

Advanced Machine Learning Techniques for Taxonomic Classification and Clustering of DNA Sequences

by

Fatemeh Alipour

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

© Fatemeh Alipour 2024

Examining Committee Membership

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

External Examiner: Dan Tulpan
Department of Animal Biosciences
University of Guelph

Supervisor(s): Lila Kari
David R. Cheriton School of Computer Science
University of Waterloo

Yang Young Lu
David R. Cheriton School of Computer Science
University of Waterloo

Internal Members: Yaoliang Yu
David R. Cheriton School of Computer Science
University of Waterloo

Olga Veksler
David R. Cheriton School of Computer Science
University of Waterloo

Internal-External Member: Brendan McConkey
Department of Biology
University of Waterloo

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 contributions of this thesis consist of the following articles:

- “DeLUCS: Deep learning for unsupervised clustering of DNA sequences” published in *PLOS ONE*. The major individual contributions are listed below.
 - **Pablo Millán Arias**: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, writing—review and editing.
 - **Fatemeh Alipour**: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing—review and editing.
 - **Kathleen A. Hill**: Conceptualization, investigation, writing—review and editing, and supervision.
 - **Lila Kari**: Conceptualization, formal analysis, investigation, methodology, supervision, writing—review and editing, and supervision.

P.M.A. and F.A. contributed equally to this work.

- “Leveraging machine learning for taxonomic classification of emerging astroviruses” published in *Frontiers in Molecular Biosciences*. The major individual contributions are listed below.
 - **Fatemeh Alipour**: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, and visualization.
 - **Connor Holmes**: Conceptualization, methodology, software, validation, formal, analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, and visualization.
 - **Yang Young Lu**: Methodology, writing—review and editing, supervision.
 - **Kathleen A. Hill**: Conceptualization, methodology, investigation, writing—review and editing, and supervision.
 - **Lila Kari**: Conceptualization, methodology, investigation, writing—review and editing, and supervision.

F.A. and C.H. are co-first authors of this work, representing computer science and biology, respectively.

- “CGRelust: Chaos game representation for twin contrastive clustering of unlabelled DNA sequences”, accepted for publication in *BMC Genomics*. The major individual contributions are listed below.
 - **Fatemeh Alipour**: Conceiving the study and writing the manuscript, designing and performing the experiments, conducting the data analysis and editing the manuscript.
 - **Kathleen A. Hill**: Conducting the data analysis and editing the manuscript, contributing biological expertise.
 - **Lila Kari**: Conceiving the study and writing the manuscript, conducting the data analysis and editing the manuscript.

Abstract

Advancements in genomic sequencing have exponentially increased the availability of DNA sequence data, presenting new opportunities and challenges in bioinformatics. This thesis addresses the critical need for scalable and precise computational tools to enhance taxonomic classification and clustering of DNA sequences through the application of advanced machine learning techniques. We introduce several novel algorithms designed to improve the accuracy and efficiency of these processes, focusing on both supervised and unsupervised machine learning approaches.

Firstly, we introduce “DeLUCS”, a deep learning-based method for the unsupervised clustering of DNA sequences. DeLUCS utilizes invariant information clustering to optimize the grouping of sequences without prior taxonomic information. We validate this model across multiple genomic datasets, including vertebrate mitochondrial genomes, bacterial genome segments, and viral sequences. DeLUCS significantly outperforms traditional methods such as K -Means and Gaussian Mixture Models, establishing its effectiveness for analyzing large, unlabelled DNA datasets.

Secondly, we explore the taxonomic classification of emerging astroviruses using a hybrid machine learning approach that effectively combines supervised and unsupervised techniques. Our novel methodology integrates k -mer composition analysis of whole genomes with host species data, which is crucial for accurately identifying and classifying novel and as-yet unclassified astroviruses. This approach addresses the challenges posed by genetic recombination and broad interspecies transmission that traditional host-based classifications fail to accommodate. By applying our method, we successfully proposed genus labels for 191 previously unclassified astrovirus genomes, and further identified potential cross-species infections, demonstrating the need for a revised understanding of astrovirus taxonomy.

