical distance that has separated us over these years. Thank you for visiting me during the cold Canadian winters and welcoming me with open arms whenever I travelled back home. I want to thank my friends in Colombia Jorge and David, who have not forgotten about me, and the Sanchez family for adding a much-needed quota of fun to all my visits and helping me decompress in stressful times. Finally, I would like to thank Juliana, my wife, best friend and the love of my life, for serving as a proofreader, project manager, and graphic design advisor while I worked on this thesis; I thank her for the emotional support she has provided me and all the sacrifices she made, including countless sleepless nights, and moving away from home to support me in this journey.

Dedication

Dedicated to the architect behind biodiversity: my God, my guide, my saviour.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The term "biodiversity" encompasses every living creature on our planet, including plants, bacteria, animals, and humans. Despite our best efforts in studying biodiversity, we have only scratched the surface of this vast landscape, with only about 1.2 million species formally identified and described [140]. Although this number might seem like there is not much left to be discovered, several studies indicate that the number of species yet to be identified varies between 2.2 million and 1 trillion for microbial biodiversity [118, 120], and between 161 million to 370 million for eukaryotes [110, 114, 210].

These estimates are enough to assert that cataloging all life on our planet is a monumental challenge that even with considerable effort, is projected to only be completed by the end of this century. Nevertheless, the pace of discovery is accelerating, with the rate of species discovered per year constantly increasing from thousands to tens of thousands [56]. Such ambitious goals and current progress demand the development of novel technologies for taxonomic identification that can manage the influx of new data and scale with the rapid pace of species discovery. Within this broader context of system development for species discovery and identification, we found the development of efficient models for taxonomic categorization to be a key area for focused study. It opens up two primary avenues for exploration: firstly, we can focus on developing models capable of *classifying* novel DNA sequences into known species or flag them as unknown. Secondly, we can focus on *clustering* DNA sequences of evolutionarily related organisms into operational taxonomic units (OTUs) without any attached semantic meaning. Considering that the taxonomic identifiers of newly discovered organisms are still uncertain, it is reasonable to suggest that the latter

strategy is arguably more appropriate for the task, and it is also helpful in comprehending the diverse evolutionary pathways leading to today's biodiversity.

Both *classification* and *clustering* are well-established concepts in machine learning and pertain to different learning paradigms. Classification is linked with supervised learning, where a model is trained to categorize unknown data points into known categories. In contrast, clustering is associated with unsupervised learning, where a model deduces patterns or properties from data, grouping similar data points together without prior knowledge of possible categories.

This dissertation attempts to put machine learning at the service of biodiversity and sets out to achieve two main objectives. Firstly, we aim to develop models and techniques for categorizing DNA sequences without supervision. Our models should be able to take DNA sequences extracted from organisms across different domains of life and group sequences of closely related organisms together without any prior knowledge of their taxonomic identifiers. Secondly, we aim to use unlabelled data to enhance supervised classification pipelines for taxonomic classification. This involves developing models that can provide a simplified representation of a DNA sequence, enabling a straightforward supervised classifier to assign a pre-defined taxonomic identifier accurately. In essence, our research explores the potential of unlabelled data in the context of DNA sequence categorization.

The methods and analyses presented in this dissertation are motivated by other general problems in bioinformatics. They may offer valuable insights into addressing them, particularly those involving unsupervised learning on genomic data. For example, our findings could facilitate the grouping of reads or contigs into genomes via non-parametric clustering, aiding genome reconstruction. They could also be used to develop systematic methods to estimate evolutionary relationships among species based on genomic signatures. Furthermore, our research underscores the necessity for developing robust, scalable systems to process the increasing volume of genomic data efficiently. The development of such systems will complement innovation in all the other processes involved in DNA barcoding. Collectively, these advancements can bring us closer to a future where we can monitor biodiversity in real time at specific locations and assess biodiversity loss due to habitat destruction and overexploitation. Moreover, these new systems could not only provide more insight into the genetic composition of groups defined by current taxonomy but also redefine boundaries to identify novel taxonomic groups.

1.2 Related work

Machine learning has significantly transformed numerous fields, delivering groundbreaking results across various scientific domains. These advancements, however, often hinge on the availability of extensive annotated datasets. Fields like computer vision and natural language processing, where data annotation is relatively cost-effective, have led the development of general machine learning algorithms, paving the way for their broader adoption. In contrast, disciplines such as the "omic" sciences (genomics, transcriptomics, proteomics), where data collection and annotation have been historically expensive, have seen much slower progress in adopting machine learning-based methodologies.

In the realm of biodiversity and taxonomic categorization, the advent of next-generation sequencing (NGS) technologies have significantly changed the landscape by producing vast amounts of genomic data. Despite these advancements, many of these novel algorithms still depend on traditional alignment-based methods, which limit the range of data they can effectively analyze. Prominent among the tools for DNA sequence classification that have found widespread application in both research and industry are BLAST (Basic Local Alignment Search Tool) [126], CLUSTAL [176], and MEGA [107]. While highly accurate for specific applications, these methods are prohibitively expensive, computationally, and inaccurate in the presence of whole genomes or non-homologous sequences.

The computational demands of alignment-based classifiers [204] and their reliance on homologous sequences have spurred the development of alignment-free methods [224, 225], presenting a viable alternative. Notably, the growing amount of data has encouraged the development of innovative machine learning-based taxonomic classifiers, often by adapting successful techniques from fields like computer vision or natural language processing. Supervised learning-based approaches have demonstrated remarkable success in classifying DNA sequences [112, 116, 213], with k-mer count-based methods emerging as both popular and efficient alternatives [224]. These methods have surpassed traditional alignment-based techniques in various applications, including whole-genome phylogenies [162], microbial community profiling [116], and species-level DNA barcoding [203]. The success of these methods is driven by the use of genomic signatures [97], a concept that encapsulates any measurable characteristic or representation for which patterns from sequences of closely related organisms are more similar to each other than patterns in sequences from distantly related organisms. The use of genomic signatures [116, 184] and suitable numerical representations [1], alongside machine learning methodologies enables alignment-free evolutionary analysis and constitutes one of the primary motivations of our unsupervised methodologies.

These highly performant supervised learning algorithms rely heavily on accurate taxonomic labels of sequences in the training set for successful classification. Naturally, errors or

disputes in the "ground truth" labels can lead to inaccuracies in subsequent classifications. This motivates the shift in our approach towards using unsupervised machine learning, as it has the potential to be more efficient for sequence categorization. Unsupervised learning, which operates on unlabelled sequences, can identify patterns in data while avoiding the propagation of labelling errors in the training data points and facilitating the classification of novel sequences by forming new clusters.

Nevertheless, the application of unsupervised learning to genomic sequence clustering has progressed more slowly than supervised classification. Previous efforts have focused mainly on applying generic algorithms like K-means or Gaussian Mixture Models (GMM) to various representations of DNA sequences, with studies exploring K-means clustering [4, 11, 25, 88, 89] and digital signal processing techniques [3, 77, 130]. Recent proposals have introduced advanced methodologies to accelerate parametric and non-parametric clustering algorithms [60, 89]. Deep-learning-based approaches have also made significant inroads into the realm of metagenomic binning, a field studying microbial communities and their functions. Several methods closely related to our work, such as those presented in [150, 154, 207, 220], have demonstrated the potential of employing neural-based clustering to group metagenomic fragments effectively.

Finally, foundation models, inspired by large language models (LLMs), represent a new paradigm in genomics. These models are pretrained in an unsupervised manner on a broad corpus of unlabelled data, learning general patterns before being fine-tuned for specific tasks. Several examples of this semi-supervised approach are provided in the literature, [38, 92, 149, 222], and our work incorporates language modelling techniques within larger architectures to embed DNA barcodes into a meaningful representation space for various downstream tasks.

1.3 Outline and contributions

In this dissertation, we explore unsupervised learning techniques for the categorization of genomic signatures across a broad spectrum of genomic data: homologous and non-homologous sequences; specific genic regions and whole genomes; mitochondrial and nuclear DNA and prokaryotic and eukaryotic genomes. Our methods bridge the performance gap between supervised and unsupervised training paradigms and correspond to meaningful steps toward the broader goal of creating a comprehensive catalogue of life on Earth.

Chapter 2 sets the foundation for our dissertation. It provides relevant biology background as well as all the mathematical background for supervised and unsupervised

deep-learning, optimization, neural network architectures and key concepts in information theory.

Chapter 3 introduces DeLUCS, a novel algorithm that learns from unlabelled genomic sequences and clusters them based on their similarity. For each input sequence, DeLUCS generates suitable data augmentations and learns to maximize the predictability of cluster assignments between the real samples and the augmented copies. It then aggregates the predictions of various independently trained networks to reduce the variance and boost overall performance. To the best of our knowledge, DeLUCS was the first deep learning-based clustering method based on genomic signatures. This chapter draws upon the collaborative work of Millán Arias, Alipour, Hill, and Kari [134], where Alipour and Millán Arias are equal first co-authors.

Chapter 4 capitalizes on the contrastive nature of DeLUCS and improves it through self-supervised representation learning, introducing *i*DeLUCS. This chapter introduces the contrastive learning framework, its core components, and the information-theoretic clustering ensemble that serves as an alternative to traditional majority voting. It also introduces an additional performance evaluation metric and explores the extension of this framework to support non-parametric clustering outcomes. The content of this chapter is based on Millán Arias, Hill, and Kari [136], where Millan Arias is the first author.

Chapter 5 presents a biological application of alignment-free methodologies and machine learning algorithms and utilizes them to investigate the patterns in the genomic signatures of prokaryotic extremophiles. First, a dataset of 693 genomes of prokaryote extremophiles is curated, and the organisms are categorized based on taxonomy and environment. Then, both supervised and unsupervised machine learning algorithms are used to find high-quality clusters that correctly approximate the data distribution. Interestingly, even without supervision, the algorithms group some organisms based on shared environmental conditions rather than solely on genetic relatedness. This finding indicates that adaptations to extreme temperatures and pH conditions may leave a discernible imprint on the genomic signatures of microbial extremophiles. The results in this chapter challenge traditional taxonomy-centric views and suggest that both environmental and taxonomic factors influence the genomic signature of specific organisms, especially in extreme conditions. This chapter is based on joint work by Millán Arias, Butler, Randhawa, Soltysiak, Hill and Kari [135], with Millán Arias being the first computer science author and Josepth Butler being the first Biology author.

Chapter 6 explores the application of self-supervised representation learning to various DNA barcode classification tasks. We pretrain our model, BarcodeBERT, on a database of $\sim$1.5 million barcodes using an auxiliary loss function. After pretraining, our model embeds barcode sequences into an expressive representation space suitable for classifying new

specimens. We evaluate our self-supervised model's embedding quality against those from supervised convolutional neural networks and fine-tuned foundational models across various classification tasks. BarcodeBERT outperforms other models, even without fine-tuning, in complex taxonomic identification tasks, such as the partial taxonomic identification of barcodes from previously unobserved invertebrate species. The content of this chapter is based on a paper by Millán Arias et al. [137], with Millán Arias, Safari and Sadjadi as equal first co-authors.

Chapter 7 summarizes our work, identifies the remaining challenges and discusses future work.

Chapter 2

Foundations

This chapter lays the groundwork necessary for understanding the interdisciplinary field that this research is situated in, at the intersection of biology and computer science, focusing on applying machine learning techniques to biological data. The chapter is divided into three main sections, each designed to provide a foundation for readers of varying backgrounds.

In Section 2.1, we introduce the basic biological concepts essential for computer scientists venturing into comparative genomics. This section is further subdivided into topics that cover the fundamental building blocks of life, including nucleic acids described in Section 2.1.1 as the "Alphabet of Life," the structure and function of genes and genomes in Section 2.1.2, and an exploration of genomes' evolutionary relatedness in Section 2.1.3. These subsections aim to equip readers with a basic understanding of molecular biology necessary for the application of computational analyses in our work.

Section 2.3 introduces a few basic concepts from information theory, setting the stage for their usage and application in subsequent chapters.

Finally, Section 2.2 shifts focus towards introducing machine learning concepts. This section begins with an overview of supervised machine learning in Section 2.2.1, followed by a more substantial dive into deep neural networks in Section 2.2.2. Section 2.2.3 discusses the critical aspect of optimization in machine learning models, and Section 2.2.4 broadens the scope to include unsupervised learning and other machine learning paradigms beyond the supervised framework. This section introduces only the machine learning concepts used in our methodologies.

2.1 Molecular biology

Despite the vast evolutionary timescale, the molecular building blocks of life have demonstrated remarkable stability [99]. Nucleic acids, comprising deoxyribonucleic acid (DNA) and ribonucleic acid (RNA), are essential macromolecules that enable most of the biological functions of all living organisms [151]. Throughout our work, as is customary in bioinformatics, we will use a simplified representation of these macromolecules as strings and, for the most part, will not deviate from this representation. However, this section explains the composition of these macromolecules, their associated terminology, and how they interact, as this information is essential to interpret the subsequent analyses in this work.

2.1.1 Nucleic acids: the alphabet of life

Both nucleic acids, DNA and RNA, are composed of chains of nucleotides called *strands*. Each nucleotide consists of three main components: a sugar, either ribose in RNA or deoxyribose in DNA, a phosphate group, and a nitrogenous base or nucleobase (adenine [A], guanine [G], cytosine [C], and thymine [T] in DNA, which is replaced by uracil [U] in RNA). There are five carbon atoms labelled from $1'$ to $5'$ in the sugar such that the nucleobase and the phosphate group are connected to the $1'$ and the $5'$ carbon, respectively. The sugar also contains a hydroxyl group (composed of one hydrogen atom and one oxygen atom) connected to its $3'$ carbon. The alternating sugars and phosphate groups are connected by covalent bonds in each single strand, forming the *backbone* DNA strand (see Figure 2.1 a-b). For each DNA single strand, the unconnected phosphate group on a nucleotide at one end is called the $5'$ end of the strand, and the other is called the $3'$ end.

DNA can appear both as a single-strand or double-stranded biomolecule, where two single DNA strands of opposite orientation bind to each other to form a chemical structure that resembles a double-helix (see Figure 2.1c). This process is made possible by the hydrogen bonds that form between the nucleotides at each position. Specifically, an A on one strand will form two hydrogen bonds with a T on the opposing strand, while a C on one strand will form three hydrogen bonds with a G on the opposing strand. This phenomenon is known as *Watson-Crick complementarity*.

Throughout our work, each DNA single strand is automatically associated with its corresponding DNA sequence as follows: Given the DNA alphabet $\Sigma = \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$, the word $b_1 b_2, \ldots, b_n \in \Sigma^*$ represents the physical DNA sequence $b_1 b_2, \ldots, b_n$ read in the $5'$ to $3'$ direction. Furthermore, given a nucleotide b, we denote its complementary nucleotide as $\bar{b}$. The reverse complement of a sequence $s = b_1 \ldots b_n$ is the sequence $\bar{s} = \bar{b}_n \ldots \bar{b}_1$ formed by taking the complement of each nucleotide in s, then reversing the resulting sequence.

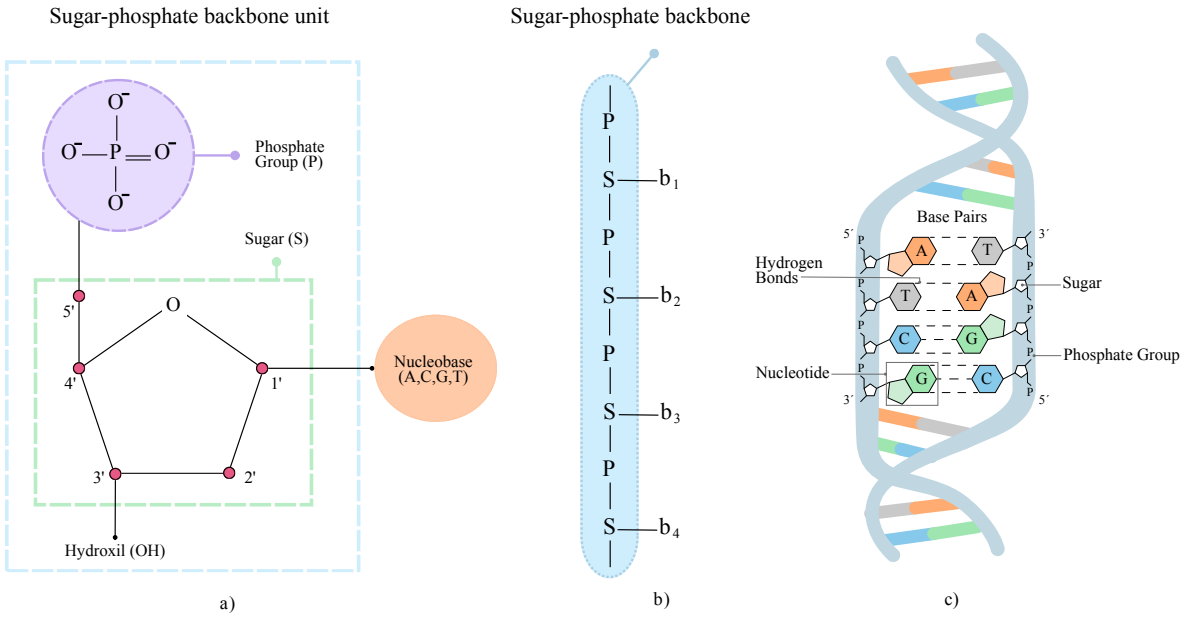


Figure 2.1: Diagrams showing the chemical structure of a DNA molecule. a) Each DNA nucleotide consists of a nucleobase (A, C, G, T), a sugar molecule (with a hydroxyl group), and a phosphate group. The sugar's five carbon atoms are labelled from $1'$ to $5'$. The nucleobase attaches to the $1'$ carbon, the hydroxyl group to the $3'$ carbon, and the phosphate group to the $5'$ carbon. b) Depiction of the interconnection of multiple nucleotides to form DNA's sugar-phosphate backbone. P represents the phosphate group, S represents the sugar and each b_i, $i \in \{1, \ldots, 4\}$, represents an arbitrary nucleobase. c) Illustration of DNA's double-helix structure and the three main components of each nucleotide. The figure illustrates the hydrogen bonds between Watson-Crick complementary nucleotide pairs: adenine-thymine and guanine-cytosine.

2.1.2 Genes, genomes and the central dogma

There are three fundamental information transfers involved in the central dogma of molecular biology: DNA replication, DNA-RNA transcription and RNA-protein translation. This theory states that there is a unidirectional flow of information between each type of molecule. Information flows from DNA to RNA and from RNA to proteins, but never the other way around [87]. It is important to note that exceptions to the central dogma have been observed, where RNA direct replication and RNA-DNA reverse transcription can occur [99].

DNA replication

Reproduction is a fundamental property of all living systems and can be observed at several levels [99]. At the molecular level, DNA replication is the most basic form of reproduction. This process involves unwinding the double helix in the DNA double-strand, with each strand serving as a template for synthesizing a new complementary strand. Replication proceeds bidirectionally, starting from specific sites known as *origins* and terminating under various biochemical conditions. This process is assisted by the DNA-polymerase enzyme, which recognizes the origins, unfolds the strands and facilitates the synthesis. The origin of replication may not be unique, so organisms with larger genomes present multiple origins and termination sites, ensuring the replication process is efficient [151].

DNA-RNA transcription

Although transcription, the synthesis of RNA from a DNA template, shares the fundamental mechanisms with replication, it is different in its execution [99]. In transcription, only one strand of DNA is transcribed into RNA, and instead of DNA polymerase, the catalyst is RNA polymerase, which unwinds the DNA to expose a segment for RNA synthesis (Figure 2.2). Unlike replication, transcription is highly selective, transcribing only specific segments of DNA, regulated by distinct start and stop signals. These signals correspond to special strands within the genome recognized by the RNA polymerase to determine the transcription region.

After transcription, another process called alternative splicing takes place. Here, the RNA transcript, or pre-mRNA, is edited before it is translated into a protein. Non-coding regions within the pre-mRNA, also called introns, are removed, and coding regions, also called exons, are joined together into genes, which serve as a template for protein synthesis. Not all transcribed RNA serves the purpose of creating genes [99]; RNA molecules are categorized based on their function: messenger RNA (mRNA) encodes proteins, ribosomal RNA (rRNA) participates untranslated in the structure of the ribosome, transfer RNA (tRNA) are untranslated strings that facilitate protein synthesis, and small nuclear RNA (snRNA) plays a role in processing mRNA.

RNA-protein translation

Before describing this cellular process, it is important to introduce another basic type of biomolecule, the *amino acid*. While a detailed exploration of their chemical structure is beyond the scope of this dissertation, we note that these are organic acids, each containing

a specific *side chain* that acts as an identifier. Among the numerous amino acids found in Nature, only 20 play a crucial role in the RNA-protein translation process; hence, we say that the amino acid alphabet has 20 symbols.

In the RNA-protein translation process, mRNA sequences are translated into amino acid sequences, also called *polypeptide chains* or proteins (See Figure 2.2). The translation process is string-to-string transduction, where the protein is the functional agent, and the

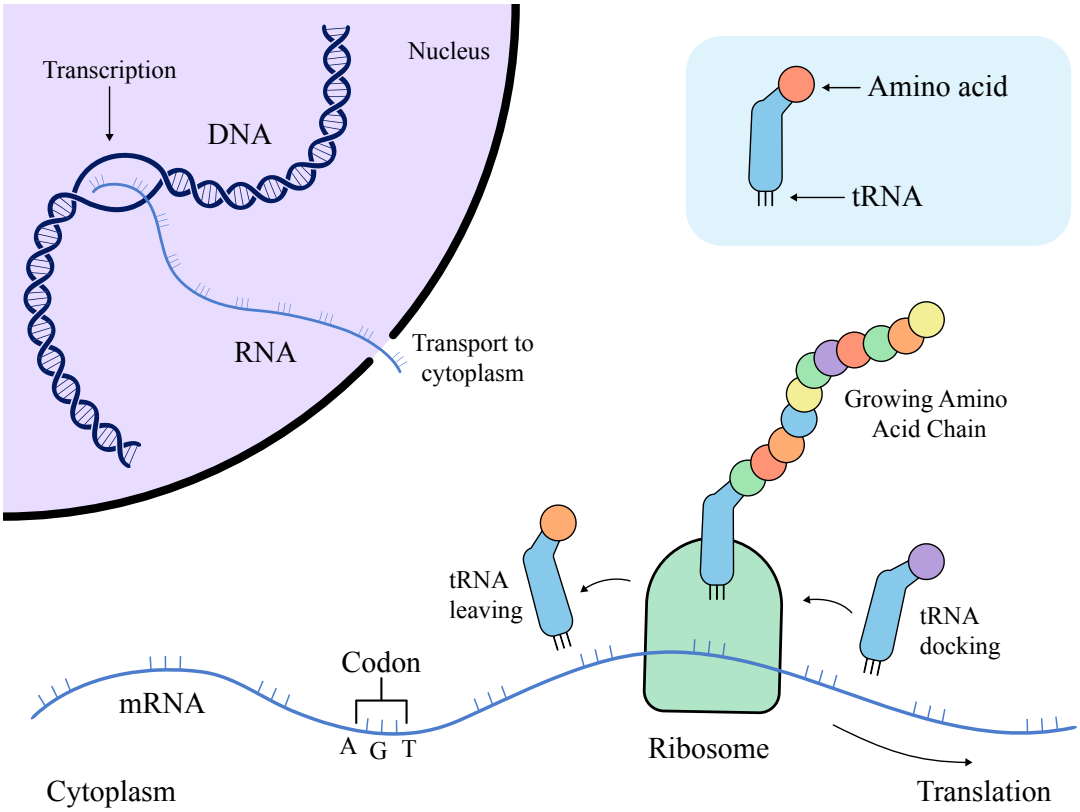


Figure 2.2: Illustration of the central dogma of molecular biology and the general flow of genetic information within a cell: from DNA to RNA and from RNA to proteins. The process begins with DNA-RNA transcription, where a segment of DNA is transcribed into messenger RNA (mRNA). This segment exits the nucleus, enters the cytoplasm, and is translated into a polypeptide chain in the ribosome with the assistance of transfer RNA (tRNA). Each tRNA molecule carries a specific amino acid to the ribosome, matching its anticodon with the corresponding codon on the mRNA strand.

polypeptide chain is to be viewed exclusively as a string over the amino acid alphabet [87, 151]. The fact that the nucleotide alphabet contains four elements and the amino acid alphabet contains 20 has forced nature to implement a translation mechanism where substrings of length $\lceil \log 20 / \log 4 \rceil = 3$ nucleotides, known as codons, encode a single amino acid. With $4^3 = 64$ possible codons and only 20 amino acids, this system exhibits redundancy, averaging 3.2 codons per amino acid. The actual map from codons to amino acids is known as the genetic code.

Translation takes place in the ribosome, a complex structure composed of proteins and rRNA. It acts as a molecular machine by reading mRNA sequences and creating proteins. To match codons on the mRNA with their corresponding amino acids, the ribosome uses tRNA as an adaptor. Each tRNA is specific to a codon-amino acid pair and carries the amino acid and an anticodon (the reverse complement of a codon triplet) that binds to the mRNA codon. Protein synthesis begins when a particular tRNA initiates the polypeptide chain, with subsequent tRNAs adding amino acids in sequence until a codon coding for a stop operation ends the process. The resulting polypeptide then folds into its functional form as a protein.

2.1.3 Evolution and the Tree of Life

The study of *evolution* sheds light on the existence of life and how living systems have changed over time. Most evolutionary theories hold that the diversity of life arose by inherited variation through an unbroken line of descent [39, 40, 47]. This concept of common ancestry, supported by extensive evidence ranging from genetic to fossil records, forms the foundation of taxonomic classifications and evolutionary theories that describe the diversity of life [40, 183]. Another related concept central to the study of early evolution and life's origin is the concept of the last universal common ancestor of all life forms (LUCA), or the progenote. In this theory, the LUCA is a hypothetical primitive entity in the evolution of life that preceded prokaryotes (single-celled organisms without a nucleus) from which all current biodiversity originated [48, 208, 211].

The evolutionary journey, from prokaryotic ancestors to today's biodiversity, is captured in the *universal tree of life*, a conceptual model that describes the relationships between organisms through time. Efforts to classify the immense variety of life have evolved from morphology-based systems using phenotypic traits as classification criteria to those incorporating genetic information, revealing relationships that are not apparent through physical characteristics alone. One of these models, based on ribosomal RNA, is the Three Domain System introduced by Woese, Kandler, and Wheelis [212] (Figure 2.3) and is the most widely adopted model for the universal tree of life. This system describes a tripartite

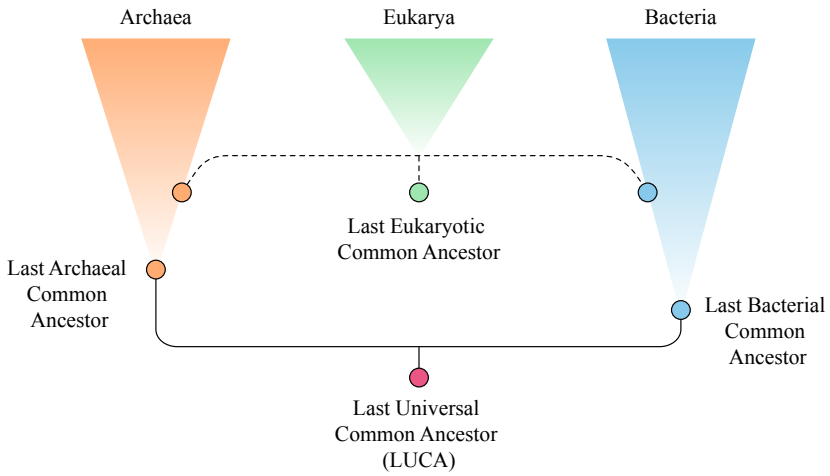


Figure 2.3: Illustration of the three-domain characterization of the Tree of Life. This system describes a tripartite division of life into Archaea, Bacteria, and Eukarya based on genetic similarities. The figure highlights the role of the last universal common ancestor (LUCA) in the tree as a hypothetical organism representing the most recent common ancestor to all organisms in the three domains.

division of life into Bacteria, Archaea, and Eukarya. This last group comprises all the organisms whose cells have a membrane-bound nucleus.

More precisely, Bacteria are prokaryotic organisms that are present almost anywhere on Earth. They comprise a wide range of organisms, from pathogens to vital symbiotic species like those that fix nitrogen in the soil or aid human digestion. Archaea, a more recently discovered group, also comprise prokaryotic organisms that surprisingly share many genetic similarities with eukaryotes despite their morphological similarities to bacteria [212]. In particular, they possess similar genes and enzymes involved in transcription and translation [185, 212]. Eukarya includes all multicellular organisms and some single-celled ones, all made of complex cells with nuclei and organelles like mitochondria and chloroplasts. Eukaryotes are subdivided into four major kingdoms: animals, plants, fungi, and protists.

2.1.4 Evolutionary relationships: Reconstructing the Tree of Life

The scientific field that studies the evolutionary history of a species or group is known as *phylogeny*. This field utilizes a "phylogenetic" tree with individual species as leaf nodes to capture the evolutionary history of specific groups. *Taxonomy*, a related field, is the discipline of categorizing organisms into distinct evolutionarily related groups at many different levels, starting from top-level domains such as Eukaryota and descending through kingdoms, phyla, classes, orders, families, and genera to finally reaching individual species. Unlike older taxonomic systems, which grouped organisms based solely on physical similarities, modern taxonomy has now adopted the use of molecular phylogenetic analyses that are based on genomic similarities to accurately assign organisms to taxonomic groups.

Comparative biology and phylogenetic analyses usually rely on the fundamental concept of homology. In genetics, "homolog" refers to a protein and its corresponding gene, which share a common ancestry. A gene inherited by two or more species from a common ancestor is considered a *homologous gene*. Although homologous genes may result in similar sequences, sequence similarity does not necessarily indicate homology [87, 99]. Conserved regions with homologous sequences are useful for estimating evolutionary distances and constructing phylogenetic gene trees using alignment-based algorithms such as the Smith-Waterman [181] for local alignment or the Needleman-Wunsch [147] global alignment. This comparison among sequences allows the classification of new sequences by comparing them to gene sequences from known taxa [51, 113, 126]. After sequence alignment, alignment information can be used to construct various phylogenetic tree topologies, each evaluated by several optimality criteria to determine its suitability in describing the data. Two commonly used reconstruction criteria are maximum parsimony and maximum likelihood. While maximum likelihood is particularly effective for analyzing homologous sequences from distantly related organisms [53], its computational complexity [32] and dependence on accurate sequence alignment make it impractical for whole genome analysis.

The advent of biochemical research in the late 20th and early 21st centuries has significantly enhanced phylogenetic analyses, primarily through advancements in DNA sequencing technologies. Researchers now have access to various methods and bioinformatics tools that provide precise estimates of species divergence. A key method enabling these advancements is the *polymerase chain reaction* (PCR), a standard laboratory technique that produces multiple copies of specific segments of double-stranded DNA, making it possible to sequence DNA more effectively. Although performed in vitro, PCR is based on natural DNA replication mechanisms, and it selectively amplifies a specific DNA region, generating millions of copies. As shown in the first row of Figure 2.4, the input of PCR is contained within a solution that includes a double-stranded DNA template, nucleotides (not shown in the figure), a special heat-stable DNA polymerase (not shown in the figure),

and the forward (F) and reverse (R) primers. These primers are short single strands of DNA that are complementary to the ends of the segment that is to be amplified and serve as sequence-specific starting points for DNA synthesis.

Figure 2.4 illustrates the steps involved in each PCR cycle: First, in *denaturation*, the hydrogen bonds between the two strands of the DNA double-helix are broken at high temperatures, usually > 94°C, resulting in two single strands (forward and reverse) called the templates. Next, the temperature is decreased to 45-68°C for the primers to attach to the templates. This process is called *annealing*. The forward (F) primer attaches to the complementary site on the reverse DNA strand while the reverse (R) primer attaches to the complementary site on the forward DNA strand; DNA synthesis is initiated at each of the two primer sites by DNA polymerase. Finally, during *extension*, the polymerase extends the newly synthesized DNA sequence by incorporating the individual nucleotides, each complementary to the corresponding nucleotide of the template, creating a new strand that is Watson-Crick complementary to the template DNA. The temperature of this step varies between 65°C and 75°C depending on the protocol [28]. After each cycle, the quantity of the DNA segment flanked by the primers doubles, producing 2^n copies of the target DNA region after n cycles. Most thermocycling protocols include 30-45 cycles.

Another highly influential technology fundamental to the success of molecular methods for phylogeny and taxonomy is *DNA sequencing*. The first of these methods, Sanger Sequencing, was invented in the 1970s and was considered the gold standard for DNA sequencing due to its high quality. The process involves using a DNA polymerase to create a complementary copy of a single-stranded DNA template. This process is initiated by a complementary primer at the 5' end of the template, and nucleotides are sequentially added according to Watson-Crick complementarity. Besides free-floating nucleotides, the input to the reaction also contains four di-deoxynucleotides (ddNTPs), each corresponding to a DNA base. While ddNTPs are similar to nucleotides and can attach to the growing chain, they lack the 3' hydroxyl group required for further DNA extension, resulting in the termination of the chain once they are incorporated. Each type of ddNTP is tagged with a unique fluorescent dye, allowing for automated reading of the DNA sequence through light emission at different wavelengths. This method generates multiple copies of DNA fragments of varying lengths, with distinct ddNTPs marking the termination of each chain at different positions in the template molecule. The original DNA sequence is determined by aggregating the fluorescent emissions of the ddNTPs.

Sanger sequencing has been slowly replaced by NGS systems over the last decade. Although with less precision, these systems enable the processing of millions or even billions of sequencing reactions simultaneously, which usually leads to producing longer DNA sequences at a fraction of the cost [66]. Although different machines, with many technical

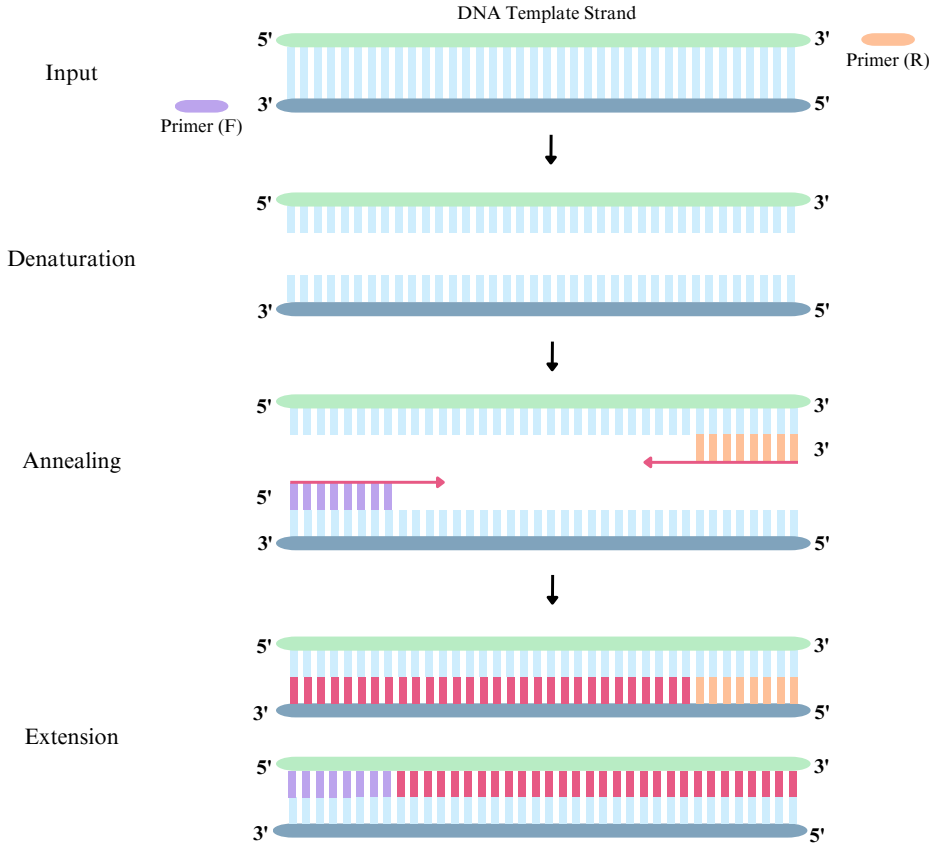


Figure 2.4: The figure illustrates one polymerase chain reaction (PCR) cycle generating two copies of the template sequence. The inputs are contained within a solution, usually called PCR mix, including a double-stranded DNA template, shown in the first row, a forward (F) and a reverse (R) DNA single-stranded primer, are also present in the first row. Free-floating nucleotides (not shown) and DNA polymerase (not shown). Each PCR cycle has three steps: denaturation, annealing, and extension. During the denaturation step, the temperature increases, and the template breaks into two single strands. During the annealing step, the temperature decreases, and the single strands from the template attach to the corresponding primers. Finally, during the extension step, the primers are extended by DNA polymerase using the free-floating nucleotides present in the solution according to the template strands and the rules of Watson-Crick complementarity.

variations, have been invented, they all share some standard features, such as sample preparation, sequence amplification, raw data collection and signal aggregation. For a more comprehensive description of modern sequencing technologies, we refer the reader to [179]. It is important to acknowledge that these are not error-free technologies. There are instances where the hardware utilized to read the signal corresponding to specific nucleobases in the DNA sequence cannot differentiate them with certainty. In such cases, sequencing technologies may generate an `N` symbol instead of the expected {`A`,`C`,`G`,`T`} symbols to account for the uncertainty. This is a recognized challenge in the field of bioinformatics, and we employ various strategies to effectively address this issue in our work.

So far, we have introduced the basic terminology and briefly mentioned the key technological advancements that have greatly improved the precision and accuracy of phylogenetic tree representations and taxonomic classifications. From all the frameworks using these technologies, we will expand on the two methodologies close to our work: DNA barcoding and alignment-free analysis using genomic signatures.

DNA barcoding

Although related to taxonomy and phylogenetic inference, DNA barcoding is a tool designed specifically for species identification with limited applicability at lower taxonomic levels. This tool leverages sequence variation in short, standardized homologous genic regions, called DNA barcodes, to discriminate species [28]. Despite being proposed in 2003 as a taxonomic identification method for organisms in the kingdom Animalia [28, 74], DNA barcoding has been successfully extended to other kingdoms within Eukarya (Plant, Fungi, Protista) [165]. The homologous DNA barcode region, or simply barcode, found to be the most effective for organisms in the kingdom Animalia is a 648-base pair (bp) fragment near the 5'-end of the mitochondrial cytochrome c oxidase subunit I (COI) gene. It was selected for identification purposes due to its advantageous properties, including a high copy number per cell, maternal inheritance without recombination, a higher nucleotide substitution rate facilitating species differentiation, and the absence of introns simplifying sequence comparison [165].

Despite its capacity to provide accurate identification at multiple taxonomic levels and contribute to understanding phylogenetic diversification, DNA barcoding was not designed to reconstruct phylogenetic relationships. Its primary function is to distinguish between species. DNA barcoding does have some limitations. For example, it may not be able to differentiate between recently diverged species, and the presence of multiple barcode variants within an individual may hinder accurate sequence recovery [28]. Nevertheless, DNA barcoding has become a dependable and effective tool for species identification and

discovery. It has significantly improved biodiversity cataloging and expanded our knowledge of species distribution. In particular, the International Barcode of Life Consortium (iBOL) and the Barcode of Life Database System [164], have enabled the collection and storage of more than 16 M total barcodes from more than 250,000 animal species, more than 72,000 plant species and more than 25,000 other species.

Genomic Signatures

Evolutionary analyses traditionally focus on studying homologous genes inherited from a common ancestor to determine the relationships between species. While this approach remains fundamental, scientists have been aware of other unique, species-specific patterns within the genome since the early 1960s. Early biochemical experiments, for example, showed evidence that the relative abundance of dinucleotides might be a unique species-specific nonrandom pattern [93]. However, the significance of these patterns as a potential tool for evolutionary analysis remained underexplored until the 1990s. It was not until this decade that Karlin and Burge [97] formally introduced the concept of a *genomic signature*, highlighting its value in studying evolutionary relationships beyond the analysis of homologous sequences. They defined a genomic signature as any numerical quantity that shows greater similarity among DNA sequences of closely related organisms compared to those of more distantly related organisms. Initially, dinucleotide relative frequencies were proposed as a robust signature that describes inter- and intra-species variations. This revelation of characteristic DNA sequence patterns through k-mers frequency profiles (normalized histograms of counts of subwords of length k) has enabled the development of alignment-free methods. Such methods identify genomic similarities without the necessity of homologous sequences, essentially capturing part of the phylogenetic information from the signature [42].