Additionally, we present “CGRclust”, a novel method employing twin contrastive learning with convolutional neural networks (CNNs) for clustering Chaos Game Representations of DNA sequences. CGRclust leverages a unique data augmentation strategy and advanced model architecture and improves upon traditional sequence classification by avoiding the need for DNA sequence alignment or taxonomic labels. This approach demonstrates robust performance, outperforming existing methods like DeLUCS and MeShClust v3.0 with superior clustering accuracy on diverse datasets, including mitochondrial genomes and viral sequences.

Our comprehensive evaluations illustrate that these methods significantly advance the accuracy and computational efficiency of genomic data analysis. Specifically, our novel algorithms, including DeLUCS and CGRclust, leverage deep learning and contrastive learning models to refine the classification of DNA sequences, without the dependency on prior

taxonomic knowledge or sequence alignment. These improvements demonstrate how machine learning can transform the field of genomic analysis, setting a new standard for taxonomic classification and providing a foundation for future explorations in evolutionary biology and biodiversity.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Professors Lila Kari and Yang Lu, for their guidance and support throughout the course of my PhD studies. Professor Kari's exceptional patience and profound kindness have greatly aided my academic and personal development. Her extensive expertise in supervising, along with her continual encouragement, has been invaluable. This thesis would not have been possible without her help, which has firmly established her as both a mentor and a true role model. I also greatly appreciate Professor Lu's expertise, insights, and guidance, which have been crucial in shaping my research direction and assisting me in overcoming the challenges of academic research.

I would like to express my sincere gratitude to Professor Kathleen Hill for her invaluable contributions to my research projects. As a mentor from the biology area, her expertise and continued encouragement have been essential throughout this journey. Her kindness and support have made her an exceptional mentor, and without her expertise in biology, this research would not have been possible.

I would also like to extend my thanks to the examining committee members: Professors Dan Tulpan, Brendan McConkey, Olga Veksler, and Yaoliang Yu. I appreciate their willingness to serve on my committee and their assistance throughout this process.

I would like to thank my fellow lab members, Monireh Safari, Pablo Millán Arias, Niousha Sadjadi, Zihao Wang, and Wanxin Li, for their support, assistance, and encouragement, which enriched my research experience. I am grateful for the collaborative efforts and friendships that have made this journey a memorable one. I am also thankful to the members of Professor Hill's lab, including Connor Holmes and Joseph Butler, for their collaboration and assistance throughout my research. Additionally, I extend my gratitude to Professors Camila de Souza and Gurjit Randhawa, for their invaluable support and guidance throughout my research.

I am profoundly grateful to my parents for their endless love, encouragement, and the many sacrifices they have made throughout my life. To my brother, Hamidreza, your love and constant support have been a great source of comfort and strength. I am also deeply grateful to my friends for their continued encouragement and the joyful moments we have shared. Your friendship has been a source of strength and happiness throughout this time.

A special thank you to my dearest husband, Pourya, the love of my life and my best friend. This thesis would not have been possible without your endless patience, care, and love. You have been my emotional anchor and the reason I have been able to keep going. Your belief in me has been my foundation, and I am deeply grateful for your presence throughout this journey.

Dedication

To my loving husband, Pourya, for your continuous support, belief in me, and for being my source of strength.

To my parents, Narges and Abbas, for your constant encouragement and love, which have supported me through every challenge.

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Statement of Contributions	iv
Abstract	vi
Acknowledgements	viii
Dedication	ix
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Motivations	1
1.2 Thesis Overview	3
1.3 Contributions	4
2 Background	6
2.1 Genomic Language of Life	7