Before these developments, in 1990, Jeffrey applied the Chaos Game Representation (CGR) to DNA sequences, uncovering hidden species-specific structural patterns [90]. CGRs offer a two-dimensional graphical representation of genomic sequences. In this construction, each nucleotide in the DNA alphabet is mapped to a corner of a unit square. Formally, the mapping w is defined by $w(A) = (0,0)$, $w(C) = (0,1)$, $w(G) = (1,1)$, $w(T) = (1,0)$. Given a DNA sequence $s = b_1 b_2 \dots b_n$, the CGR sequence $\mathbb{S} = \{(x_i, y_i) \mid 1 \leq i \leq n\}$ of s consists of the collection of points with coordinates recursively defined as follows:

$$\mathbf{x}_0 = \left(\frac{1}{2}, \frac{1}{2}\right) \qquad \mathbf{x}_i = \frac{1}{2}\left(\mathbf{x}_{i-1} + w(b_i)\right).$$

This marked the first instance of genome sequences being visualized, thereby illuminating their local and global characteristics. Notably, CGRs and k-mer frequency vectors are

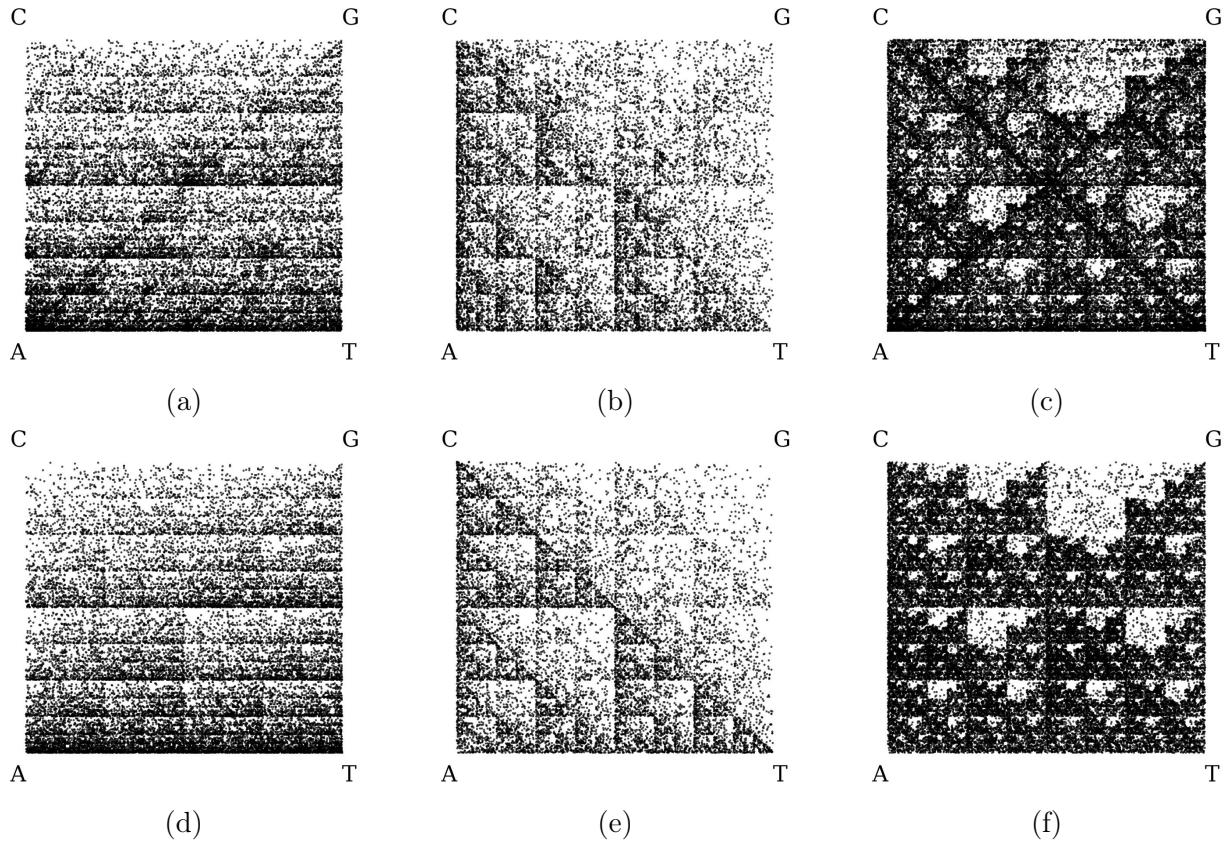


Figure 2.5: Frequency chaos game representation at a resolution $k = 9$ of (a) The first 25,000 bp of the *Bacillus mycoides genome* in domain Bacteria — Accession ID: NZ_CP009691.1; (b) The complete mitochondrial genome of *Equus Caballus* in kingdom Animalia –Accession ID:NC_001640; (c) Chromosome 1 in genome assembly GRCh38.p14 of the *Homo sapiens* in kingdom Animalia; (d) Random DNA sequence avoiding letters G and C with high probability; (e) Random DNA sequence avoiding letter G with high probability; (f) Random DNA sequence avoiding sub-string CG with high probability.

interconnected through the $fCGR_k$ representation, which discretizes the continuous CGR into a two-dimensional unit square image. The intensity of each pixel within this image represents the frequency of a specific k-mer in the sequence [43]. This renders $fCGR_k$ both as a graphical and numerical representation, encoding patterns characteristic of each species' genome. For example, consider the representations in the top row of Figure 2.5. These correspond to fCGRs of real DNA sequences at a resolution $k = 9$, each having a specific fractal pattern (genomic signature). On the other hand, consider the representations in the bottom row of Figure 2.5. These correspond to the representations of mathematical sequences, whose construction is based on sub-string avoidance. This comparison shows that visual patterns in fCGRs from real DNA sequences can be used to infer properties of the original sequence. Note that comparison by visual inspection is only possible for distantly related organisms; more careful mathematical and computational models are necessary to compare closely related organisms. These alignment-free methods based on genomic signatures present a competitive alternative to alignment-based methods both in phylogenetic [163, 224, 225] and taxonomic classification studies [7, 112, 116] and are an inspiration for the methods developed in this dissertation.

2.2 Information theory

In this section, we introduce the basic concepts from information theory that are relevant to our work. For a more thorough exposition of the field, the reader is referred to [36, 123].

2.2.1 Entropy

Given a discrete random variable $\mathbf{x}$ that takes values $\boldsymbol{x} \in \mathcal{X}$ and has probability mass function $p(\boldsymbol{x}) = P(\mathrm{x} = \boldsymbol{x})$, the entropy $H(\mathbf{x})$ is a measure of the average uncertainty in the random variable and is defined by

$$H(\mathbf{x}) = -\sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) \log_2 p(\boldsymbol{x}). \tag{2.1}$$

The base of the log used is typically 2, so the information is measured in *bits* (short for binary digits). If the base of the log is used in e, the unit of information is called a *nat*. Throughout the rest of this chapter, we will measure information in nats to omit the base for convenience. $H(\mathbf{x})$ represents the average number of bits/nats required to describe the random variable $\mathbf{x}$.

For distributions p and q over the same event space $\mathcal{X}$, the cross entropy H_{CE} is defined as:

$$H_{CE} = -\sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) \log q(\boldsymbol{x}) \tag{2.2}$$

serving as a measure of the dissimilarity between q and the true distribution p and commonly used as a loss function in machine learning.

2.2.2 Mutual information

For a second random variable $\tilde{\mathbf{x}}$ that takes values $\tilde{\boldsymbol{x}} \in \tilde{\mathcal{X}}$, we define the conditional entropy $H(\mathbf{x}|\tilde{\mathbf{x}})$, for a pair sampled from a joint probability distribution $p(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = P(\mathbf{x} = \boldsymbol{x}, \tilde{\mathbf{x}} = \tilde{\boldsymbol{x}})$, as the entropy of a random variable $\mathbf{x}$ conditional on having some knowledge about the variable $\tilde{\mathbf{x}}$ as:

$$H(\mathbf{x}|\tilde{\mathbf{x}}) = -\sum_{\boldsymbol{x}\in\mathcal{X},\tilde{\boldsymbol{x}}\in\tilde{\mathcal{X}}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x}, \tilde{\boldsymbol{x}})}{p(\boldsymbol{x})} \tag{2.3}$$

The reduction in the uncertainty of $\mathbf{x}$ introduced by the additional knowledge provided by $\tilde{\mathrm{x}}$ is called *mutual information*, and it is defined by

$$MI(\mathbf{x}, \tilde{\mathbf{x}}) = H(\mathbf{x}) - H(\mathbf{x}|\tilde{\mathbf{x}}) = \sum_{\boldsymbol{x}\in\mathcal{X},\tilde{\boldsymbol{x}}\in\tilde{\mathcal{X}}} p(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \log \frac{p_(\boldsymbol{x}, \tilde{\boldsymbol{x}})}{p(\boldsymbol{x})p(\tilde{\boldsymbol{x}})}. \tag{2.4}$$

Mutual information measures the dependence between the two random variables and represents the amount of information one random variable contains about another. $I(\mathrm{x}, \tilde{\mathrm{x}})$ is symmetric, always non-negative, and is equal to zero if and only if x and $\tilde{\mathrm{x}}$ are independent.

2.2.3 Kullback-Leibler divergence

For two arbitrary distributions p and q, it is also useful to define a relative measure to identify the proximity between them. We say that $D(p, q)$ is a *divergence measure* if $D(p, q) \geq 0$, with equality if and only if $p = q$. Note that to be a metric, D must also satisfy the triangle inequality $D(p, q) \leq D(p, r) + D(r, q)$ and be symmetric $D(p, q) = D(q, p)$. There are many possible divergence measures used to determine the proximity of distributions. Here, we focus on a particular one, the *Kullback-Leibler (KL) divergence*, also known as the relative entropy between two distributions p and q. For discrete distributions, the KL divergence is defined as:

$$D_{KL}(p||q) = \sum_{x\in\mathcal{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{2.5}$$

Note that the previous expression can be re-written in terms of the other quantities as:

$$D_{KL}(p||q) = \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) \log p(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) \log q(\boldsymbol{x}) \tag{2.6}$$

$$= -H(p) + H_{CE}(p, q). \tag{2.7}$$

For distributions p and q of a continuous random variable x, the KL divergence is defined as:

$$D_{KL}(p||q) = \int_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} dx. \tag{2.8}$$

Finally, we note that the mutual information between random variables $\mathbf{x}$ and $\tilde{\mathbf{x}}$ can now be expressed in terms of the KL of their respective marginal and conditional distributions $p(\boldsymbol{x})$, $p(\tilde{\boldsymbol{x}})$, $p(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ as:

$$MI(\mathbf{x}, \mathbf{x}) = D_{KL}(p(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \;||\; p(\boldsymbol{x})q(\tilde{\boldsymbol{x}})) \tag{2.9}$$

emphasizing the information gained from using the joint distribution over independent marginal distributions.

2.3 Machine learning

This section provides the necessary technical background on machine learning and artificial neural networks (ANN), but it is not meant to be a thorough exposition of the field. For a more exhaustive presentation, the reader is referred to [63, 65, 143].

2.3.1 Supervised machine learning

Supervised machine learning is a fundamental approach in the vast landscape of machine learning methodologies. The primary goal is to learn a parameterized mapping function, $f(\boldsymbol{x}; \boldsymbol{\theta})$, that connects inputs $\boldsymbol{x} \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$. Inputs, also known as features, covariates, or predictors, are typically represented as fixed-dimensional vectors of numerical values. For example, in genomics, these inputs could be DNA or RNA sequences. In classification tasks within supervised learning, the output space is a discrete set of mutually exclusive labels, $\mathcal{Y} = \{1, 2, \ldots, K\}$, where each label corresponds to a distinct class. In genomics, these could correspond to the presence or absence of a pattern in the input sequence.

Formally, the training data for a supervised learning model consists of a set of $N \in \mathbb{N}$ input-output pairs $\mathbb{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$, known as the training set. The model learns from this data to accurately predict the output y given a new input $\boldsymbol{x}$. From a probabilistic standpoint, the function $f(\boldsymbol{x}; \boldsymbol{\theta})$ encapsulates the conditional probability of the target variable y given the input $\boldsymbol{x}$, symbolized as $p(y|\boldsymbol{x}, \boldsymbol{\theta})$. Conceptually, this is equivalent to approximating the posterior predictive distribution of the target based on new input data grounded in an assumed probabilistic model.

Binary classification is a particular case of this problem where Y can take on one of two possible values, e.g., $y \in \{0, 1\}$. An example of binary classification in bioinformatics can be identifying patterns that serve functional roles within the genome where labels 0 and 1 correspond to the presence or absence of the pattern. Multi-class classification is also applied in the context of genomics and is very close to our work. In this context, the set $\mathcal{X}$ consists of DNA sequences that must be classified, for example, into different taxonomic categories comprising the set $\mathcal{Y}$.

2.3.2 Deep neural networks

Inspired by the structure of the human brain [125], deep neural networks consist of a particular composition of functions called layer functions or simply layers. These layers are interconnected, modelling the stimuli propagation of brain synapses [111]. More specifically, given input and output spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, the function $f_l = f(\boldsymbol{x}; \boldsymbol{\theta}_l)$ is called a parameterized "layer function". An ANN is a mapping $y = f(\boldsymbol{x}; \boldsymbol{\theta})$ defined as a composition of a finite number $L \in \mathbb{N}$ of layer functions.

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = f_L \circ \cdots \circ f_1(\boldsymbol{x}, \boldsymbol{\theta}) \tag{2.10}$$

The function f_1 is called the *input layer*, f_L is called the *output layer*, and all the others are referred to as the *hidden layers*. Neural networks are usually called *deep* as the number of hidden layers increases and are categorized according to the behaviour of their main layers, interconnections, and architectures. In this subsection, we briefly describe the generalities for each class of networks related to our work.

Multi-Layer Perceptron

In a Multi-Layer Perceptron (MLP) [169] or deep feedforward network, each layer function consists of a linear operator parameterized by a *weight matrix* $\mathbf{W}_l$. These layers are also called fully connected layers because the output depends on all the components of the input

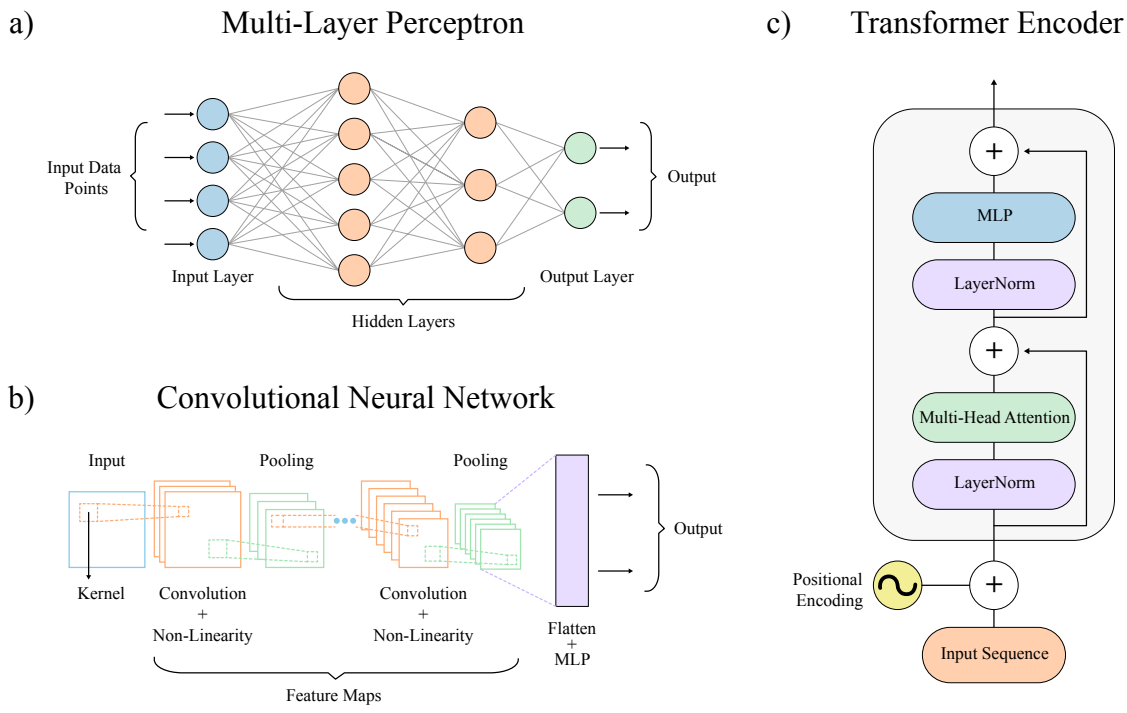


Figure 2.6: The figure illustrates three distinct neural network architectures utilized in this dissertation: a) the multi-layer perceptron (MLP) or feedforward neural network, which consists of multiple layer functions that process inputs sequentially; b) the convolutional neural network (CNN) architecture, as a composition of convolution and pooling layers, usually followed by an MLP for prediction; and c) the transformer encoder architecture as a stack of encoder layers each consisting of a multi-head attention block and an MLP using residual connections and layer normalization (Adapted from [201]).

via the weight matrix. Each fully connected layer is followed by a non-linear *activation function* $\psi(\cdot)$. More specifically, each layer's function f_l, combines the previous layer's output, or *representation* z_{l-1}, with its own non-linear activation ψ_l:

$$z_l = f_l(z_{l-1}) = \psi_l(\mathbf{W}_l z_{l-1}) \tag{2.11}$$

Though any differentiable function could be an activation function, non-linearity is essential for approximating arbitrary functions [81]. In practice, several activation functions have been proposed in the literature [63], but the most commonly used are the Rectified

Linear Units (ReLU) [63], defined as:

$$\psi(x) = \text{ReLU}(x) = \max(x, 0) \tag{2.12}$$

Appropriate non-linear activation functions ensure that MLP can act as universal function approximations [81]. Note that although an MLP with a single hidden layer can achieve this, deeper networks have been empirically and theoretically proven to surpass architectures with fewer layers [111].

Convolutional Neural Networks

The expressive power of MLPs is theoretically enough to approximate arbitrary functions. However, learning the optimal parameters becomes harder as the number of parameters increases, which may occur with an increase in the dimensionality of input space or when more layers are added to the network. MLPs struggle with high-dimensional data modalities such as images or raw audio. Moreover, an extra level of complexity of these modalities is that salient patterns can appear in different "positions" across the input, so the model's prediction must be invariant to the translation of these patterns across the input. Convolutional neural networks (CNNs) became popular in this context, where translational invariance and shared parameters are essential to effectively learn patterns in the input data [143]. The idea behind this type of architecture is to replace the linear operators $\mathbf{W}_l$ in MLPs with a more versatile linear operator with fewer parameters that could act as a template across the input. That could be achieved through the convolutional operator, which will be detailed below.

Given two functions $h, g : \mathbb{R}^D \to \mathbb{R}$, the convolution $(h * g) : \mathbb{R}^D \to \mathbb{R}$ is defined as:

$$(h * g)(\boldsymbol{u}) = \int_{\mathbb{R}^D} h(\boldsymbol{u} - \boldsymbol{z}) \cdot g(\boldsymbol{z}) d\mathbf{z} \tag{2.13}$$

For discrete signals $x, w : \mathbb{Z} \to \mathbb{R}$, the convolution operation is defined by

$$(x * w)[n] = \sum_{-\infty}^{\infty} x[n - m] \cdot w[m] \tag{2.14}$$

where $x[n]$ is the input signal and $w[n]$ is called the *convolution kernel*, represented by an array of parameters, or *weights*, optimized through the learning process. In the context of CNNs, the main layer functions become

$$f_l[z_{l-1}] = \psi(z_{l-1} * w_l),$$

where the outputs of each layer, or hidden representations $z_l = f_l[z_{l-1}]$, are often called the *feature maps*.

Convolutions can retain useful positional information, but for tasks like image classification where location invariance is desired (an object can be present anywhere in the image), convolution layers must be alternated with *pooling layers*. A max-pooling layer selects the maximum value from its inputs. In contrast, an average pooling calculates the mean in fixed-sized windows, summarizing the feature maps and making the output unaffected by the input pattern's location. A typical CNN design alternates between convolutional and pooling layers and uses a linear output layer. This type of architecture, as seen in Figure 2.6, is specialized in learning patterns from data with a known grid-like topology, i.e., contains spatial structure, because the parameters in the convolution kernel can be shared and optimized to extract meaningful information throughout the entire input. Furthermore, they can massively compute and combine feature maps to derive non-linear input-output relationships, proving effective in visual applications for classification and feature extraction [111], as well as in analyzing short homologous DNA sequences [52, 189].

Transformer Models

In the neural networks described so far, hidden representations $\mathbf{z}_l$ are obtained through linear transformations, followed by non-linear activation functions as $\mathbf{z}_l = \psi(\mathbf{W}_l\mathbf{z}_{l-1})$. However, transformer models [201] use a sophisticated way to quantify the dependency of one part of the input on other parts of the input and scale the weight matrices accordingly. This *attention mechanism* introduces a more flexible approach where the weights depend on the input, allowing for dynamic interactions between input elements and enhancing the model's capability to learn long-range dependencies in the data. Given this novel feature, transformer models became state-of-the-art in many tasks involving sequential data, such as machine translation [201], language modelling [44], text summarization [24] and protein sequence generation [124]. For a more comprehensive list of applications and efficient implementation of transformer models, see [190].

The transformer model comprises stacked encoder layers, each consisting of multi-headed attention, residual connections, feedforward layers, and layer normalization, as illustrated in Figure 2.6-c. We will describe each of the components below. These blocks enable the transformer to process sequential data effectively, making it a powerful tool for various sequence generation tasks.

Self-Attention:

Given a sequence of n input embeddings $\boldsymbol{x}_i$, each represented as a row of a matrix $X \in \mathbb{R}^{n \times d}$, the self-attention mechanism aims to model interactions between each input and every other input in the sequence. This mechanism enables the capture of contextual relationships in data, regardless of their positional distances within the sequence.

The self-attention function $\text{sa}(\cdot, X)$ for an input $\boldsymbol{x}_i$ is defined as a weighted sum of all inputs embeddings, where the weights encode the relevance of each embedding:

$$\text{sa}(\boldsymbol{x}_i, X) = \sum_{j=1}^{n} a[\boldsymbol{x}_i, \boldsymbol{x}_j] \cdot \Phi_v \boldsymbol{x}_j^T \tag{2.15}$$

Here, Φ_v is a linear transformation applied to the input embeddings $\boldsymbol{x}_j$, and the attention weights $a[\boldsymbol{x}_i, \boldsymbol{x}_j]$ are computed as follows:

$$a[\boldsymbol{x}_i, \boldsymbol{x}_j] = \frac{\exp(\Phi_q \boldsymbol{x}_i \cdot (\Phi_k \boldsymbol{x}_j)^T)}{\sum_{l=1}^{n} \exp(\Phi_q \boldsymbol{x}_i \cdot (\Phi_k \boldsymbol{x}_l)^T)} \tag{2.16}$$

This expression can be understood as a soft look-up table where the terms $\Phi_q \boldsymbol{x}_i$ and $\Phi_k \boldsymbol{x}_j$ represent the query and key vectors, respectively, obtained by applying linear transformations Φ_q and Φ_k to the inputs. The dot product $\Phi_q \boldsymbol{x}_i \cdot (\Phi_k \boldsymbol{x}_j)^T$ measures the similarity between the query and key.

To efficiently compute self-attention for all inputs simultaneously, we leverage matrix operations:

$$\text{sa}_{QKV}(X) = \text{softmax}\left(\frac{XQ(XK)^T}{\sqrt{d_k}}\right) XV \tag{2.17}$$

where X is the matrix with inputs i as rows, and Q, K, and V are parameter matrices for queries, keys, and values, respectively. The factor $\sqrt{d_k}$ is used for scaling the dot products to control the gradient's variance during training.

In practice, several attention matrices are learned simultaneously to aggregate different relevant dependencies in the data. **Multi-head attention** computes l independent attention functions (or "heads") $T_i(\cdot)$ and then concatenates their outputs:

$$\text{Multi-Head}(X) = \text{Concat}(T_1, \ldots, T_l) W_o \tag{2.18}$$

$$T_i(X) = \text{softmax}\left(\frac{XQ_i(XK_i)^T}{\sqrt{d}}\right)XV_i \tag{2.19}$$

W_o is a linear transformation applied to the concatenated outputs, integrating information across heads. This approach enables the model to capture a richer representation of the input data.

Residual connections, feedforward layers, and layer normalization

Besides the attention mechanism, transformer layers incorporate more components for improved training stability and enhanced testing performance. In particular, *layer normalization* standardizes the activations within a layer, reducing training time and improving the model's generalization. *Residual connection*, corresponding to bypass connections in the network architecture that enable gradient flow directly across layers, improving training efficiency and preventing gradient vanishing during SGD optimization. To complete the transformer layer function, an additional MLP$(\cdot)$ layer is included to capture more complex dependencies within the data, along with an extra residual connection and layer normalization. Formally, these operations are expressed as:

$$\boldsymbol{z} = \text{LayerNorm}(\boldsymbol{x} + \text{MultiHeadAttention}(\boldsymbol{x})),$$

$$\boldsymbol{z} = \text{LayerNorm}(\boldsymbol{z} + \text{MLP}(\boldsymbol{z})).$$

These components of the transformer layer function are stacked together. In practice, the most used architecture, popularized by [201], consists of 12 encoder layers and 12 attention heads.

Positional encodings

Even with attention, the previously described transformer encoder layers lack an inherent understanding of sequence order. This means that we could permute the input elements (*tokens*), and that would be irrelevant to the model. If sequence order is important for our applications, as it usually is, we need to provide positional information to the model. For this, sinusoidal functions are used to encode a continuous pattern that helps the model discern the relative positions of the input elements. We represent these positional embeddings as a position matrix $P \in \mathbb{R}^{n \times d}$ calculated as:

$$P(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{2.20}$$

$$P(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{2.21}$$

The oscillations allow the model to understand sequential order without introducing discontinuities. Alternatively, positional embeddings can be learned as part of the model's parameters. In either case, positional encoding is the first component of the transformer model, and the input is encoded before being fed through the stack of encoder layers.

Note that we have briefly introduced the encoder layers of the model, as these are the only ones used in this dissertation. For a more comprehensive description of an encoder-decoder transformer model, see [201].

2.3.3 Optimization

In machine learning, parameter estimation is a pivotal goal, where we aim to identify the optimal parameters $\boldsymbol{\theta} \in \Theta$ that minimize or maximize an objective function, such as a loss function $\mathcal{L}(\theta)$ or a reward function $\gamma(\theta)$, respectively [143]. The parameter space Θ is typically assumed to be continuous, lying within $\mathbb{R}^D$, with D representing the dimensionality of the parameters. Optimization methods commonly rely on first-order derivatives of the objective function to determine "downhill" directions, albeit without considering the curvature information:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta_t \mathbf{d}_t \tag{2.22}$$

In this equation, η_t denotes the learning rate or step size, and $\mathbf{d}_t$ represents a descent direction, typically the negative gradient, given by $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_t}$. Stochastic optimization is often used to optimize the expected value of the function:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x})}[\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x})] \tag{2.23}$$

This approach involves using random training examples x from the training set. With the assumption that the distribution q is independent of the parameters, unbiased gradient estimates can utilize $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}_t(\boldsymbol{\theta}_t)$, leading to the stochastic gradient descent (SGD) method that converges to a stationary point where the gradient equals zero.

Many methods have been proposed to accelerate the optimization process. One such method is the momentum approach, which accelerates the optimization process by pushing updates toward consistently favourable directions while mitigating oscillations in the directions where the gradient changes abruptly, much like a rolling ball gaining momentum on a downhill. The *AdaGrad* [50] method adapts learning rates for each parameter

to accommodate the sparsity of gradients, improving optimization for convex functions. Although this method dynamically adjusts learning rates, the effective learning rate may decrease significantly over time due to the cumulative nature of the denominator term. The *RMSProp* method addresses this by employing an exponentially weighted moving average of squared gradients. Finally, the *Adam* optimizer [101] combines the concepts of momentum and adaptive gradients, making it an efficient choice for large-scale and quick-converging optimization tasks in machine learning. It calculates moving averages of gradients and squared gradients, adjusting updates accordingly.

The integration of momentum and adaptive learning rates into the Adam optimizer positions it as a preferred method for various machine learning optimization challenges. It balances the benefits of SGD and advanced heuristics for improved optimization efficiency.

2.3.4 Beyond supervised machine learning

Labelled data fuel supervised machine learning. Nevertheless, even without labelled training samples, we can still uncover useful patterns within the data. For example, unsupervised *representation learning* focuses on training a network to produce as output meaningful intermediate representations $\boldsymbol{z}_l$ that could potentially be used for subsequent analyses. These trained models, or *feature extractors*, can generate representations of novel data points without categorizing them into predefined classes. If a new model is later trained on the representations in a supervised way, this approach is called semi-supervised learning. Another example of how to make use of unlabelled data is *clustering*, which consists of the categorization of the training data points based on their similarity. In this section, we introduce foundational unsupervised learning techniques, including autoencoders and variational autoencoders. This lays the groundwork for more advanced methodologies like self-labelling, contrastive methods, and transformer model-based semi-supervised learning, which will be detailed in subsequent chapters.

Autoencoders

Autoencoders, a class of deep neural networks, are designed to learn compressed representations of data through two primary components: an encoder $f(\boldsymbol{x};\boldsymbol{\theta})$ that maps an input $\boldsymbol{x}$ to a representation $\boldsymbol{z} = f(\boldsymbol{x})$ in a lower dimensional space, and a decoder g_φ that aims to reconstruct the input from this representation, yielding $\hat{\boldsymbol{x}} = g(\boldsymbol{z})$. The encoder and decoder can either be MLPs or CNNs and are trained simultaneously to minimize a reconstruction loss $\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}})$, ideally preserving significant properties of the original data within a compact lower-dimensional space. Autoencoders find applications in denoising and dimensionality

reduction, showcasing their utility in extracting and preserving salient features from the data.

The latent space learned by autoencoders may lack continuity, meaning that arbitrary points in the latent space do not necessarily translate back into plausible data points. Generative models, particularly Variational Autoencoders (VAE) [102], address this by ensuring that the latent space is continuous and structured, enhancing the model's ability to generate new data instances. VAEs model the encoder and decoder as conditional distributions, with the encoder approximating a simple, fixed distribution (e.g., a Gaussian with mean zero and unit variance). The decoder, or generator, samples from this distribution to reconstruct the original training samples with high fidelity. The loss function for VAEs combines a reconstruction term with a Kullback-Leibler divergence term (see Section 2.2), encouraging the latent space to adhere to the prior distribution:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] + D_{\text{KL}}\left(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right). \tag{2.24}$$

Clustering

Clustering aims to partition data into groups of "similar" items without predefined labels. This is especially relevant in bioinformatics, as clustering can be used to identify DNA or RNA sequences with similar patterns. In the context of DNA, the clustering problem is formally stated as follows [64].

Let $\Sigma = \{A, C, G, T\}$ and L the length of the sequences to be clustered. The goal of the sequence clustering problem is to assign N sequences into a maximum of K clusters, assuming that K is a given parameter.

In the more specific problem of characterization of biodiversity, it is required that each cluster c corresponds to a unique species or OTU with distribution p_c. More precisely, the goal is to find a decision rule $\delta : \Sigma^L \to \{1, \ldots, K\}$ which correctly maps each DNA sequence to its respective genome bin or OTU. In contrast with the "taxonomic classification" or a more general clustering problem, the subgroups or bins will remain unlabelled, the clustering should be as fine-grained as possible, and the assessment will be done using clustering-specific metrics. Unlike general domains, one of the most challenging characteristics of this problem is the complexity of the output space, since there could be thousands of species present in a sample. Here, we present some of the earliest and more general approaches to unsupervised clustering, K-means, and GMMs. These are classic and versatile algorithms we often use as baselines throughout our work. These are parametric clustering algorithms, where the computation of the cluster label assignments for each training sample is selected from a fixed number of `n_clusters` (K), given as an extra input to the models.

K**-means** algorithm seeks to minimize the sum of squared distances between data points and their assigned cluster centroids. Its objective function can formally express this:

$$\mathcal{L} = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \left\| \boldsymbol{x}_j^{(i)} - \boldsymbol{\mu}_i \right\|^2, \tag{2.25}$$

where K is the number of clusters, n_i is the number of points in cluster i, $\boldsymbol{x}_j^{(i)}$ is the jth point in cluster i, and $\boldsymbol{\mu}_i$ is the centroid of cluster i. The algorithm iteratively updates cluster assignments and centroids until convergence, aiming for a partition that minimizes within-cluster variances.

Gaussian Mixture Model (GMM) is a probabilistic model that assumes data points are generated from a mixture of several Gaussian distributions, each representing a cluster. The model's parameters are optimized using the Expectation-Maximization algorithm [131], which iteratively updates the probabilities of cluster assignments and the Gaussian parameters to maximize the likelihood of the data. The GMM can accommodate clusters of different sizes and shapes, making it more flexible than K-means in capturing the complexity of data distributions, with the caveat that it can be more computationally expensive.

Through these unsupervised learning methodologies, we can discern meaningful patterns and structures in datasets where direct annotations or labels are unavailable.

Clustering evaluation metrics

Given the available information at test time, clustering results can be evaluated *post hoc* using external or internal validation methods. The key distinction is that external methods evaluate performance against the ground truth of the data, and internal methods do not use any ground truth and measure other data properties instead. We focus on external evaluation methods, as achieving agreement with the ground truth (taxonomy) is ultimately one of the main goals of this research.

For a DNA sequence dataset $\mathcal{S} = \{s_n\}_{n=1}^{N}$, we say that a clustering partition of $\mathcal{S}$, $\pi_C = \{C_1, \ldots, C_R\}$, represents a set of R clusters such that $\pi_C(s_i) = c_i$ corresponds to the clustering assignment of sequence s_i, and each $C_r = \{s \in \mathcal{S} \mid \pi_C(s) = r\}$ corresponds to the set of all sequences assigned to cluster r. The true labelling partition $\pi_L = \{L_1, \ldots L_K\}$ of $\mathcal{S}$, represents a set of K clusters, such that $\pi_L(s_i) = l_i$ corresponds to the true label of sequence s_i and $L_k = \{s \in \mathcal{S} \mid \pi_L(s) = k\}$ is the set of all sequences with the true label k.

Based on this notation, we consider the following clustering external measures:

- **Unsupervised Clustering Accuracy**: Proposed in [214], this metric is considered to be the primary evaluation criteria in our analysis as it is based on the computation of the optimal mapping ξ between numerical cluster label assignments and true taxonomic labels. Formally, this metric is defined as:

$$ACC = \max_{\xi} \frac{\sum_{i=1}^{N} \mathbf{1}[l_i = \xi(c_i)]}{N}, \tag{2.26}$$

where $\mathbf{1}[\text{condition}] \in \{0, 1\}$ is an indicator function equal to 1 if and only if the condition is true, and $\xi(c_i)$ denotes the true taxonomic label assigned by ξ. The mapping ξ ranges over all possible one-to-one mappings between clusters and labels and can be calculated, for example, by the Hungarian algorithm [105]. A value of $ACC = 1$ stands for a perfect match, and $ACC = 0$ indicates that all samples were wrongly assigned. A larger value is correlated with a better matching with the ground truth.

- **Homogeneity**: Proposed in [168], it measures the extent to which each cluster contains only samples belonging to a single class. It is defined as:

$$\text{Homogeneity}(\pi_L, \pi_C) = \begin{cases} 1 & \text{if } H(\pi_L, \pi_C) = 0 \\ 1 - \frac{H(\pi_L|\pi_C)}{H(\pi_L)} & \text{otherwise} \end{cases}$$

$H(\pi_L|\pi_C)$ is the conditional entropy of the class distribution given by the clustering partition, and $H(\pi_L)$ is the entropy of the true class labels. The entropies are calculated as:

$$H(\pi_L|\pi_C) = -\sum_{C \in \pi_C} \sum_{L \in \pi_L} \frac{|L \cap C|}{N} \log \frac{|L \cap C|}{|C|}$$

$$H(\pi_L) = -\sum_{L \in \pi_L} \frac{|L|}{N} \log \frac{|L|}{N} \tag{2.27}$$

The homogeneity score ranges between 0 and 1, with 1 indicating perfect homogeneity.

- **Completeness**: Proposed in [168], it measures whether or not all data points that belong to a given class are assigned to the same cluster. In other words, a clustering result satisfies completeness if all data points from a single class are assigned to a single cluster. Formally:

$$\text{Completeness}(\pi_L, \pi_C) = \begin{cases} 1 & \text{if } H(\pi_C, \pi_L) = 0 \\ 1 - \frac{H(\pi_C|\pi_L)}{H(\pi_C)} & \text{otherwise} \end{cases}$$

The entropies are calculated as:

$$H(\pi_L|\pi_C) = -\sum_{C\in\pi_C}\sum_{L\in\pi_L}\frac{|L\cap C|}{N}\log\frac{|L\cap C|}{|L|}$$

$$H(\pi_C) = -\sum_{c\in\pi_C}\frac{|C|}{N}\log\frac{|C|}{N} \tag{2.28}$$

The completeness score ranges from 0 to 1, with higher values indicating better clustering performance.

- **Normalized Mutual Information (NMI)**: Measures the amount of overlap between the clustering partition π_C and the ground truth partition π_L. We define the probability that a random element in $\mathcal{S}$ whose true label is l gets assigned to cluster C as $P(C,L) = \frac{|C\cap L|}{N}$. Similarly, we define the probability that a random element gets assigned to C or has a label L independently as $P(C) = \frac{|C|}{N}$ and $P(L) = \frac{|L|}{N}$, respectively. These definitions allow the computation of the mutual information between the two labellings according to equation 2.4:

$$MI(\pi_C,\pi_C) = -\sum_{C\in\pi_C}\sum_{L\in\pi_L}p(C,L)\log\frac{P(C,L)}{P(C)P(L)} \tag{2.29}$$

The value determined by equation 2.29 is between 0 and $\min\{H(\pi_C), H(\pi_L)\}$. That means that it is possible to achieve the maximum value with a partition with multiple small clusters. For this reason, the normalized version of the metric is preferred.

$$NMI(\pi_C,\pi_L) = \frac{2MI(\pi_C,\pi_L)}{H(\pi_C)+H(\pi_L)} \tag{2.30}$$

The value determined by equation 2.30 is between 0, indicating no mutual information, and 1 indicating a perfect correlation between the clustering and the ground truth partitions.

- **Adjusted Rand Index (ARI)**: Also measures the agreement between two dataset partitions. Unlike previous information-theoretic metrics, ARI is a combinatorial measure focusing on the correct or incorrect assignment of sample pairs to the same or different clusters. It is given by the equation:

$$RI = \frac{TP+TN}{\binom{N}{2}}, \tag{2.31}$$

where TP (True Positives) represents the count of sample pairs correctly placed in the same category both by the clustering partition π_C and the ground truth partition π_L; TN (True Negatives) denotes the count of sample pairs correctly placed in different categories by both π_C and π_L. The sum of these values is divided by the total number of pairs of samples in the dataset. The formula in equation 2.31 measures the proportion of correct decisions made by the clustering partition; however, its lowest possible value of zero rarely occurs in practice. The Adjusted Rand Index (ARI) [161] is introduced, applying normalization to account for randomness:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \tag{2.32}$$

where $\mathbb{E}[RI]$ is the expected RI under random chance. This adjustment ensures that ARI has a value near 0 for random label assignments, irrespective of the number of clusters or sample size, and a value of exactly 1 for perfectly matching clustering. ARI can also adopt values below 0, indicating clustering agreements worse than random and with a lower bound of -0.5.

Chapter 3

Alignment-free neural information-based clustering of DNA sequences

This chapter is an adaptation and extension of the paper "DeLUCS: Deep Unsupervised Clustering of DNA Sequences." [134], of which I was first co-author. Here, we explore a novel methodology for deep clustering of DNA sequences.
Section 3.1 describes the related work. Particularly, Section 3.1.1 explores the evolution of general machine learning-based methods in genomics, and Section 3.1.2 describes new neural methodologies and motivates their adaptation to genomics. In Section 3.2, the core aspects of the methodology for the adaptation of unsupervised information-based frameworks to genomics are given. The first key component is to generate appropriate data augmentations, as detailed in Section 3.2.1. The second major component involves training a neural network by enforcing the mutual predictability of the original sequences' cluster assignments and the assignments of their corresponding augmentations. Different approaches to this procedure are described in Section 3.2.2. The last step aggregates the predictions of various independently trained networks to reduce the variance and boost overall performance, as detailed in Section 3.2.3. Section 3.3 describes the experimental setup we followed, including the compilation of datasets and the training details.
Section 3.4 contains updated results obtained after following the proposed methodology, and Section 3.5 discusses the proposed method's strengths and limitations.

3.1 Related work

3.1.1 DNA sequence classification and clustering

We have discussed that with the advent of NGS technologies, many innovative machine learning-based taxonomic classifiers have emerged, sometimes by adopting successful methodologies in other fields such as computer vision or natural language processing. These methodologies have demonstrated remarkable success in classifying DNA sequences, matching or surpassing traditional alignment-based techniques in various applications, including whole-genome phylogenies [7, 162], microbial community profiling [112, 116] and general taxonomic classification [52, 184, 203, 219]. Despite their applicability and utility, supervised machine learning algorithms are limited by their dependence on the stability of taxonomic labels, as any errors in the "ground truth" can be perpetuated in subsequent classifications. Moreover, despite the great utility of expert annotations, they suffer from instability due to occasional inaccuracies and temporary assignments, particularly in cases with limited information or characterization (e.g., [6, 158, 182]). This issue is compounded by the occasional absence of a definitive "ground truth" in taxonomic labelling, leading to classification disputes. For example, the field of microbial taxonomy has undergone considerable changes, most notably through the recent efforts of the Genome Taxonomy Database (GTDB) [29, 157], in which, for example, over 32,000 genomes had their species names updated in the last release [156] with respect to the corresponding National Center for Biotechnology Information (NCBI) taxonomy. Considering the challenges presented by the labour-intensive and time-consuming task of assigning taxonomic labels and biological annotations to raw sequences, a significant bottleneck in the field, as highlighted by [133], our approach pivoted towards unsupervised machine learning. This strategic shift is further justified by the advancements in sequence acquisition techniques, which require more precise and efficient classification methodologies.

Unsupervised learning, operating on unlabelled sequences, has the potential to infer patterns from the data without the biases of pre-existing labels. It avoids propagating labelling errors and can categorize novel sequence types by dynamically forming new clusters. However, clustering large datasets using unsupervised learning is a challenging problem, and the progress in using unsupervised learning for the clustering of genomic sequences has not been as rapid as that of its supervised classification counterparts [96]. Previous efforts mostly focused on applying generic algorithms like K-means or GMMs to various numerical representations of DNA sequences. Several studies have explored K-means clustering with different DNA sequence representations [4, 11, 25, 88, 89]. Other approaches have employed digital signal processing techniques [3, 77, 130]. Despite their versatility, these methods face limitations in high-dimensional spaces. For instance, K-means assumes

spherical clusters and often struggles with the "curse of dimensionality," where both the increase in the number of dimensions and in the number of samples lead to a significant rise in computational complexity and a decrease in the distinctiveness of nearest neighbours. GMM, while more flexible regarding data distribution, encounters difficulties in parameter estimation and convergence in high-dimensional spaces, often requiring dimensionality reduction techniques for effective clustering [2, 19, 63]. These challenges have prompted the exploration of advanced methodologies for speeding up classic non-parametric algorithms for general clustering [60, 89] or, more specifically, in the field of metagenomic binning, where deep-learning-based approaches [150, 207, 220] have begun to show promise in clustering metagenomic fragments, underscoring the potential of neural-based clustering methods.

3.1.2 Neural unsupervised clustering

The general application of neural networks in unsupervised learning tasks dates back to the origins of neural networks themselves [73, 104]. Although applications were not widely adopted until the last decade, recent advancements have highlighted their potential.

Self-labelling approaches

One of the earliest and most notable approaches to deep learning-based clustering was proposed by [214]. The method, termed Deep Embedded Clustering (DEC), simultaneously learns the parameters of a neural encoder that maps feature data points $\boldsymbol{x}_i \in \mathcal{X}$ into a lower dimensional embedding space $\mathcal{Z}$, and a set of cluster centroids $\{\boldsymbol{\mu}_j\}_{j=1}^{k}$ in the embedding space [214]. DEC iteratively optimizes a clustering objective based on soft K-means assignments and the KL divergence between them and an auxiliary target distribution. Each soft assignment is computed using the Student's t-distribution as a kernel to measure the similarity between an embedding point $\boldsymbol{z}_i$ and a centroid $\boldsymbol{\mu}_j$:

$$q_{ij} = \frac{\left(1 + \left\|\boldsymbol{z}_i - \boldsymbol{\mu}_j\right\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'} \left(1 + \left\|\boldsymbol{z}_i - \boldsymbol{\mu}_{j'}\right\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}} \tag{3.1}$$

where $z_i = f(\boldsymbol{x}_i; \boldsymbol{\theta}) \in \mathcal{Z}$ corresponds to $\boldsymbol{x}_i \in \mathcal{X}$ after the embedding, α are the degrees of freedom of the Student's t-distribution and q_{ij} can be interpreted as the probability of assigning sample i to cluster j (i.e., a soft assignment). The training objective is the KL divergence between the soft assignments q_i and an auxiliary distribution p_i. This auxiliary distribution p_i corresponds to a soft refinement of q_i that puts more weight into highly confident predictions and normalizes the contribution of each centroid to prevent large

clusters from distorting the embedding space. Formally, the parameters of the distribution are calculated as:

$$p_{ij} = \frac{\tilde{q}_{ij}}{\sum_{j'} \tilde{q}_{ij'}}, \tag{3.2}$$

where $\tilde{q}_{ij} = \frac{q_{ij}^2}{\sum_i q_{ij}}$. This method inspired a new class of deep learning-based self-learning methods that attempted to refine their performance based on high confidence predictions [115, 199].

Information based techniques

Neural information-based clustering is rooted in two fundamental principles: fairness [23] (sometimes referred to as balance [62, 83]) and decisiveness [23] (or separation [62, 83]). A robust clusterer should assign samples to all available clusters, avoiding collapse to trivial solutions and exhibiting certainty in its decisions for all samples. The previous desiderata can be achieved by maximizing the mutual information between the input and the output of a classifier, assuming its output are soft labels [23].

Given a random variable $\mathbf{x}$ with sample space $\mathcal{X}$ and distribution $p(\boldsymbol{x})$ and its corresponding class assignment y with distribution $p(y)$ and sample space $\mathcal{Y}$, their mutual information can be calculated combining equations 2.8 and 2.9 as:

$$MI(\mathrm{y}, \mathbf{x}) = \iint p(y, \boldsymbol{x}) \log \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})p(y)} \, dy \, d\boldsymbol{x} \tag{3.3}$$

By applying Bayes' theorem, the previous expression can be rewritten as:

$$MI(\mathrm{y}, \mathbf{x}) = \iint p(y|\boldsymbol{x})p(\boldsymbol{x}) \log \frac{p(y|\boldsymbol{x})}{p(y)} \, dy \, d\boldsymbol{x} \tag{3.4}$$

Considering that y is a discrete random variable, taking values in $\mathcal{Y} = \{1, \ldots, K\}$, the expression can be further refined as:

$$\begin{aligned} MI(\mathrm{y}, \mathbf{x}) &= \iint p(y|\boldsymbol{x})p(\boldsymbol{x}) \log p(y|\boldsymbol{x}) dy dx - \iint p(y|\boldsymbol{x})p(\boldsymbol{x}) \log p(y) dy d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \left[\sum_{j=1}^{K} p\left(y_j|\boldsymbol{x}\right) \log p\left(y_j|\boldsymbol{x}\right) \right] - \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \left[\sum_{j=1}^{K} p\left(y_j|\boldsymbol{x}\right) \log p\left(y_j\right) \right] \end{aligned} \tag{3.5}$$

Here, K is the number of clusters in the categorical distribution for y, $\boldsymbol{x}_i$ are the sampled points from the distribution $p(\boldsymbol{x})$, and y_j represents each cluster in the categorical distribution of y. The inner sum over j (from 1 to K) accounts for the contribution of each category in the distribution of y for a given $\boldsymbol{x}_i$. Assuming that N is the number of samples drawn from $p(\boldsymbol{x})$, the expression can be approximated using Montecarlo integration:

$$\begin{aligned} MI(\mathrm{y}, \mathbf{x}) &= \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} p\left(y_j|\boldsymbol{x}_i\right) \log p\left(y_j|\boldsymbol{x}_i\right) - \sum_{j=1}^{K} \log p\left(y_i\right) \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}\left[p\left(y_j|\boldsymbol{x}\right)\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} p\left(y_j|\boldsymbol{x}_i\right) \log p\left(y_j|\boldsymbol{x}_i\right) - \sum_{j=1}^{K} \log \left(\frac{1}{N} \sum_{i=1}^{N} p\left(y_i|\boldsymbol{x}_i\right) \right) \cdot \frac{1}{N} \sum_{i=1}^{N} p\left(y_j|\boldsymbol{x}_i\right) \\ &= -\overline{H\left(\sigma_\theta^i\right)} + H\left(\overline{\sigma_\theta^i}\right) \end{aligned} \tag{3.6}$$

Equation 3.6 demonstrates that the mutual information between the inputs and corresponding labels can be estimated by computing the difference between the entropy of the average of the outputs and their average entropy. Both averages are computed over the training set (or mini-batch in practice). It is worth noting that $H(\overline{\sigma})$ reaches its maximum value when the clusters are assigned an approximately equal number of samples (fairness), and $\overline{H(\sigma)}$ reaches its minimum value when the model is confident in all its predictions (decisiveness). Therefore, finding the parameters that maximize the mutual information between the input and the assigned labels is equivalent to determining the parameters of a discriminative classifier that satisfies both fairness and decisiveness. This can be accomplished using stochastic gradient descent and the negative of the expression in equation 3.6 as the loss function. This method is also known as entropy-based clustering.

In practice, optimization of this function is not sufficient to escape from degenerate solutions (all data points assigned to the same cluster). Several successful techniques based on different regularization criteria have been proposed to improve the performance of entropy-based clustering. In particular, one of the most successful strategies is enforcing consistency of the representations of multiple views of a given training sample [62, 83]. Some of these strategies are specific to clustering and will be explored later in this chapter in the context of clustering DNA sequences. The next chapter will detail a strategy related to self-supervised representation learning.

3.2 DeLUCS: Deep Learning for Unsupervised Clustering of DNA Sequences

This section builds upon the previous discussion on the evolution of classification and clustering methodologies for genomic signatures inspired by advances in other fields. We focus on adapting neural approaches for effective clustering of DNA sequences, emphasizing information-based techniques over self-labelling strategies due to their superior performance, versatility, and scalability, as will be later discussed.

The general pipeline of DeLUCS, illustrated in Figure 3.1, consists of three main steps:

1. Each DNA sequence in the dataset is augmented with several artificial *mimic sequences*, presumed to be in the same cluster. These mimics are generated through a probabilistic model employing transitions and transversions. Subsequently, *k-mer* counts for the original, and its corresponding mimic sequences are computed to produce the k-mer feature vectors.

2. Pairs of feature vectors are used to train multiple ANNs independently. This training employs an information-based loss function, focusing on maximizing the mutual predictability of cluster assignments for each training pair.

3. Due to the observed high variance in the training outcomes of the ANNs, a majority voting mechanism is implemented. This approach aggregates the results from step 2, assigning each sequence to a final cluster based on the consensus among the various ANNs.

DeLUCS leverages normalized k-mer frequency vectors as input features for all computational experiments, specifically opting for $k = 6$. This choice is empirically determined to strike an optimal balance between clustering performance and computational efficiency.

3.2.1 Mimic sequences: Data augmentations for learning taxonomic information from DNA sequences

Regularization is critical in improving neural information-based clustering and preventing model collapse. The most effective approach, as supported by the literature [62, 83, 115], involves enforcing consistency in representations across augmented versions of input samples. While popular in computer vision, this approach presents unique challenges in genomics due to the complexity of defining suitable augmentation schemes.

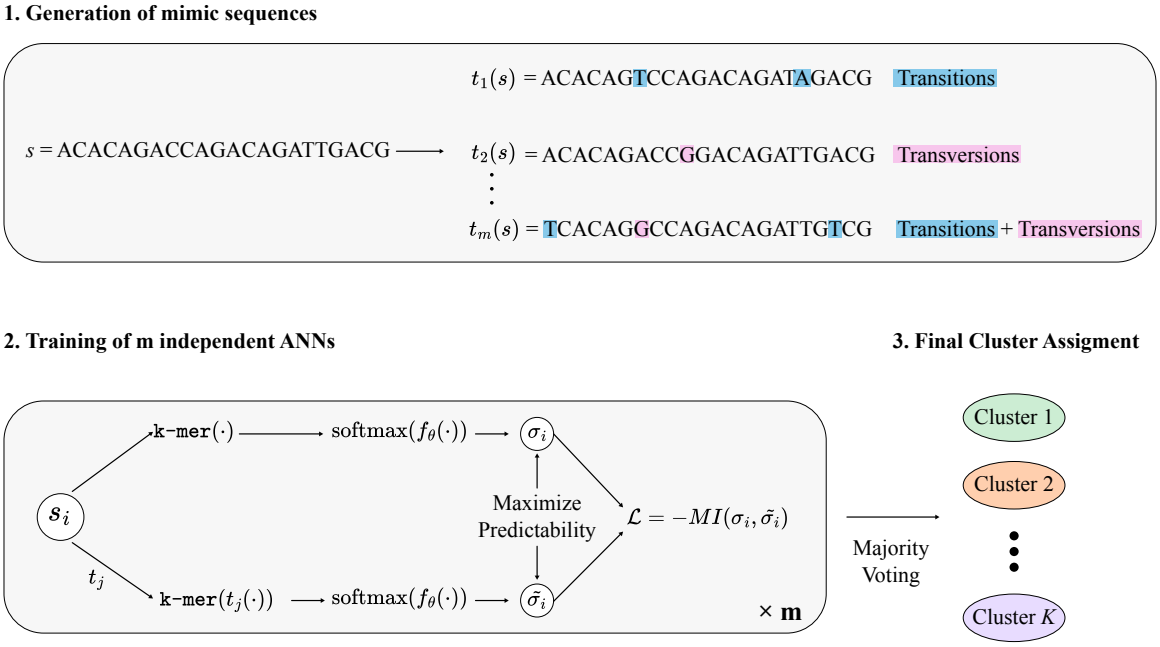


Figure 3.1: Overview of the DeLUCS pipeline. The process begins with the original DNA sequences intended for clustering. Step 1 generates artificial mimic sequences from the original sequences using a probabilistic model (t_j) of transitions and transversions. In step 2, normalized k-mer frequency vectors for all original and mimic sequences are calculated. Then, m independent neural networks $f(\boldsymbol{x}; \boldsymbol{\theta})$ are trained, guided by an information-based loss function enforcing the consistency of the network predictions for a sequence and its mimic. Finally, step 3 employs majority voting to finalize each sequence's cluster assignment.

In general, given a data set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the goal is to construct the auxiliary training set of paired data

$$\mathbb{A} = \left\{ \left(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1^1\right), \left(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1^2\right), \left(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1^3\right), \ldots, \left(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i^m\right) \mid 1 \leq i \leq n \right\},$$

where the data points in each pair $(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i^j)$ are considered similar according to some criteria based on prior knowledge, e.g., invariance to distortions or spatial proximity. Each original sample $\boldsymbol{x}_i$ corresponds to a true sample in the original dataset, and $\tilde{\boldsymbol{x}}_i^j$ corresponds to an augmentation that may not be present in the original dataset.

In the particular context of DNA sequences, the proposed data augmentations correspond to creating m artificial mimic sequences per original sequence using a probabilistic model based on transitions and transversions while preserving the original sequences. For each

sequence s_i, $1 \leq i \leq N$, we use a simple probabilistic model $t_j(\cdot)$ based on DNA substitution mutations (transitions and transversions) to produce different mimic sequences, as follows. Given a sequence s_i and a particular position ι in the sequence, we fixed independent transition and transversion probabilities $p_{ts}[\iota]$ and $p_{tv}[\iota]$ respectively. Next, we produce the following mimic sequences, probabilistically: $t_1(s)$ with only transitions, $t_2(s)$ with only transversions, and $t_j(s)$ with both transitions and transversions, for all $3 \leq j \leq m$. The parameter m is determined for each experiment based on the particulars of its dataset. Its default value is 3 to account for the use of the two individual substitution mutations and their combination. Still, it may have to be increased if the number of available sequences per cluster is insufficient to obtain a high classification accuracy.

After computing the sequence mimics, each original sequence and its corresponding mimics are converted into a numerical vector containing the counts of all of its k-mers, where a k-mer is defined as a subsequence of length k that does not contain the symbol `N`. Finally, each k-mer count vector is converted into a k-mer frequency vector by dividing its k-mer counts by the total length of the sequence minus the number of `N` symbols in the original DNA sequence. This operation is represented as $\boldsymbol{x}_s = \texttt{k-mer}(s)$, with the subindex usually omitted to lighten the notation.

Although the use of transition and transversion probabilities for generating mimic sequences is biologically inspired it is applied here as a mathematical tool, focusing on creating minimally divergent sequences through random base substitutions. The selected probabilities are empirically determined to optimize classification accuracy, and we consider the prior that transitions are twice as frequent in nature than transversions [99]. While these rates draw from biological concepts, they do not claim biological precision due to inherent variability in mutation rates across different regions, species, and estimation methods [5, 144, 172]. In practice, selecting species-specific mutation rates is not feasible without taxonomic labels.

3.2.2 Leveraging data augmentations to enhance information-based loss functions

The information bottleneck principle [193, 194], which is part of the information-theoretic approach to clustering, suggests that effective clustering should capture relevant semantic information from the input while discarding irrelevant information. This objective can be realized through the explicit maximization of the mutual information between the input and the discrete output probability distribution, which, as we have discussed in section 3.1.2, is approximated by equation 3.6. If $\tilde{\mathbf{x}}$ corresponds to a random variable, sampled from the space of augmentations $\tilde{\mathcal{X}}$, a regularization term can be added to enforce the

consistency of the assignments for several views of a given sample. The regularized loss function becomes

$$\mathcal{L} = -MI(\mathrm{y}|\mathbf{x}) + \mathcal{R}(\mathbf{x}, \tilde{\mathbf{x}}). \tag{3.7}$$

Here, $\mathcal{R}$ penalizes dissimilarity in representations between original and augmented data points, using penalty measures like KL divergence or cross-entropy [83].

An alternative formulation, Invariant Information Clustering (IIC), introduced by [91] for computer vision tasks, also aims to learn from paired data, *i.e.*, from pairs of samples $(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \in \mathcal{X} \times \tilde{\mathcal{X}}$ drawn from a joint probability distribution $p(\boldsymbol{x}, \tilde{\boldsymbol{x}})$. This formulation does not directly maximize the mutual information between input and output random variables. Instead, artificial copies $\tilde{\boldsymbol{x}}$ of each true sample $\boldsymbol{x}$ are created, and it aims to learn a mapping $\Phi(\boldsymbol{x}; \boldsymbol{\theta})$ of a discriminative classifier that retains commonalities between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ while discarding information that is not relevant for categorization. The resulting space $\mathcal{Y} = \Phi(\mathcal{X})$ is then a compressed representation space that encodes the semantic clusters present in $\mathcal{X}$.

The mapping Φ can be implemented by an ANN $f(\boldsymbol{x}, \boldsymbol{\theta})$ and a softmax output layer, $\Phi(\boldsymbol{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}(x))$. For a dataset with K expected clusters, we define $\boldsymbol{\sigma} = \Phi(\boldsymbol{x}) \in [0, 1]^K$ as the distribution of a discrete random variable z over the K clusters for each sample $\boldsymbol{x}$. Specifically, $\sigma_c = P(z = c|\boldsymbol{x})$ represents the probability that sample $\boldsymbol{x}$ is assigned to cluster c. It is possible to maximize the predictability of z_i from $\tilde{z}_i = \Phi(\tilde{\boldsymbol{x}}_i)$ across all samples, by maximizing the mutual information $MI(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}})$. This can be estimated using the alternative definition in equation 2.9:

$$MI(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}) = D_{KL}(P(z, \tilde{z})||P(z)P(\tilde{z})). \tag{3.8}$$

Here, $P(z)$, $P(\tilde{z})$ are the marginal distributions, and $P(z, \tilde{z})$ is the discrete joint probability distribution. The mutual information is calculated as follows:

$$MI(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}})_\lambda = \sum_{j=1}^{K} \sum_{k=1}^{K} P(z_i = j, \tilde{z}_i = k|\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \log \frac{P(z_i = j, \tilde{z}_i = k|\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)}{[P(z_i = j)P(\tilde{z}_i = k)]^\lambda} \tag{3.9}$$

The joint probability is calculated assuming independence and symmetrized to prevent clustering collapse. During neural network training, we minimize the negative weighted mutual information. From equation 2.4, the loss function derived from this formulation, is thus:

$$\mathcal{L}(\mathrm{x}, \tilde{\mathrm{x}}) = -MI(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}})_\lambda \approx -(2\lambda - 1)H(\boldsymbol{\sigma}) + H(\boldsymbol{\sigma} \mid \tilde{\boldsymbol{\sigma}}) \tag{3.10}$$

The hyper-parameter $\lambda \geq 1$ is introduced to weigh the contribution of the entropy term in equation 2.4. Note that the loss function derived from this alternative formulation

aligns with the original design principles of the information-based theory for discriminative clustering described in Section 3.1.2. The entropy term $H(\boldsymbol{\sigma})$ in equation 3.10 measures the amount of randomness present in the network's output; this value is maximized when all clusters are assigned the same number of samples (fairness). The conditional entropy term $H(\boldsymbol{\sigma} \mid \tilde{\boldsymbol{\sigma}})$ in equation 3.10 measures the uncertainty in the original sample $\boldsymbol{x}$, given its counterpart $\tilde{\boldsymbol{x}}$. This uncertainty must be minimized since the original sample $\boldsymbol{x}$ should be perfectly predictable from $\tilde{\boldsymbol{x}}$ (decisiveness).

3.2.3 Determination of final assignment: Majority voting

During training, since a definitive ground truth is not available, it becomes crucial to standardize the outputs from various neural network models, especially since labels may undergo permutations across different training runs. To ensure consistent labelling, our approach uses the initial set of predictions as a reference and applies the Hungarian algorithm [105] to match and adjust subsequent model labels to this reference. This creates a standardized label assignment across models. Once labels are aligned, we take the mode of the predictions for each sample in the dataset to aggregate final predictions. This approach also mitigates outliers and reduces overall variance, making our predictions more robust even in the absence of a ground truth.

Although the Hungarian algorithm is effective for label alignment, exploring alternatives like ensemble learning methods could enhance performance. These alternatives, offering potentially more refined alignment or aggregation, will be discussed in the next chapter, examining their efficacy in improving prediction accuracy without ground truth.

3.3 Experimental setup

3.3.1 Datasets

Three types of datasets were used in our proof-of-concept experiments: mitochondrial genomes, bacterial genome segments, and viral genes. The datasets, detailed in Table 3.1, Table 3.2, and Table 3.3, were sourced from public databases.

Mitochondrial genomes: We analyzed vertebrate mitochondrial genomes, selecting 13,300 sequences from Geneious 2020.2.4, constrained to lengths between 14,000 bp and 24,500 bp. The sequences were retrieved from NCBI as of November 16, 2020; these sequences allow the assessment of DeLUCS across various taxonomic levels, focusing on

the largest cluster at each level. Being a deep entropy-based clustering method, DeLUCS performs optimally with balanced clusters, defined by size uniformity and a minimum of 20 sequences. Clusters failing to meet the minimum size were excluded, as were sequences without a sub-taxon identifier. Oversized clusters were randomly sampled to meet size constraints, resulting in test-dependent cluster compositions. We followed a decision-tree approach to direct the selection of clusters for computational Tests 1 to 6 within Vertebrata, with the sequence count varying across tests due to the min/max cluster size parameters, see Table 3.1.

For example, in Test 3, only 250 out of the total 2,723 available Ostariophysi sequences were selected due to the need to achieve cluster balance, while the min/max cluster size parameters of Test 4 allowed for 383 Ostariophysi sequences to be used. The remaining Ostariophysi sequences could not be selected in either test since most belonged to the over-represented Order Cypriniformes (2,171 available sequences). Tests 1 and 2 illustrate a different scenario: 500 of the total 7,876 Actinopterygii sequences were used in Test 1 since this cluster size was sufficient for high accuracy. In contrast, in Test 2, only 113 Actinopterygii sequences could be used due to the under-representation of class Polypteriformes (33 available sequences) and over-representation of class Neopterygii (7,715 available sequences).

Table 3.1: Details of the datasets used in computational tests 1 through 6 (full vertebrate mitochondrial genomes).

Test #	**Dataset**	**Total no.of seq.**	**Min clus. size**	**Max clus. size**	**Min. seq.len. (bp)**	**Avg. seq.len. (bp)**	**Max seq.len. (bp)**
1	Subphylum Vertebrata (Fish: 500, Amphibians: 500, Birds: 500, Mammals: 500, Reptiles: 500)	2,500	500	500	14,127	16,951	24,317
2	Class Actinopterygii (Neopterygii: 40, Polypteriformes: 33, Chondrostei: 40)	113	33	40	15,531	16,623	18,062
3	Subclass Neopterygii (Ostariophysi: 250, Clupeomorpha: 250, Elopomorpha: 226, Acanthopterygii: 250, Paracanthopterygii: 249 Protacanthopterygii: 250)	1,475	226	250	15,564	16,688	19,801
4	Superorder Ostariophysi (Cypriniformes: 130, Characiformes: 123, Siluriformes: 130)	383	123	130	15,664	16,635	17,998
5	Order Cypriniformes (Cyprinidae: 80, Cobitidae: 80, Balitoridae: 75, Nemacheilidae: 80, Xenocyprididae: 80, Acheilognathidae: 70, Gobionidae: 80)	545	70	80	16,061	16,610	17,282
6	Family Cyprinidae (Acheilognathus: 47, Acrossocheilus: 46, Carassius: 45, Labeo: 45, Microphysogobio: 35, Notropis: 26, Onychostoma: 29, Rhodeus: 28, Schizothorax: 31, Sarcocheilichthys: 45, Cyprinus: 43, Sinocyclocheilus: 27)	447	26	47	16,070	16,632	17,426

Bacterial genomic fragments: The second dataset comprises 3,200 bacterial DNA

segments from eight families across three phyla, as studied in [77]. We pursued a balanced representation of diversity, adhering to GTDB (release 95) guidelines [156] and selecting a uniform cluster size of 400 sequences per family. The selection process involved random sampling of genomes or segments, ensuring family representation within the dataset. A random selection of up to 400 species or genomes was made for families with sufficient species or genomes. Otherwise, contigs were divided into segments up to 500 kbp, creating a pool from which 400 segments were chosen. The composition of each cluster is detailed in Table 3.2.

In addition to the inter-phylum classification of bacterial sequences into families (Test 7), we assessed the performance of DeLUCS for an intra-phylum classification into families within the Proteobacteria phylum only (Test 8). The dataset for Test 8 was simply the subset of the dataset in Test 7, including only the segments from genomes in bacterial families from phylum Proteobacteria. Test 7 comprises randomly selected genome segments from bacterial families across several phyla (min. segment length 150,499 bp, average segment length 433,613 bp, max. segment length 500,000 bp). Test 8 consists of the genome segments in Test 7 that belong to phylum Proteobacteria (min. segment length 150,499 bp, average segment length 434,150 bp).

Table 3.2: Details of the datasets used in computational tests 7 and 8 containing randomly selected bacterial genome segments. The datasets in these computational tests contain 400 segments per family, each of length between 150 kbp and 500 kbp.

Test #	**Phylum**	**Family**	**No. Species**	**No. Genomes**	**Total No. of Seg.**	**Avg.No. Seg./ Genome**	**Avg. Seg.len. (bp)**
7	Spirochaetes	*Treponemataceae*	46	153	400	2.3	387,939
	Firmicutes	*Bacillaceae*	47	400	400	1	493,999
		Clostridiaceae	136	400	400	1	443,267
		Staphylococcaceae	77	400	400	1	404,889
	Proteobacteria	*Enterobacteriaceae*	379	400	400	1	446,887
		Rhodobacteriaceae	400	400	400	1	464,632
		Desulfovibrionaceae	73	99	400	2.5	359,337
		Burkholderiaceae	400	400	400	1	465,707
8	Proteobacteria	*Enterobacteriaceae*	379	400	400	1	446,887
		Rhodobacteriaceae	400	400	400	1	464,632
		Desulfovibrionaceae	73	99	400	2.5	359,337
		Burkholderiaceae	400	400	400	1	465,707

Viral genomes: The third dataset includes viral DNA sequences, with cluster size determined like that of the mitochondrial DNA datasets. Test 9 features 949 sequences of segment 6 of the Influenza A virus genome, sourced from NCBI [13]. Test 10 consists of 1,633 full Dengue virus genomes from NCBI [68], and Test 11 includes 1,562 full Hepatitis B virus genomes from the Hepatitis Virus Database [69]. Each dataset represents different

virus subtypes, with descriptions provided in Table 3.3.

Table 3.3: Details of the datasets used in computational tests 9, 10 and 11 (*Influenza virus* NA-encoding gene, *Dengue virus* full genomes, *Hepatitis B virus* full genomes).

Test #	Dataset	Total no.of seq.	Min clus. size	Max clus. size	Min. seq.len. (bp)	Avg. seq.len. (bp)	Max seq.len. (bp)
9	Influenza A (NA-encoding gene) (Subtypes H1N1: 191, H2N2: 187, H5N1: 188, H7N3: 193, H7N9: 190)	949	187	193	1,345	1,409	1,469
10	Dengue complete genomes (Subtypes 1: 409, 2: 409, 3: 408, 4: 407)	1,633	407	409	10,161	10,559	10,991
11	Hepatitis B complete genomes (Subtypes A: 258, B: 262, C: 263, D: 260, E: 261, F: 258)	1,562	258	263	3,182	3,210	3,227

3.3.2 Training details

Given the relatively smaller dataset sizes in our proof-of-concept experiments compared to typical datasets in machine learning, we observed that commonly effective deep-learning architectures for visual or natural language processing tasks were not entirely suitable for our genomic data. Consequently, we employed a simpler yet versatile architecture tailored for clustering DNA sequences. The input to the network are pairs $(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = (\texttt{k-mer}(s), \texttt{k-mer}(\tilde{s}))$ representing the k-mer frequency vectors of original DNA sequences s and their mimic sequences $\tilde{s}$. The architecture, depicted in Figure 3.2, comprises two fully connected layers, *Linear (512 neurons)* and *Linear (64 neurons)*, each one followed by a *ReLU* and a *Dropout* layer with a dropout rate of 0.5. The output layer *Linear (K_ clusters)*, where K is a numerical parameter representing the upper bound of the number of clusters, is followed by a *softmax* activation function. The ReLU layers mitigate vanishing gradient issues during SGD optimization, while the Dropout layers prevent overfitting. This is crucial in unsupervised learning to avoid degenerate solutions like assigning all samples to a single cluster. Finally, the softmax layer gives as output a K-dimensional vector $\boldsymbol{\sigma} = \Phi(\boldsymbol{x}) \in [0,1]^K$, such that $\boldsymbol{\sigma}_j$ represents the probability that an input sequence s belongs to a particular cluster j.

Note that this general architecture was designed to successfully cluster all the diverse datasets presented in this study. Nonetheless, the DeLUCS pipeline is flexible enough to accommodate other architectures, including those leveraging the two-dimensional nature of fCGR patterns, like convolutional neural networks, for specific genomic data types.

The training hyperparameters were selected for optimal performance in the mitochondrial DNA datasets. Network initialization used the Kaiming method [70] to maintain input

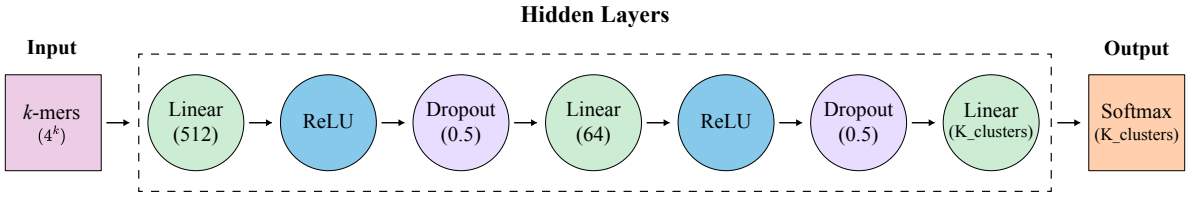


Figure 3.2: The ANN architecture used by DeLUCS. It receives k-mer frequency vectors as input. Each linear layer indicates neuron counts, except for the output layer, which is parameterized by the expected number of clusters K. The dropout rate is specified in each case, and the output is a probability distribution via the softmax function.

magnitude consistency. This is crucial for our method because poor initialization may lead to degenerate solutions, as one of the terms in the loss function becomes dominant. Training employed the Adam optimizer [101] at a learning rate of 5×10^{-5} for 150 epochs without early stopping. The batch size, set at 512, was crucial for accurate estimates of the output distribution. The hyperparameter λ was fixed at 2.5, as per equation 3.10.

3.4 Results

For each dataset, we compare the performance of DeLUCS with that of two classic clustering algorithms, the K-means algorithm and a GMM. We also consider DEC, a self-labelling, deep-learning-based algorithm (see Section 3.1.2), as our deep learning-based baseline. DeLUCS's methodology can be implemented with any information-based loss function that enforces consistency of the augmentations (*mimic sequences*). We consider the loss in equation 3.6 with a cross-entropy regularization penalty (DeLUCS$_{i/o}$), and the IIC method with the loss function in equation 3.10 (DeLUCS$_{IIC}$), referred to as DeLUCS hereafter for consistency with our published work [134]. We clustered the eleven datasets detailed in Section 3.3.1, and assessed the performance of each algorithm using metrics from Section 2.3.4, especially focusing on the ACC metric (equation 2.26) for its correspondence with taxonomic labels.

In the vertebrate mtDNA dataset (Table 3.4), our methods consistently outperformed other unsupervised algorithms, achieving mean accuracies of 87% for DeLUCS and 86% for DeLUCS$_{i/o}$, outperforming K-means, the best classical clustering algorithm, by ∼10% and DEC, our deep learning baseline, by ∼16%. This performance advantage over DEC highlights the effectiveness of our information-based approaches. The necessity of increasing

the number of mimic sequences for datasets with fewer than 150 sequences per cluster was noted to maintain classification accuracy. It is noteworthy that k-means outperforms DEC, suggesting that the DEC could also benefit from the regularization of overconfident predictions using augmentations.

Table 3.4: Performance of different clustering algorithms on the mtDNA datasets in Table 3.1, tests 1 to 6. The reported values for all the metrics: homogeneity, completeness, normalized mutual information (NMI), adjusted Rand index (ARI) and unsupervised clustering accuracy (ACC) correspond to the average over 10 runs of the algorithms.

Dataset	**Model**	**No. Mimics**	**Homogeneity**	**Completeness**	**NMI**	**ARI**	**ACC**
Vertebrata	K-means	-	0.65	0.67	0.66	0.64	0.81
	GMM	-	0.59	0.64	0.62	0.54	0.66
	DEC	-	0.52	0.54	0.53	0.47	0.64
	DeLUCS	3	**0.84**	**0.84**	**0.84**	**0.83**	0.92
	DeLUCS$_{i/o}$	3	0.83	0.83	0.83	0.83	**0.92**
Actinopterygii	K-means	-	0.66	0.68	0.66	0.63	0.85
	GMM	-	0.66	0.68	0.66	0.63	0.85
	DEC	-	0.58	0.60	0.58	0.57	0.80
	DeLUCS	8	**0.96**	**0.96**	**0.96**	**0.97**	**0.99**
	DeLUCS$_{i/o}$	8	0.95	0.95	0.95	0.96	0.99
Neopterygii	K-means	-	0.59	0.66	0.62	0.48	0.64
	GMM	-	0.48	0.52	0.50	0.41	0.59
	DEC	-	0.46	0.55	0.50	0.30	0.59
	DeLUCS	3	0.67	0.68	0.67	0.61	0.80
	DeLUCS$_{i/o}$	3	**0.68**	**0.69**	**0.68**	**0.63**	**0.83**
Ostariophysi	K-means	-	0.56	0.56	0.56	0.49	0.68
	GMM	-	0.58	0.58	0.58	0.51	0.74
	DEC	-	0.15	0.16	0.15	0.15	0.54
	DeLUCS	8	**0.66**	**0.67**	**0.66**	**0.67**	**0.87**
	DeLUCS$_{i/o}$	8	0.57	0.58	0.57	0.57	0.81
Cypriniformes	K-means	-	0.66	0.68	0.66	0.56	0.76
	GMM	-	0.67	0.68	0.67	0.57	0.76
	DEC	-	0.30	0.31	0.29	0.21	0.48
	DeLUCS	8	0.68	0.69	0.68	0.58	**0.77**
	DeLUCS$_{i/o}$	8	**0.69**	**0.70**	**0.69**	**0.58**	0.77
Cyprinidae	K-means	-	**0.88**	0.86	**0.89**	0.83	0.87
	GMM	-	0.86	**0.91**	0.88	0.76	0.81
	DEC	-	0.62	0.63	0.60	0.47	0.62
	DeLUCS	8	0.87	0.88	0.87	**0.84**	**0.88**
	DeLUCS$_{i/o}$	8	0.86	0.87	0.85	0.78	0.86

For bacterial DNA, DeLUCS and DeLUCS$_{i/o}$ again surpassed other algorithms in clustering long bacterial genome fragments, with average accuracies of 78% and 74%, respectively, over the two datasets. Notably, this is an average improvement of $> 14\%$ over the best classical clustering algorithm (GMM) and $> 25\%$ compared to the DEC

baseline. That said, the overall accuracy was lower than that observed for mitochondrial DNA, highlighting the complexity of this task.

This dataset's heterogeneity, coupled with recent taxonomic reclassifications, posed a significant challenge for clustering methods. Yet, DeLUCS's methodology effectively coped with these complexities, as misclassifications predominantly occur among previously related families now reclassified into separate phyla [156]. Further analysis within the Proteobacteria phylum confirms the hypothesis that dataset heterogeneity impacts classification accuracy. By focusing solely on intra-phylum clustering, DeLUCS's accuracy improves significantly, underscoring its capability in more homogeneously composed datasets.

Table 3.5: Performance of different clustering algorithms on the bacterial datasets in Table 3.2, Test 7 and 8. The reported values for all the metrics: homogeneity, completeness, normalized mutual information (NMI), adjusted Rand index (ARI) and unsupervised clustering accuracy (ACC) correspond to the average over 10 runs of the algorithms.

Dataset	**Model**	**No. Mimics**	**Homogeneity**	**Completeness**	**NMI**	**ARI**	**ACC**
Bacteria	K-means	-	0.57	0.60	0.59	0.44	0.60
	GMM	-	0.66	**0.71**	**0.68**	0.56	0.71
	DEC	-	0.59	0.63	0.61	0.45	0.59
	DeLUCS	3	**0.66**	0.67	0.66	**0.57**	**0.73**
	DeLUCS$_{i/o}$	3	0.64	0.67	0.65	0.55	0.71
Proteobacteria	K-means	-	0.21	0.26	0.23	0.16	0.43
	GMM	-	0.45	0.51	0.48	0.38	0.59
	DEC	-	0.20	0.25	0.22	0.15	0.40
	DeLUCS	3	**0.68**	**0.68**	**0.68**	**0.66**	**0.83**
	DeLUCS$_{i/o}$	3	0.61	0.63	0.62	0.57	0.78

In clustering viral sequences, including *Influenza A*, *Dengue*, and *Hepatitis B* virus genomes into subtype-based clusters, both K-means and DeLUCS methods exhibited almost perfect performance despite the sequences' close similarities, with K-means marginally outperforming DeLUCS by 2% in the Influenza-A dataset.

These results demonstrate DeLUCS's capability to discern meaningful clusters from unlabelled, diverse DNA sequences, outperforming classical unsupervised methods that rely on k-mer counts. In addition to quantitative assessments, we evaluated DeLUCS's clustering ability qualitatively through a visual inspection of the training process. Figure 3.3 shows the ANN's training progress in identifying clusters over different training epochs. Each epoch represents a complete pass through the dataset. In Figure 3.3, clusters are represented as vertices of a regular polygon (with $c = 5$ vertices in this case), each potentially corresponding to a taxonomic label. Initially, sequences are equally likely to be assigned to any cluster, as indicated by their central location. As training progresses, sequences increasingly align

Table 3.6: Performance of different clustering algorithms on the viral sequences datasets, Table 3.3, Tests 9, 10, and 11. The reported values for all the metrics: homogeneity, completeness, normalized mutual information (NMI), adjusted Rand index (ARI) and unsupervised clustering accuracy (ACC) correspond to the average over 10 runs of the algorithms.

Dataset	**Model**	**No. Mimics**	**Homogeneity**	**Completeness**	**NMI**	**ARI**	**ACC**
Dengue	K-means	-	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	GMM	-	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	DEC	-	1.00	1.00	1.00	1.00	1.00
	DeLUCS	3	1.00	1.00	1.00	1.00	1.00
	DeLUCS$_{i/o}$	3	1.00	1.00	1.00	1.00	1.00
HBV	K-means	-	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	GMM	-	0.87	0.93	0.90	0.77	0.76
	DEC	-	0.85	0.91	0.88	0.76	0.78
	DeLUCS	3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	DeLUCS$_{i/o}$	3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Influenza-A	K-means	-	**0.98**	**0.98**	**0.98**	**0.98**	**0.99**
	GMM	-	0.81	0.90	0.85	0.71	0.73
	DEC	-	0.56	0.92	0.70	0.49	0.59
	DeLUCS	3	0.91	0.93	0.89	0.88	0.97
	DeLUCS$_{i/o}$	3	0.69	0.84	0.75	0.65	0.75

with their respective clusters. The qualitative and quantitative results affirm DeLUCS's efficacy in unsupervised learning from unlabelled DNA sequences, marking its advancement over traditional clustering methods.

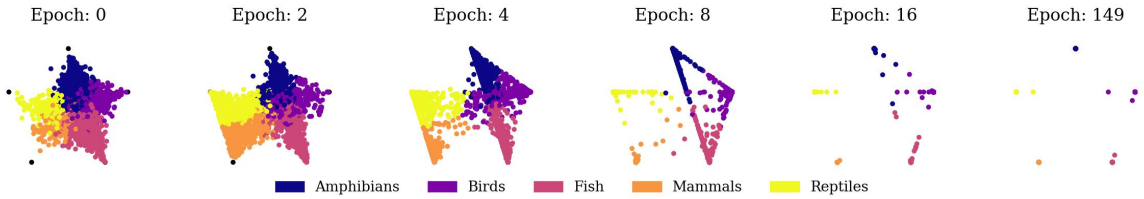


Figure 3.3: Visualization of the clustering process for 2,500 vertebrate mtDNA full genomes into five clusters. Each point represents a sequence, with its position reflecting the probability of belonging to a particular cluster. Initially, sequences are equally probable for all clusters (centred) but gradually align with specific vertices (clusters) as training progresses. Overlap occurs for sequences with identical probability vectors.

3.4.1 Ablation studies

In this section, we empirically study the DeLUCS methodology to understand the impact of different components within our framework. Table 3.7 presents model variants as different components of DeLUCS are added or removed. Initial observations indicate that the IIC loss function marginally outperforms the entropy-based loss function. However, the latter seems to be invariant with respect to the addition or elimination of the other components of our methodology. Additionally, we compare these clustering methods against a supervised learning classification method. For this purpose, the same neural network architecture described in the previous section is trained using labelled data and the cross-entropy loss function. The classification accuracy is calculated by first taking 70% of the data for training and 30% of the data for testing, and it is defined as the ratio of the number of correctly predicted testing sequence labels to the total number of testing sequences.

Table 3.7: Optimal performance metrics from ten trials per dataset under different inclusion/exclusion scenarios. The average accuracy for each dataset was calculated, and these were averaged across all datasets to calculate a unified measure for each configuration.

	Method	**Mimics**	**Noise**	**Majority Voting**	**Loss**	**ACC**
1	DeLUCS	✓	✓	✓	IIC	**0.92 ± 0.08**
2		✗	✓	✓	IIC	-
3		✓	✗	✓	IIC	0.83 ± 0.13
4		✓	✓	✗	IIC	0.88 ± 0.15
5	DeLUCS$_{i/o}$	✓	✓	✓	HC	0.90 ± 0.08
6		✗	✓	✓	HC	0.89 ± 0.09
7		✓	✗	✓	HC	0.88 ± 0.11
8		✓	✓	✗	HC	0.89 ± 0.13
9	Supervised	-	-	-	CE	0.96 ± 0.06

IIC: Invariant Information Clustering, HC: Entropy-based clustering, CE: Cross-Entropy.