2.2	Evolutionary Taxonomy	10
2.3	DNA Representations	12
2.3.1	Ordinal Encoding	12
2.3.2	One-hot Encoding	13
2.3.3	Frequency of k -mers	14
2.3.4	Chaos Game Representation	15
2.3.5	Subword Tokenization Methods	19
2.3.6	Additional Methods for DNA Representation	20
2.4	Sequence Alignment and Comparison Techniques	20
2.4.1	Global Alignment	21
2.4.2	Local Alignment	22
2.4.3	Multiple Sequence Alignment	23
2.4.4	BLAST (Basic Local Alignment Search Tool)	24
2.4.5	Sequence Comparison Techniques Beyond Traditional Alignment	25
2.5	Supervised Machine Learning	26
2.5.1	Support Vector Machines (SVMs)	27
2.5.2	Artificial Neural Networks	28
2.6	Unsupervised Machine Learning	32
2.6.1	K -Means	33
2.6.2	Gaussian Mixture Models	34
2.6.3	Hierarchical Clustering	34
2.6.4	Contrastive Clustering	35
2.6.5	Dimensionality Reduction	36
2.7	Machine Learning Evaluation Metrics	38
2.7.1	Classification Accuracy	38
2.7.2	Clustering Accuracy	39
2.7.3	Normalized Mutual Information (NMI)	40

2.7.4	Adjusted Rand Index (ARI)	41
2.7.5	Silhouette Coefficient	41
2.8	Recent Applications and Relevant Technologies	42
2.8.1	Machine Learning with Digital Signal Processing (ML-DSP)	42
2.8.2	Viral Genome Deep Classifier (VGDC)	43
2.8.3	MeShClust v3.0	44
3	DeLUCS: Deep Learning for Unsupervised Clustering of DNA Sequences	45
3.1	Introduction	46
3.2	Related Work	48
3.2.1	Supervised Machine Learning Approaches	49
3.2.2	Unsupervised Machine Learning Approaches	50
3.3	Materials and Methods	52
3.3.1	Method Overview	52
3.3.2	Invariant Information Clustering (IIC)	54
3.3.3	Generation of Mimic Sequences	56
3.3.4	Artificial Neural Network (ANN) Architecture	56
3.3.5	Evaluation	58
3.3.6	Implementation	59
3.3.7	Datasets	59
3.4	Results	64
3.4.1	Qualitative and Quantitative DeLUCS Performance Results	64
3.4.2	Methodology Results	73
3.5	Discussion	76
3.6	Conclusions	78

4	Machine Learning Methods for Taxonomic Classification of Emerging Astroviruses	80
4.1	Introduction	81
4.2	Materials and Methods	83
4.2.1	Datasets	84
4.2.2	Overview of the Methodological Pipeline	85
4.2.3	An Optional Initial Component of Eliminating Recombinant Sequences	87
4.2.4	Prong 1: Supervised Machine Learning	89
4.2.5	Prong 2: Unsupervised Machine Learning	92
4.2.6	Selection of k -mer Size	95
4.2.7	Astrovirus Near-Neighbour Analysis (Potyvirus)	97
4.2.8	Computational Setup	98
4.3	Results	99
4.3.1	Identification and Elimination of Recombinant Sequences	99
4.3.2	Classification of Unclassified Astroviruses with Mammal and Avian Hosts	99
4.3.3	Classification of Unclassified Astroviruses with Hosts Other than Mammals and Avian Species	104
4.3.4	A Closer Look at Genera Mamastrovirus and Avastrovirus	105
4.4	Discussion	109
5	CGRclust—Advanced Clustering of DNA Sequences Using Twin Contrastive Learning	112
5.1	Introduction	113
5.2	Related Work	114
5.3	Materials and Methods	115
5.3.1	Datasets	116
5.3.2	Method Overview	119
5.3.3	DNA Data Augmentation: Mimic Sequences	123

5.3.4	Twin Contrastive Learning	125
5.3.5	Backbone Model Architecture	129
5.3.6	Majority Voting Scheme	129
5.3.7	Confidence Interval for Clustering Accuracy	131
5.3.8	Experimental Settings and Implementation	133
5.4	Results	134
5.4.1	Qualitative Performance of Twin Contrastive Learning	134
5.4.2	Quantitative Performance Analysis and Comparison with Other Methods	135
5.4.3	Summative Observations	140
5.5	Discussion	141
5.6	Conclusions	144
6	Summary and Future Directions	145
	References	149
	APPENDICES	182
A	Proposed Classification for As-yet Unclassified Astroviruses	183
A.1	3PCM's Proposed Classification with Inclusion of Candidate Recombinants	184
A.2	3PCM's Proposed Classification with Exclusion of Candidate Recombinants	186
B	Comparative Evaluation of CGRclust with Phylonium	188