Mimic Sequences: In the DeLUCS framework, the role of mimic sequences is crucial, particularly when integrated with the IIC loss function, as its use is impossible without them. When considering the entropy loss, Table 3.7 incorporating mimic sequences alongside a regularization cross-entropy term enforcing the consistency of the predictions has a marginal impact on performance. The optimal quantity of mimic sequences is determined through empirical evaluation. We have established three mimic sequences per sample as the standard setting for our methodologies. This baseline was chosen because increasing the number of mimics beyond this threshold does not substantially enhance accuracy in datasets with an ample number of sequences per cluster. Our findings indicate that for datasets anticipated to contain fewer than 150 sequences per cluster, setting the number of mimic sequences to at least eight per sample is necessary to achieve competitive outcomes.

Adding Gaussian Noise: Gaussian noise was introduced to the network parameters at intervals of every 30 epochs to address the model's tendency to converge to suboptimal solutions. This strategy empirically improved accuracy, as demonstrated in rows 3 and 7 in Table 3.7 and in Figure 3.4. The Figure showcases the learning curves for a single ANN with and without Gaussian noise. Introducing noise periodically prevents convergence to less optimal solutions, boosts classification accuracy, and is more relevant for the IIC metric.

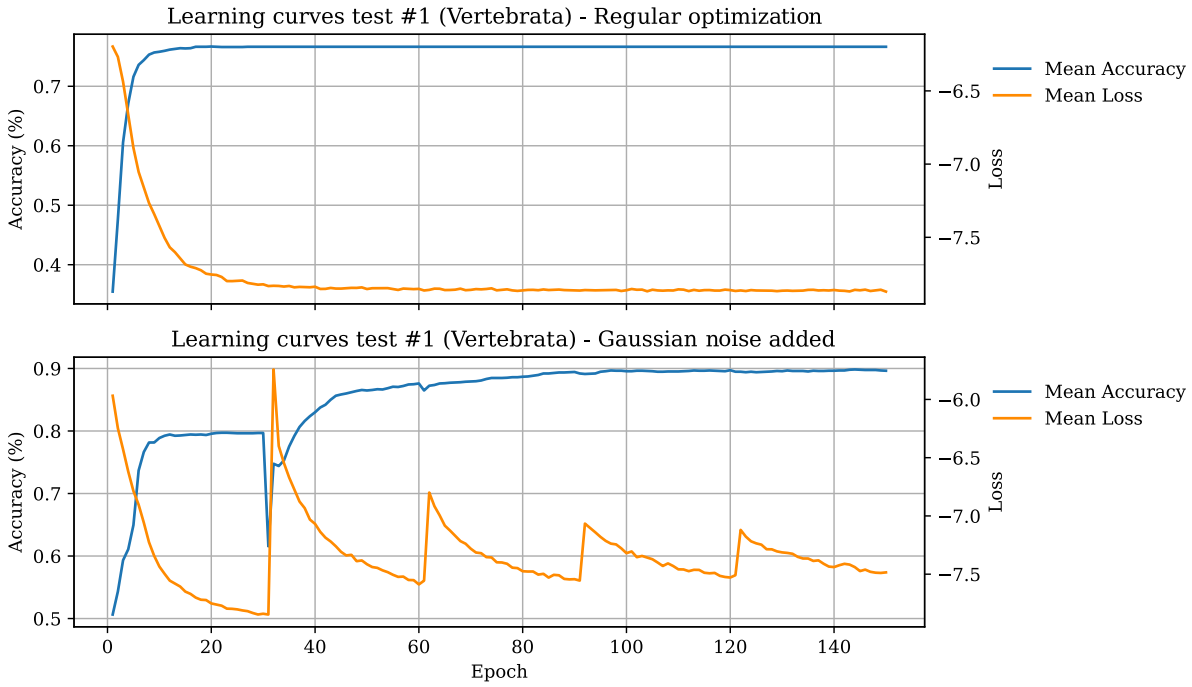


Figure 3.4: Learning curves for a single ANN during the training process, showing the effect of Gaussian noise addition on classification accuracy for the vertebrate mtDNA genome dataset (Test 1). The top graph illustrates training without noise, and the bottom graph with noise addition highlights improving classification accuracy from approximately 82% to 96%

Majority Voting for Variance Mitigation: Given the high variance observed in the outcomes of training multiple ANNs independently, a majority voting scheme was employed by training five ANNs for each dataset and combining the results. This approach reduced prediction variance and improved classification accuracies across all tests, as shown

in Figure 3.5 and rows 4 and 8 in Table 3.7.

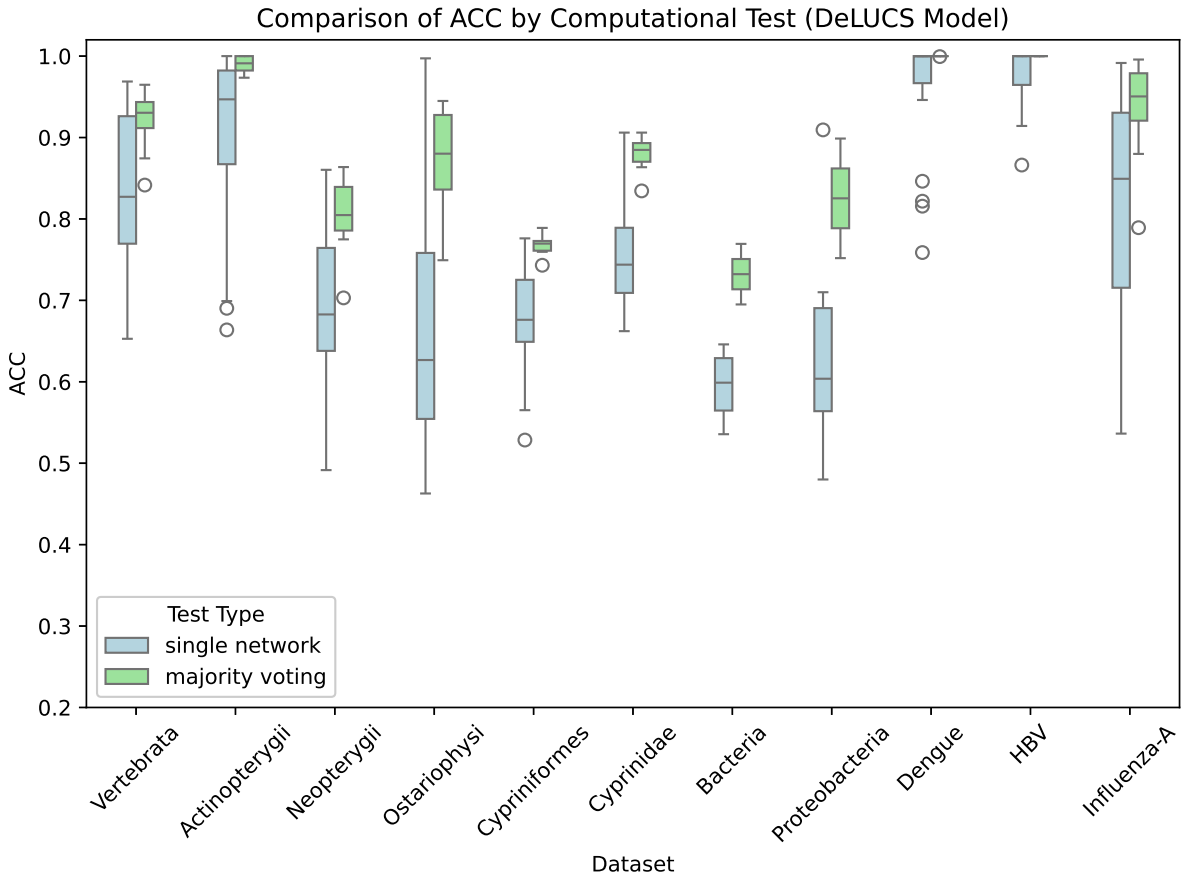


Figure 3.5: Accuracy comparison between single ANN training (light blue) and an ensemble of 5 ANNs (light green) with majority voting. Each test was conducted one hundred times to evaluate variance, with the ensemble approach showing both reduced variance and increased accuracy in all cases.

After assessing the significance of each component in DeLUCS, a final observation from the results obtained in Table 3.7 is that, on average, the performance of DeLUCS is still behind that of a model trained using taxonomic annotations by approximately 4%. Nevertheless, the integration of entropy-based clustering loss functions holds promise in narrowing this performance gap even further with the introduction of better optimization techniques and improved regularization. Moreover, while supervised methods may demonstrate performance superiority, this advantage can be diminished by the requirement for excessive manual

annotations. Finally, it is crucial to acknowledge the potential for incorrect annotations, which poses a challenge when comparing unsupervised methods using external evaluation metrics. Some clustering assignments obtained through DeLUCS initially deemed incorrect, might actually be correct, but they have been evaluated against an incorrect assumed-to-be-real taxonomic assignment. These results emphasize the nuanced balance between classification accuracy, algorithmic complexity, and efficiency.

3.5 Discussion

DeLUCS represents a pioneering effort in applying unsupervised deep learning for clustering unlabelled DNA sequences. This method represents a breakthrough for processing large and diverse datasets, which often pose challenges for conventional clustering methods due to the lack of homology or incomplete biological annotations. Importantly, DeLUCS brings the principles of entropy-based clustering, previously confined to computer vision, into computational biology, promising to revolutionize our approach to genomic data analysis. That being said, DeLUCS has some limitations. For example, being an entropy-based clustering technique, it performs best when the clusters are evenly distributed. However, this scenario is not typically achieved in practice where some taxonomic groups are overrepresented. Additionally, there is a high variance in results across multiple runs of the training process, making the method highly sensitive to initialization and reducing both performance and reproducibility. Finally, as a parametric clustering technique, DeLUCS requires a well-defined range for the expected number of clusters, which may not always be possible in practice.

There are various ways to enhance DeLUCS and overcome its current limitations. One crucial area of focus is to improve the loss function to handle unbalanced datasets more effectively. Additionally, developing more advanced clustering ensemble techniques could also lead to improved performance. Another promising approach is to optimize parameter initialization, which has the potential to enhance results and reduce the variance in each training run. Ideally, this could even reduce the need for clustering ensembles, making clustering faster and more efficient using a single neural network. Finally, one can enforce the consistency of the intermediate representations learned by each discriminative model and utilize these representations as part of non-parametric clustering methodologies. Some of these ideas will be explored in the upcoming chapter.

Implementation and code availability

Some of the code implementing the DeLUCS methodology is available on GitHub https://github.com/millnap95/DeLUCS as part of the Supplementary information in [134], enabling users to replicate our results or cluster new sequences. Testing was conducted on the Cedar cluster of Compute Canada, which was equipped with Intel E5-2650 v4 CPUs, 32 GB RAM, and 1 NVIDIA P100 Pascal GPU.

Chapter 4

Improving DNA sequence clustering with contrastive self-supervision

We have discussed how clustering algorithms can play a fundamental role in bioinformatics, as they are used to study the structural composition of DNA sequence datasets, discover novel OTUs, and complement phylogenetic analysis. In the previous chapter, we presented DeLUCS, a methodology that leverages deep neural networks, to significantly outperform classical unsupervised learning algorithms in discovering genomic-signature-based clusters at different taxonomic levels. These promising initial results motivated the development of *i*DeLUCS, which also uses mimic sequences and enforces the consistency of the predicted labels. A key observation is that it is also possible to enforce the consistency of the intermediate representations learned by the network. *i*DeLUCS considers this observation and other implementation optimizations to cluster datasets comprising more than 400 Mbp. Finally, *i*DeLUCS exhibits several novel features that enhance the interpretability of its results, and a graphical user interface (GUI) was developed to improve the method's applicability for bioinformatics practitioners.

The contents of this chapter are an adaptation of a paper I co-authored titled "*i*DeLUCS: A deep learning interactive tool for alignment-free clustering of DNA sequences" [136]. In Section 4.1, the contrastive learning framework and its integral components are presented. Following this, Section 4.2 delves into our novel method, delineating the main distinctions from the DeLUCS approach. Section 4.4.1 describes the components of our methodology in light of the contrastive learning framework. Concurrently, Section 4.2.2 presents an information-theoretic clustering ensemble posited as an alternative to the conventional majority voting scheme.

Section 4.3 explores the seamless extension of the framework to accommodate non-

parametric clustering outcomes, simultaneously spotlighting the versatile applications of representations derived through contrastive learning in various domains. Section 4.4 presents the updated results derived from applying the proposed methodology, including an additional evaluation metric employed in this chapter to assess performance more rigorously. Concluding the chapter, Section 4.5 offers a reflective discussion of the method's strengths, limitations, and implications of these findings.

4.1 Related work

Our methodology is motivated by self-supervised representation learning, where models are encouraged to learn the similarity between semantically transformed versions of the input data points (augmentations). This principle aligns with the underlying design principles considered for our clustering methodology in DeLUCS. In particular, contrastive self-supervised methods learn useful representations by mapping data into an embedding space where representations of augmented views (positive pairs) must be close to each other and far from the rest of the data (negative pairs). Although several of these algorithms have been proposed, they all follow the same framework, now known as the contrastive-learning framework [30]. It contains three major components:

- *A dataset of paired data.* Given a data set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the goal is to construct the auxiliary training set of augmentations

$$\mathbb{A} = \left\{ \left(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_1^1\right), \left(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_1^2\right), \left(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_1^3\right), \ldots, (\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_i^m) \mid 1 \leq i \leq n \right\},$$

  where the data points in each pair $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_i^j)$ are considered similar according to some criteria based on prior knowledge, e.g., invariance to distortions or spatial proximity. Note that in this framework, samples $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_i^j$ may not be present in the original dataset and can be both an augmented version of the original sample $\boldsymbol{x}_i$. We refer to this module as the data augmentation module, consisting of a set $\mathcal{T}$ of augmentation functions t, such that each $t \sim \mathcal{T}$ applies a different augmentation to a given sample in the original dataset.

- *A neural encoder.* To learn from the paired dataset, the goal is to learn a mapping that encodes only what is common between $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_i^j$ while dropping all the irrelevant information. If such a mapping Φ is found, the image $\mathcal{Y} = \Phi(\mathcal{X})$ becomes a lower dimensional representation of the original space $\mathcal{X}$ and can be used for downstream tasks. The best candidate for Φ is a deep neural network Φ_θ as its parameters θ can be optimized via SGD.

- *A contrastive loss function.* Depending on the unsupervised learning task of interest, any "pretext" task that attempts to minimize the distance between representations of pairs of samples $(\boldsymbol{z}_i, \tilde{\boldsymbol{z}}_i)$ can be used as inspiration for the loss function. A general formulation of unified contrastive losses was proposed by [192] as a family of loss functions

$$\mathcal{L}_{\phi,\psi}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \phi \left(\sum_{j \neq i} \psi \left(\|\boldsymbol{z}_i - \tilde{\boldsymbol{z}}_i\|_2^2 - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 \right) \right), \tag{4.1}$$

  where ϕ and ψ are monotonously increasing, differentiable scalar functions and .

  Another more informal but perhaps more intuitive characterization of a contrastive loss is described in [31]. Here, any contrastive function can be written as

$$\mathcal{L}_{\text{contrastive}} = w \cdot \mathcal{L}_{\text{alignment}} + (1 - w) \cdot \mathcal{L}_{\text{distribution}}, \tag{4.2}$$

  where $\mathcal{L}_{\text{alignment}}$ encourages representations of paired samples to be consistent and $\mathcal{L}_{\text{distribution}}$ encourages representations to match a target distribution. w is the weighting parameter defining the importance of each term in the final loss.

4.2 Proposed method: *i*DeLUCS

*i*DeLUCS builds upon the pipeline proposed in the previous chapter, consisting of: *(i)* calculating the k-mer frequencies for each DNA sequence, *(ii)* computing the data augmentations (mimic sequences), *(iii)* training multiple deep neural networks to learn the cluster assignments, and *(iv)* computing the majority voting cluster assignment for each sequence. In addition to multiple algorithmic optimizations to the pipeline, *i*DeLUCS significantly extends it in four main aspects. First, it uses the contrastive learning framework introduced in the previous section and incorporates an additional contrastive term in the loss function, which enforces the consistency of the hidden representations learned by the artificial neural networks. These hidden representations are learned simultaneously with the cluster assignments via backpropagation. Second, it replaces the majority voting scheme with a more robust clustering ensemble based on information theory, which reduces the variance and boosts the accuracy. Third, it uses the information provided by the ensemble and the consistency of the hidden representations to provide an intrinsic quantitative assessment of the clustering assignment (silhouette coefficient), as well as to output the confidence score for the cluster assignment of each sequence in the dataset. The new contrastive learning framework can be combined with non-parametric clustering algorithms, such as HDBSCAN ([127]), to automatically determine the number of clusters.

4.2.1 Contrastive learning-based pipeline

In this subsection, we illustrate how the methodology proposed in *i*DeLUCS fits into the contrastive learning framework. We describe its main components and compare them against the pipeline presented in Chapter 3.

Data augmentation

Inspired by DeLUCs, we use the same probabilistic model based on DNA substitution mutations (transitions and transversions) to produce different mimic sequences. For each DNA sequence in the dataset, this process produces $2m$ artificially created training samples, considered positive training pairs, even though they are not in the original training dataset. Specifically, given a sequence s_i and a particular position j in the sequence, we fix independent transition and transversion probabilities $p_{ts}[j]$ and $p_{tv}[j]$ respectively. Each augmentation function $t \in \mathcal{T}$ applies three different augmentations sequentially: Two random DNA substitution mutations, i.e., transitions and transversions with fixed independent substitution probabilities $p_{ts} = 10^{-3}$ and $p_{tv} = 5^{-3}$ respectively, and a random assignment of $r = 20$ nucleotides to symbol `N` (representing an unidentified nucleotide). This composition of augmentations incorporates robustness into the model and allows the networks to learn the structure of more complex datasets. Each augmented training sequence is then converted into a numerical vector containing the counts of all of its k-mers, where a k-mer is defined as a subsequence of length k that does not contain the symbol `N`. Finally, each k-mer count vector is converted into a k-mer frequency vector by dividing its k-mer counts by the total length of the sequence minus the number of `N` symbols in the original DNA sequence.

Neural network - Base encoder

As it is illustrated in Figure 4.1, we divide our architecture into a base encoder $f_\theta(\cdot)$ that extracts a meaningful lower dimensional representation and a clustering layer $g_\theta(\cdot)$ such that $\Phi_\theta(\boldsymbol{x}) = \text{softmax}(g_\theta(f_\theta(\boldsymbol{x})))$. For a mini-batch $X_{\mathcal{MB}} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)\}_{i=1}^{N}$ of $2N$ augmented k-mer vectors, all the k-mer vectors are passed through the encoder, that consists of two fully connected layers, *Linear (512 neurons)* and *Linear (64 neurons)*, each one followed by a *ReLU* and a *Dropout* layer with dropout rate of 0.5. The hidden representation $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ are then passed through the clustering layer *Linear (K_ clusters)*, where K represents the upper bound of the number of clusters. The distribution over the K output clusters is calculated using a *softmax* activation function.

Contrastive loss function

The negative weighted mutual information loss function, as described in equation 3.10, aligns with the general principles of a contrastive loss within the contrastive learning paradigm described in the previous section (equation 4.2). Minimizing the conditional entropy $H(\Phi(\mathbf{x}) \mid \Phi(\tilde{\mathbf{x}}))$, enforces sample $\mathbf{x}$ to be precisely predictable from its augmented counterpart $\tilde{\mathbf{x}}$. Moreover, maximizing the entropy term in equation 3.10 can be interpreted as maximizing the KL divergence between the output distribution and a uniform distribution across clusters, $H(\Phi(\mathbf{x})) = D_{KL}(\Phi(\mathbf{x}) \mid\mid Unif(y))$, where y spans the set of possible cluster assignments $\Phi(\mathcal{X})$. This interpretation of the loss shows how it satisfies the two principles of alignment and distribution.

*i*DeLUCS leverages this framework to learn cluster assignments concurrently with a hidden representation that quantifies distances between samples in distinct clusters. It achieves this by integrating the weighted mutual information loss in equation 3.10, with a consistency-enforcing loss function for intermediate representations during training. Specifically, the normalized temperature-scaled cross entropy (NT-Xent) [30] loss is employed for this purpose, as defined by:

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{2N} \sum_{(i,j)\in\mathbb{P}} \log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbf{1}[k \neq i] \exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)}, \tag{4.3}$$

where $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ represents the learned representation, the cosine similarity $\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)$ measures the distance between sample representations, $\mathbb{P}$ is the set of pairs of indices representing positive examples in the mini-batch, $\mathbf{1}[k \neq i]$ is an indicator function, and τ is the temperature parameter fine-tuning the similarity range, set to $\tau = 1$ in our case.

The collective loss function of *i*DeLUCS is articulated as:

$$\mathcal{L} = w \cdot \mathcal{L}_I + (1 - w) \cdot \mathcal{L}_{\text{NT-Xent}}, \tag{4.4}$$

with the hyper-parameter w adjusting the balance between the loss components. Figure 4.1 illustrates how *i*DeLUCS incorporates the additional contrastive term into the final loss to enforce the consistency of the hidden representations learned by the artificial neural networks. This provides robustness with respect to unbalanced datasets, as the learned representation of sequences in the same cluster are close to each other but far from the sequences in other clusters.

4.2.2 Information theoretic clustering ensemble

For a given dataset $\mathcal{X} = \{x_1, \ldots, x_N\}$, a partition π can be represented as a set of K clusters $\pi = \{L_1, \ldots, L_K\}$, such that $\pi(x_i)$ denotes the cluster label assigned to x_i by the

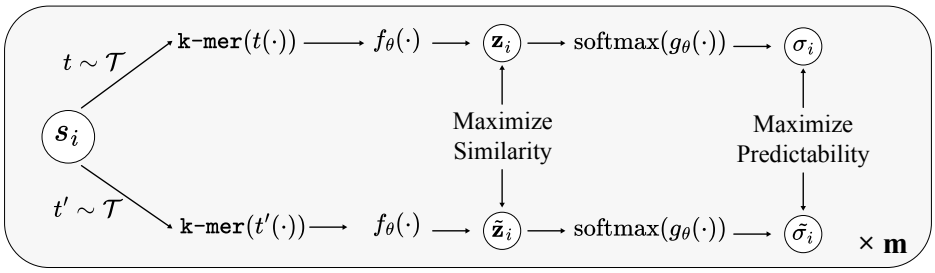


Figure 4.1: *i*DeLUCS maximizes the mutual information between the corresponding soft assignments $\boldsymbol{\sigma}$ and $\tilde{\boldsymbol{\sigma}}$ of the augmentations $\mathbf{x}$, $\tilde{\mathbf{x}}$ from each training sequence after random mapping t, while maximizing the similarity of the hidden representations $\mathbf{z}$ and $\tilde{\mathbf{z}}$.

partition. Suppose we are given a set $\Pi = \{\pi_1, ..., \pi_T\}$ of T partitions of the data set $\mathcal{X}$. The problem of clustering combination is to find the consensus partition π_C that best summarizes the information present in Π. In general, combining multiple partitions in an unsupervised setting is a challenging problem, as each partition in the combination is represented as a set of labels assigned by an independent clustering algorithm with no trivial mapping between the assignments. Here, we provide a detailed explanation of the concepts used in *i*DeLUCS, which are a combination of the work presented in [138, 186, 195]. We use an information theoretic approach to the problem that does not compute an explicit mapping between the assignments. In this framework, the quality of the consensus partition $\pi_C = \{C_1, \ldots, C_K\}$ is determined by the amount of information it shares with all the partitions $\pi_i = \{L_j^i \mid 1 \leq j \leq K\} \in \Pi$, where L_j^i is the $j-th$ cluster in the i-th partition. The best possible partition is then determined by

$$\pi_C^{\text{best}} = \arg\max_{\pi_C} \sum_{i=1}^{T} MI(\pi_C, \pi_i) \quad \text{, where}$$

$$MI(\pi_C, \pi_i) = \sum_{r=1}^{K} \sum_{r=1}^{K} p(C_r, L_j^i) \log \left(\frac{p(C_r, L_j^i)}{p(C_r)p(L_j^i)} \right) \tag{4.5}$$

is the classical Shannon mutual information between partitions. The previous optimization problem represents a difficult combinatorial problem [186]. However, the work in [195] shows that it is possible to consider a generalized definition of mutual information to simplify the problem. The generalized entropy of degree s for a discrete probability distribution

$P = (p_1, \ldots, p_n)$ is defined as

$$H^s(P) = (2^{1-s} - 1)^{-1} \left(\sum_{i=1}^{n} p_i^s - 1 \right), \; s > 0, \; s = 1.$$

Hence, the generalized quadratic mutual information ($s = 2$) becomes:

$$\begin{aligned} I^2(\pi_C, \pi_i) &= H^2(\pi_i) - H^2(\pi_i \mid \pi_C) \\ &= -2 \left(\sum_{j=1}^{K} p\left(L_j^i\right)^2 - 1 \right) + 2 \sum_{r=1}^{K} p\left(C_r\right) \left(\sum_{j=1}^{K} p\left(L_j^i \mid C_r\right)^2 - 1 \right) \\ &= 2 \sum_{r=1}^{K} p\left(C_r\right) \sum_{j=1}^{K} p\left(L_j^i \mid C_r\right)^2 - 2 \sum_{j=1}^{K} p\left(L_j^i\right)^2 . \end{aligned} \tag{4.6}$$

With the following estimates, $p(C_r) = |C_r|/N$, $p(L_j^i) = |L_j^i|/N$, and $p(L_j^i \mid C_r) = |L_j^i \cap C_r|/|C_r|$, the quadratic mutual information $I^2(\pi_C, \pi_i)$ can be expressed in terms of the category utility score Δ as $I^2(\pi_C, \pi_i) = 2\Delta(\pi_C, \pi_i)$ [186]. This is relevant because in [138], Mirkin showed that a solution to the optimization problem of the utility function can be obtained by transforming the categorical labels into standardized binary features. *i*DeLUCS uses the same transformation, replacing each partition π_i by K binary features and standardizing each binary feature to a zero mean. More specifically, for each data point x, and each partition $\pi_i \in \Pi$, the values of the new features are calculated as $y_{ij} = \mathbf{1}[L_j^i = \pi_i(x)] - p(L_j^i)$. The final solution of the consensus partition problem can be obtained by a classic clustering algorithm operating over the new features y_{ij}. This clustering ensemble technique introduces robustness into the method and provides a better estimate of the confidence score of *i*DeLUCS for each sequence in the dataset.

4.3 Clustering and representation learning: A general framework to cluster DNA sequences

For scenarios where the number of expected clusters may be unknown, non-parametric clustering tools that automatically identify the number of clusters may be preferred. Fortunately, the contrastive learning framework can be seamlessly integrated with non-parametric clustering algorithms when some homology is expected. In this context, we have enhanced *i*DeLUCS with an additional option to infer the number of clusters using the classical non-parametric clustering algorithm HDBSCAN [127]. To achieve this, we set

the invariant information component of the loss in equation 4.4, responsible for network assignment, to zero, while the predominant component becomes the NT-Xent loss. The learned 64-dimensional latent features are then used as input to HDBSCAN to compute the final clustering. We recommend using this feature only when the resulting clusters are expected to correspond to the lowest possible taxonomic level since HDBSCAN is a density-based method and higher taxonomic groups usually contain several subclusters.

4.4 Experimental setup

4.4.1 Datasets

To evaluate the efficacy and versatility of *i*DeLUCS, we conducted tests across a diverse range of datasets from real and simulated data with known ground-truth annotations. Besides the datasets described in the previous chapter (described in (a)), we test the performance of *i*DeLUCS over 3 additional mitochondrial datasets compiled from NCBI in June 2022 (described in (b)); one dataset of metagenomic reads simulated from eight microbial genomes using the Pacific Biosciences SMRT error model for long metagenomic reads [209] (described in (c)), and 12 synthetic datasets totalling 246,625 artificial DNA sequences (described in (d)). Each dataset was selected for its unique characteristics, as described herein.

(a) *Eight datasets from Kingdom Animalia, Kingdom Bacteria, and three datasets of viral sequences, obtained from [134].* Six mitochondrial DNA datasets of vertebrates at taxonomic levels from Subphylum to Family; two bacterial datasets to be clustered into families; and three viral datasets (Dengue, Influenza-A, Hepatitis B) clustered into virus subtypes. The maximum number of clusters per dataset is 12, and the maximum cluster size is 500 sequences, with an average sequence length of 16,700 bp for mtDNA, 433,882 bp for bacterial, and 5,058 bp for viral sequences. The composition of this datasets is summarized in Tables 3.1, 3.2 and 3.3, in the previous chapter

(b) *Three new mitochondrial DNA datasets* created to enhance the representation across Kingdoms of Life: A dataset of 2,581 mitochondrial genomes from Kingdom Protista (average sequence length 17,141 bp) clustered into three phyla/subphyla; a dataset of 9,027 mitochondrial genomes from class Insecta (average sequence length 15,841 bp) clustered into seven orders; and a dataset of 1,759 mitochondrial genomes from Kingdom Fungi (average sequence length 62,644 bp), clustered into three phyla/subphyla. The composition of this dataset is summarized in Table 4.1

Table 4.1: Description of the new mitochondrial DNA datasets (b), and the simulated metagenomic reads from eight microbial genomes introduced by [209]. Note that there is a balanced version of each new dataset (Fungi, Protists, Insects), where the number of sequences per cluster in the balanced version was selected according to the number of sequences available in the smallest cluster.

Dataset	Total no. sequences	Min. seq. length (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)	Total no. clusters	Cluster min. size	Cluster avg. size	Cluster max. size
Insects	9,027	14,602	15,841	26,613	7	652	1,290	1,976
Fungi	1,759	20,063	62,644	99,976	3	335	586	889
Protists	2,581	5,493	17,141	69,503	3	315	860	1,642
Insects–balanced	4,550	14,602	15,897	25,011	7	650	650	650
Fungi–balanced	1,005	21,684	60,657	99,976	3	335	335	335
Protists–balanced	945	5,498	24,697	69,503	3	315	315	315
Simulated reads	432,333	5,000	8,511	37,216	8	8,538	54,042	119,330

(c) *One dataset of simulated metagenomic reads from eight microbial genomes, obtained from [209].* This dataset comprises 432,333 sequencing reads to be clustered into eight species (seven Bacteria and one Archaea). The reads were simulated using the PacBio sequencing simulation parameters, with a maximum cluster size of 119,330 sequences and an average sequence length of 8,511 bp. The composition of this dataset is summarized in the last row in Table 4.1.

(d) *12 synthetic datasets from [60].* These are artificial datasets, each consisting of 100 random template sequences representing the true clusters and a random number of mutated copies that were generated from each template according to a predefined identity threshold. Each dataset contains at most 25,000 sequences, with a minimum dataset size of 18,210. The maximum number of clusters for each dataset is 12, the maximum cluster size is 400 sequences, and the average sequence length is 20,552 bp. The composition of this dataset is summarized in Table 4.2

4.4.2 Evaluation metrics

In addition to the external validation methods described in Section 2.3.4, which are the most appropriate for our application domain, as discussed in the previous chapter, we consider an intrinsic evaluation metric to assess the impact of the learned representations in the clustering. The results are calculated in the representation space and not in the k-mer space, following [121].

Table 4.2: Summary of the twelve synthetic datasets from [60] included in the study. The number in the name of each dataset represents an identity score threshold, indicating that each sequence in a cluster is within this threshold from the cluster center.

Dataset	Total no. sequences.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)	Total no. clusters	Cluster min. size	Cluster avg. size	Cluster max. size
Medium-60	18,210	653	1,365	2,062	100	13	182	397
Medium-70	18,731	678	1,359	2,027	100	7	187	399
Medium-80	20,939	664	1,425	2,043	100	17	209	383
Medium-90	21,266	730	1,340	2,016	100	9	213	398
Medium-95	24,039	724	1,446	2,038	100	18	240	396
Medium-97	20,772	736	1,358	2,022	100	6	208	399
Long-60	20,885	1,393	2,758	4,039	100	10	209	400
Long-70	18,558	1,441	2,754	4,062	100	6	186	399
Long-80	20,525	1,396	2,639	3,974	100	9	205	396
Long-90	22,518	1,489	2,586	3,964	100	6	225	397
Long-95	20,222	1,461	2,890	4,049	100	14	202	401
Long-97	19,960	1,486	2,715	3,988	100	9	200	396

- **Silhouette Coefficient:** This measure compares the cluster assignment of a sequence with the assignment of the closest sequence assigned to a different cluster. Specifically,

$$\text{Silhouette} = \frac{1}{N}\sum_{i=1}^{N}\frac{b-a}{\max(a,b)} \tag{4.7}$$

  where N is the number of sequences in the dataset, b is the distance between the representation of a sequence and the representation of the nearest sequence in a cluster it does not belong to, and a is the mean intra-cluster distance. The best possible score is 1, which decreases as the cluster overlap increases. Scores with negative values indicate that most of the sequences have been placed in the wrong cluster, with the worst possible value being -1.

4.4.3 Results

In this section, we build on the proven effectiveness of DeLUCS over classical clustering algorithms and the DEC baseline. Here, we focus on comparing the performance of *i*DeLUCS on the datasets in (a) and (b) against its predecessor. This comparative analysis is detailed in Table 4.4 for the new datasets and Table 4.3 for the eleven benchmarking datasets.

In particular, *i*DeLUCS outperforms DeLUCS on the real datasets in (a) and (b), most of which consist of non-homologous sequences. For example, for the mitochondrial genome datasets, the average accuracy (ACC) of *i*DeLUCS is 92.4%, a substantial increase compared to DeLUCS, for which the average accuracy was only 75.42%. This difference is mainly due

Table 4.3: Comparison of the performance of *i*DeLUCS against DeLUCS on the benchmark datasets (a), using an intrinsic cluster evaluation metric (silhouette coefficient) and external evaluation metrics (homogeneity, completeness, adjusted Rand index ARI, normalized mutual information NMI, and unsupervised clustering accuracy ACC). The boldface indicates the best result, and "balanced" indicates the balanced version of the datasets.

Dataset	**Model**	**No. Mimics**	**Silhouette Score**	**Homogeneity**	**Completeness**	**NMI**	**ARI**	**ACC**
Vertebrata	DeLUCS	3	0.37	0.85	0.86	0.85	0.86	0.94
	iDeLUCS	3	**0.54**	**0.93**	**0.93**	**0.93**	**0.94**	**0.97**
Actinopterygii	DeLUCS	8	**0.35**	**0.99**	**0.98**	**0.98**	**0.99**	**1.00**
	iDeLUCS	7	0.22	0.95	0.95	0.95	0.95	0.98
Neopterygii	DeLUCS	3	0.34	0.69	0.70	0.70	0.64	0.83
	iDeLUCS	3	**0.44**	**0.75**	**0.75**	**0.75**	**0.71**	**0.87**
Ostariophysi	DeLUCS	8	**0.13**	**0.73**	**0.74**	**0.73**	**0.75**	**0.90**
	iDeLUCS	8	0.08	0.68	0.69	0.68	0.69	0.85
Cypriniformes	DeLUCS	8	0.26	0.68	0.69	0.68	0.58	0.77
	iDeLUCS	8	**0.48**	**0.72**	**0.72**	**0.71**	**0.63**	**0.80**
Cyprinidae	DeLUCS	8	0.33	**0.88**	**0.89**	**0.87**	**0.80**	**0.89**
	iDeLUCS	8	**0.66**	0.87	0.87	0.86	0.78	0.87
Bacteria	DeLUCS	3	0.30	0.67	0.68	0.67	0.59	0.74
	iDeLUCS	3	**0.42**	**0.78**	**0.79**	**0.79**	**0.72**	**0.83**
Proteobacteria	DeLUCS	3	**0.16**	0.67	0.67	0.67	0.65	0.83
	iDeLUCS	3	0.11	**0.75**	**0.76**	**0.75**	**0.74**	**0.86**
Dengue	DeLUCS	3	**0.75**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	iDeLUCS	3	0.07	0.99	0.99	0.99	1.00	1.00
HBV	DeLUCS	3	**0.77**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	iDeLUCS	3	0.59	0.99	0.99	0.99	0.99	1.00
Influenza-A	DeLUCS	3	0.58	0.97	0.97	0.97	0.98	0.99
	iDeLUCS	3	**0.69**	**0.98**	**0.98**	**0.98**	**0.98**	**0.99**

to the drop in performance of DeLUCS on the new mitochondrial datasets, chosen to be from particularly challenging taxa.

As seen in Table 4.1, for the newly introduced mitochondrial datasets, *i*DeLUCS showcases clustering accuracies between 78% and 89.7%, outperforming DeLUCS, which ranges from 60.1% to 88.1%. Indeed, although *i*DeLUCS performs better overall on balanced datasets, both the improved clustering ensemble and the new contrastive loss function provide robustness with respect to unbalanced datasets, as equation (4.4) is not dominated by the entropy term in favour of a uniform output distribution.

For simulated data, we explore *i*DeLUCS's capabilities against various baselines on previously untested data types. Hence, for the dataset of simulated reads (c), we include a comparison with DeLUCS, classical clustering algorithms (k-means and GMM), and a specialized binning algorithm for long metagenomic reads [209], and the results are summarized in 4.5. Additionally, for synthetic data (d), compiled from [60], we benchmark against MeShCLust v.3.0, a leading alignment-assisted method requiring a predefined similarity threshold. Results are detailed in Tables 4.6 and 4.7.

In the clustering of the dataset of simulated long metagenomic reads (c), the accuracy of *i*DeLUCS is 84%, $\sim$ 16% higher than that of DeLUCS and $\sim$ 7% higher than that of

Table 4.4: Comparison of the performance of *i*DeLUCS against DeLUCS on the new mtDNA datasets (b), using an intrinsic cluster evaluation metric (silhouette coefficient) and external evaluation metrics (homogeneity, completeness, adjusted Rand index ARI, normalized mutual information NMI, and unsupervised clustering accuracy ACC). The boldface indicates the best result, and "balanced" indicates the balanced version of the datasets.

Dataset	**Model**	**No. Mimics**	**Silhouette Score**	**Homogeneity**	**Completeness**	**NMI**	**ARI**	**ACC**
Insects	DeLUCS	3	0.25	0.64	0.63	0.63	0.60	0.73
	iDeLUCS	3	**0.52**	0.78	**0.76**	**0.79**	**0.80**	**0.84**
Fungi	DeLUCS	3	**0.46**	0.50	0.48	0.49	0.40	0.63
	iDeLUCS	3	0.28	0.67	**0.63**	**0.64**	**0.56**	**0.78**
Protists	DeLUCS	3	0.51	0.54	0.45	0.49	0.36	0.62
	iDeLUCS	3	**0.81**	0.60	**0.79**	**0.56**	**0.65**	**0.81**
Insects – balanced	DeLUCS	3	0.37	0.67	0.68	0.67	0.59	0.78
	iDeLUCS	3	**0.57**	**0.82**	**0.83**	**0.82**	**0.80**	**0.89**
Fungi – balanced	DeLUCS	3	0.32	0.52	0.52	0.52	0.50	0.76
	iDeLUCS	3	0.26	**0.88**	**0.88**	**0.88**	**0.91**	**0.97**
Protists – balanced	DeLUCS	3	**0.53**	**0.70**	**0.70**	**0.70**	**0.70**	**0.88**
	iDeLUCS	3	0.37	0.66	0.67	0.67	0.65	0.86

Table 4.5: Comparison of the performance of *i*DeLUCS against K-means, GMM, DeLUCS and LRBinner on the dataset of simulated metagenomic reads from eight microbial genomes introduced by [209], using an intrinsic cluster evaluation metric (silhouette coefficient) and external evaluation metrics (homogeneity, completeness, adjusted Rand index ARI, normalized mutual information NMI, and unsupervised clustering accuracy ACC). The boldface indicates the best result. **Note:** GMM did not converge in this experiment.

Dataset	Model	No. Mimics	Silhouette Score	Homogeneity	Completeness	NMI	ARI	ACC
Simulated reads	K-means	-	0.055	0.86	0.81	0.79	0.75	0.77
	DeLUCS	3	0.68	0.65	0.70	0.68	0.64	0.67
	LRBinner	-	**0.91**	**0.97**	**0.99**	**0.98**	**0.97**	**0.98**
	*i*DeLUCS	3	0.80	0.90	0.86	0.90	0.87	0.83

Table 4.6: Comparison of the performance of *i*DeLUCS + HDBSCAN (*i*DeLUCS – auto) against MeShClust v3.0 clustering algorithms on the **medium synthetic datasets** introduced by [89], using external evaluation metrics (homogeneity, completeness, adjusted Rand index ARI, normalized mutual information NMI, and unsupervised clustering accuracy ACC). The boldface indicates the best result. "MeShCLust" denotes MeShCLust v3.0 run with the option of automatically identifying the identity threshold parameter.

Dataset	Model	No. Clusters	Homogeneity	Completeness	NMI	ARI	ACC
MediumTest-60	MeShCLust	100	**1.00**	1.00	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	68	0.93	**1.00**	0.96	0.84	0.89
MediumTest-70	MeShCLust	100	**1.00**	1.00	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	79	0.97	**1.00**	0.98	0.95	0.94
MediumTest-80	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	92	0.99	1.00	0.99	0.98	0.97
MediumTest-90	MeShCLust	100	**1.00**	1.00	1.00	1.00	1.00
	*i*DeLUCS-auto	100	1.00	**1.00**	**1.00**	**1.00**	**1.00**
MediumTest-95	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
MediumTest-97	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

K-means. However, it trails the specialized metagenomic binning baseline by $\sim 16\%$. That said, the potential inclusion of coverage information could further enhance *i*DeLUCS's performance in this domain.