List of Figures

1.1 Mitochondrial DNA Chaos Game Representation of human, chimpanzee, and goldfish	3
2.1 Double helix and chemical structure of DNA	8
2.2 The central dogma of molecular biology	10
2.3 Visualization of the k -mer counting method for a DNA sequence	14
2.4 CGR square labelling and sequence representation	17
2.5 Frequency chaos game representation of real and computer-generated DNA sequences	18
2.6 Illustration of linear SVM classifier separating two classes	27
2.7 Architecture of a neural network	29
2.8 Architecture of a convolutional neural network	31
3.1 Machine learning-based alignment-free methods for classification/clustering of DNA sequences	49
3.2 DeLUCS pipeline	53
3.3 Architecture of the deep artificial neural network used in DeLUCS pipeline	57
3.4 DeLUCS learning process for the clustering of mtDNA full genomes into five different clusters	65
3.5 DeLUCS confusion matrices for mtDNA dataset	67
3.6 DeLUCS confusion matrices for bacteria dataset	69
3.7 DeLUCS confusion matrices for viral dataset	71

3.8	The effect of the number of mimic sequences on clustering accuracy	74
3.9	Learning curves for the training process of DeLUCS with, and without the addition of Gaussian noise to the parameters of the networks	75
3.10	Comparison between training a single ANN, versus using majority voting between 10 ANNs to decide the final cluster label in DeLUCS	76
4.1	An overview of Three-Pronged Classification Method (3PCM)	88
4.2	Confusion matrix for the clustering of astrovirus whole genomes into the Avastrovirus and Mamastrovirus genera using K -Means++ algorithm in Prong 2 of 3PCM	94
4.3	Effect of k -mer size in the performance of 3PCM's Prong 1 and Prong 2 . .	96
4.4	Taxonomic classification of class Stelpaviricetes	97
4.5	Confusion matrices for classification of unclassified astroviruses, using Prong 1 and Prong 2 of 3PCM	101
4.6	A comparison of as-yet unclassified sequences obtained from hosts Amphibia, Actinopterygii, and Reptilia with 875 astrovirus sequences belonging to Mamastrovirus or Avastrovirus	106
4.7	3D PCA data visualizations of Mamastrovirus and Avastrovirus sequences k -mer frequencies	107
5.1	Overview of the CGRclust pipeline	122
5.2	Effect of the weight parameter α in the CGRclust training loss function . .	128
5.3	3D visualization of CGRclust temperature parameters optimization	130
5.4	Architecture of CGRclust's backbone model	131
5.5	Effect of majority voting scheme in CGRclust pipeline	132
5.6	CGRclust's evolution of clustering Cypriniformes mtDNA sequences	136

List of Tables

3.1	Details of the datasets used in computational tests 1 through 6 (full vertebrate mitochondrial genomes) of DeLUCS	61
3.2	Details of the datasets used in computational tests 7 and 8 (randomly selected bacterial genome segments) of DeLUCS	63
3.3	Details of the datasets used in computational tests 9, 10 and 11 (viral genes/genomes) of DeLUCS.	64
3.4	Clustering/classification accuracies for the mtDNA datasets	66
3.5	Clustering/classification accuracies for the bacterial datasets	68
3.6	Clustering/classification accuracies for the viral datasets	70
3.7	Comparison of DeLUCS, K -Means++, and GMM across all eleven computational tests	72
4.1	Description of Dataset 1 containing 992 viral genomes belonging to the family <i>Astroviridae</i>	85
4.2	Description of Dataset 2, a subset of Dataset 1 consisting of sequences with available ground truth.	85
4.3	Distribution of the hosts of the 308 astrovirus genomes in Dataset 1 that are as-yet unclassified at the genus level.	86
4.4	Description of Dataset 3, consisting of astrovirus genomes in Dataset 1 that are as-yet unclassified and have mammalian/avian hosts	86
4.5	Comparative classification accuracy of fifteen supervised classifier candidates for 3PCM	91
4.6	Comparative classification accuracy of 3PCM's Prong 1 vs. ML-DSP and VGDC methods	92

4.7	Performance of Prong 2 in clustering astrovirus whole genomes using <i>K</i> -Means++, GMM, and hierarchical clustering algorithms	93
4.8	Performance of 3PCM's Prong 2 against three state-of-the-art alignment-free machine-learning viral genome clustering methods (DeLUCS, <i>i</i> DeLUCS, and MeShClust v3.0)	95
4.9	Description of dataset comprising 2,422 Stelpaviricetes viral genomes	98
4.10	Performance of Prong 1 and Prong 2 applied to the classification/clustering of RNA sequences of the Stelpaviricetes class into family <i>Astroviridae</i> and family <i>Potyviridae</i>	98
4.11	Performance of Prong 1 and Prong 2 applied to the classification/clustering unclassified astroviruses into Mamastroviruses and Avastroviruses	102
4.12	Number of as-yet unclassified viral genomes for which two or all three, of the prongs, agree in their genus label prediction	103
4.13	Description of the dataset including astrovirus genomes belonging to Avastrovirus and Mamastrovirus genera, augmented with astrovirus genomes of unknown genus and having one of the Reptilia, Actinopterygii, or Amphibia hosts	105
4.14	Performance of Prong 1 and Prong 2 applied to the classification/clustering of Mamastrovirus sequences into <i>HAsTV</i> and <i>Non-HAsTV</i> Mamastrovirus .	108
4.15	Performance of Prong 1 and Prong 2 applied to the classification/clustering of Avastrovirus sequences into <i>GoAsTV</i> and <i>Non-GoAsTV</i> Avastrovirus . .	110
5.1	Details of CGRclust Group 1 dataset (Cypriniformes full mitochondrial genomes)	117
5.2	Details of CGRclust Group 2 dataset (viral whole genomes)	119
5.3	Details of CGRclust Group 3 dataset (mtDNA of Insects, Protists, and Fungi)	120
5.4	Details of CGRclust Group 4 dataset (synthetic sequences)	121
5.5	Comparison of various augmentation techniques in CGRclust pipeline	125
5.6	Optimal threshold values across datasets for MeShClust v3.0	135
5.7	CGRclust performance of clustering the Group 1 dataset (Cypriniformes mtDNA)	137
5.8	CGRclust performance of clustering the Group 2 dataset (viral whole genomes)	138

5.9	CGRclust performance of clustering the Group 3 dataset (mtDNA of Insects, Protists, and Fungi)	139
5.10	CGRclust performance of clustering the Group 4 dataset (the synthetic datasets)	140
5.11	CGRclust training time	142
A.1	Accession IDs of the proposed Avastroviruses when all three prongs agreed on the classification.	184
A.2	The Accession IDs of the proposed Avastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.	184
A.3	Accession IDs of the proposed Mamastroviruses when all three prongs agreed on the classification.	185
A.4	The Accession IDs of the proposed Mamastroviruses when there was agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.	185
A.5	Accession IDs of the proposed Avastroviruses when all three prongs agreed on the classification	186
A.6	The Accession IDs of the proposed Avastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.	186
A.7	Accession IDs of the proposed Mamastroviruses when all three prongs agreed on the classification.	187
A.8	The Accession IDs of the proposed Mamastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.	187
B.1	Comparison of clustering accuracies between CGRclust and phylonium for five out of twenty-five datasets where phylonium successfully generated distance matrices.	189

Chapter 1

Introduction

1.1 Motivations

The rapidly evolving field of genomics generates vast amounts of DNA sequence data, presenting both unique opportunities and significant computational challenges [248]. Biodiversity, defined as the variety of life in the world or in a particular habitat or ecosystem, is a critical area of study that benefits from this rapid evolution in genome sequencing technology. Recent estimates suggest that we have only described a small percentage of Earth's total biodiversity: For eukaryotes, only about 14% of species on Earth and 9% of species in the ocean have been indexed in central databases [205]. The percentage for microbes is likely even lower, which highlights the scale of unexplored genetic diversity [183].