For the synthetic datasets in (d), *i*DeLUCS attains near-parity with MeShCLust v3.0, securing an average accuracy of 98.5% when the expected number of clusters is given as a

Table 4.7: Comparison of the performance of *i*DeLUCS + HDBSCAN (*i*DeLUCS – auto) against MeShClust v3.0 clustering algorithms on the **long synthetic datasets** introduced by [89], using external evaluation metrics (homogeneity, completeness, adjusted Rand index ARI, normalized mutual information NMI, and unsupervised clustering accuracy ACC). The boldface indicates the best result. "MeShCLust" denotes MeShCLust v3.0 run with the option of automatically identifying the identity threshold parameter.

Dataset	Model	No. Clusters	Homogeneity	Completeness	NMI	ARI	ACC
LongTest-60	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	99	0.99	1.00	0.99	0.99	0.99
LongTest-70	MeShCLust	100	**1.00**	0.92	0.95	0.97	0.93
	*i*DeLUCS-auto	82	0.98	**1.00**	**0.99**	**0.98**	**0.97**
LongTest-80	MeShCLust	100	**1.00**	1.00	1.00	**1.00**	**1.00**
	*i*DeLUCS-auto	100	1.00	**1.00**	**1.00**	1.00	0.99
LongTest-90	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
LongTest-95	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
LongTest-97	MeShCLust	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	*i*DeLUCS-auto	100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

parameter and 97.3% with automatic cluster determination via HDBSCAN. This is slightly below MeShCLust v3.0's 99.3% average accuracy, highlighting *i*DeLUCS's competitive edge even with alignment-assisted non-parametric methods.

Overall, *i*DeLUCS has a robust performance across these very different types of datasets: small (113 sequences) or large (432,000 reads); real, simulated, or synthetic; at different taxonomic levels ranging from phyla to subtypes; with balanced clusters or with unbalanced clusters; with cluster number varying from 3 to 100 clusters; comprising long sequences (500,000 bp) or short sequences (650 bp); consisting of homologous sequences or of non-homologous sequences. On these datasets, the unsupervised clustering accuracy (ACC) obtained by *i*DeLUCS ranges from 78% to 100%, with an average accuracy of 90%.

4.4.4 Ablation studies

We studied the impact of the specific changes in *i*DeLUCS over the previous pipeline. Our study assessed the performance of *i*DeLUCS with and without each change made across several non-simulated datasets under different scenarios. We removed each change introduced in *i*DeLUCS one at a time and averaged the accuracies over ten trials to establish

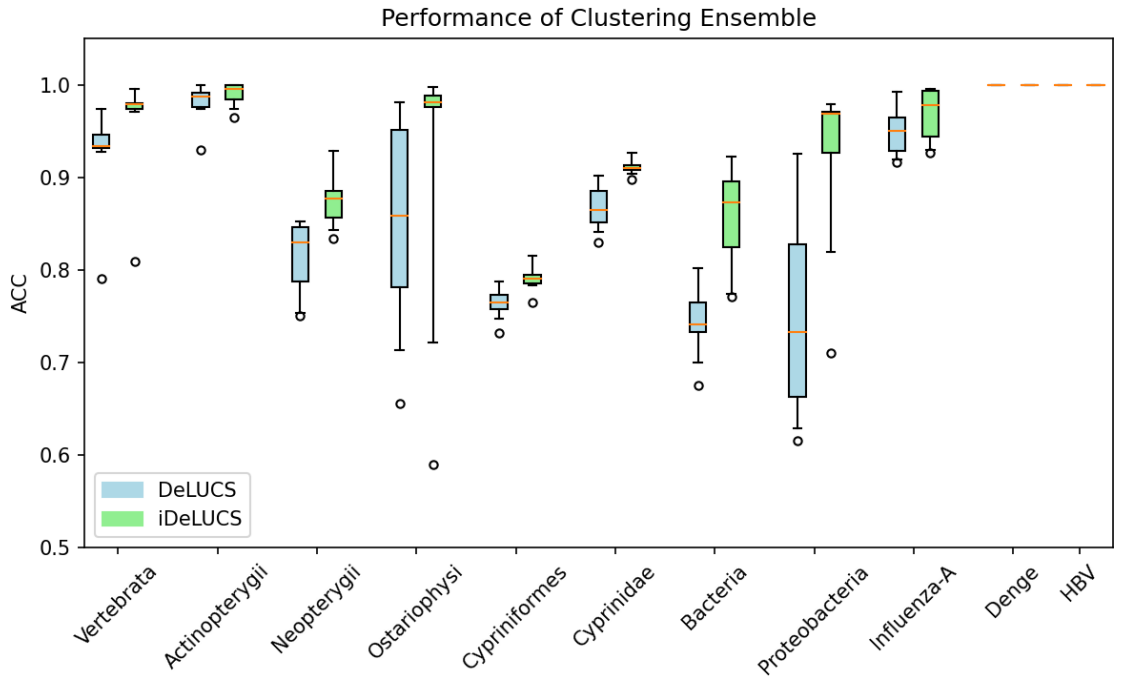
a consolidated performance measure. These values were then averaged across all datasets to determine a unified effectiveness measure for each configuration.

Table 4.8: Optimal performance metrics derived from ten trials for each non-simulated dataset across different scenarios, selecting the best result per dataset. These top performances were averaged across all datasets to establish a unified effectiveness measure for each configuration.

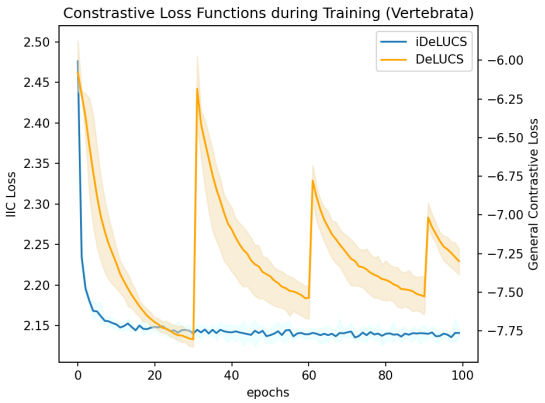
	Method	**Contrastive Loss**	**Clustering Ensemble**	**ACC**
1	*i*DeLUCS	✓	✓	0.89 ± 0.07
2		✓	✗	0.87 ± 0.11
3		✗	✓	0.84 ± 0.16
4	DeLUCS	✗	✗	0.84 ± 0.12

We first examine the effects of the improved contrastive loss by training the network using the new loss function on the new datasets and maintaining the naive majority voting scheme followed by DeLUCS. Our results show that this change alone leads to an improvement of ∼3% over the DeLCUS pipeline, as seen in the second row of Table 4.8. Interestingly, removing the contrastive term in the loss function while keeping the information-theoretic clustering ensemble was only marginally better than DeLUCS on average. However, our findings indicate that the newly introduced clustering ensemble provided lower variance when compared to majority voting on the 11 benchmarking datasets introduced in the previous chapter, as illustrated in Figure 4.2-a).

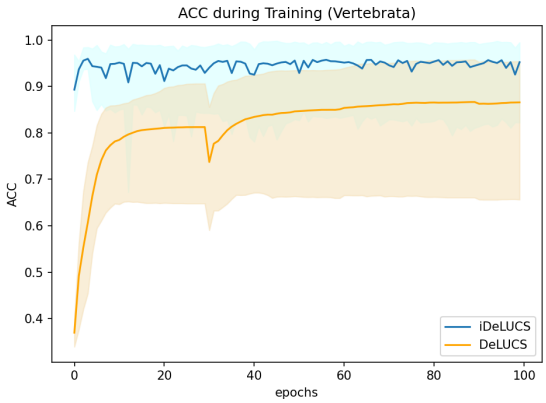
Furthermore, we observed that enforcing the consistency of the hidden representations in *i*DeLUCS provided robustness to local optima. Unlike the previous pipeline, *i*DeLUCS does not require the addition of external noise to the network parameters during training. The network learned an embedding where the representations of sequences in the same cluster were close to each other but distant from the representations of sequences in other clusters. Figure 4.2 b-c) illustrates the reduction in the variance of independent runs of the clustering algorithm. The figure aggregates the learning curves for fifty independently trained ANNs on the mtDNA Vertebrates (Table 3.1). The same behaviour is observed across all the eleven datasets presented in the previous chapter. *i.e.,* mitochondrial, bacterial and viral DNA. Overall, both incorporations reduced the variance for independent runs of the algorithm by $\sim 18\%$.



(a)



(b)



(c)

Figure 4.2: Comparison of the performance of *i*DeLUCS against the performance of DeLUCS on 11 benchmark datasets. (a) The box plot represents the performance of the clustering ensemble of *i*DeLUCS against the majority voting used in DeLUCS. Fifty models with five voters were trained over the eleven benchmark datasets using both strategies. (b) Contrastive loss as a function of the training epoch for 100 runs of the training algorithm on the Vertebrata dataset. (c) Unsupervised clustering accuracy as a function of the training epoch for 100 runs of the training algorithm on the Vertebrata dataset.

4.5 Software description

*i*DeLUCS is a standalone software tool that exploits deep learning capabilities to cluster genomic sequences. It is agnostic to the data source, making it suitable for genomic sequences taken from any organism in any kingdom of life. *i*DeLUCS assigns a cluster identifier to every DNA sequence present in a dataset while incorporating several built-in visualization tools that provide insights into the underlying training process and the composition of the datasets, as illustrated in Figure 4.3. *i*DeLUCS offers an evaluation mode to compare the dataset sequences' ground-truth label assignments (or hypothesized label assignments) with their discovered cluster labels. This is accompanied by a visual qualitative assessment of the clustering, using the uniform manifold approximation (UMAP, see [128]) of the learned lower dimensional embedding. Finally, *i*DeLUCS outputs confidence scores for all cluster-label predictions for enhanced interpretability. The software was developed using Python 3.9 and can be deployed with or without a graphics processing unit (GPU)

Note: At the moment of writing this dissertation, completely reproducible results are not guaranteed across PyTorch releases, individual commits, or platforms. Furthermore, results may not be reproducible between CPU and GPU executions, even when using identical seeds. That said, users may attempt to produce results similar to the ones obtained in this paper using the default parameters. Additional information about extra hyper-parameters and test scripts can be found in the Examples folder of the paper repository. All of the tests were performed on one of the nodes of the Beluga cluster of the Digital Research Alliance of Canada (16 x Intel Gold 6148 Skylake @ 2.4 GHz CPU, 32 GB RAM) with NVIDIA V100SXM2 (16 GB memory).

4.6 Conclusion

Overall, our analysis shows that *i*DeLUCS is an accurate and scalable clustering method, performant on datasets of long, homology-free DNA sequences, not tractable via alignment-based methods due to either lack of alignment or excessive time complexity. The modifications introduced by *i*DeLUCS are representative as it still outperforms other algorithms in clustering sizeable datasets of unlabelled DNA sequences while improving interpretability and speed. Future work is needed to systematically test and optimize *i*DeLUCS for datasets with short reads or datasets where more than 200 clusters are expected.

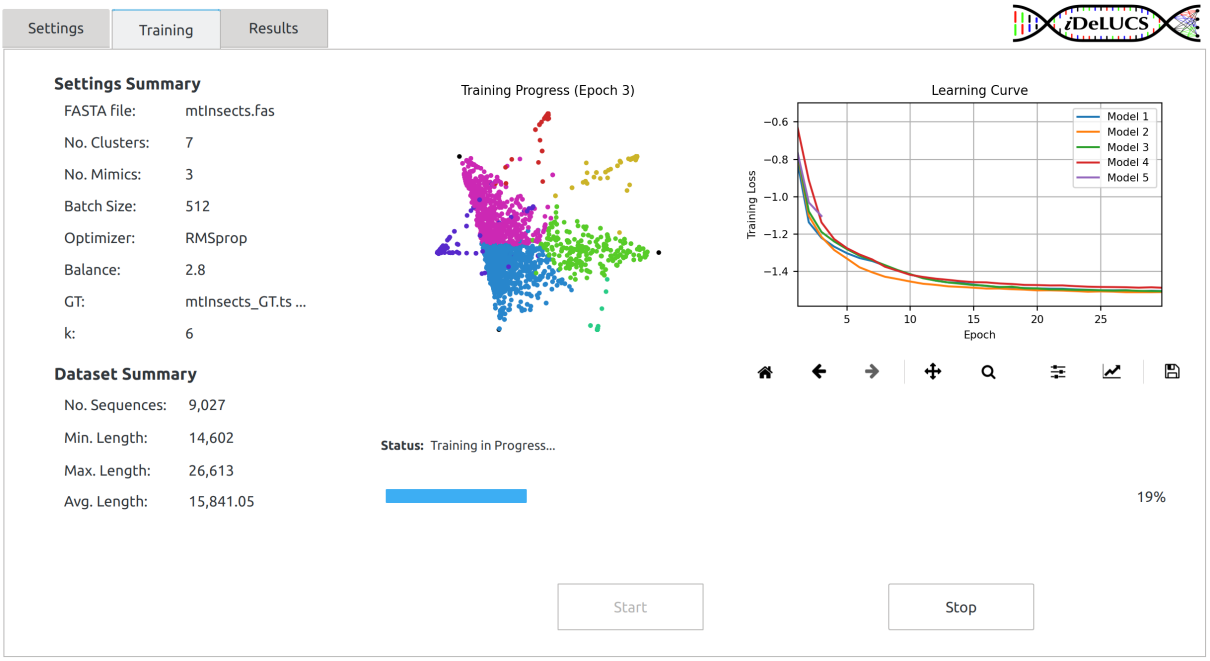


Figure 4.3: Snapshot of the training tab of *i*DeLUCS as it learns to cluster 9,027 mitochondrial genomes of insects into 7 different clusters. The left panel displays a summary of the main training parameters, as well as some statistics about the dataset under study. The center panel contains a qualitative assessment of the learning progress. The right panel contains a dynamic plot with the learning curves of the different models. Four models have been trained for thirty epochs each, and the training process of the fifth model is going through the third epoch.

Chapter 5

Case study: Discovering traces of convergent evolution in the genomic signatures of microbial extremophiles

This chapter is an adaptation and extension of a paper [135] of which I was co-author, titled "*Environment and taxonomy shape the genomic signature of microbial extremophiles.*" We leverage alignment-free methodologies and machine learning algorithms to unearth evidence suggesting that adaptations to extreme temperatures and pH conditions leave a discernible mark on the genomic signatures of microbial extremophiles.

Section 5.1 outlines our study's context and goals, emphasizing the exploration of genomic analysis to understand extremophiles' adaptations. Section 5.2 details our approach, from dataset compilation (Section 5.2.1) and supervised sequence classification (Section 5.2.2) to unsupervised sequence clustering (Section 5.2.3), setting the stage for our analytical exploration.

Section 5.3 presents our analysis, which demonstrates the capabilities of supervised learning in identifying features of genomic adaptation, as detailed in Subsection 5.3.1. Further, it explores the application of both parametric and non-parametric clustering methods (Sections 5.3.2 and 5.3.3) to uncover genomic patterns and potential candidates for convergent evolution. The Discussion (Section 5.4) reflects on our findings' evolutionary significance, comparing them with existing literature and presenting some of their implications for understanding extremophiles' adaptability.

Section 5.5 concludes the chapter and summarizes our contributions to extremophile genomics, underscoring the importance of genomic analysis in revealing life's adaptability to extreme conditions and suggesting future research directions.

5.1 Introduction

It is hypothesized that all life forms on our planet originated from a common ancestor and that organisms in each domain branched at different times during evolutionary history [39, 48]. The now standard three-domain system classifies organisms into three domains based on evolutionary relatedness: Archaea, Bacteria, and Eukarya. Although extremophiles are present in all three domains, as illustrated in Figure 5.1, microbial ones (Bacteria and Archaea) have garnered special attention for their unique phenotypic traits and largely unexplored genomic landscapes. Investigating the genomic organization and diversity of extremophiles could shed light on their adaptation strategies and the evolution of life under extreme conditions, with potential implications for biotechnology and astrobiology [80, 153, 198, 217].

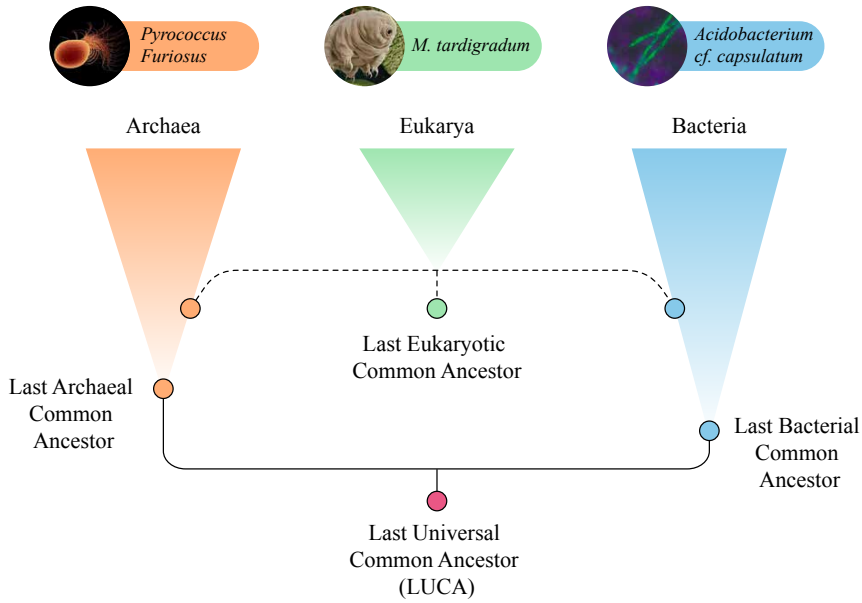


Figure 5.1: Illustration of the three-domain characterization of the tree of life, including examples of representative extremophiles from each domain: *M. tardigradum* in phylum *Tardigrada* as representative of Eukaryotes, *P. furiosus* in phylum *Euryarchaeota* as a representative of Archaea, and *Acidiobacterium cf capsulatum* in phylum *Acidobacteriota* as a representative of Bacteria

.

These organisms have developed many structural, biochemical, and metabolic strategies

to ensure cell viability in inhospitable environments. These adaptations, potentially resulting from convergent evolution across different taxa, manifest both at proteomic and genomic levels. At the proteomic level, organisms in extreme environments exhibit a significant amino acid compositional bias, attributed to convergent proteomic adaptations [59, 173, 218]. Distinctive codon usage patterns have also been linked to the selective pressures exerted by extreme environmental conditions [117, 178, 205]. Finally, at the genomic level, the genomes of microbial extremophiles are still subject to superimposed large changes in composition due to mutational biases [57, 58].

The concept of genomic signatures, defined as pervasive quantitative measures along a genome that discriminates between different species, has been effectively employed in genome analysis, comparison, and sensitive taxonomic classification, even in unsupervised contexts [43, 95, 97, 116, 162, 184, 225] (See Section 2.1.4). These studies have reinforced the notion of a robust phylogenetic signal within genomic signatures, offering an alternative perspective to alignment-based taxonomic analyses.

In this chapter, we set out to provide a comprehensive quantitative analysis suggesting that microbial extremophiles' adaptation to extreme temperatures or pH is reflected in their genomic signatures. Defined here as the k-mer frequency vector of a 500 kbp DNA fragment, representing a genome where k is a fixed positive integer ($1 \leq k \leq 6$), we investigate the genomic signatures across a dataset of 693 high-quality genomes from bacterial and archaeal organisms adapted to extreme conditions. Employing supervised machine learning, we first analyzed genomic signatures labelled with taxonomic or environmental category labels to explore the taxonomic and environmental components. The classification accuracies obtained support the presence of an environmental component and a well-established taxonomic component. Further, using interpretability tools on supervised learning algorithms enabled the identification of specific k-mers that are most relevant for environmental category classification.

The presence of the environmental component was also independently verified through unsupervised clustering analysis. By assessing the ability of various unsupervised algorithms to discern the taxonomic structure of unlabelled data, we identified several organisms with genomic signatures indicating convergent adaptations despite significant taxonomic divergences. Hyperthermophile bacteria and archaea, such as *Thermocrinis ruber*, *Pyrococcus furiosus*, *Thermococcus litoralis*, and *Pyrococcus chitonophagus*, emerged as exemplars of organisms whose genomic signatures were consistently grouped together across all machine learning analyses, regardless of their taxonomic disparities.

Overall, the results of machine learning analyses, corroborated in the exemplar cases by observations of shared characteristics of the isolating environments, suggest the existence of an *environmental component* that co-exists with a strong *taxonomic component* in the

genomic signatures of organisms living in extreme temperatures or extreme pH conditions. The work presented in [135] is among the most detailed examinations of prokaryotic extremophiles' genomic signatures, offering new insights into the coexistence of environmental and taxonomic components within the genomic signatures of organisms adapted to extreme conditions.

5.2 Materials and methods

5.2.1 Datasets

All the data were collected through a systematic literature search focused on identifying extremophilic microbes adapted to environments of extreme temperature and pH. The search was conducted on the PubMed Database (accessed September 2022) and Google Scholar (accessed September 2022) for primary research articles and reviews, and 768 microbial species or strains for which extremophilic characteristics were recorded were identified. Subsequently, these species/strains were identified in the Genome Taxonomy Database (GTDB; release R207 April 8, 2022, Accessed February 2023), the gold-standard database for taxonomy [197], and only GTDB species representative genomes with reported completeness of over 95%, and contamination of under 5% were selected. Species/strains were mapped to their identified extremophilic characteristic(s), along with genome assembly numbers provided by GTDB for each given organism. The extremophilic characteristic(s) was validated for each organism by searching PubMed with the given strain/species name and identifying a primary article/review or reliable BacDive database (accessed February 2023) entry to confirm the accuracy of the characteristic(s). Entries lacking consistent observations related to the growth characteristics of the respective microbe were removed from the dataset.

We used the following definitions, based on the Optimal Growth Temperature (OGT), respectively Optimal Growth pH (OGpH): Psychrophile (OGT of $< 20^{o}$C)[132], mesophile (OGT of 20-45^{o}C)[132], thermophile (OGT of $45-80^{o}$C)[132], and hyperthermophile (OGT of $> 80^{o}$C)[132], acidophile (OGpH $<$ pH 5)[132] and alkaliphile (OGpH $>$ pH 9)[132]. The dataset was then curated for 154 descriptors that comply with the temperature and pH intervals used in the above definitions. Fourteen entries could not be validated and were discarded from the dataset.

This selection process resulted in 693 annotated high-quality extremophile microbial genome assemblies. These high-quality assemblies were then used to form two datasets according to two extremophilic characteristic(s), as follows. The first dataset, called the

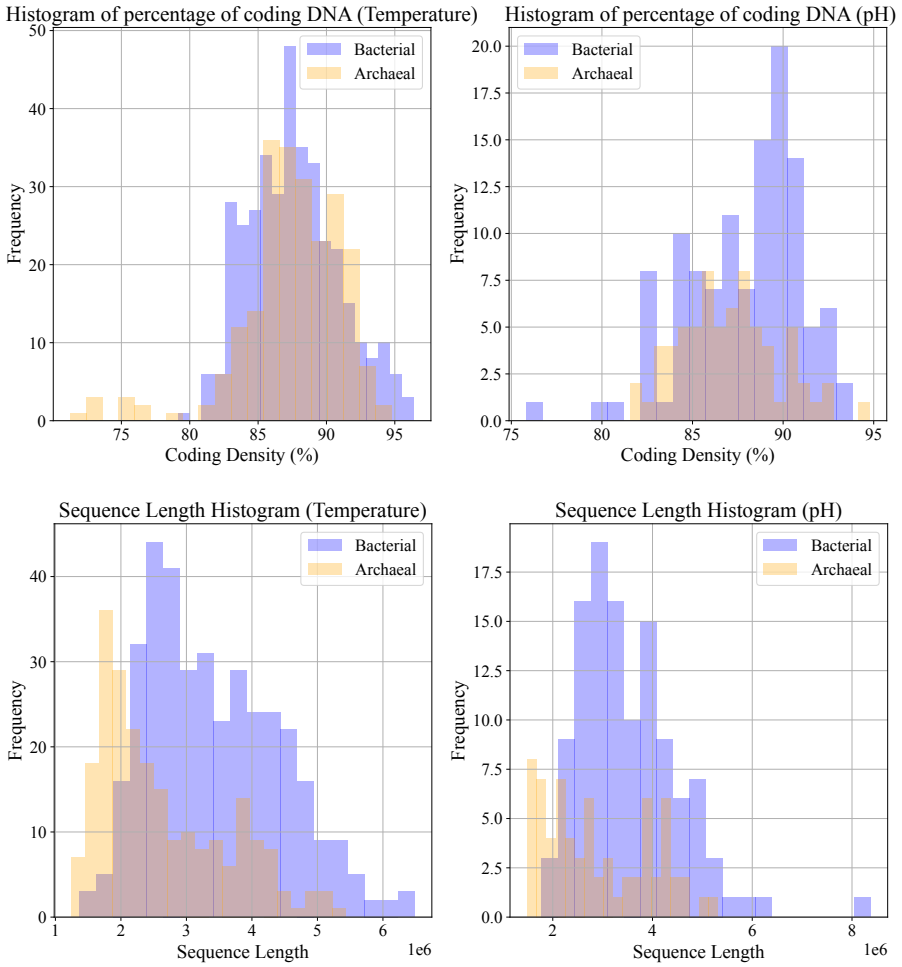


Figure 5.2: Histograms depicting the coding DNA density and sequence length across genomes of microbial extremophiles: Bacteria (blue) and Archaea (yellow), with brown representing an overlap between the two histograms. The figures are arranged by dataset type: Temperature (left panels) and pH (right panels). The top histograms illustrate the coding DNA density. On average, over 85% of the sequences consist of coding DNA, which could impact the presence of a genome-wide pervasive environmental component in the genomic signature. The bottom histograms correspond to the sequence length. These are used to select a suitable threshold for the maximum length of non-specific fragments for genomic signature analysis, given the notable differences in genome lengths between bacterial and archaeal extremophiles.

Temperature Dataset, is composed of 148 psychrophile genomes (8 archaeal, 140 bacterial), 190 mesophile genomes (84 archaeal, 106 bacterial), 183 thermophile genomes (67 archaeal, 116 bacteria), and 77 hyperthermophile genomes (70 archaeal, seven bacterial) for a total of 598 organism genomes (229 archaeal, 369 bacterial) (Table 5.1). The second dataset, called the pH Dataset, is composed of 100 acidophile genomes (39 archaeal, 61 bacterial) and 86 alkaliphile genomes (30 archaeal, 56 bacterial) for a total of 186 organisms (69 archaeal, 117 bacterial) (Table 5.2). Note that 91 organisms were identified to belong to both the Temperature Dataset and the pH Dataset.

Selecting a genomic fragment s to represent an organism's genome is a process that has to consider several factors, including fragment length, taxonomic level, and computational complexity of the algorithms used. For methods that rely on k-mer frequency for sequence classification, some studies [20, 42] suggest the relation $k = log_4(\mid s \mid)$, where $\mid s \mid$ is the minimum length of sequence s that is necessary, in theory, to obtain statistical significance. However, in practice, longer sequences are needed. For example, another study [134] used sequence length of 500 kbp in conjunction with $k = 6$ to cluster bacterial sequences at the family level, even though in theory a length of 4,096 bp would have sufficed for this value of k. Each genome (assembly) was represented by a single, arbitrarily selected 500 kbp DNA fragment. The values for k, namely $1 \leq k \leq 6$, were empirically chosen to balance the trade-off between classification accuracy and computational complexity and to explore multiple scales of the k-mer analysis.

A DNA fragment was arbitrarily selected for each DNA genome/assembly. First, the contigs of the assembly were sorted by length, from the longest to the shortest. Then, if the longest contig was longer than 500 kbp, a 500 kbp fragment was randomly selected as the *representative DNA sequence* for that genome. Otherwise, the sorted contigs were concatenated one by one until the desired length of 500 kbp was reached, which became the DNA representative sequence for that genome. The k-mers were counted starting from the beginning of the representative DNA sequence by using a sliding window with step size 1. To avoid spurious k-mers that could arise from the concatenation of contigs, the N character was added as a separator between contigs or contig fragments. Still, no k-mers that contain N were considered when calculating k-mer counts. Also, note that the inserted letters N were not counted towards the length of the DNA sequence representing each genome/assembly. To eliminate the variable of the strand orientation of the uploaded DNA sequences, the final k-mer frequency vector of a sequence was computed as the sum between the vector of its k-mer counts and the corresponding vector of k-mer counts of its reverse complement. [103] In the remainder of the chapter, a k-mer and its reverse complement will be considered to be indistinguishable. Only the *canonical* k-mer of a pair (the first, in alphabetical order, of the two reverse complementary k-mers) will be listed.

Table 5.1: Composition of the Temperature Dataset: 598 DNA fragments from microbial genomes/species (369 DNA fragments from bacterial genomes, and 229 DNA fragments from archaeal genomes).

Domain	**Temperature Category**	**# Phyla**	**# Classes**	**# Orders**	**# Families**	**# Genera**	**# Species**
Archaea	Psychrophiles	2	4	4	5	7	8
	Mesophiles	4	6	7	20	45	84
	Thermophiles	6	11	14	21	41	67
	Hyperthermophiles	5	6	8	15	31	70
Bacteria	Psychrophiles	4	4	6	13	19	140
	Mesophiles	3	3	6	10	14	106
	Thermophiles	15	19	24	27	47	116
	Hyperthermophiles	5	5	5	5	5	7

Table 5.2: Composition of the pH Dataset: 186 DNA fragments from microbial genomes/species (117 DNA fragments from bacterial genomes and 69 DNA fragments from archaeal genomes).

Domain	**pH Category**	**# Phyla**	**# Classes**	**# Orders**	**# Families**	**# Genera**	**# Species**
Archaea	Acidophiles	4	5	7	11	24	39
	Alkaliphiles	2	5	5	9	18	30
Bacteria	Acidophiles	10	12	13	13	32	61
	Alkaliphiles	12	14	25	30	36	56

5.2.2 Supervised machine learning for sequence classification

To test the hypothesis of the existence of an environmental component in the genomic signature of microbial extremophiles, the two previously described datasets (Temperature and pH) were classified using supervised machine learning algorithms, and the average accuracy of each classification was computed. For each dataset, computational experiments were performed using six different classifiers and different values of k, as detailed below. In addition, for each computational experiment, three different scenarios for labelling the training dataset were analyzed as follows:

(1) All DNA sequences used in training were labelled taxonomically by their domain

(bacteria or archaea),

(2) All DNA sequences used in training were labelled by their environment category (psychrophile, mesophile, acidophile, etc.),

(3) All DNA sequences used in training were labelled with pseudo-labels sampled from a discrete uniform distribution. More specifically, sequences in the Temperature Dataset are given a random pseudo-label sampled from $Unif(0, 3)$ because there are four possible environmental labels in this dataset. Respectively, sequences in the pH Dataset are given a random pseudo-label sampled from $Unif(0, 1)$ because there are two possible environmental labels in this dataset. This third scenario was introduced as a control, and it was expected to result in predictions of the correct pseudo-labels with probabilities equal to the sampling probability for each dataset.

The six classifiers used for these classification tasks were selected as representative algorithms of four main categories in the classification of DNA sequences. Support Vector Machines (SVM) were selected as a representative of *Kernel Methods*, with a radial basis function kernel [200]. Random Forest was selected to represent *Tree-Based Methods*, with the Gini index as the classification criteria [76]. The third algorithm was an *Artificial Neural Network (ANN)*, with a simple and versatile architecture consisting of two fully connected layers, Linear (512 neurons) and Linear (64 neurons), each one followed by a Rectified Linear Unit (ReLU) and a Dropout layer with a dropout rate of 0.5. Lastly, a *Digital Signal Processing* framework[162] was considered, whereby pairwise distances between numerical representations of DNA sequences are computed and then used in conjunction with *Linear Discriminant* (MLDSP-1), with *Quadratic SVM* (MLDSP-2), or with *Subspace Discriminant* (MLDSP-3) machine learning algorithms.

Two different types of computational experiments were performed for each combination of (a) the two datasets (Temperature and pH), (b) the supervised machine learning classifier (six classifiers), (c) the value of k ($1 \leq k \leq 6$), and (d) training data labelling (taxonomy, environment category, random).

In the first type of experiment, called *restriction-free*, the predictive power of the algorithms was tested using standard stratified 10-fold cross-validation, as follows. The dataset was split into ten distinct subsets, called *folds*, and a model was trained using 9 of the folds as training data; the resulting model was validated on the remaining part of the data (i.e., it was used as a validation set to compute a performance measure such as accuracy). The performance measure reported by 10-fold cross-validation was calculated as the average of the classification accuracy for each of the ten possible validation sets.

The second type of experiment, called *restricted*, or *non-overlapping genera*, was designed to address the possibility that a contributing taxonomic component may influence a correct environment category label classification. For example, one goal was to ensure that a DNA sequence was not classified as a hyperthermophile simply due to its similarity to DNA sequences of the same genus that happened to belong to the same hyperthermophile category. To this end, we adopted a grouped 10-fold cross-validation approach, whereby all sequences of the same genus appeared in exactly one fold. At the same time, to align with the principles of stratified cross-validation, the distribution of the labels in each fold was kept the same as the distribution of the corresponding labels in the entire dataset. In this *restricted* (*non-overlapping genera*) scenario, if a DNA sequence is in the test set, then no other sequence of the same genus is present in the training set. This approach attempts to disentangle, at the genus level, the taxonomic component from the environmental component of the genomic signature.

As an independent method for assessing the environmental component of the genomic signature, we employed interpretability tools for machine learning methods. Global interpretability tools were preferred as they help understand the general mechanisms in the data through a global importance measure. Given the high correlation between the k-mers, the mean decrease in impurity (MDI) for Random Forest was selected as a k-mer global importance measure and then used to learn the actual k-mers relevant to the environment category classification. (This measure was preferred over the widely adopted global-agnostic Permutation Feature Importance method, as that method is not suitable for handling highly correlated features [187].) The methodology used to determine the relevant k-mers is as follows. First, a one-vs-all classifier was trained for each environment category in the dataset using stratified 10-fold cross-validation. Second, the MDI algorithm was used to compute the global importance of each k-mer in each fold, and the average taken over all folds was used to create a ranked list of k-mers in decreasing order of their contribution. Finally, for each environment category, the "most relevant subset of k-mers" was computed, defined as the subset of the ranked k-mer list that was sufficient to classify the dataset with the same classification accuracy as when *all* k-mers were used in that classification.

5.2.3 Unsupervised learning for sequence clustering

In unsupervised learning, no labels are provided for the DNA sequences in the dataset, and various algorithms are used to cluster similar genomic signatures and explore the structure of the space of the genomic signatures in the dataset.

Two groups of tests with unsupervised learning algorithms were performed: parametric clustering algorithms (that take the number of expected clusters as an input parameter)

and non-parametric clustering algorithms (that automatically determine the number of clusters). In the first group, four parametric clustering algorithms were used: K-means, GMM, K-medoids, and DeLUCS [134]. The computation of the cluster label assignments for each sequence in the Temperature and the pH Datasets was performed with various values of the parameter `n_clusters` (the expected number of clusters) in each algorithm, `n_clusters` $\in \{2, 4, 8\}$ for the Temperature Dataset, and respectively `n_clusters` $\in \{2, 4\}$ for the pH Dataset, based on the number of potential true clusters in each dataset.

For each dataset, the strength of each of the two components of the signature (taxonomic, environmental) was assessed by comparing the clustering accuracies in two scenarios, the first where the clustering was evaluated against the true taxonomic groups and the second when the clustering was assessed against the true environment category groups. The performance was evaluated in each case using the unsupervised clustering accuracy metric in equation 2.26.

In the second type of test, we assessed whether non-parametric clustering algorithms can discover the clusters of each dataset at the lowest possible taxonomic level (genus). For this purpose, we used two non-parametric clustering algorithms, HDBSCAN [127] and iterative medoids [150], combined with three different dimensionality reduction techniques: VAE, Deep Contrastive Learning (CL), and UMAP [128]. We also used *i*DeLUCS [136], which is semi-parametric, in the sense that its parameter `n_clusters` (herein = 300) represents an upper limit of the number of clusters found by the algorithm. These seven clustering algorithms were used to recover the lowest taxonomic groups. The following metrics were defined to assess the quality of the found clusters: the *completeness* of each cluster (defined as the number of occurrences of the most common genus present in the cluster, divided by the total number of sequences of that genus in the dataset), and the *contamination* of each cluster (defined as the number of sequences that belong to the most common genus in the cluster, divided by the cluster size). The overall quality of each clustering algorithm was then calculated as the total number of clusters that are at least 50% complete and at most 50% contaminated.

5.3 Results

5.3.1 Supervised machine learning analysis

Supervised classification by taxonomy, environment category, and random label assignment

We conducted several computational tests using supervised machine learning to classify genomic signatures from the Temperature and pH Datasets based on taxonomy labels, environment category labels, and randomly assigned environment category labels. These tests employed six machine learning algorithms over k-mer lengths $1 \leq k \leq 6$, under two scenarios: *(a) restriction-free* with stratified 10-fold cross-validation, and *(b) restricted* with stratified 10-fold cross-validation ensuring non-overlapping genera.

For the *restriction-free* case, summarized in Table 5.3, taxonomy label-based training achieved high classification accuracies, exceeding 97.49% and 94.18% for the Temperature and pH Datasets, respectively, for $k = 6$. Environment category label-based training yielded medium-high accuracies, with over 77.59% for the Temperature Dataset and 84.95% for the pH Dataset. Random label assignments, as expected, resulted in low accuracies, not surpassing 28.09% and 50.06% for the Temperature and pH Datasets, respectively.

In the *restricted* case, detailed in Table 5.4, taxonomy label-based classifications maintained high accuracies over 95.30% and 91.90% for the Temperature and pH Datasets, respectively. Environment label-based classifications showed a dip to medium accuracies for the Temperature Dataset (61.90%) and medium-high for the pH Dataset (79.24%). Random label assignments remained low, aligning with the probabilities of environment category labels.

In both the restriction-free and the restricted cases, the classification of genomic signatures for $k = 1$ corresponds precisely to a classification based on the G+C content of the sequences (this is due to k-mers being counted from a DNA fragment together with its reverse complement). As seen from Table 5.3, the supervised classification accuracies for $k = 1$ were relatively low for taxonomic classifications and even lower for the environment category classifications. These results suggest that previous observations[173] of high G+C content of archaeal tRNA sequences being correlated with DNA stability in high-temperature environments ($\geq 60°C$) may not generalize to pervasive genomic signatures and larger datasets. This inference is also supported by the single nucleotide composition summary for the datasets (Figure 5.3).

Overall, we first note that the classification accuracy improved with higher values of k for both datasets. Second, we observe that for both datasets, the classification

accuracies obtained when using a random label assignment were approximately equal to the probabilities that a sequence had one of the environment category labels (around 25% in the case of the four temperature labels and around 50% in the case of the two pH labels). Third, note that the classification accuracies in the restricted scenario were slightly lower than in the restriction-free scenario for both the taxonomic and the environment category classifications. This decrease could be partly attributed to the reduction in the amount of training data in the restricted scenario. This being said, even in the restricted scenario, the

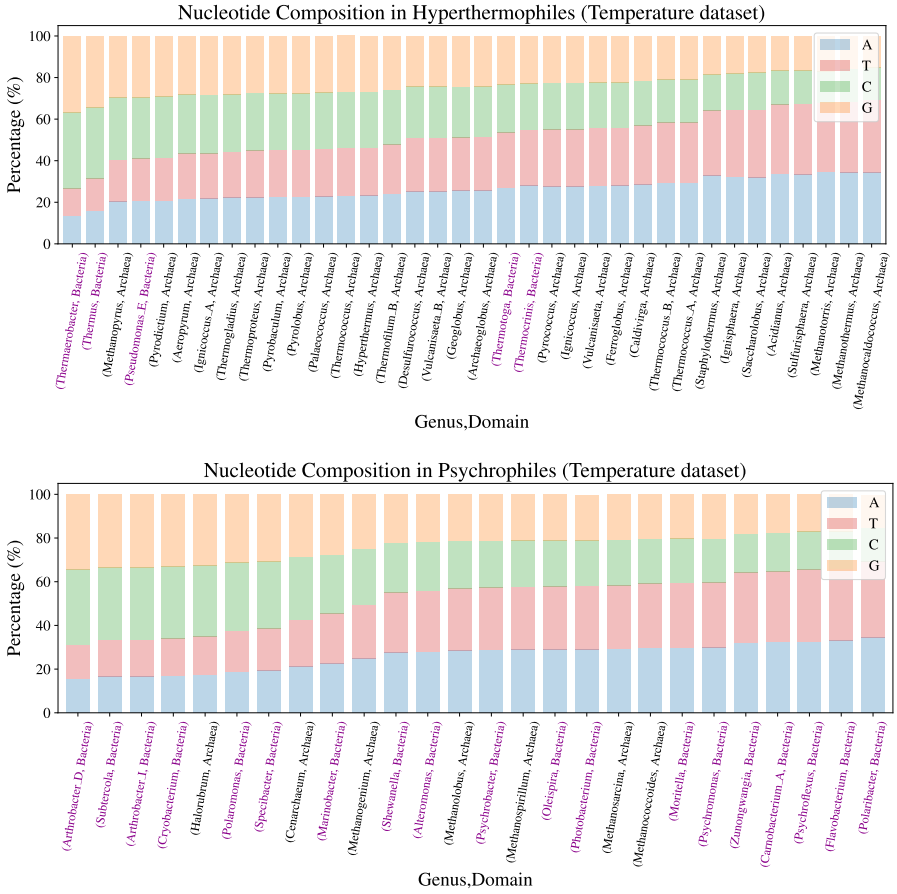


Figure 5.3: Single nucleotide composition of the sequences in the temperature Dataset, separated by extremophile environment: Hyperthermophile environment (top) and psychrophile environment (bottom). The nucleotide composition is averaged over the different genera, and the color of each genus and domain pair represents the specific domain, either bacteria (black) or archaea (magenta).

environment category classification accuracies were significantly higher than those for the random label assignment scenario.

These supervised machine learning classification experiments suggest the presence of an environmental component in the genomic signature of temperature and pH microbial extremophiles, able to provide discriminating power for k-mer values $3 \leq k \leq 6$. This environmental component of the genomic signature appears to co-exist with a more robust taxonomic component, providing discriminating power for k-mer values $2 \leq k \leq 6$.

Sets of k-mers relevant to environment category classifications computed by interpretability tool of supervised learning algorithm

Of the six supervised classifiers used in the previous section, in this section, we use the Mean Decrease in Impurity (MDI) algorithm for the Random Forest classifier to compute a global measure of feature importance. This interpretability tool provides insight into each feature's relative contribution (k-mer) to the successful classification.

To this end, we first conducted 10-fold cross-validation on a one-vs-all Random Forest classifier, achieving specific accuracy for each environment category. In the four computational experiments associated with the Temperature Dataset, the psychrophile category was correctly separated from the other sequences in the Temperature Dataset with 86.31% accuracy, the mesophile category with 71.14% accuracy, the thermophile category with 75.22% accuracy, and the hyperthermophile category with 89.62% accuracy. Similarly, in the two computational experiments associated with the pH Dataset, the alkaliphile category was classified with 86.45% accuracy and the acidophile category with 83.76% accuracy.

We then used the trained models obtained in these computational experiments in conjunction with Random Forest's interpretability tool, the MDI algorithm, to compute a *global importance* measure for each k-mer (for $k = 6$, the maximum value analyzed) to determine their relative contribution to the one-vs-all environment category classification. This global importance can be visualized using the $fCGR_k$ to identify potential patterns, as seen in Figure 5.4. A visual inspection of Figure 5.4 suggests that the set of 6-mers that is relevant for distinguishing DNA sequences from a given environment category from the rest of the dataset (darker k-mers) is specific to that environment category.

To confirm these findings and supplement the analysis with previous observations on codon usage patterns and amino acid compositional biases in extremophiles, we also examined the value $k = 3$. Note that not all the 3-mers identified by our method as relevant to the classification are codons because 3-mers are not counted only from coding sequences or translation frames. For each environment category, the MDI algorithm was used to

Table 5.3: Classification accuracies of six supervised learning classifiers trained on the Temperature Dataset and pH Dataset, in the *restriction-free* scenario, for three different label assignments (taxonomy, environment category, and random label assignment), and values of $1 \leq k \leq 6$. The classification accuracy in each cell is calculated using standard stratified 10-fold cross-validation.

Dataset	**k-value**	**Class Labelling Type**	**Classification Model Accuracy (%)**					
			RBF SVM	**Random Forest**	**ANN**	**MLDSP-1**	**MLDSP-2**	**MLDSP-3**
Temperature	$k=1$	Taxonomy	62.88	53.87	62.21	47.99	54.85	59.03
		by Environment	39.97	35.29	38.65	26.92	32.27	31.44
		Random	22.26	29.42	31.77	27.59	26.92	27.59
	$k=2$	Taxonomy	96.65	95.14	96.14	86.79	92.64	86.79
		by Environment	74.58	76.91	74.42	46.49	68.06	46.32
		Random	23.25	28.10	27.09	26.42	25.08	25.75
	$k=3$	Taxonomy	98.82	97.99	97.32	92.64	96.82	92.64
		Environment	82.11	81.59	75.41	71.91	74.58	71.24
		Random	23.58	25.08	27.76	25.59	26.09	24.58
	$k=4$	Taxonomy	99.50	98.33	98.66	98.16	97.16	98.16
		Environment	83.29	84.11	82.28	78.43	75.08	80.43
		Random	25.06	23.74	27.59	25.42	26.92	23.58
	$k=5$	Taxonomy	99.50	98.16	99.33	97.32	97.32	98.16
		Environment	83.27	84.76	83.29	69.23	77.26	81.77
		Random	24.08	20.23	23.07	26.09	25.42	24.25
	$k=6$	Taxonomy	99.50	98.50	99.33	99.16	97.49	98.83
		Environment	83.46	83.94	84.12	79.60	77.59	82.44
		Random	27.24	22.91	26.58	28.09	25.59	24.25
pH	$k=1$	Taxonomy	65.2	66.70	62.37	52.69	56.99	58.06
		by Environment	56.52	58.10	51.14	54.30	53.23	54.30
		Random	51.20	53.39	50.53	49.46	53.23	50.54
	$k=2$	Taxonomy	95.15	93.48	95.09	84.95	91.40	84.41
		by Environment	87.72	83.33	85.00	80.65	82.26	81.72
		Random	51.14	52.72	51.67	54.84	52.69	55.91
	$k=3$	Taxonomy	97.34	94.09	96.78	94.62	96.24	94.62
		Environment	90.94	90.94	90.38	81.18	83.87	80.11
		Random	44.15	52.72	55.91	54.84	46.77	44.62
	$k=4$	Taxonomy	97.87	96.29	96.81	93.01	95.16	97.85
		Environment	90.44	88.80	91.58	84.95	86.02	89.78
		Random	49.42	47.84	49.01	44.62	44.62	47.85
	$k=5$	Taxonomy	98.42	96.81	95.79	95.70	96.24	98.92
		Environment	91.55	88.30	87.81	88.17	86.02	90.32
		Random	55.35	53.77	52.13	48.39	46.24	46.24
	$k=6$	Taxonomy	98.42	94.71	94.18	98.92	96.77	98.39
		Environment	91.99	88.30	86.70	92.47	84.95	92.47
		Random	47.81	49.06	50.06	50.00	45.70	46.77

Table 5.4: Classification accuracies of six supervised learning classifiers trained on the Temperature Dataset and pH Dataset, in the *restricted* scenario, for three different label assignments (taxonomy, environment category, and random label assignment), and values of $1 \leq k \leq 6$. The classification accuracy in each cell is calculated using stratified 10-fold cross-validation with *non-overlapping genera*.

Dataset	**k-value**	**Class Labelling Type**	**Classification Model Accuracy (%)**					
			RBF SVM	**Random Forest**	**ANN**	**MLDSP-1**	**MLDSP-2**	**MLDSP-3**
Temperature	$k = 1$	Taxonomy	60.05	49.49	58.99	50.20	53.30	58.50
		by Environment	30.87	29.72	26.38	23.70	30.80	31.30
		Random	23.91	25.12	25.15	24.20	23.40	28.30
	$k = 2$	Taxonomy	94.11	91.12	93.79	85.80	90.80	85.60
		by Environment	57.30	53.75	54.99	33.30	48.20	33.10
		Random	22.59	27.56	25.80	24.20	24.40	24.10
	$k = 3$	Taxonomy	98.82	95.13	97.14	87.00	94.60	87.00
		Environment	65.57	63.25	58.10	44.80	53.30	44.50
		Random	24.93	21.40	26.12	26.60	27.10	26.60
	$k = 4$	Taxonomy	99.16	96.13	97.81	95.00	94.50	97.20
		Environment	70.55	63.75	63.29	54.00	56.70	59.90
		Random	25.94	26.60	27.22	26.40	26.90	25.40
	$k = 5$	Taxonomy	99.16	96.13	98.82	92.50	94.50	97.20
		Environment	72.21	64.13	66.89	50.00	62.70	65.40
		Random	26.74	23.23	22.49	24.20	26.80	26.40
	$k = 6$	Taxonomy	99.16	96.47	97.81	99.20	95.30	98.00
		Environment	74.17	65.48	67.88	61.90	64.70	67.90
		Random	24.20	26.74	24.59	24.90	24.10	27.90
pH	$k = 1$	Taxonomy	65.09	67.31	62.37	51.10	50.50	58.60
		by Environment	53.30	49.91	47.75	51.10	52.70	59.10
		Random	41.78	55.89	48.60	47.80	50.00	52.70
	$k = 2$	Taxonomy	92.98	90.29	94.09	79.60	86.00	79.60
		by Environment	75.09	75.15	82.66	80.60	79.60	81.20
		Random	51.52	54.14	45.79	55.90	46.80	55.90
	$k = 3$	Taxonomy	97.37	93.54	96.78	88.70	92.50	88.20
		Environment	79.24	86.73	84.04	73.70	76.30	74.20
		Random	54.91	48.57	55.96	43.50	54.30	44.10
	$k = 4$	Taxonomy	97.37	96.20	96.78	88.20	92.50	94.10
		Environment	81.43	83.51	85.61	73.10	79.60	80.60
		Random	46.83	41.74	47.31	52.70	48.40	49.50
	$k = 5$	Taxonomy	97.89	97.28	96.23	94.60	92.50	96.80
		Environment	80.91	88.83	83.01	77.40	79.60	83.90
		Random	46.73	54.83	52.44	46.80	50.00	50.00
	$k = 6$	Taxonomy	98.42	96.23	95.73	97.30	91.90	96.80
		Environment	83.54	86.70	79.24	81.70	80.10	86.60
		Random	53.15	48.69	55.62	47.30	50.50	52.70

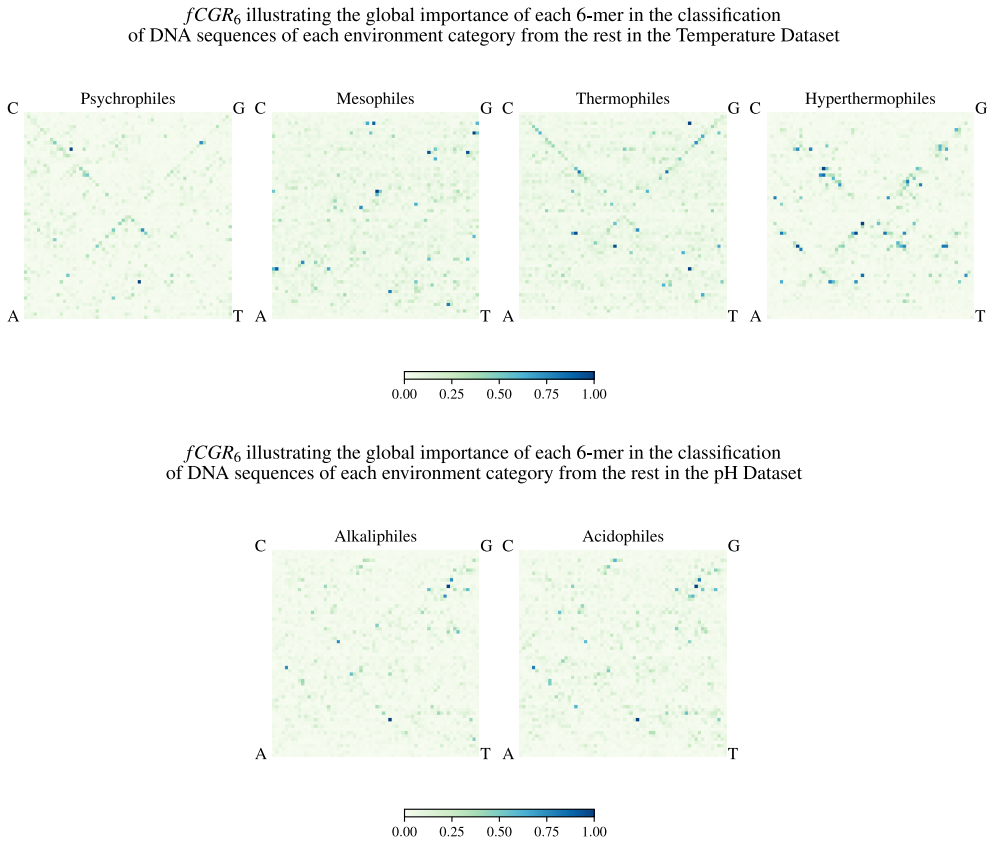


Figure 5.4: Frequency chaos game representation ($fCGR_k$) of the global importance of various 6-mers in the classification of DNA sequences of each environment category from the rest of the dataset. The top panel shows the $fCGR_k$ for the Temperature Dataset, and the bottom panel shows the $fCGR_k$ for the pH Dataset, both for $k = 6$. The colour and intensity of each pixel represent the relative importance (relevance) of its corresponding 6-mer (dark blue pixels represent the most relevant 6-mers, etc., as described in the colour bar legend).

identify the specific 3-mers that are relevant for each of the one-vs-all Random Forest environment category classifications.

To investigate further the concept of *relevance* and explore its connection with the over-representation and under-representation of codons/amino acids as described in the literature, we computed the histograms of the 3-mers' deviation from the dataset mean for

each dataset and environment category. Figures 5.5 and 5.6 display these histograms and single out (in green) the 3-mers relevant for each environment category in the Temperature Dataset (Figure 5.5) and the pH Dataset (Figure 5.6). To complement this analysis, Table 5.5 and Table 5.6 list the sets of relevant 3-mers displayed in Figures 5.5 and 5.6, respectively, alongside with the relevant literature on biological observations of codon/amino acid compositional biases associated with extreme temperature and pH environments. Note that each set of relevant 3-mers is listed in an environment category panel in Figure 5.5 (Figure 5.6), ordered left-to-right alphabetically on the x-axis of the panel, corresponds to a set of relevant 3-mers in a matching environment category column in Table 5.5 (Table 5.6), ordered top-to-bottom alphabetically by the abbreviation of the amino acid they would encode if they were codons.

As seen in the tables, most of our findings regarding over- and under-representing 3-mers match existing observations in the literature about codon/amino acid bias in extremophiles' genomic sequences. Disagreements could be due to several factors. First, the 3-mers are not codons: they are counted from an arbitrarily selected 500 kbp DNA fragment representing a genome, and their frequency profile (the genomic signature) is quasi-constant along a genome. Thus, some 3-mers could be relevant for the one-vs-all temperature/pH category classification in ways unrelated to transcriptional or proteomic adaptations. Second, the fact that a 3-mer is relevant for a temperature/pH category indicates that it belongs to a set of 3-mers that *collectively* contribute to distinguishing sequences in that temperature/pH category from the rest of the dataset. In this sense, the concept of "relevant k-mer set" is more general, and the fact that a k-mer belongs to the relevant set of k-mers for classification does not necessarily imply that it is over- or under-represented in the genomic sequences of that environment category.

5.3.2 Parametric unsupervised clustering

The supervised learning computational experiments suggested the existence of an environmental component in the genomic signature of microbial extremophiles in both a restriction-free scenario and a restricted scenario where sequences from the same genus as the test sequence were absent from training.

It should be noted that the considered datasets are not comprehensive since the discovery and sequencing of genomes of extremophilic organisms is an ongoing complex process given the challenging environments in which they are found, which are difficult to reproduce to culture and further characterize microbial extremophiles [223]. In particular, the datasets' sparsity and sampling bias do not allow computational experiments in restricted scenarios at taxonomic levels higher than the genus level. This is because such restrictions could

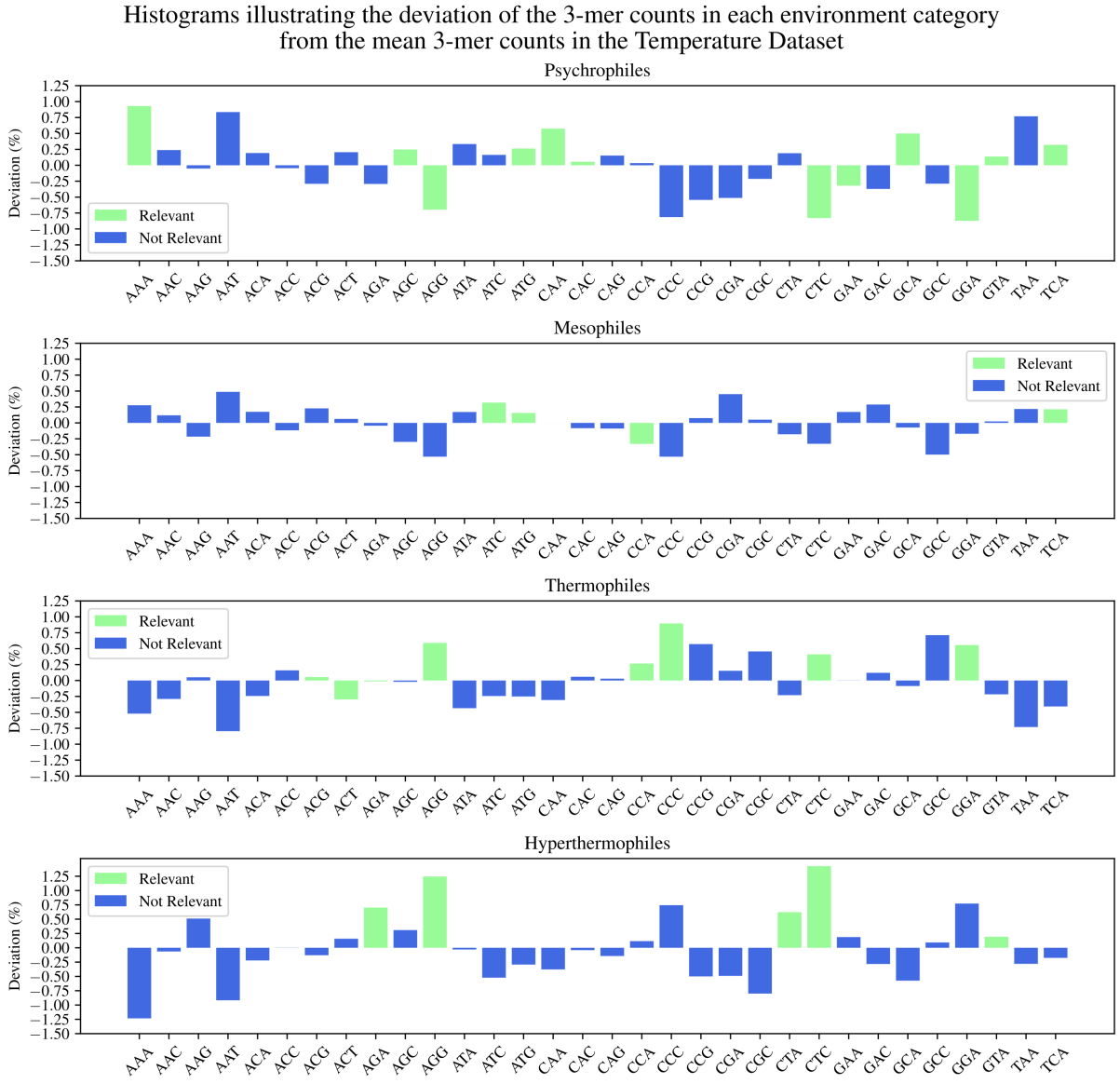


Figure 5.5: Histograms of the deviation of 3-mer counts in each environment category from the Temperature Dataset mean. A 3-mer and its reverse complement are considered to be indistinguishable, and only canonical 3-mers are listed. Relevant 3-mers for the one-vs-all classification are highlighted in green. The height of each bar represents the difference between a 3-mer's count in that temperature category and the mean of that 3-mer's counts over the entire Temperature Dataset (in percentage points).

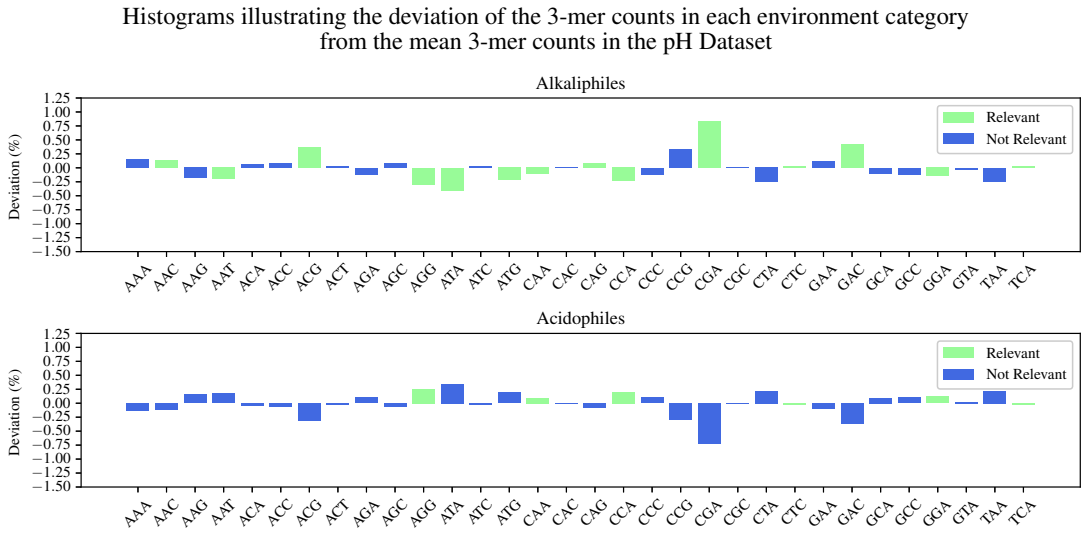


Figure 5.6: Histograms of the deviation of 3-mer counts in each environment category from the pH Dataset mean. A 3-mer and its reverse complement are considered to be indistinguishable, and only canonical 3-mers are listed. Relevant 3-mers for the one-vs-all classification are highlighted in green. The height of each bar represents the difference between a 3-mer's count in that pH category and the mean of that 3-mer's counts over the entire pH Dataset (in percentage points).

eliminate many of the labelled sequences from the cross-validation training sets, rendering them insufficient in size for supervised learning purposes.

To address this challenge, in this section, we explore the genomic signatures of the Temperature Dataset and pH Dataset through an unsupervised clustering approach. In unsupervised clustering, no taxonomic or environment category labels for DNA sequences are used during the entire learning process, and ground-truth labels are used exclusively to evaluate the quality of clustering (if applicable). In the first set of tests, we applied *parametric* unsupervised algorithms for the task of clustering both datasets with different values for the parameter `n_cluster` (the number of clusters). When compared to the highest taxonomic level (domain), the ACC measure (Equation 2.26) for the clustering assignments computed by each algorithm suggests that for the Temperature Dataset, all algorithms can partially cluster sequences according to their real taxonomic labels at `n_clusters` = 2, with *i*DeLUCS (68%) outperforming the others by a small margin (see Table 5.7 for accuracies). For the pH Dataset, all algorithms are unsuccessful at separating by the domain (see Table 5.8 for accuracies). For values of the parameter `n_clusters` greater

Table 5.5: Over- and under-representation of the relevant 3-mers, found by our method to be collectively associated with genomic signatures of temperature-adapted prokaryotic extremophiles. The symbol ↑ (↓) indicates over-representation (under-representation) of a 3-mer/codon. Matched arrows, e.g., $(\uparrow, \uparrow^{ref})$ indicate that our method and reference ref agree in their finding. Mismatched arrows indicate disagreement. See Supplementary Table S4 for details on the observations in biological literature.

Psychrophiles	**Mesophiles**	**Thermophiles**	**Hyperthermophiles**	**Corresponding Amino acid**
GCA (↑,↑ [22])				Ala
AGG (↓,↓ [167])		AGG (↑,↑ [218]) AGA (↑,↑ [218])	AGG (↑,↑ [41]) AGA (↑,↑ [41])	Arg
CAA (↑,↑ [173])				Gln
GAA (↓,↑ [16])				Glu
GGA (↓,↓ [22, 61, 166])		GGA (↑,↑ [41])		Gly
CAC (↑,↓ [167])				His
	ATC (↑,↓ [188])			Ile
CTC (↓,↓ [173])		CTC (↑,↑ [218])	CTC (↑,↑ [61]) CTA (↑,↑ [61])	Leu
AAA (↑,↓ [61])				Lys
ATG (↑,↑ [61])	ATG (↑,↑ [155])			Met
	CCA (↓,↓ [106])	CCA (↑,↑ [67]) CCC (↑,↑ [67])		Pro
AGC (↑,↑ [166]) TCA (↑,↑ [166])	TCA (↑,↑ [188])			Ser
		ACT (↓,↓ [67, 173]) ACG (↑,↓ [67, 173])		Thr
GTA (↑,↑ [22])			GTA (↑,↑ [41])	Val

than 2, the accuracy increases for both datasets. Still, the increase is more significant for the pH Dataset where the ACC increases by $\sim 30\%$, which suggests a good separation by environment category within each domain in the pH Dataset. Overall, the unsupervised clustering accuracy computed using taxonomic labels as ground truth is higher than when calculated using environment category labels as ground truth. This confirms the supervised machine learning results in the previous section, suggesting that the taxonomic component is stronger than the environmental component of genomic signatures.

In a second set of tests, six different *non-parametric* algorithms (the number of clusters is discovered by the algorithm instead of being given as a parameter) and the semi-parametric algorithm *i*DeLUCS were employed to cluster both datasets. Subsequently, all clusters

obtained from each algorithm were compared with GTDB labels at the genus level, hereafter referred to as *true genera*, and only those clusters meeting the predefined quality criteria ($> 50\%$ completeness and $< 50\%$ contamination, see Methods) were selected for evaluation.

The outcomes, presented in Figure 5.7, speak to the effectiveness of deep learning clustering methodologies in accurately recovering the true genera and illustrate the importance of choosing appropriate algorithms for specific datasets. For the datasets considered in this chapter, the VAE with the Iterative Medoids method (VAE+IM) demonstrated superior performance in recovering clusters that meet the predefined quality criteria. Specifically, VAE+IM successfully recovered 61 out of 93 true genera represented by more than two sequences in the Temperature Dataset and 31 out of 37 true genera represented by more than two sequences in the pH Dataset.

Based on this analysis, the five algorithms that were able to recover at least 20% of the total number of true genera were VAE+HDBSCAN, CL+HDBSCAN, VAE+IM,

Table 5.6: Over- and under-representation of the relevant 3-mers, found by our method to be collectively associated with genomic signatures of pH-adapted prokaryotic extremophiles. The symbol ↑ (↓) indicates over-representation (under-representation) of a 3-mer/codon. Matched arrows, e.g., $(\downarrow, \downarrow^{ref})$ indicate that both our method and reference *ref* agree in their finding. Mismatched arrows indicate disagreement. See Supplementary Table S5 for details of observations in biological literature.

Alkaliphiles	**Acidophiles**	**Corresponding Amino Acid**
AGG (↓,↑ [79]) CGA (↑,↑ [100])	AGG (↑,↓ [100])	Arg
AAC (↑,↑ [100, 142]) AAT (↓,↑ [100])		Asn
GAC (↑,↑ [100])		Asp
CAG (↑,↑ [100])		Gln
CAA (↓,↓ [100])	CAA (↑,↑ [100])	Glu
GGA (↓,↓ [100])	GGA (↑,↑ [100])	Gly
ATA (↓,↓ [100])		Ile
CTC (↑,↑ [100])	CTC (↓,↓ [100])	Leu
ATG (↓,↓ [100])		Met
CCA (↓,↑ [100])	CCA (↓,↑ [142])	Pro
TCA (↑,↓ [100])	TCA (↓,↓ [100])	Ser
ACG (↑,↑ [100])		Thr

Table 5.7: Accuracies (ACC) of the unsupervised clustering of the Temperature Dataset, for several parametric clustering algorithms, and several values of the pre-specified number of clusters. For each value of the number of clusters parameter, the unsupervised clustering accuracies are computed using the taxonomic labels as ground truth (top row) and the environment category labels as ground truth (bottom row).

No. Clusters	**Labelling**	**Unsupervised Clustering Accuracy (%)**			
		K**-means**	K**-medoids**	**GMM**	i**DeLUCS**
2	Tax	63.84	63.92	63.23	68.97
	Env	36.27	36.50	36.26	38.23
4	Tax	63.99	77.65	68.45	75.44
	Env	34.37	40.81	38.30	48.31
8	Tax	87.81	82.13	77.79	81.48
	Env	50.99	49.63	53.74	56.77

Table 5.8: Accuracies (ACC) of the unsupervised clustering of the pH Dataset for several parametric clustering algorithms and several values of the pre-specified number of clusters. For each value of the number of clusters parameter, the unsupervised clustering accuracies are computed using the taxonomic labels as ground truth (top row) and the environment category labels as ground truth (bottom row).

No. Clusters	**Labelling**	**Unsupervised Clustering Accuracy (%)**			
		K**-means**	K**-medoids**	**GMM**	i**DeLUCS**
2	Tax	52.22	52.66	51.08	56.72
	Env	50.89	50.94	51.04	50.53
4	Tax	78.69	80.45	76.72	87.43
	Env	63.56	74.23	67.81	75.59

UMAP+HDBSCAN, and *i*DeLUCS for the Temperature Dataset, respectively VAE+HDBSCAN, CL+HDBSCAN, VAE+IM, CL+IM, and *i*DeLUCS for the pH Dataset. These five algorithms were thus selected as sources of information for subsequent analysis since they performed best when compared to true genera groupings.

5.3.3 Non-parametric clustering: Finding candidates of convergent evolution

Following the selection of the five top-performing unsupervised clustering algorithms in the previous section, the clusters discovered by these algorithms were used in conjunction with a majority voting scheme to determine concrete "candidates" that is, concrete exemplars of taxonomically different organisms that were clustered together presumably due to the environmental component of their genomic signatures. This computational process identified a list of pairs of hyperthermophilic, alkaliphilic, and acidophilic *candidate sequences*, each belonging to a different taxonomic domain, which were nevertheless grouped together by the majority of the aforementioned unsupervised clustering algorithms.

Of these candidates, we then proceeded to select sequences for which the unexpected results of the clustering could be independently confirmed by *(i)* supervised machine learning for the prediction of environment category, by *(ii)* supervised machine learning for the prediction of the taxonomic labels, and by *(iii)* observations of shared characteristics of their isolating environments. In these experiments, thermophiles and hyperthermophiles were treated as part of a single environment category called "high-temperature" to enhance the rigour of the confirmation procedure, given the lack of definitive knowledge of the precise threshold separating these two environment categories from each other.

The goal of the experimental design was to devise challenging scenarios that would demonstrate the presence of the environmental component in the genomic signature of each candidate. To this end, for each candidate sequence to be tested, a challenge training set was created by selecting all DNA sequences of organisms from the opposite domain (i.e., archaea or bacteria) and sequences within the same domain but under a different environment category. The classifiers were then trained to perform two different tasks.

In experiments *(i)*, a classifier was trained to predict the environment category of a candidate test sequence, as follows. For instance, if the test sequence was of a hyperthermophilic bacteria, the training set comprised all archaeal sequences (different domain) and mesophilic and psychrophilic bacterial sequences (same domain, different environment category). The objective was to determine if the hyperthermophilic bacterial test sequence would be assigned the correct label "high-temperature," despite the training set's absence

of high-temperature bacterial sequences. If this were the case, it would indicate that the correct temperature label assignment was due to the similarity of this bacterial sequence to other high-temperature archaeal sequences in the dataset, further suggesting that the environmental component overrides the taxonomical component in the genomic signature of the candidate sequence.

In experiments *(ii)*, a classifier was trained to predict the domain of each candidate test sequence, as follows. For example, if the candidate test sequence was of hyperthermophilic archaea, the training set comprised all bacteria sequences (different domain) and all the mesophilic and psychrophilic archaeal sequences (same domain, different environment category). The objective was to determine if the hyperthermophilic archaeal sequence would be assigned the incorrect label "Bacteria". If this were indeed the case, it would indicate that the assignment of this archaeal sequence to domain Bacteria was likely due to its similarity to the high-temperature bacterial sequences, further suggesting that the environmental component overrides the taxonomic component of the candidate sequence.

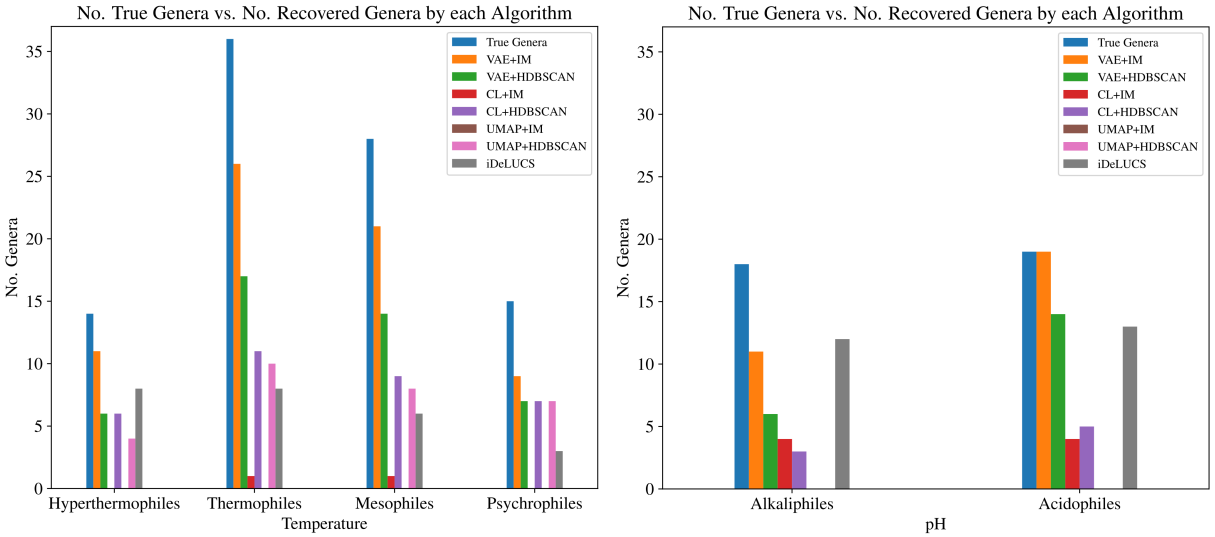


Figure 5.7: Number of true genera (blue) vs. the number of genera identified by seven clustering algorithms for each environment category in the Temperature Dataset (left), respectively the pH Dataset (right). Only true genera represented by more than two sequences in the respective dataset (Temperature or pH) are considered, and only clusters meeting the quality criteria are counted.

All candidate sequences generated by the unsupervised clustering experiment underwent both computational experiments *(i)* and *(ii)*. Of these, the following four sequences were assigned by the majority of the classifiers (SVM, Random Forest, ANN, MLDSP) to the correct environment category in experiment *(i)*, and to the incorrect domain in experiment *(ii)*: the bacterial sequence *Thermocrinis ruber* – Accession ID: GCA_000512735.1, and the three archaeal sequences, *Pyrococcus furiosus DSM 3638* (formerly *Pyrococcus sp000211475*) – Accession ID: GCA_000007305.1, *Thermococcus litoralis DSM 5473 (formerly Thermococcus litoralis NS-C)* – Accession ID: GCA_000246985.3, and *Pyrococcus chitonophagus (formerly known as Thermococcus chitonophagus)* – Accession ID: GCA_002214605.1. Note that the current release of Genome Taxonomy Database (GTDB release R214 April 28, 2023) defines *Thermococcus litoralis* as a strain type of species *Thermococcus alcaliphilus*. We refer to it as "*Thermococcus litoralis*" given its classification in the database version used to create the dataset.

Indeed, in experiments *(i)*, all environment-trained classifiers correctly predicted these four microbial sequences belonging to the high-temperature environment category. However, all genomic sequences used to train the classifier to predict temperature conditions were from a domain different from the test sequence. Moreover, in experiments *(ii)*, all taxonomy-trained classifiers erroneously predicted the genomic sequences of these microbial extremophiles as belonging to a different domain, likely due to their environmental characteristic.

For biological corroboration *(iii)*, a literature search was undertaken to correlate the candidate species to the context of phenotypic traits and the characteristics of the isolating environments. It was determined that few phenotypic traits were congruent between candidates, including gram-negative cell walls, OGpH falling within the neutrophilic range (pH 5.0 to 9.0) for each candidate, presence of intergenic sequences, and emissions of light hydrocarbons from the nearby environment[17, 26, 33, 54, 85, 86, 148, 152]. However, several more phenotypic traits display dissimilarities between organisms.The particular environments from which each species was initially isolated were analyzed in greater detail. It was found that two Joint Genome Institute's Genomes OnLine Database-derived (JGI-GOLD) ecosystem classifiers describe the isolating environment of all four species, as follows: ID 4027 for *P. furiosus* and *P. chitonophagus*, and ID 3991 for *T. litoralis* and *T. ruber* [18, 86, 141]. The descriptors for these classifiers are "aquatic marine hydrothermal vent" and "aquatic thermal hot springs," respectively [141].

Although these environments are classified differently by JGI-GOLD, as ID 3991 and ID 4027, respectively, the descriptors accurately describe these environments due to the presence of hydrothermal systems [152, 175]. Note that *T. litoralis* (ID 3991) has been recently isolated from the Guaymas Basin, albeit from a geographic site of the Guaymas

Basin that was different from the isolation site of *P. chitonophagus*[221]

For additional insight, the pairwise distance matrix of all genomic signatures generated by ML-DSP[162] for each dataset, with $k = 6$, was analyzed. The pairwise distance matrix of the Temperature Dataset revealed that the DNA fragment with the shortest distance from that of *Thermocrinis ruber* (bacterium) belonged to *Thermococcus_A litoralis* (archaea) with a distance value of 0.0327 (the distance ranges between 0 and 1, with 0 the minimum distance, between identical sequences, and one the maximum distance).

5.4 Discussion

We note that the six supervised machine learning algorithms produced highly accurate taxonomic classifications of extremophile prokaryotic genome sequences and medium to medium-high accurate environment category classifications of the same sequences. These results suggest that, in addition to the taxonomic information present in the genomic signatures of extremophiles, a distinct k-mer frequency profile associated with each environment category also exists. Thus, if the bacteria and archaea sequences in the training set are labelled by environment category, then the supervised learning algorithms will likely assign a new sequence to its correct environment category, regardless of its taxonomy. Also, note that the classification accuracies obtained when the datasets were taxonomy-labelled and environment category-labelled were significantly higher than those obtained when the same datasets were assigned random labels. These findings are consistent with the claim that these taxonomic and environment category classifications are not due to chance and support the hypothesis of the presence of both a taxonomic and an environmental component in the genomic signatures of microbial extremophiles.

Additional analyses revealed that the classification accuracies obtained in restriction-free supervised classification scenarios were higher than those obtained in the restricted (non-overlapping genera) supervised classification scenarios. However, even in the restricted scenario, the accuracy of environmental category classifications was higher than those in the control "random label" scenario. Together, these results suggest that the taxonomic component of the genomic signature is stronger than the environmental component but that the latter is discernible and provides discriminating power.

Note that while the subsets of 3-mers relevant for the environment category classification identified by the MDI algorithm provide insights into the relations between genomic signatures and extreme environmental conditions, caution should be taken when interpreting the results. This is because the experiment prioritizes accuracy, and the identified subsets of relevant 3-mers may partially reflect a correlation between taxonomy and environment. In

other words, especially due to the bias and sparsity of both datasets, it is likely that some taxonomic information may also have influenced the process of computational discovery of these subsets of relevant 3-mers. This being said, the overlap between the subsets mentioned above of relevant 3-mers and codon usage patterns and amino acid compositional biases found to be associated with extreme environments in the biological literature still suggest a detectable environmental component of genomic signatures in temperature and pH-adapted microbial extremophiles.

The use of unsupervised learning algorithms for exploring the space of genomic signatures holds significant value, as these algorithms effectively discover clusters of genomic fragments possessing similar genomic signatures, free from the influence of any human annotations. Since the precise definition of the term "genomic signature" entails the differentiation of genetically distant organisms from each other, a high-performing clustering algorithm should primarily yield clusters corresponding to the true genera within the dataset. That being said, ascertaining causality for fragments assigned to erroneous clusters proves challenging, given the potential for similar genomic signatures to coincide with taxonomic information at a lower level and the inherent systematic errors in each algorithm. Therefore, in the present study, identifying pairs exhibiting a similar environmental component in their genomic signature based on the clustering assignments relies predominantly on the consensus of the high-performing clustering algorithms. Furthermore, only pairs of fragments originating from organisms in different domains were retained. Additional confirmation steps by supervised learning in challenging scenarios were applied to the remaining pairs, and four hyperthermophilic exemplars successfully passed all these stringent tests. Thus, other candidates from the list identified by unsupervised clustering could be viable, such as pairs for which only some supervised tests yielded successful results. One such example is the pair of acidophilic organisms *Thermoanaerobacterium thermosaccharolyticum* (bacterium) and *Caldisphaera lagunensis* (archaea) in the pH Dataset, which were clustered together despite their domain-level taxonomic differences. Further analysis is needed to confirm such additional pairs by, e.g., an analysis that utilizes, as a genome representative, multiple DNA fragments combined into a single genomic signature.

5.5 Conclusion

In this chapter, we have demonstrated the successful application of supervised machine learning algorithms for highly accurate taxonomic classifications of extremophile prokaryotic genome sequences and medium to medium-high accurate classifications of the same sequences based on their environment category (hyperthermophile, psychrophile, acidophile, alkaliphile, etc.). The use of k-mer frequency vectors of arbitrarily selected 500 kbp DNA fragments as genomic signatures reveal a strong taxonomic component for $2 \leq k \leq 6$

and a discernible environmental component for $3 \leq k \leq 6$. Furthermore, specific k-mer profiles associated with distinct environment categories are identified, partially agreeing with previous observations in the literature using alignment-based analyses. Finally, these findings are confirmed using unsupervised learning clustering algorithms, revealing specific exemplar organisms whose environmental component appears to be as strong as the taxonomic component of their genomic signature. When applied to a substantial dataset, this multi-pronged approach significantly strengthens the hypothesis of an environmental component in the genomic signature of microbial extremophiles adapted to extreme temperature or pH environmental conditions.

Chapter 6

Encoding DNA barcodes with transformer models and self-supervision

This chapter explores the application of pretrained transformer models as feature extractors for the taxonomic classification of DNA barcodes. The content in this chapter is an adaptation and extension of a paper titled "*BarcodeBERT: Transformers for Biodiversity Analysis*" presented at the 4^{th} *Workshop on Self-Supervised Learning: Theory and Practice* at the NeurIPS 2023 conference.