Traditional methods for analyzing DNA sequence data, which often rely on sequence alignment, are computationally expensive and may fail as data volumes expand and sequence diversity increases [153]. Additionally, these methods can struggle with the precision needed for detailed taxonomic classification at the microbial level, which is essential for understanding biodiversity and evolutionary biology [323, 76]. In response to these challenges, alignment-free methods have emerged as computationally more efficient alternatives, allowing for the rapid analysis of large genomic datasets. This is particularly valuable in biodiversity studies, where researchers often deal with massive amounts of sequence data from numerous species. Importantly, these methods facilitate inclusive analyses across the entire tree of life, allowing comparisons that span multiple domains and enhance our understanding of evolutionary relationships [333].

Advances in genomics require new technologies for identifying species that can handle the increasing amount of data and keep up with the fast discovery of new species. This

thesis is motivated by the urgent need for scalable, accurate, and efficient computational tools that leverage the power of modern machine learning to provide deeper insights into genomic data. These tools must overcome the limitations inherent in traditional methods, facilitating rapid and precise classification—determining the exact category or taxon of DNA sequences—and clustering—grouping sequences based on genetic similarities without predefined categories. Both processes are critical for advancing our understanding of genetic relationships and biodiversity, with clustering playing a vital role in revealing evolutionary dynamics and the emergence of new genotypes, strains, and species.

With the increasing complexity of genomic data, advanced techniques are needed to handle large amounts of information effectively. This complexity arises from the extensive length of genomes, the diversity across the tree of life, the varying degrees of genetic divergence, and the internal sequence characteristics such as repeats and rearrangements. These factors demand robust computational tools to handle the scale, diversity, and dynamics of genomic data efficiently [107]. Machine learning, particularly deep learning, is well-suited for this task due to its ability to detect complex, underlying patterns in large datasets [204]. Both classification and clustering are fundamental concepts in machine learning, addressing different learning paradigms that enable accurate identification and grouping of DNA sequences.

Genomic signatures are distinctive patterns or subsequences found within the DNA of organisms that uniquely characterize different species [147]. These signatures are crucial for accurately identifying and differentiating species, providing a basis for genetic analysis and classification. An example of such a signature is the Chaos Game Representation (CGR), a graphical method for representing DNA sequences [21, 136]. This DNA representation is considered a genomic signature since the CGR of sequences from closely related organisms are more similar than that between distantly related ones. Some examples of CGRs of real DNA sequences are illustrated in Figure 1.1. By leveraging such signatures, machine learning models can significantly enhance their predictive accuracy and provide deeper insights into genetic diversity and evolutionary relationships [309].

By developing and implementing new machine learning models specifically designed for classifying and clustering DNA sequences, this research aims to bridge the gap between biological data collection and its analytics and interpretation, contributing to the fields of computational biology and bioinformatics. These advanced models enhance our ability to detect, identify, and understand complex biological patterns quickly and accurately, directly from primary DNA sequences without relying on prior annotations. Moreover, by automating and improving the accuracy of DNA classification and clustering, we can make genetic research much more efficient, improving the efficiency and resources needed to extract useful information from large datasets. Ultimately, this work aims to provide