Section 6.1 highlights the significance of DNA barcodes in biodiversity studies and the role of computational methods in their classification. Section 6.2 reviews existing methodologies, including those using supervised convolutional neural networks and foundation models using different DNA encodings and self-supervised learning strategies.

Section 6.3 outlines the datasets (Section 6.3.1) used in our analysis, providing a basis for the subsequent introduction of our method. Section 6.4 describes our approach, detailing the network architectures (Section 6.4.1), training and optimization strategies (Section 6.4.2), and our comprehensive evaluation and experimental setup (Section 6.4.3). BarcodeBERT represents the first self-supervised method specifically designed for general biodiversity analysis on DNA barcodes, leveraging a vast reference library of invertebrate DNA barcodes.

Finally, Section 6.5 demonstrates the efficacy of BarcodeBERT, particularly in the taxonomic classification of DNA barcodes (Section 6.5.1) and its application in Bayesian zero-shot learning (BZSL) for image analysis using DNA as side information (Section 6.5.2). Our findings demonstrate the superiority of BarcodeBERT in species and genus-level identification tasks, outperforming existing models without the need for fine-tuning.

6.1 Introduction

In this dissertation, we have emphasized that the quest to map and comprehend Earth's biodiversity presents an enduring challenge. While traditional taxonomic methods have been a bottleneck for discovering new species given the vast amounts of sequencing data, BIOSCAN is a pioneering global initiative focused on species discovery and identification based on DNA barcodes. The project strives to overcome the limitations of traditional approaches, such as the high computational complexity introduced by the alignment of complete genes, to develop a DNA-based system for species discovery and identification at a global scale [164]. The aim is not only to isolate and identify species in laboratories but also to dynamically identify them using DNA barcodes (see Section 2.1.4) as species discriminators. This methodology bypasses both the experimental challenge of sequencing whole genomes and the computational challenge of excessive time complexity of multiple-sequence alignment algorithms. Ultimately, the project envisions tracking ecosystem dynamics and cataloging our planet's vast array of multicellular life.

Our work in this chapter closely aligns with the ambitious objectives of BIOSCAN, and we build on its leading technology, DNA barcoding, a successful species-level specimen identification tool. For animals, DNA barcoding uses a 648 base pair segment of the COI gene, which we refer to as *barcode* and has become an invaluable tool in biodiversity studies [122]. Although the technologies utilized for DNA barcoding have evolved and expanded, the underlying process remains constant [28]. Figure 6.1 illustrates the entire DNA barcoding workflow. Initially, DNA is extracted from an individual specimen using a small sample or tissue like an insect leg, hair/feathers, or mouth swab. The second stage involves amplification with appropriate primers in the PCR (see Section 2.1.4) to replicate the specific barcode region millions of times to prepare it for sequencing, which is the third stage. The amplified DNA sample is fed into a DNA sequencing platform to provide the nucleotide sequence representing the barcode as output. The final step entails all computational analyses and is this chapter's primary focus.

These DNA barcodes can be easily stored, and new sequences can be compared against a reference library using alignment-based techniques [74], as specified in Section 2.1.4. Most barcoding biodiversity data is publicly available in the Barcode of Life Database (BOLD) [164]. This library is BIOSCAN's multi-modal powerhouse, and it incorporates genomic data and visual and geographic information for each isolated specimen. Currently, BOLD contains more than 16 M total barcodes from more than 250,000 animal species and more than 72,000 plant species [164]. This comprehensive database enables efficient species-level identification and acts as a continuously growing catalogue of global biodiversity, providing an unmatched resource for scientific research.

Among the numerous taxonomic groups present in BOLD, arthropods stand out as an incredibly diverse and taxonomically complex group, where multiple new species are described daily [15, 129]. Hence, they provide an excellent testbed for evaluating new algorithms to be incorporated in the DNA barcoding workflow. We leverage the extensive and high-quality arthropod data publicly available on BOLD and explore advanced machine learning techniques with two main objectives: design efficient algorithms that allow us to gain insight into arthropod diversity, and test the general suitability of these algorithms to be included in the general DNA barcoding workflows. Previous efforts have laid the groundwork

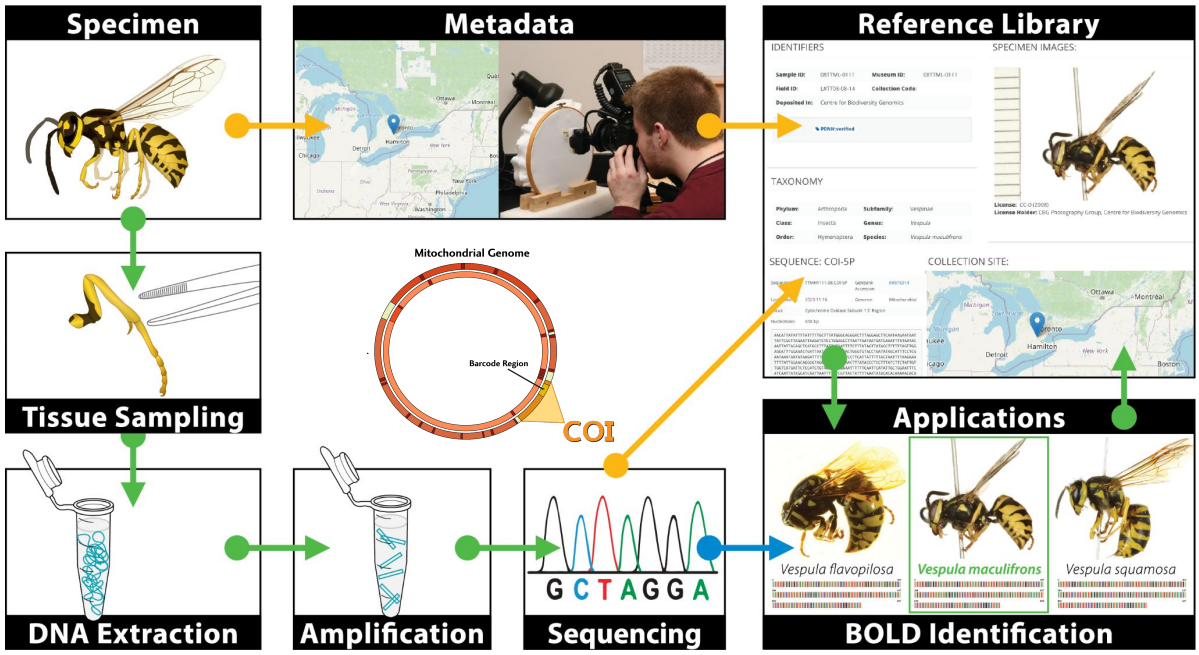


Figure 6.1: Description of the essential stages in DNA barcoding, starting with DNA extraction from specimens, followed by DNA amplification via PCR to amplify the barcode region for sequencing. The nucleotide sequence is then obtained through a DNA sequencing platform, after which different computational methods can be used for taxonomic identification and classification. For example, data can be filtered for inclusion in the reference library or can be used as a query for taxonomic identification. (This figure is adapted from figure 9 in [28].)

in this context by employing CNNs and transformer models for species and genus-level identification over images, demonstrating notable successes. Introducing BarcodeBERT, we present a self-supervised learning approach specifically designed for DNA barcoding. Our method is trained on a reference library containing 1.5 million invertebrate barcodes [45] and represents a significant leap forward in the field. It learns meaningful data embeddings for efficient species-level classification and showcases the potential of self-supervised pretraining in enhancing barcode-based identification accuracy across various taxonomic levels.

We provide a comprehensive comparison between BarcodeBERT, fine-tuned foundation models and a successful CNN baseline trained in a supervised manner. We evaluate their performance in several taxonomic classification tasks, including Bayesian zero-shot classification of insect images using DNA as side information [8]. Our method is the first successful SSL approach for taxonomic identification using the COI gene. By harnessing the capabilities of transformer-based models, we significantly improve the taxonomic identification process, supporting the global efforts of BIOSCAN and beyond.

6.2 Related work

Various algorithmic approaches can expedite the taxonomic categorization of novel specimens. For example, a natural approach is to embed the sequences into a vector space such that the geometric distance in the target space allows a faster computation of a similarity measure between each uncategorized sequence and sequences in a database [34]. Grouping the barcodes into different OTUs based on sequence similarity is also possible. Each OTU acts as an algorithmic proxy for species, particularly useful for species without consensus on the taxonomy [49, 165].

Given the classification-oriented nature of these tasks, machine learning provides many methods that can be applied to biodiversity analyses on DNA barcodes. A recent study [9] proposes a Bayesian framework based on CNNs which, when combined with visual information, achieves high accuracies in species-level identification of seen species and genus-level inference of novel species in a dataset of ∼32,000 insect DNA barcodes. This method uses supervised learning to compute meaningful embeddings that can be used as side information in a two-layer Bayesian zero-shot learning framework.

Transformer-based models [201], which have demonstrated superiority over convolutional neural networks (CNNs) in various tasks [27, 196], are known for their ability to capture complex patterns in sets and sequences. These models have found applications across diverse domains thanks to their effectiveness in learning from large unlabelled datasets [24, 196]. Transformers pretrained with self-supervised learning (SSL) at scale, also referred to as

"foundation models," are often task-agnostic and expected to perform well after fine-tuning for various downstream tasks. Yet, their application for taxonomic identification using DNA barcodes had not been extensively explored until now. Foundation models for DNA primarily target human sequences [38, 92, 222], which intuitively makes them unsuitable for barcode data. While DNABERT employs masked token prediction and same-segment prediction tasks, its successor, DNABERT-2, introduces Byte Pair Encoding for versatile tokenization, pretrained on a vast multi-species genomic dataset. However, adapting these models to barcode data has encountered hurdles, mainly due to the fact that DNA barcodes correspond to a very specific region in mitochondrial DNA, and patterns learned from other regions in the genome might be irrelevant to this type of genomic data.

Efforts to use foundation models to map DNA barcodes into vector embeddings revealed that although models such as DNABERT could be fine-tuned for species classification, the computational demand significantly exceeded that of training simpler CNN models. This highlighted the need for an efficient approach that combines the pattern recognition capabilities of transformers with the computational efficiency of CNNs. This backdrop motivates the design of our new tool, BarcodeBERT, a novel transformer-based approach meant to address the limitations of other methods by being both efficient and effective in transforming DNA barcodes into meaningful encodings. We utilize the extensive resources available through BOLD and address the computational obstacles that previously hindered the integration of transformers into DNA barcoding. We leverage the transformer's exceptional pattern recognition capabilities and make this technology readily accessible for integration into DNA barcoding workflows, which could potentially facilitate the completion of BIOSCAN's ambitious goals.

6.3 Methods

In this section, we outline the key elements of our methodology. We begin with a detailed account of our data processing pipeline, where we include all the steps taken to curate our dataset from the reference library, followed by a brief description of the architectures and hyper-parameters used. Finally, we describe our evaluation framework and the downstream tasks used for testing.

6.3.1 Dataset

The primary source of data for this study is the reference library for Canadian invertebrates [45], containing 1.5 M DNA samples, which was directly queried from the barcode of life

database (BOLD) [164].

Data Pre-Processing: To ensure data integrity and consistency, we performed a series of pre-processing steps over this dataset. First, empty entries were removed, and IUPAC Ambiguity Codes (non-`ACGT` symbols), including alignment gaps, were uniformly replaced with the symbol `N`. Duplicated sequences, even with different identifiers, were removed to avoid redundancy and increase the complexity of the training and pretraining tasks. Sequences with trailing `N`'s were truncated. Finally, sequences falling below 200 base pairs or exhibiting over 50% `N` content were excluded.

Table 6.1: The distribution of barcode sequences used in the pretraining phase.

Phylum name	# ID	# BIN	# Class	# Order	# Family	# Genus	# Species	# Sequences
Annelida	2102	516	2	16	48	150	329	2102
Arthropoda	888934	61328	14	67	929	6211	13991	888934
Brachiopoda	20	2	1	2	2	2	2	20
Bryozoa	5	4	3	3	3	2	2	5
Chordata	289	102	5	18	37	67	89	289
Cnidaria	112	46	4	10	24	25	24	112
Echinodermata	276	79	5	17	26	43	74	276
Hemichordata	4	2	1	1	1	2	1	4
Mollusca	1912	372	6	30	97	162	271	1912
Nematoda	24	8	2	5	10	5	2	24
Nemertea	56	22	3	2	5	5	5	56
Platyhelminthes	1	1	0	0	0	0	0	1
Porifera	7	5	1	3	4	4	3	7
Priapulida	1	1	1	1	1	1	1	1
Tardigrada	1	1	1	1	1	0	0	1

Data Split: After pre-processing, 965,289 sequences were obtained. The dataset was divided into three distinct subsets for various evaluation purposes: (*i*) Supervised Seen: This dataset was curated for assessing the model's efficacy in classifying known species. It comprises 1,390 species, each represented by at least 10 and at most 50 barcodes. These sequences were further partitioned into training (70%), testing (20%), and validation (10%) subsets. (*ii*) Unseen: This dataset was created to emulate the real-world scenario of encountering previously unknown species during testing. It includes 4,278 sequences from 1,826 species that are absent from the training data. Only species with a minimum of 50 records were included, and exactly 50 DNA sequences per species were chosen for this dataset for genus-level identification. (*iii*) Unsupervised pretraining: The remaining sequences, including sequences with incomplete taxonomic annotations at different levels, constitute this dataset. To benchmark against prior works, we additionally use the INSECT dataset as introduced in [8], henceforth referred to as Badirli *et al.*

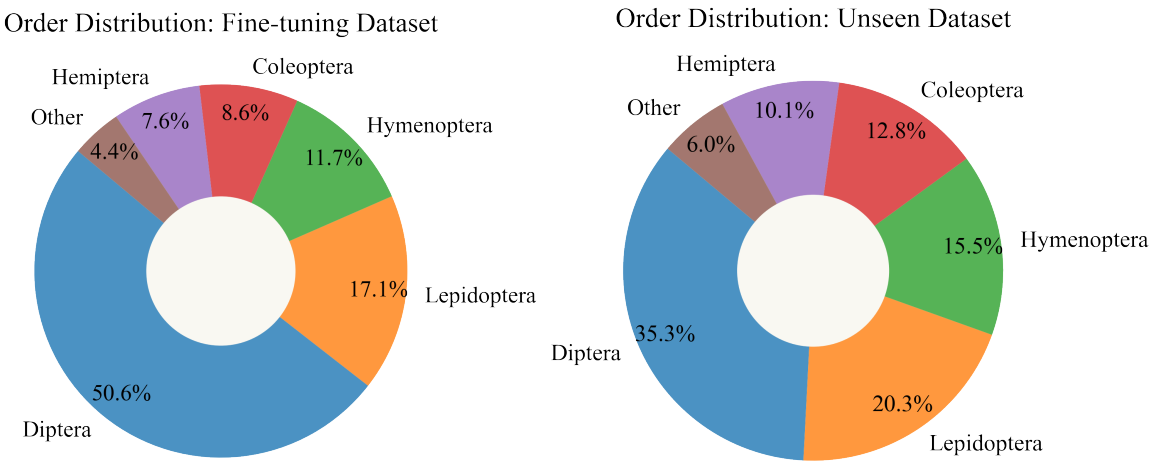


Figure 6.2: Distribution of orders in the Fine-tuning (left) and unseen (right) datasets.

6.3.2 Proposed method: BarcodeBERT

This section presents an account of the data processing pipeline, including the steps taken to curate the dataset from the reference library and a description of the architectures and hyperparameters used. In addition, it describes the evaluation framework and the downstream tasks used for testing.

Network architectures

In this section, we introduce the key network architectures employed in our study for DNA sequence analysis.

CNN baseline: Adapted from [8], it comprises three convolutional layers, each followed by batch normalization and max-pooling. The output of the third convolutional layer is flattened, batch normalized, and connected to a linear layer with 500 units that are finally connected to the output layer.

Foundation models: Our comparison includes two pretrained foundation models based on the Bidirectional Encoder Representations from Transformers model (BERT). These are capable of converting sequence inputs into embedding vectors, and they can be further trained using self-supervised and/or supervised objectives. Within this transformer-based architecture, multi-head attention units play a vital role in capturing relations among

input sequences at various scales, encompassing both small-scale and large-scale interactions. The first model, DNABERT, captures global and transferable genomic understanding by leveraging nucleotide contexts using an overlapping k-mer window for tokenization. The model is highly accurate at predicting splicing and transcriptor factor binding sites. The second, DNABERT2, pioneers the use of Byte-Pair Encoding (BPE) in this domain and overcomes inefficiencies in genomic tokenization through non-overlapping k-mers.

Our model: Also inspired by the BERT architecture, it features 12 attention heads, 12 layers, and a maximum sequence length of 512. After DNA barcodes are segmented into non-overlapping k-mers, the BERT model encodes the sequence of k-mers into a sequence of d-dimensional vectors ($d = 768$). Since our primary objective is to generate an embedding vector that encapsulates information across the entire DNA barcode, following a self-supervised training phase, we merge these d-dimensional vectors for each DNA sequence to create a comprehensive vector representation for the entire sequence using global average pooling.

Training and optimization

As previously mentioned, our method entails the segmentation of each DNA barcode into a series of non-overlapping k-mers. The standard DNA alphabet comprises the nucleotides `A`, `C`, `G`, and `T`. However, note that specific DNA barcodes may incorporate other symbols, such as `N`'s or alignment gaps '-' within their sequences, denoting ambiguity. Our vocabulary encompasses all possible combinations of k-length strings derived from the nucleotide alphabet, supplemented by two special tokens: `<MASK>` and `<UNK>`. The `<MASK>` token is utilized for masking k-mers during the training phase, and k-mers containing any symbol that is not present in the nucleotide alphabet are assigned the `<UNK>` token. Consequently, the total vocabulary size is determined by the expression $4^k + 2$.

We implement the BERT model using the Hugging Face Transformers library and PyTorch. During training, we focused exclusively on masked token prediction, masking 50% of the input tokens and optimizing the network with a cross-entropy loss. We utilize the AdamW optimizer [119] and incorporate a linear scheduler with an initial learning rate of 1×10^{-4} during the optimization process. Additionally, we performed experiments across different k-mer lengths ($4 \leq k \leq 6$) to observe the impact of k-mer length on embedding quality.

Evaluation and experimental setup

To explore the applicability of transformer architectures for DNA barcode-based biodiversity analyses, we employ different SSL evaluation strategies [10] and contrast their performance

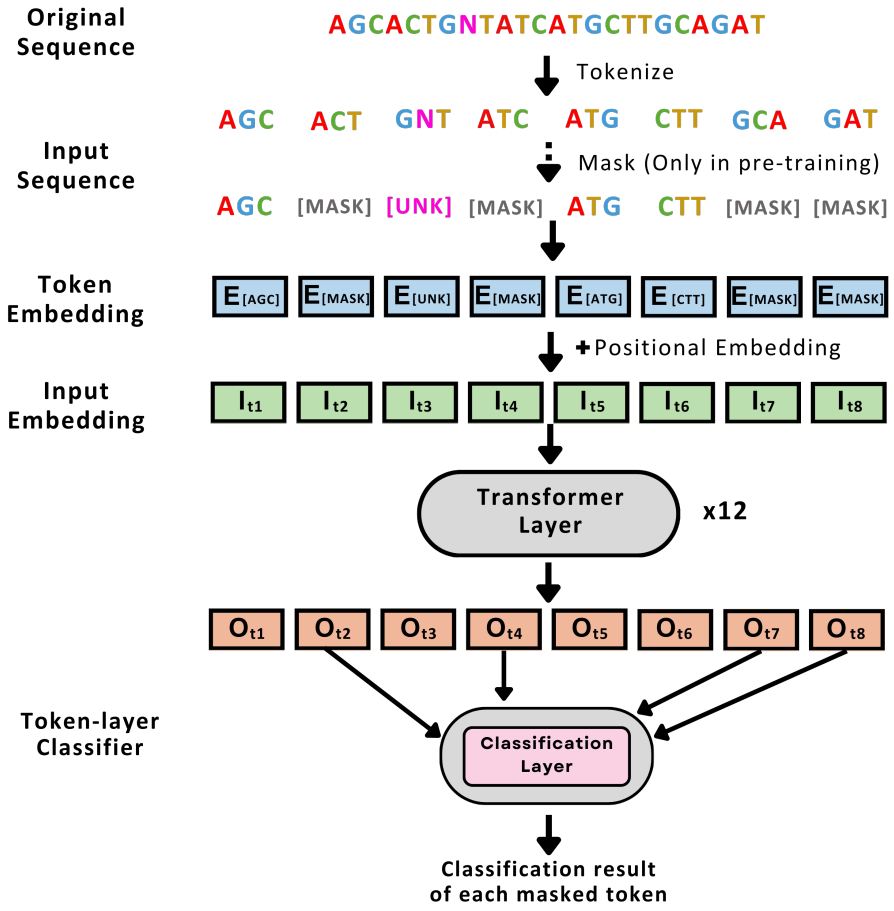


Figure 6.3: Architecture of BarcodeBERT, a transformer-based model employing a self-supervised learning strategy. The model is trained on non-overlapping k-mers from DNA sequences as tokens. Any token containing a character that is not in the nucleotide vocabulary is replaced by the `<UNK>` token. Pretraining involves masking certain input parts and predicting these masked elements using a linear classification layer. The masking is implemented using the `<MASK>` token to represent masked k-mers during pretraining. Following the notation in [92], E_t, I_t and O_t denote the positional encoding, the input embedding and the last hidden state at token t, respectively.

against a supervised baseline. Initially, we perform task-specific fine-tuning, *i.e.,* we fine-tune the models on the supervised training dataset and assess their performance at species-level classification. Second, we gauge the influence of pretraining on DNA barcodes by using the models as feature extractors.

We first implement genus-level 1-NN probing on sequences from unseen species, providing insights into the models' ability to generalize to new taxonomic groups. Additionally, we perform species-level classification using a linear classifier trained on embeddings from the pretrained models. Throughout the evaluation process, our pipeline remains consistent with the training process. DNA barcodes are tokenized into non-overlapping k-mers, and the sequence of tokens is fed into the model.

To generate an overall embedding for the entire DNA barcode, we calculate the average vector obtained from the constituent non-overlapping k-mers within that specific barcode. Finally, in a novel exploration, following [8], we evaluate our model's performance in the context of Bayesian zero-shot learning on the INSECT dataset for species classification as a downstream task, applying a Bayesian model that generates a posterior predictive distribution (PPD) for both seen and unseen categories with image features as prior and DNA features as side information. Due to the absence of unseen categories' image features in the training set, to allow the BZSL model to generate the PPD for each unseen category, the model selects the K-nearest seen categories of the unseen category in the DNA feature space and uses their image features as a local prior. We consider both employing the DNA feature embeddings directly from the pretrained BERT models and fine-tuning the models through supervised learning of the species classification task on the INSECT dataset using the DNA barcodes as input. We utilize image features from the INSECT dataset mentioned in [8], pre-extracted using ResNet-101 [71], to ensure that our results can be compared effectively with those in [8]. We compare our model's performance to the supervised CNN used in [8] as well as pretrained DNABERT [92] and DNABERT-2 [222] models. We tokenize the barcode data using overlapping k-mers for DNABERT, $k = 6$, and the BPE tokenizer for DNABERT-2.

6.4 Results

6.4.1 Taxonomic classification of DNA barcodes

In our evaluation based on traditional classification setups, detailed in Table 1, fine-tuning revealed no significant performance gap, with the CNN baseline marginally outperforming all transformer models. Genus-level 1-NN probing displayed a similar trend. Linear probing,

however, favoured the pretrained model by a slight margin. It's noteworthy that both our model and DNABERT2 consistently outperformed DNABERT. This likely stems from the non-overlapping tokenization approach and the fact that DNABERT2 was not exclusively pretrained on human data. Although the baseline model performed well, the transformer-based models demonstrate their potential to contribute significantly to DNA barcode analysis.

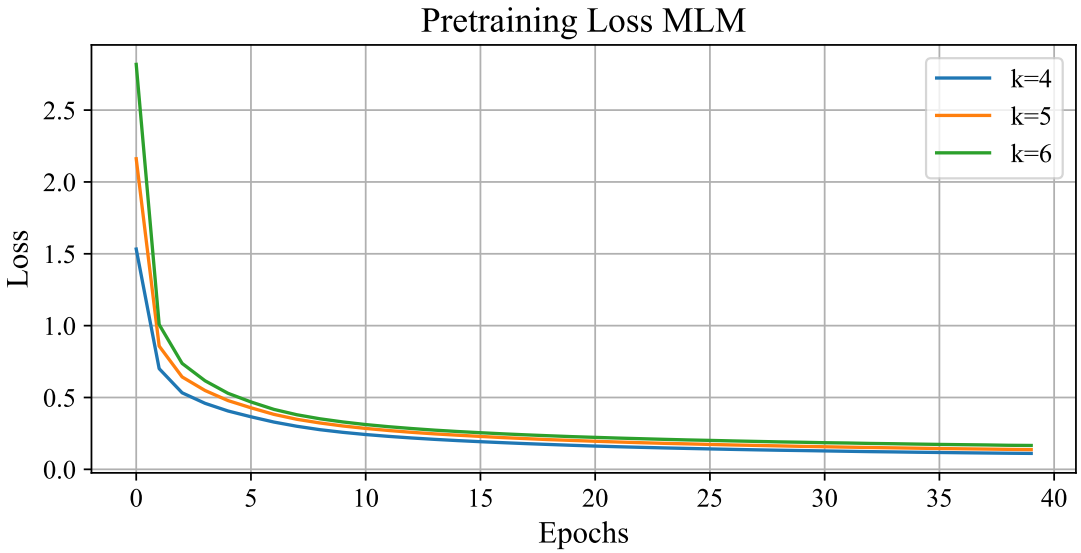


Figure 6.4: Mask prediction loss over 40 training epochs for different k-mer lengths.

As detailed in Table 6.2, fine-tuning revealed no significant performance gap, with DNABERT-2 marginally outperforming all other models. In the genus-level 1-NN probing task, BarcodeBERT and DNABERT-2 outperformed the baseline, with DNABERT-2 performing less competitively. Linear probing, however, favoured our pretrained models and DNABERT-2 over the baseline and DNABERT. It is noteworthy that both BarcodeBERT and DNABERT-2 outperformed DNABERT in two out of three tasks. This likely stems from the non-overlapping tokenization approach and the fact that DNABERT-2 was not exclusively trained on human data. Although the baseline model performed well, the transformer-based models demonstrate their potential to contribute significantly to DNA barcode analysis.

Table 6.2: Classification accuracy of DNA barcode models under different SSL evaluation strategies. Some models supported variable stride length; for these, we show results at several k-mer lengths.

	Species-level acc (%) of seen species						Genus-level acc (%) of unseen species		
Model	Fine-tuned			Linear-probe			1-NN probe		
CNN baseline	98.2			51.8			47.0		
DNABERT-2	**98.3**			87.2			40.9		
k-mer length	$k=4$	$k=5$	$k=6$	$k=4$	$k=5$	$k=6$	$k=4$	$k=5$	$k=6$
DNABERT	96.3	96.9	97.4	47.1	38.4	41.2	38.2	41.6	48.5
BarcodeBERT (ours)	98.6	98.5	98.7	**93.0**	88.6	84.0	49.0	**58.4**	57.6

6.4.2 Bayesian zero-shot learning of images with DNA as side information

While we experimented with the alignment of barcodes mentioned in [8], we found in practice that further alignment of DNA barcodes did not significantly affect the results. Therefore, the DNA barcodes we used for experiments with all the models are not aligned and taken as-is from the BOLD database. For each model before and after fine-tuning, we perform a grid search over the same hyperparameter space used by [8] for Bayesian zero-shot learning. The resulting accuracy for seen and unseen test species, as well as the harmonic mean, are presented in Table 6.3.

Even without fine-tuning, BarcodeBERT substantially outperforms DNABERT and DNABERT-2 on unseen species, regardless of whether they had been fine-tuned previously or not. BarcodeBERT achieves similar performance to the reported baseline CNN results [8] and improves on the harmonic mean score by 1.2% and unseen accuracy by 1.9%, respectively. We thus find that in the zero-shot learning task of predicting insect species, employing BERT-like models that have also been trained on insect DNA barcodes as DNA encoders can improve performance.

6.5 Conclusions

Our research shows that pretraining masked language models on DNA barcode data, as demonstrated by BarcodeBERT, is both effective and essential for arthropod species identification. This underscores the need to diversify datasets beyond human DNA sequences

Table 6.3: Evaluation of DNA barcode models in a Bayesian zero-shot learning task on the INSECT dataset. The pretraining and fine-tuning data source is indicated by the respective DNA type, and '–' signifies the absence of training for that type. We also indicate the most specific taxon subset. For the baseline CNN encoder, we report the original paper result (left) and reproduced result (right).

	Data sources		Species-level acc (%)		
Model	SSL pretraining	Fine-tuning	Seen	Unseen	Harmonic Mean
CNN encoder	–	Insect	38.3 / 39.4	20.8 / 18.9	**27.0** / 25.5
DNABERT	Human	–	35.0	10.3	16.0
DNABERT	Human	Insect	**39.8**	10.4	16.5
DNABERT-2	Multi-species	–	36.2	10.4	16.2
DNABERT-2	Multi-species	Insect	30.8	8.6	13.4
BarcodeBERT (ours)	Arthropod	–	38.4	16.5	23.1
BarcodeBERT (ours)	Arthropod	Insect	37.3	**20.8**	**26.7**

to advance the field of biodiversity science. While we have made strides in improving the classification of arthropod species using both DNA sequences and images, our findings point to a wealth of untapped data, *e.g.*, the BOLD dataset, currently comprising 14 million DNA barcodes, continuously augmented by data from previously seen or unseen species. Future work includes further investigation of such DNA barcode data to develop more robust and scalable self-supervised models for taxonomic classification.

Chapter 7

Summary and future work

One of the central motivations for this dissertation stems from the ongoing debates regarding taxonomic identifiers for certain organisms and the uncertainty surrounding identifiers of newly discovered species. This uncertainty and lack of consensus make it reasonable to suggest that strategies less reliant on taxonomic labels may be better suited to achieve the overarching goal of creating a comprehensive record of life on our planet. Two different aspects of this challenge have been explored throughout the topics of this thesis. First, we explored how to effectively categorize DNA sequences without relying on traditional taxonomic labels, and group together DNA sequences from closely related organisms across different domains of life. Second, we sought to use unlabelled data to enhance supervised classification pipelines. Ultimately, our research aims to use unlabelled genomic data to improve our understanding of biodiversity and facilitate a more efficient categorization of DNA sequences.

We start by developing DeLUCS, our deep learning-based unsupervised clustering method. To cluster a given sequence dataset, DeLUCS first generates artificial mimic sequences from the original sequences using a probabilistic model, and calculates normalized k-mer frequency vectors for both the original and the mimic sequences. Then, using an information-based loss function, m independent neural networks are trained to maximize the mutual predictability of cluster assignments for a sequence and its corresponding mimic sequences. Finally, we employ majority voting to finalize each sequence's cluster assignment.

Through the development of DeLUCS, we have shown that it is possible to train a discriminative neural network to identify significant taxonomic clusters in datasets of mitochondrial DNA from eukaryotes or fragments of nuclear DNA from prokaryotes. This method, pioneering in its application of unsupervised deep learning for clustering unlabelled DNA sequences, marks a significant advance in analyzing large and diverse datasets. These

datasets, often challenging for traditional unsupervised methods due to their size and lack of DNA sequence homology, are now more accessible thanks to DeLUCS. This approach introduces the principles of contrastive learning to comparative bioinformatics and sets a new benchmark for unsupervised DNA sequence clustering, promising a transformative impact on genomic data analysis.

Refining DeLUCS to enable good performance, even in the case of unbalanced datasets, and developing more sophisticated clustering ensemble techniques were key areas with room for further improvement. Building on this, we introduced *i*DeLUCS, an improvement of DeLUCS that uses self-supervised representation learning. The *mimic* sequences are now used as part of a more general contrastive framework, where the consistency of both the final cluster assignments and the intermediate representations learned by the network are enforced during training. These learned representations are also suitable for non-parametric clustering of long DNA sequences, as our software tool matched or surpassed the performance of alignment-based methodologies in synthetic datasets. As a standalone tool, *i*DeLUCS exemplifies the flexibility of the contrastive learning framework across various genomic datasets, facilitating insightful visualizations and evaluations of the training process and dataset compositions. Various avenues could be explored to improve *i*DeLUCS, such as adapting the contrastive learning framework to enable the processing of raw DNA sequences and other DNA sequence representations, such as CGRs. We are also actively searching for more precise mathematical formulations for calculating mimic sequence augmentations that are dataset-independent. Lastly, better optimization or initialization techniques could be implemented to eliminate the need for a clustering ensemble.

The emergence of high-performing unsupervised learning algorithms such as *i*DeLUCS inspired an investigation into the genomic signatures of microbial extremophiles. This case study showcases how the joint use of both supervised and unsupervised machine learning-based methodologies can lead to meaningful biological discoveries. We explore the hypothesis that an organism's genome could contain other information beyond ancestry or taxonomy and found evidence of a pervasive, genome-wide environmental component in the genomes of some extremophiles. These findings offer a new lens through which to view adaptations to extreme conditions and could potentially redefine the concept of genomic signature. Future work includes a systematic selection and compilation of the genomic signatures to guarantee uniform coverage across the genome of both bacterial and archaeal organisms. Furthermore, similar analyses can be performed for larger datasets and other extremophilic characteristics.

Finally, we attempt to bridge the performance gap between supervised and semi-supervised taxonomic classification using DNA barcodes. Our model, BarcodeBERT, is a transformer model pretrained on a 1.5 million barcode database using a masked language

modelling loss function. After pretraining, BarcodeBERT embeds barcode sequences into an expressive representation space that can be exploited for various classification tasks. We compare the quality of the embeddings learned by our self-supervised model against the ones produced by supervised convolutional neural networks and fine-tuned foundation models across different classification tasks: supervised fine-tuning over a dataset with novel species, kNN probing for the classification of unseen species into seen genera, and linear probing for the classification of novel specimens from seen species. Our results show that BarcodeBERT outperforms all other models, even without fine-tuning, in complex identification tasks.

Through the development of our model, BarcodeBERT, we illustrate the essential role of pretraining on DNA barcode data for species identification. This approach highlights the importance of expanding our datasets to encompass a broader spectrum of biodiversity, pointing to the future of robust, self-supervised models for taxonomic classification. Future directions will incorporate data augmentation techniques into the training process to increase the robustness of the model. Future work also includes investigating the different pretext tasks and masking strategies, such as the ones used by recently proposed foundation models in genomics. Ultimately, BarcodeBERT could also be coupled with a transformer-based decoder to exploit the learned representations for hierarchical, fine-grained taxonomic classification.

In summary, in this dissertation we have investigated how to unite the fields of biodiversity and taxonomic categorization with the field of deep unsupervised learning. We successfully trained neural networks for representation learning of DNA barcodes and unsupervised clustering of genomic signatures. The neural nature of our methodologies makes our software tools stand out in terms of robustness and scalability, making them suitable for the ever-increasing volume of new genomic data being generated. We conclude that meaningful information can be learned without reliance on labels, and our work not only introduces novel methodologies to do so but also paves the way for future explorations into different neural architectures and comprehensive analyses of biodiversity data.

References

[1] Emmanuel Adetiba and Oludayo O. Olugbara. Classification of eukaryotic organisms through cepstral analysis of mitochondrial DNA. In *International Conference on Image and Signal Processing*, pages 243–252. Springer, 2016.

[2] Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, 1st edition, 2013.

[3] Mahmood Akhtar, Eliathamby Ambikairajah, and Julien Epps. GMM-based classification of genomic sequences. In *2007 15th International Conference on Digital Signal Processing*, pages 103–106, 2007.

[4] Nasssima Aleb and Narimane Labidi. An improved K-means algorithm for DNA sequence clustering. In *2015 26th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 39–42, 2015.

[5] Remi Allio, Stefano Donega, Nicolas Galtier, and Benoit Nabholz. Large variation in the ratio of mitochondrial to nuclear mutation rate across animals: Implications for genetic diversity and the use of mitochondrial DNA as a molecular marker. *Molecular Biology and Evolution*, 34(11):2762–2772, 2017.

[6] Wendy L. Applequist. A brief review of recent controversies in the taxonomy and nomenclature of *Sambucus nigra* sensu lato. In *ActaHortic.*, pages 25–33. International Society for Horticultural Science (ISHS), Leuven, Belgium, 2015.

[7] Jorge Avila Cartes, Santosh Anand, Simone Ciccolella, Paola Bonizzoni, and Gianluca Della Vedova. Accurate and fast clade assignment via deep learning and frequency Chaos Game Representation. *GigaScience*, 12:giac119, 2022.

[8] Sarkhan Badirli, Zeynep Akata, George Mohler, Christine Picard, and Mehmet M. Dundar. Fine-grained zero-shot learning with DNA as side information. In *Advances*

in Neural Information Processing Systems, volume 34, pages 19352–19362. Curran Associates, Inc., 2021.

[9] Sarkhan Badirli, Christine Johanna Picard, George Mohler, Frannie Richert, Zeynep Akata, and Murat Dundar. Classifying the unknown: Insect identification with deep hierarchical Bayesian learning. *Methods in Ecology and Evolution*, 14(6):1515–1530, 2023.

[10] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023. arXiv: 2304.12210.

[11] Junpeng Bao, Ruiyu Yuan, and Zhe Bao. An improved alignment-free model for DNA sequence similarity metric. *BMC Bioinformatics*, 15:321, 2014.

[12] Qiyu Bao, Yuqing Tian, Wei Li, Zuyuan Xu, Zhenyu Xuan, Songnian Hu, Wei Dong, Jian Yang, Yanjiong Chen, Yanfen Xue, Yi Xu, Xiaoqin Lai, Li Huang, Xiuzhu Dong, Yanhe Ma, Lunjiang Ling, Huarong Tan, Runsheng Chen, Jian Wang, Jun Yu, and Huanming Yang. A complete sequence of the *T. tengcongensis* genome. *Genome Res*, 12(5):689–700, 2002.

[13] Yiming Bao, Pavel Bolotov, Dmitry Dernovoy, Boris Kiryutin, Leonid Zaslavsky, Tatiana Tatusova, Jim Ostell, and David Lipman. The Influenza virus resource at the National Center for Biotechnology Information. *Journal of Virology*, 82(2):596–601, 2008.

[14] Sailen Barik. Evolution of protein structure and stability in global warming. *International Journal of Molecular Sciences*, 21(24):9662, 2020.

[15] Yves Basset, Lukas Cizek, Philippe Cuénoud, Raphael K. Didham, François Guilhaumon, Olivier Missa, Vojtech Novotny, Frode Ødegaard, Tomas Roslin, Jürgen Schmidl, Alexey K. Tishechkin, Neville N. Winchester, David W. Roubi, Henri-Pierre Aberlenc, Johannes Bail, Héctor Barrios, Jon R. Bridle, Gabriela Castaño-Meneses, Bruno Corbara, Gianfranco Curletti, Wesley Duarte da Rocha, Domir De Bakker, Jacques H. C. Delabie, Alain Dejean, Laura L. Fagan, Andreas Floren, Roger L. Kitching, Enrique Medianero, Scott E. Miller, Evandro Gama de Oliveira, Jérôme Orivel, Marc Pollet, Mathieu Rapp, Sérvio P. Ribeiro, Yves Roisin, Jesper B. Schmidt, Line Sørensen, and Maurice Leponce. Arthropod diversity in a tropical forest. *Science*, 338(6113):1481–1484, 2012.

[16] Cédric Bauvois, Lilian Jacquamet, Adrienne L. Huston, Franck Borel, Georges Feller, and Jean-Luc Ferrer. Crystal structure of the cold-active aminopeptidase from *Colwelliapsychrerythraea*, a close structural homologue of the human bifunctional leukotriene A4 hydrolase. *Journal of Biological Chemistry*, 283(34):23315–23325, 2008.

[17] Dennis A. Bazylinski, John W. Farrington, and Holger W. Jannasch. Hydrocarbons in surface sediments from a Guaymas Basin hydrothermal vent site. *Organic Geochemistry*, 12(6):547–558, 1988.

[18] Shimshon Belkin, Carl O. Wirsen, and Holger W. Jannasch. A new sulfur-reducing, extremely thermophilic eubacterium from a submarine thermal vent. *Applied and Environmental Microbiology*, 51(6):1180–1185, 1986.

[19] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[20] Vincenzo Bonnici and Vincenzo Manca. Informational laws of genome structures. *Scientific Reports*, 6(1):28840, 2016.

[21] Ernesto Borrayo, E. Gerardo Mendizabal-Ruiz, Hugo Vélez-Pérez, Rebeca Romo-Vázquez, Adriana P. Mendizabal, and J. Alejandro Morales. Genomic signal processing methods for computation of alignment-free distances from DNA sequences. *PLOS ONE*, 9(11):1–13, 2014.

[22] Jeff S. Bowman and Jody W. Deming. Alkane hydroxylase genes in psychrophile genomes and the potential for cold active catalysis. *BMC Genomics*, 15(1):1120, 2014.

[23] John S. Bridle, Anthony J. R. Heading, and David J. C. MacKay. Unsupervised classifiers, mutual information and 'phantom targets'. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 1096–1101, San Francisco, CA, USA, 1991.

[24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[25] Alhadi Bustamam, Hengki Tasman, Nunung Yuniarti, Frisca Frisca, and Ichsani Mursidah. Application of K-means clustering algorithm in grouping the DNA sequences of Hepatitis B Virus (HBV). *AIP Conference Proceedings*, 1862(1):030134, 2017.

[26] Bruno Capaccioni, Franco Tassi, and Orlando Vaselli. Organic and inorganic geochemistry of low temperature gas discharges at the Baia di Levante beach, Vulcano Island, Italy. *Journal of Volcanology and Geothermal Research*, 108(1-4):173–185, 2001.

[27] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020.

[28] University of Guelph Centre for Biodiversity Genomics. The global taxonomy initiative 2020: A step-by-step guide for DNA barcoding. Technical series no. 94, Secretariat of the Convention on Biological Diversity, 2021.

[29] Pierre-Alain Chaumeil, Aaron J. Mussig, Philip Hugenholtz, and Donovan H. Parks. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database. *Bioinformatics*, 36(6):1925–1927, 2019.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[31] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *Advances in Neural Information Processing Systems*, volume 34, pages 11834–11845, 2021.

[32] Benny Chor and Tamir Tuller. Maximum likelihood of evolutionary trees: hardness and approximation. *Bioinformatics*, 21(*suppl 1*):i97–i106, 2005.

[33] Charles G. Clifton, Clifford Walters, and Bernd R. T. Simoneit. Hydrothermal petroleums from Yellowstone National Park, Wyoming, U.S.A. *Applied Geochemistry*, 5(1-2):169–191, 1990.

[34] Gabriele Corso, Zhitao Ying, Michal Pándy, Petar Veličković, Jure Leskovec, and Pietro Liò. Neural distance embeddings for biological sequences. In *Advances in Neural Information Processing Systems*, volume 34, pages 18539–18551. Curran Associates, Inc., 2021.

[35] Mark J. Costello, Simon Wilson, and Brett Houlding. Predicting total global species richness using rates of species description and estimates of taxonomic effort. *Systematic Biology*, 61(5):871–871, 2011.

[36] Thomas. M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, New York, 1991.

[37] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[38] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, 2023.

[39] Charles Darwin. *On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life*. Modern Library of the World's Best Books. Modern Library, New York, 1936.

[40] Richard Dawkins. *The Selfish Gene*. Oxford paperbacks. Oxford University Press, 1989.

[41] Sávio Torres de Farias and Maria Christina Manhães Bonato. Preferred codons and amino acid couples in hyperthermophiles. *Genome Biology*, 3(8):preprint0006.1, 2002.

[42] Rebeca De la Fuente, Wladimiro Díaz-Villanueva, Vicente Arnau, and Andrés Moya. Genomic signature in evolutionary biology: A review. *Biology*, 12(2), 2023.

[43] Patrick J. Deschavanne, Alain Giron, Joseph Vilain, Guillaume Fagot, and Bernard Fertil. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. *Molecular Biology and Evolution*, 16(10):1391–1399, 1999.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019.

[45] Jeremy R. deWaard, Sujeevan Ratnasingham, Evgeny V. Zakharov, Alex V. Borisenko, Dirk Steinke, Angela C. Telfer, Kate H. J. Perez, Jayme E. Sones, Monica R. Young, Valerie Levesque-Beaudin, Crystal N. Sobel, Arusyak Abrahamyan, Kyrylo Bessonov, Gergin Blagoev, Stephanie L. deWaard, Chris Ho, Natalia V. Ivanova, Kara K. S. Layton, Liuqiong Lu, Ramya Manjunath, Jaclyn T. A. McKeown, Megan A. Milton, Renee Miskie, Norm Monkhouse, Suresh Naik, Nadya Nikolova, Mikko Pentinsaari, Sean W. J. Prosser, Adriana E. Radulovici, Claudia Steinke, Connor P. Warne, and Paul D. N. Hebert. A reference library for Canadian invertebrates with 1.5 million barcodes, voucher specimens, and DNA samples. *Scientific Data*, 6(1):308, 2019.

[46] Betsey Dexter Dyer, Kahnand Michael J., and LeBlanc Mark D. Classification and regression tree (CART) analyses of genomic signatures reveal sets of tetramers that discriminate temperature optima of archaea and bacteria. *Archaea*, 2:159–167, 2008.

[47] Theodosius Grygorovych Dobzhansky. *Genetics and the Origin of Species.* Columbia University Press, 1941.

[48] W. Ford Doolittle and James R. Brown. Tempo, mode, the progenote, and the universal root. *Proceedings of the National Academy of Sciences*, 91(15):6721–6728, 1994.

[49] Andrew Dopheide, Leah K. Tooman, Stefanie Grosser, Barbara Agabiti, Birgit Rhode, Dong Xie, Mark I. Stevens, Nicola Nelson, Thomas R. Buckley, Alexei J. Drummond, and Richard D. Newcomb. Estimating the biodiversity of terrestrial invertebrates on a forested island using DNA barcodes and metabarcoding data. *Ecological Applications*, 29(4):e01877, 2019.

[50] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[51] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[52] Anna Fabijańska and Szymon Grabowski. Viral genome deep classifier. *IEEE Access*, 7:81297–81307, 2019.

[53] Joseph. Felsenstein. *Inferring phylogenies*. Sinauer Associates, 1st edition, 2003.

[54] Gerhard Fiala and Karl O. Stetter. *Pyrococcus furiosus* sp. nov., represents a novel genus of marine heterotrophic archaebacteria growing optimally at 100 °C. *Archives of Microbiology*, 145(1):56–61, 1986.

[55] Antonino Fiannaca, Laura La Paglia, Massimo La Rosa, Giosue' Lo Bosco, Giovanni Renda, Riccardo Rizzo, Salvatore Gaglio, and Alfonso Urso. Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC Bioinformatics*, 19(7):198, 2018.

[56] Benoît Fontaine, Kees van Achterberg, Miguel Angel Alonso-Zarazaga, Rafael Araujo, Manfred Asche, Horst Aspöck, Ulrike Aspöck, Paolo Audisio, Berend Aukema, Nicolas Bailly, Maria Balsamo, Ruud A. Bank, Carlo Belfiore, Wieslaw Bogdanowicz, Geoffrey Boxshall, Daniel Burckhardt, Przemysław Chylarecki, Louis Deharveng, Alain Dubois, Henrik Enghoff, Romolo Fochetti, Colin Fontaine, Olivier Gargominy, Maria Soledad Gomez Lopez, Daniel Goujet, Mark S. Harvey, Klaus-Gerhard Heller, Peter van Helsdingen, Hannelore Hoch, Yde De Jong, Ole Karsholt, Wouter Los, Wojciech Magowski, Jos A. Massard, Sandra J. McInnes, Luis F. Mendes, Eberhard Mey, Verner Michelsen, Alessandro Minelli, Juan M. Nieto Nafrıa, Erik J. van Nieukerken, Thomas Pape, Willy De Prins, Marian Ramos, Claudia Ricci, Cees Roselaar, Emilia Rota, Hendrik Segers, Tarmo Timm, Jan van Tol, and Philippe Bouchet. New species in the old world: Europe as a frontier in biodiversity exploration, a test bed for 21st century taxonomy. *PLOS ONE*, 7(5):1–7, 2012.

[57] Donald R. Forsdyke. Neutralism versus selectionism: Chargaff's second parity rule, revisited. *Genetica*, 149(2):81–88, 2021.

[58] Donald R. Forsdyke and Sheldon J. Bell. Purine loading, stem-loops and Chargaff's second parity rule: a discussion of the application of elementary principles to early chemical observations. *Applied Bioinformatics*, 3(1):3–8, 2004.

[59] Robert Friedman, John W. Drake, and Austin L. Hughes. Genome-wide patterns of nucleotide substitution reveal stringent functional constraints on the protein sequences of thermophiles. *Genetics*, 167(3):1507–1512, 2004.

[60] Hani Z. Girgis. Meshclust v3.0: high-quality clustering of DNA sequences using the mean shift algorithm and alignment-free identity scores. *BMC Genomics*, 23(1):423, 2022.

[61] Richard A. Goldstein. Amino-acid interactions in psychrophiles, mesophiles, thermophiles, and hyperthermophiles: insights from the quasi-chemical approximation. *Protein Science: A Publication of the Protein Society*, 16(9):1887–1895, 2007.

[62] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Proceedings of the 23rd International*

Conference on Neural Information Processing Systems - Volume 1, NIPS'10, page 775–783, Red Hook, NY, USA, 2010.

[63] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 1. MIT Press Cambridge, 2016.

[64] Grant Greenberg and Ilan Shomorony. The metagenomic binning problem: Clustering Markov sequences. In *2019 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2019.

[65] Joe G. Greener, Shaun M. Kandathil, Lewis Moffat, and David T. Jones. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, 23(1):40–55, 2022.

[66] Nidhi Gupta and Vijay K. Verma. Next-generation sequencing and its application: Empowering in public health beyond reality. In *Microorganisms for Sustainability*, pages 313–341. Springer Singapore, Singapore, 2019.

[67] Suman Hait, Saurav Mallik, Sudipto Basu, and Sudip Kundu. Finding the generalized molecular principles of protein thermal stability. *Proteins: Structure, Function, and Bioinformatics*, 88(6):788–808, 2020.

[68] Eneida L. Hatcher, Sergey A. Zhdanov, Yiming Bao, Olga Blinkova, Eric P. Nawrocki, Yuri Ostapchuck, Alejandro A. Schäffer, and J. Rodney Brister. Virus Variation Resource – improved response to emergent viral outbreaks. *Nucleic Acids Research*, 45(D1):D482–D490, 2016.

[69] Juliette Hayer, Fanny Jadeau, Gilbert Deléage, Alan Kay, Fabien Zoulim, and Christophe Combet. HBVdb: a knowledge database for Hepatitis B Virus. *Nucleic Acids Research*, 41(D1):D566–D570, 2012.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[72] Lily He, Rui Dong, Rong Lucy He, and Stephen S-T Yau. A novel alignment-free method for HIV-1 sub-type classification. *Infection, Genetics and Evolution*, 77:104080, 2020.

[73] Donald O. Hebb. *The organization of behavior: A neuropsychological heory.* Wiley, New York, 1949.

[74] Paul D. N. Hebert, Alina Cywinska, Shelley L. Ball, and Jeremy R. deWaard. Biological identifications through DNA barcodes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1512):313–321, 2003.

[75] Geoffrey E. Hinton, Terrence Joseph Sejnowski, Tomaso A. Poggio, et al. *Unsupervised Learning: Foundations of Neural Computation.* MIT Press, Cambridge, MA 02142-1209, 1999.

[76] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[77] Tung Hoang, Changchuan Yin, Hui Zheng, Chenglong Yu, Rong Lucy He, and Stephen S.-T. Yau. A new method to cluster DNA sequences using Fourier power spectrum. *Journal of Theoretical Biology*, 372:135 – 145, 2015.

[78] Antoine Hocher, Guillaume Borrel, Khaled Fadhlaoui, Jean-François Brugère, Simonetta Gribaldo, and Tobias Warnecke. Growth temperature and chromatinization in archaea. *Nature Microbiology*, 7(11):1932–1942, 2022.

[79] Koki Horikoshi. Alkaliphiles: Some applications of their products for biotechnology. *Microbiology and Molecular Biology Reviews*, 63(4):735–750, 1999.

[80] William H. Horne, Robert P. Volpe, George Korza, Sarah DePratti, Isabel H. Conze, Igor Shuryak, Tine Grebenc, Vera Y. Matrosova, Elena K. Gaidamakova, Rok Tkavc, Ajay Sharma, Cene Gostinčar, Nina Gunde-Cimerman, Brian M. Hoffman, Peter Setlow, and Michael J. Daly. Effects of desiccation and freezing on microbial ionizing radiation survivability: Considerations for Mars sample return. *Astrobiology*, 22(11):1337–1350, 2022.

[81] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[82] En-Ze Hu, Xin-Ran Lan, Zhi-Ling Liu, Jie Gao, and Deng-Ke Niu. A positive correlation between GC content and growth temperature in prokaryotes. *BMC Genomics*, 23(1):110, 2022.

[83] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning*, pages 1558–1567, 2017.

[84] Xiaoju Hu, Zhuxuan Xu, and Subhajyoti De. Characteristics of mutational signatures of unknown etiology. *NAR Cancer*, 2(3):zcaa026, 2020.

[85] Robert Huber, Wolfgang Eder, Stefan Heldwein, Gerhard Wanner, Harald Huber, Reinhard Rachel, and Karl O. Stetter. *Thermocrinis ruber* gen. nov., sp. nov., A pink-filament-forming hyperthermophilic bacterium isolated from Yellowstone National Park. *Applied and Environmental Microbiology*, 64(10):3576–3583, 1998.

[86] Robert Huber, Josef Stöhr, Sabine Hohenhaus, Reinhard Rachel, Siegfried Burggraf, Holger W. Jannasch, and Karl O. Stetter. *Thermococcus chitonophagus* sp. nov., a novel, chitin-degrading, hyperthermophilic archaeum from a deep-sea hydrothermal vent environment. *Archives of Microbiology*, 164(4):255–264, 1995.

[87] Lawrence Hunter. Molecular biology for computer scientists. *Artificial Intelligence and Molecular Biology*, 177:1–46, 1993.

[88] Zhiguang Huo and George Tseng. Integrative sparse K-means with overlapping group lasso in genomic applications for disease subtype discovery. *The Annals of Applied Statistics*, 11(2):1011–1039, 2017.

[89] Benjamin T. James, Brian B. Luczak, and Hani Z. Girgis. MeShClust: an intelligent tool for clustering DNA sequences. *Nucleic Acids Research*, 46(14):e83–e83, 2018.

[90] H. Joel Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170, 1990.

[91] Xu Ji, Andrea Vedaldi, and João F. Henriques. Invariant information clustering for unsupervised image classification and segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9864–9873, 2018.

[92] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V. Davuluri. DNABERT: Pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[93] John Josse, Armin Dale Kaiser, and Arthur Kornberg. Enzymatic synthesis of deoxyribo-nucleic acid. VIII. frequencies of nearest neighbor base sequences in deoxyribonucleic acid. *Journal of Biological Chemistry*, 236:864–875, 1961.

[94] Rallis Karamichalis, Lila Kari, Stavros Konstantinidis, and Steffen Kopecki. An investigation into inter-and intragenomic variations of graphic genomic signatures. *BMC Bioinformatics*, 16(1):246, 2015.

[95] Rallis Karamichalis, Lila Kari, Stavros Konstantinidis, Steffen Kopecki, and Stephen Solis-Reyes. Additive methods for genomic signatures. *BMC Bioinformatics*, 17(1):313, 2016.

[96] Rezaul Karim, Oya Beyan, Achille Zappa, Ivan G. Costa, Dietrich Rebholz-Schuhmann, Michael Cochez, and Stefan Decker. Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, 22(1):393–415, 2020.

[97] Samuel Karlin and Chris Burge. Dinucleotide relative abundance extremes: a genomic signature. *Trends in Genetics*, 11(7):283 – 290, 1995.

[98] Samuel Karlin, Jan Mrázek, and Allan M. Campbell. Compositional biases of bacterial genomes and evolutionary implications. *Journal of Bacteriology*, 179(12):3899–3913, 1997.

[99] Gerald Karp, Janet Iwasa, and Wallace Marshall. *Karp's Cell and Molecular Biology*. Wiley, 2020.

[100] Mohd Faheem Khan and Sanjukta Patra. Deciphering the rationale behind specific codon usage pattern in extremophiles. *Scientific Reports*, 8(1):15548, 2018.

[101] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic gradient descent. In *ICLR: International Conference on Learning Representations*, pages 1–15. ICLR US., 2015.

[102] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada*, 2014.

[103] Andrey Kislyuk, Srijak Bhatnagar, Jonathan Dushoff, and Joshua S. Weitz. Unsupervised statistical clustering of environmental shotgun sequences. *BMC Bioinformatics*, 10(1):316, 2009.

[104] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[105] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[106] Sandeep Kumar, Chung-Jung Tsai, and Ruth Nussinov. Factors enhancing protein thermostability. *Protein Engineering, Design and Selection*, 13(3):179–191, 2000.

[107] Sudhir Kumar, Masatoshi Nei, Joel Dudley, and Koichiro Tamura. MEGA: A biologist-centric software for evolutionary analysis of DNA and protein sequences. *Briefings in Bioinformatics*, 9(4):299–306, 2008.

[108] Hon Keung Kwan and Swarna Bai Arniker. Numerical representation of DNA sequences. In *2009 IEEE International Conference on Electro/Information Technology*, pages 307–310. IEEE, 2009.

[109] Perry J. Lao and Donald R. Forsdyke. Thermophilic bacteria strictly obey Szybalski's transcription direction rule and politely purine-load RNAs with both Adenine and Guanine. *Genome Research*, 10(2):228–236, 2000.

[110] Brendan B. Larsen, Elizabeth C. Miller, Matthew K. Rhodes, and John J. Wiens. Inordinate fondness multiplied and redistributed: the number of species on earth and the new pie of life. *The Quarterly Review of Biology*, 92(3):229–265, 2017.

[111] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[112] Wanxin Li, Lila Kari, Yaoliang Yu, and Laura A. Hug. MT-MAG: Accurate and interpretable machine learning for complete or partial taxonomic assignments of metagenome-assembled genomes. *PLOS ONE*, 18(8):1–22, 2023.

[113] Weizhong Li and Adam Godzik. CD-HIT: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[114] Xin Li and John J Wiens. Estimating global biodiversity: The role of cryptic insect species. *Systematic Biology*, 72(2):391–403, 2022.

[115] Yunfan Li, Mouxing Yang, Dezhong Peng, Taihao Li, Jiantao Huang, and Xi Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9):2205–2221, 2022.

[116] Qiaoxing Liang, Paul W. Bible, Yu Liu, Bin Zou, and Lai Wei. DeepMicrobes: taxonomic classification for metagenomics with deep learning. *NAR Genomics and Bioinformatics*, 2(1), 2020.

[117] J.R. Lobry and A. Necşulea. Synonymous codon usage and its potential link with optimal growth temperature in prokaryotes. *Gene*, 385:128–136, 2006.

[118] Kenneth J. Locey and Jay T. Lennon. Scaling laws predict global microbial diversity. *Proceedings of the National Academy of Sciences*, 113(21):5970–5975, 2016.

[119] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[120] Stilianos Louca, Florent Mazel, Michael Doebeli, and Laura Wegener Parfrey. A census-based estimate of earth's bacterial and archaeal diversity. *PLOS Biology*, 17(2):1–30, 2019.

[121] Scott C Lowe, Joakim Bruslund Haurum, Sageev Oore, Thomas B. Moeslund, and Graham W. Taylor. Zero-shot clustering of embeddings with pretrained and self-supervised learnt encoders. In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models Worshop at the Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[122] Dave Lunt, De-Xing Zhang, Jacek M. Szymura, and Godfrey M. Hewitt. The insect cytochrome oxidase I gene: evolutionary patterns and conserved primers for phylogenetic studies. *Insect Molecular Biology*, 5(3):153–165, 1996.

[123] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, U.K, 2003.

[124] Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023.

[125] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[126] Scott McGinnis and Thomas L. Madden. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res.*, 32(Web Server issue):W20–5, 2004.

[127] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[128] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[129] Rudolf Meier, Kwong Shiyang, Gaurav Vaidya, and Peter K. L. Ng. DNA barcoding and taxonomy in diptera: A tale of high intraspecific variability and low identification uccess. *Systematic Biology*, 55(5):715–728, 2006.

[130] Gerardo Mendizabal-Ruiz, Israel Román-Godínez, Sulema Torres-Ramos, Ricardo A Salido-Ruiz, Hugo Vélez-Pérez, and J. Alejandro Morales. Genomic signal processing for DNA sequence clustering. *PeerJ*, 6:e4264, 2018.

[131] Xiao-Li Meng and David Van Dyk. The EM Algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Methodological)*, 59(3):511–567, 01 2002.

[132] Nancy Merino, Heidi S. Aronson, Diana P. Bojanova, Jayme Feyhl-Buska, Michael L. Wong, Shu Zhang, and Donato Giovannelli. Living at the extremes: Extremophiles and the limits of life in a planetary context. *Frontiers in Microbiology*, 10:780, 2019.

[133] Michael L. Metzker. Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.

[134] Pablo A. Millán Arias, Fatemeh Alipour, Kathleen A. Hill, and Lila Kari. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. *PLOS ONE*, 17(1):e0261531, 2022.

[135] Pablo A. Millán Arias, Joseph Butler, Gurjit S. Randhawa, Maximillian P. M. Soltysiak, Kathleen A. Hill, and Lila Kari. Environment and taxonomy shape the genomic signature of prokaryotic extremophiles. *Scientific Reports*, 13(1):16105, 2023.

[136] Pablo A. Millán Arias, Kathleen A. Hill, and Lila Kari. *i*DeLUCS: A deep learning interactive tool for alignment-free clustering of DNA sequences. *Bioinformatics*, 39(9):btad508, 2023.

[137] Pablo A. Millán Arias, Niousha Sadjadi, Monireh Safari, ZeMing Gong, Austin T. Wang, Scott C. Lowe, Joakim Bruslund Haurum, Iuliia Zarubiieva, Dirk Steinke, Lila Kari, Angel X. Chang, and Graham W. Taylor. BarcodeBERT: Transformers for biodiversity analysis. In *4th Workshop on Self-Supervised Learning: Theory and Practice. Neural Information Processing Systems (NeurIPS)*, 2023.

[138] Boris Mirkin. Reinterpreting the category utility function. *Machine Learning*, 45(2):219–228, 2001.

[139] Florian Mockand, Fleming Kretschmerand, Anton Kriese, Sebastian Böcker, and Manja Marz. Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. *Proceedings of the National Academy of Sciences*, 119(35):e2122636119, 2022.

[140] Camilo Mora, Derek P. Tittensor, Sina Adl, Alastair G. B. Simpson, and Boris Worm. How many species are there on earth and in the ocean? *PLOS Biology*, 9(8):1–8, 2011.

[141] Supratim Mukherjee, Dimitri Stamatis, Cindy Tianqing Li, Galina Ovchinnikova, Jon Bertsch, Jagadish Chandrabose Sundaramurthi, Mahathi Kandimalla, Paul A. Nicolopoulos, Alessandro Favognano, I-Min A. Chen, Nikos C. Kyrpides, and T. B. K. Reddy. Twenty-five years of Genomes OnLine Database (GOLD): Data updates and new features in v.9. *Nucleic Acids Research*, 51(D1):D957–D963, 2023.

[142] Salma Mukhtar, Naeem Rashid, Muhammad Farhan Ul Haque, and Kauser Abdulla Malik. Metagenomic approach for the isolation of novel extremophiles. In *Microbial Extremozymes*, pages 55–66, 2022.

[143] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[144] Benoit Nabholz, Sylvain Glémin, and Nicolas Galtier. Strong variations of mitochondrial mutation rate across mammals—the longevity hypothesis. *Molecular Biology and Evolution*, 25(1):120–130, 2007.

[145] Hiroshi Nakashima, Satoshi Fukuchi, and Ken Nishikawa. Compositional changes in RNA, DNA and proteins for bacterial adaptation to higher and lower temperatures. *The Journal of Biochemistry*, 133(4):507–513, 2003.

[146] Hiroshi Nakashima and Yuka Kuroda. Differences in dinucleotide frequencies of thermophilic genes encoding water soluble and membrane proteins. *Journal of Zhejiang University SCIENCE B*, 12(6):419–427, 2011.

[147] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[148] Annemarie Neuner, Holger W. Jannasch, Shimshon Belkin, and Karl. O. Stetter. *Thermococcus litoralis* sp. nov.: A new species of extremely thermophilic marine archaebacteria. *Archives of Microbiology*, 153(2):205–207, 1990.

[149] Eric Nguyen, Michael Poli, Marjan Faizi, Armin W. Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton M. Rabideau, Yoshua Bengio, Stefano Ermon, Christopher Re, and Stephen Baccus. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[150] Jakob Nybo Nissen, Joachim Johansen, Rosa Lundbye Allesøe, Casper Kaae Sønderby, Jose Juan Almagro Armenteros, Christopher Heje Grønbech, Lars Juhl Jensen, Henrik Bjørn Nielsen, Thomas Nordahl Petersen, Ole Winther, and Simon Rasmussen. Improved metagenome binning and assembly using deep variational autoencoders. *Nature Biotechnology*, 39(5):555–560, 2021.

[151] Shawn T. O'Neil. *A Primer for Computational Biology.* Oregon State University Press, 2017.

[152] Vaselli Orlando, Tassi Franco, Tedesco Dario, Poreda J. Robert, and Caprai Antonio. Submarine and inland gas discharges from the Campi Flegrei (Southern Italy) and the Pozzuoli Bay: Geochemical clues for a common hydrothermal-magmatic source. *Procedia Earth and Planetary Science*, 4:57–73, 2011.

[153] Emanuel Ott, Yuko Kawaguchi, Denise Kölbl, Elke Rabbow, Petra Rettberg, Maximilian Mora, Christine Moissl-Eichinger, Wolfram Weckwerth, Akihiko Yamagishi, and Tetyana Milojevic. Molecular repertoire of *Deinococcus radiodurans* after 1 year of exposure outside the International Space Station within the Tanpopo mission. *Microbiome*, 8(1):150, 2020.

[154] Shaojun Pan, Chengkai Zhu, Xing-Ming Zhao, and Luis Pedro Coelho. A deep siamese neural network improves metagenome-assembled genomes in microbiome datasets across different environments. *Nature Communications*, 13(1):2326, 2022.

[155] Anindya S. Panja, Smarajit Maiti, and Bidyut Bandyopadhyay. Protein stability governed by its structural plasticity is inferred by physicochemical factors and salt bridges. *Scientific Reports*, 10(1):1822, 2020.

[156] Donovan H. Parks, Maria Chuvochina, Pierre-Alain Chaumeil, Christian Rinke, Aaron J. Mussig, and Philip Hugenholtz. A complete domain-to-species taxonomy for bacteria and archaea. *Nature Biotechnology*, 38(9):1079–1086, 2020.

[157] Donovan H. Parks, Maria Chuvochina, David W. Waite, Christian Rinke, Adam Skarshewski, Pierre-Alain Chaumeil, and Philip Hugenholtz. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, 36(10):996–1004, 2018.

[158] Gregory B. Pauly, David M. Hillis, and David C. Cannatella. Taxonomic freedom and the role of official lists of species names. *Herpetologica*, 65(2):115 – 128, 2009.

[159] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[160] Giorgio Perrella, Anna Zioutopoulou, Lauren R Headland, and Eirini Kaiserli. The impact of light and temperature on chromatin organization and plant adaptation. *Journal of Experimental Botany*, 71(17):5247–5255, 2020.

[161] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[162] Gurjit S. Randhawa, Kathleen A. Hill, and Lila Kari. ML-DSP: Machine Learning with Digital Signal Processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels. *BMC Genomics*, 20(267), 2019.

[163] Gurjit S. Randhawa, Maximillian P.M. Soltysiak, Hadi El Roz, Camila P.E. de Souza, Kathleen A. Hill, and Lila Kari. Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *PLOS ONE*, 15(4):e0232391, 2020.

[164] Sujeevan Ratnasingham and Paul D. N. Hebert. BOLD: The barcode of life data system. *Molecular Ecology Notes*, 7(3):355–364, 2007.

[165] Sujeevan Ratnasingham and Paul D. N. Hebert. A DNA-based registry for all animal species: The barcode index number (BIN) system. *PLOS ONE*, 8(7):1–16, 2013.

[166] Isabelle Raymond-Bouchard, Jacqueline Goordial, Yevgen Zolotarov, Jennifer Ronholm, Martina Stromvik, Corien Bakermans, and Lyle G. Whyte. Conserved genomic and amino acid traits of cold adaptation in subzero-growing Arctic permafrost bacteria. *FEMS Microbiology Ecology*, 94(4), 2018.

[167] Monica Riley, James T. Staley, Antoine Danchin, Ting Zhang Wang, Thomas S. Brettin, Loren J. Hauser, Miriam L. Land, and Linda S. Thompson. Genomics of an extreme psychrophile, *Psychromonas ingrahamii*. *BMC Genomics*, 9(1):210, 2008.

[168] Andrewand Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, 2007.

[169] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958.

[170] Lynn J. Rothschild and Rocco L. Mancinelli. Life in extreme environments. *Nature*, 409(6823):1092–1101, 2001.

[171] Travis J. Sanders, Craig J. Marshall, and Thomas J. Santangelo. The role of archaeal chromatin in transcription. *Journal of Molecular Biology*, 431(20):4103–4115, 2019.

[172] Cristina Santos, Rafael Montiel, Blanca Sierra, Conceiçcão Bettencourt, Elisabet Fernandez, Luis Alvarez, Manuela Lima, Augusto Abade, and M. Pilar Aluja. Understanding differences between phylogenetic and pedigree-derived mtDNA mutation rate: A model using families from the Azores Islands (Portugal). *Molecular Biology and Evolution*, 22(6):1490–1505, 2005.

[173] Neil F.W. Saunders, Torsten Thomas, Paul M.G. Curmi, John S. Mattick, Elizabeth Kuczek, Rob Slade, John Davis, Peter D. Franzmann, David Boone, Karl Rusterholtz, Robert Feldman, Chris Gates, Shellie Bench, Kevin Sowers, Kristen Kadner, Andrea Aerts, Paramvir Dehal, Chris Detter, Tijana Glavina, Susan Lucas, Paul Richardson, Frank Larimer, Loren Hauser, Miriam Land, and Ricardo Cavicchioli. Mechanisms of thermal adaptation revealed from the genomes of the antarctic *archaea methanogenium frigidum* and *methanococcoides burtonii*. *Genome Research*, 13(7):1580–1588, 2003.

[174] Isabel Schwende and Tuan D. Pham. Pattern recognition and probabilistic measures in alignment-free sequence analysis. *Briefings in Bioinformatics*, 15(3):354–368, 2014.

[175] Laura S. Sherman, Joel D. Blum, Darrell K. Nordstrom, R. Blaine McCleskey, Tamar Barkay, and Costantino Vetriani. Mercury isotopic composition of hydrothermal systems in the Yellowstone Plateau volcanic field and Guaymas Basin sea-floor rift. *Earth and Planetary Science Letters*, 279(1-2):86–96, 2009.

[176] Fabian Sievers, Andreas Wilm, David Dineen, Toby J. Gibson, Kevin Karplus, Weizhong Li, Rodrigo Lopez, Hamish McWilliam, Michael Remmert, Johannes Söding, Julie D. Thompson, and Desmond G. Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Mol. Syst. Biol.*, 7(1):539, 2011.

[177] Gregory E. Sims, Se-Ran Jun, Guohong A. Wu, and Sung-Hou Kim. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106(8):2677–2682, 2009.

[178] Gregory A.C. Singer and Donal A. Hickey. Thermophilic prokaryotes have characteristic patterns of codon usage, amino acid composition and nucleotide content. *Gene*, 317:39–47, 2003.

[179] Barton E. Slatko, Andrew F. Gardner, and Frederick M. Ausubel. Overview of next-generation sequencing technologies. *Current Protocols in Molecular Biology*, 122(1):e59, 2018.

[180] Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017.

[181] Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[182] Vincent S. Smith. DNA Barcoding: Perspectives from a "Partnerships for Enhancing Expertise in Taxonomy" (PEET) Debate. *Systematic Biology*, 54(5):841–844, 2005.

[183] Vassiliki Betty Smocovitis. Unifying biology: The evolutionary synthesis and evolutionary biology. *Journal of the History of Biology*, 25(1):1–65, 1992.

[184] Stephen Solis-Reyes, Mariano Avino, Art Poon, and Lila Kari. An open-source k-mer based machine learning tool for fast and accurate sub-typing of HIV-1 genomes. *PLOS ONE*, 13(11):e0206409, 2018.

[185] Anja Spang, Jimmy H. Saw, Steffen L. Jørgensen, Katarzyna Zaremba-Niedzwiedzka, Joran Martijn, Anders E. Lind, Roel van Eijk, Christa Schleper, Lionel Guy, and Thijs J. G. Ettema. Complex archaea that bridge the gap between prokaryotes and eukaryotes. *Nature*, 521(7551):173–179, May 2015.

[186] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[187] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, 2008.

[188] András Szilágyi and Péter Závodszky. Structural differences between mesophilic, moderately thermophilic and extremely thermophilic protein subunits: Results of a comprehensive survey. *Structure*, 8(5):493–504, 2000.

[189] Ardi Tampuu, Zurab Bzhalava, Joakim Dillner, and Raul Vicente. ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PLOS ONE*, 14(9):1–17, 2019.

[190] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6), 2022.

[191] Andreas Teske, Dirk de Beer, Luke J. McKay, Margaret K. Tivey, Jennifer F. Biddle, Daniel Hoer, Karen G. Lloyd, Mark A. Lever, Hans Røy, Daniel B. Albert, Howard P. Mendlovitz, and Barbara J. MacGregor. The Guaymas Basin hiking guide to hydrothermal mounds, chimneys, and microbial mats: Complex seafloor expressions of subsurface hydrothermal circulation. *Frontiers in Microbiology*, 7, 2016.

[192] Yuandong Tian. Understanding deep contrastive learning via coordinate-wise optimization. In *Advances in Neural Information Processing Systems*, 2022.

[193] Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Allerton Conference on Communication, Control and Computation*, volume 49, pages 368–377, 1999.

[194] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[195] Alexander Topchy, Anil K. Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.

[196] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 10347–10357, 2021.

[197] Ming-Hsin Tsai, Yen-Yi Liu, Von-Wun Soo, and Chih-Chieh Chen. A new genome-to-genome comparison approach for large-scale revisiting of current microbial taxonomy. *Microorganisms*, 7(6):161, 2019.

[198] Pernilla Turner, Gashaw Mamo, and Eva Nordberg Karlsson. Potential and utilization of thermophiles and thermostable enzymes in biorefining. *Microbial Cell Factories*, 6(1):9, 2007.

[199] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020.

[200] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer: New York, 2000.

[201] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:6000–6010, 2017.

[202] Sergey V. Venev and Konstantin B. Zeldovich. Massively parallel sampling of lattice proteins reveals foundations of thermal adaptation. *The Journal of Chemical Physics*, 143(5):055101, 2015.

[203] Duong Vu, Marizeth Groenewald, and Gerard Verkley. Convolutional neural networks improve fungal classification. *Scientific Reports*, 10(1):12628, 2020.

[204] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[205] Quanhui Wang, Zhen Cen, and Jingjing Zhao. The survival mechanisms of thermophiles at high temperatures: An angle of omics. *Physiology*, 30(2):97–106, 2015.

[206] Yingwei Wang, Kathleen A. Hill, Shiva Singh, and Lila Kari. The spectrum of genomic signatures: From dinucleotides to chaos game representation. *Gene*, 346:173 – 185, 2005.

[207] Ziye Wang, Ronghui You, Haitao Han, Wei Liu, Fengzhu Sun, and Shanfeng Zhu. Effective binning of metagenomic contigs using contrastive multi-view representation learning. *Nature Communications*, 15(1):585, 2024.

[208] Madeline C. Weiss, Martina Preiner, Joana C. Xavier, Verena Zimorski, and William F. Martin. The last universal common ancestor between ancient earth chemistry and the onset of genetics. *PLOS Genetics*, 14(8):1–19, 2018.

[209] Anuradha Wickramarachchi and Yu Lin. Binning long reads in metagenomics datasets using composition and coverage information. *Algorithms for Molecular Biology*, 17(1):14, 2022.

[210] John J. Wiens. How many species are there on earth? progress and problems. *PLOS Biology*, 21(11):1–4, 2023.

[211] Carl R. Woese and George E. Fox. The concept of cellular evolution. *Journal of Molecular Evolution*, 10(1):1–6, 1977.

[212] Carl R. Woese, Otto Kandler, and Mark L. Wheelis. Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576–4579, 1990.

[213] Derrick E. Wood and Steven L. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, 2014.

[214] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 478–487, 2016.

[215] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[216] Shaohua Xu, Jiayan Wang, Zixiao Guo, Ziwen He, and Suhua Shi. Genomic convergence in the adaptation to extreme environments. *Plant Communications*, 1(6):100117, 2020.

[217] Benjamin M. Zeldes, Matthew W. Keller, Andrew J. Loder, Christopher T. Straub, Michael W. W. Adams, and Robert M. Kelly. Extremely thermophilic microorganisms as metabolic engineering platforms for production of fuels and industrial chemicals. *Frontiers in Microbiology*, 6:1209, 2015.

[218] Konstantin B. Zeldovich, Igor N. Berezovsky, and Eugene I Shakhnovich. Protein and DNA sequence determinants of thermophilic adaptation. *PLOS Computational Biology*, 3(1):1–11, 2007.

[219] A-B Zhang, Derek S. Sikes, C. Muster, and Shuqiang Li. Inferring species membership using DNA sequences with back-propagation neural networks. *Systematic Biology*, 57(2):202–215, 2008.

[220] Pengfei Zhang, Zhengyuan Jiang, Yixuan Wang, and Yu Li. Clmb: Deep contrastive learning for robust metagenomic binning. In *Research in Computational Molecular Biology: 26th Annual International Conference, RECOMB*, page 326–348, 2022.

[221] Zhichao Zhou, Yang Liu, Wei Xu, Jie Pan, Zhu-Hua Luo, and Meng Li. Genome- and community-level interaction insights into carbon utilization and element cycling functions of *Hydrothermarchaeota* in hydrothermal sediment. *mSystems*, 5(1):e00795–19, 2020.

[222] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V. Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024.

[223] Daochen Zhu, Wasiu Adewale Adebisi, Fiaz Ahmad, Sivasamy Sethupathy, Blessing Danso, and Jianzhong Sun. Recent development of extremophilic bacteria and their application in biorefinery. *Frontiers in Bioengineering and Biotechnology*, 8:483, 2020.

[224] Andrzej Zielezinski, Hani Z. Girgis, Guillaume Bernard, Chris-Andre Leimeister, Kujin Tang, Thomas Dencker, Anna Katharina Lau, Sophie Röhling, Jae Jin Choi, Michael S. Waterman, Matteo Comin, Sung-Hou Kim, Susana Vinga, Jonas S. Almeida, Cheong Xin Chan, Benjamin T. James, Fengzhu Sun, Burkhard Morgenstern, and Wojciech M. Karlowski. Benchmarking of alignment-free sequence comparison methods. *Genome Biology*, 20(1):144, 2019.

[225] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, 18(1):186, 2017.

APPENDICES

Appendix A

Default Notation

We follow the notation from the *Deep Learning* [63] textbook:

Numbers and Arrays	
a	A scalar (integer or real)
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets

$\mathbb{A}$	A set
$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{R}$	The set of real numbers
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	The arbitrary domains/codomains of a given function
$\{0, 1\}$	The set containing 0 and 1
$\{a, \ldots, b\}$	The set of all integers between a and b
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b

Functions

$f : \mathcal{X} \to \mathcal{Y}$	The function f with domain $\mathcal{X}$ and range $\mathcal{Y}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$ (Sometimes we write $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to lighten notation)
$\log x$	Natural logarithm of x
$\text{softmax}(\boldsymbol{x})$	Softmax function, $\frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_j)}$
$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	L^2 norm of $\boldsymbol{x}$
$\mathbf{1}[\text{condition}]$	is 1 if the condition is true, 0 otherwise

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{x \sim P}[f(x)]$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P \Vert Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Glossary of Biology Concepts

alternative splicing Cellular process whereby exons from the same gene are assembled in various combinations, resulting in multiple, related mRNA transcripts. These transcripts can then be translated into different proteins with unique structures and functions. 10

central dogma The central dogma of molecular biology, introduced by Francis Crick in 1958, is a theory that proposes a one-way transfer of genetic information from DNA to RNA to protein. It asserts that information cannot be transferred from a protein back to nucleic acids or to another protein, establishing a strict pathway for genetic information flow in biological systems. However, subsequent scientific discoveries have identified numerous exceptions to this theory. 9

codon A codon is a DNA or RNA sequence of three nucleotides that encodes for a specific amino acid or a stop signal for protein synthesis. Among the 64 possible codons, 61 encode the 20 amino acids that form proteins, and the remaining three are stop signals. 12

deoxyribonucleic acid Deoxyribonucleic acid (DNA) is the macromolecule responsible for encoding all the genetic information necessary for the development and functioning of all living organisms. It is made of two single strands that wind around each other. These strands comprise a sugar (deoxyribose) and phosphate backbone, with one of four nucleobases (adenine, cytosine, guanine, or thymine) attached to each sugar. Adenine pairs with thymine and cytosine with guanine through chemical bonds, connecting the two strands. 8

exon Region of the genome that will form an mRNA molecule after introns have been removed by RNA splicing. The term exon refers to the DNA sequence within a gene and the corresponding sequence in RNA transcripts. 10

gene The gene is considered the basic unit of inheritance. Genes are passed from parents to offspring and contain the information needed to specify physical and biological traits. Most genes code for specific proteins, or segments of proteins, which have differing functions within the body. Humans have approximately 20,000 protein-coding genes 10

intron A region of the genome that is part of a gene but does not code for amino acids, as it is removed from the final mature mRNA molecules after transcription. 10

nucleotide Basic building block of nucleic acids. A nucleotide consists of a sugar molecule, either ribose in RNA or deoxyribose in DNA, attached to a phosphate group and a nitrogenous nucleobase. The nucleobases present in DNA are adenine (A), cytosine (C), guanine (G) and thymine (T). In RNA, thymine is replaced by uracil (U). 8

progenote The most recent common ancestor to all organisms now living on Earth; specifically, the most recent progenitor of all archaea, bacteria and eukaryotes. 12

ribonucleic acid Ribonucleic acid (RNA), structurally similar to DNA, exists in all living cells, often as a single strand. An RNA molecule has a backbone of alternating ribose sugar and phosphate groups, with one of four nucleobases (adenine, uracil, cytosine, guanine) attached to each ribose. RNA types include messenger RNA (mRNA), ribosomal RNA (rRNA), and transfer RNA (tRNA), which have roles in gene expression regulation. Some viruses also use RNA as their genetic material. 8

transitions DNA substitution mutations in which nucleobases with similar chemical structures get interchaneged. This involves changes between two-ring nucleobases, also known as purines (A and G), or one-ring nucleobases, also known as pyrimidines (C and T). 42

transversions DNA substitution mutations in which nucleobases with different chemical structures get interchanged. This involves changes between a two-ring nucleobase (A or G) and a one-ring nucleobase (C or T). 42