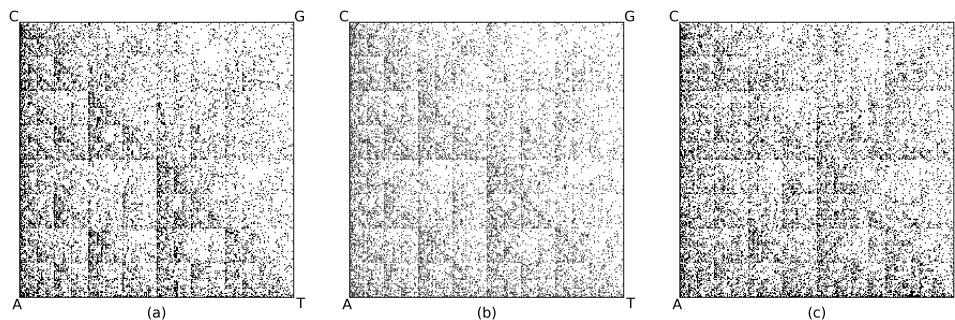


Figure 1.1: Chaos Game Representation (CGR) of mitochondrial DNA sequences showcasing genetic diversity across species. (a) CGR of *Homo sapiens* (human) mitochondrial DNA, 16,569 bp (accession ID: KX228192.1). (b) CGR of *Pan troglodytes* (chimpanzee) mitochondrial DNA, 16,554 bp (accession ID: NC_001643.1). (c) CGR of *Carassius langsdorfii* (goldfish) mitochondrial DNA, 16,578 bp (accession ID: AB006953.1). These representations visually encapsulate the unique genomic signatures of each species, enabling comparative analysis of their genetic structures and evolutionary relationships. Note the relative similarity between (a) and (b).

scientists with useful tools for studying the diversity of life, potentially aiding in the discovery and classification of new taxonomic groups, and offering insights that may contribute to healthcare-related research.

1.2 Thesis Overview

This dissertation is structured to address the computational challenges in taxonomic classification and clustering of DNA sequences through advanced machine learning approaches. It is organized into six chapters as follows:

- **Chapter 1: Introduction** — This chapter sets the stage by discussing the motivations behind the thesis, the importance of advanced computational methods in genomics, and outlines the structure of the thesis. It also highlights the key contributions of this research, including the development of novel machine learning models for taxonomic classification and clustering of DNA sequences.
- **Chapter 2: Background** — Provides a comprehensive overview of the biological and computational foundations necessary for understanding DNA sequence analysis.

It covers the basic principles of genomic structures, evolutionary taxonomy, various DNA representations, and traditional as well as modern computational techniques used for sequence alignment and comparison.

- **Chapter 3: DeLUCS: Deep Learning for Unsupervised Clustering of DNA Sequences** — Introduces “DeLUCS”, a novel deep learning-based method designed for the unsupervised clustering of DNA sequences. This chapter details the methodology, implementation, and validation of the model using diverse datasets.
- **Chapter 4: Machine Learning Methods for Taxonomic Classification of Emerging Astroviruses** — Discusses the application of supervised and unsupervised machine learning methods to classify emerging astroviruses, detailing the datasets, methodological approaches, and the results from computational experiments.
- **Chapter 5: CGRclust—Advanced Clustering of DNA Sequences Using Twin Contrastive Learning** — Presents “CGRclust”, a novel method utilizing twin contrastive learning for clustering DNA sequences. This chapter explores the algorithm’s design, its experimental setup, and the comparative analysis with other clustering methods.
- **Chapter 6: Summary and Future Directions** — Summarizes the findings of the thesis, discusses the implications of the research, and suggests future research paths based on the results of the current study.

Each chapter builds upon the previous ones, creating a coherent narrative that demonstrates how machine learning enhances the methods and processes used in genomic analysis. It also highlights the development of computational tools that facilitate significant biological discoveries, advancing our understanding of genetic relationships and biodiversity.

1.3 Contributions

This thesis makes several significant contributions to the field of bioinformatics, particularly in the taxonomic classification and clustering of DNA sequences using advanced machine learning techniques. The key contributions of this research are as follows:

1. **Development of DeLUCS:** A novel deep learning framework designed for unsupervised clustering of DNA sequences. DeLUCS leverages the power of invariant

information clustering to enhance the accuracy and efficiency of unsupervised learning in genomic contexts. This framework has been tested across diverse datasets, demonstrating superior performance compared to existing methods.

2. **Novel classification of emerging astroviruses:** Introduction of a three-pronged methodology combining both supervised and unsupervised machine learning techniques with the incorporation of information about host species to accurately classify emerging astroviruses. This approach includes a novel initial component that enhances the robustness of the classification by systematically eliminating recombinant sequences that could skew the analysis.
3. **CGRclust algorithm:** Presentation of CGRclust, an advanced clustering technique utilizing twin contrastive learning. This method significantly improves upon traditional clustering algorithms by incorporating a novel DNA data augmentation process and a backbone model architecture tailored for high-dimensional genomic data. CGRclust refines the clustering process and provides a methodological framework that can be adapted for other complex biological data types.
4. **Comprehensive evaluation and Benchmarking:** Extensive evaluation of the proposed methods against state-of-the-art techniques using standard metrics such as classification accuracy, clustering accuracy, Normalized Mutual Information (NMI), and the Adjusted Rand Index (ARI). This comprehensive benchmarking provides a robust validation of the methods and highlights their potential applicability in broader genomic studies.
5. **Application and implications:** Demonstration of the practical applications of the developed techniques in real-world biological data analysis. This includes insights into the evolutionary biology of viruses, potential identification of novel viral species, and contributions to the understanding of genetic diversity through more refined taxonomic classification.

The methodologies developed in this thesis represent a significant advancement in the capabilities of machine learning applications in genomics, offering scalable, accurate, and efficient tools that are crucial for the modern demands of biological research and discovery.

Chapter 2

Background

This chapter provides an overview of the basic biological concepts and computational methods relevant to the study of classification and clustering of DNA sequences using machine learning. This overview serves as the foundation for subsequent chapters of this thesis. This chapter is divided into eight sections:

- **Genomic Language of Life:** This section introduces the fundamental biological structure of DNA and its significance in biological processes. It explains the chemical structure of DNA, the role of nucleotides, and the principles of the double helix configuration. The concept of the central dogma of molecular biology is discussed, highlighting the processes of transcription and translation which convert genetic information from DNA to proteins.
- **Evolutionary Taxonomy:** This section focuses on the classification of organisms based on evolutionary relationships and morphological characteristics. It covers traditional approaches and the integration of molecular data from DNA sequencing, enhancing the accuracy and depth of taxonomic classifications.
- **DNA Representations:** The transition from biological sequences to a format suitable for computational analysis is explored in this section. Multiple methods of DNA representations, such as ordinal encoding, one-hot encoding, and the use of k -mers, are discussed, detailing how these representations preserve biological information while preparing data for computational processing and machine learning.
- **Sequence Alignment and Comparison Techniques:** This section details the computational methods used to identify similarities in genomic sequences, which

suggest functional or evolutionary relationships. It explains both global and local alignment techniques, including the Needleman-Wunsch and Smith-Waterman algorithms, as well as the application of multiple sequence alignment for comparative genomic studies.

- **Supervised Machine Learning:** This section outlines the application of supervised machine learning algorithms to genomic data analysis, discussing specific models such as Support Vector Machines (SVMs) and neural networks, and their relevance to DNA sequence classification.
- **Unsupervised Machine Learning:** This section explores clustering and other unsupervised techniques that identify patterns and structures in unlabelled genomic data. It covers approaches from traditional clustering algorithms such as K -Means to more advanced methods incorporating deep learning. This section also discusses dimensionality reduction techniques that simplify the analysis of high-dimensional data, crucial for enhancing computational efficiency and visualization.
- **Machine Learning Evaluation Metrics:** This section provides a comprehensive guide to the metrics used to evaluate the performance of machine learning models in both supervised and unsupervised contexts.
- **Recent Applications and Relevant Technologies:** The section reviews recent applications and technologies in genomic research, including advanced tools for DNA sequence classification and clustering, and will set the stage for the methodologies developed in this thesis.

2.1 Genomic Language of Life

Deoxyribonucleic acid (DNA) is a molecule that contains the genetic materials used in the growth, development, functioning, and reproduction of all known living organisms and many viruses. During reproduction, DNA, along with the instructions it contains, is passed from adult organisms to their offspring [128]. As shown in Figure 2.1, double-stranded DNA commonly consists of two polynucleotide strands that twist around each other to form a structure known as a *double helix*, although it can also exist as a single strand during certain biological processes. Every single-stranded DNA sequence is made up of simpler units known as *nucleotides*, each composed of a phosphate group, a five-carbon sugar called deoxyribose, and one of four nitrogenous bases: adenine (A), thymine (T), cytosine (C), and guanine (G). The two strands of DNA are oriented in opposite

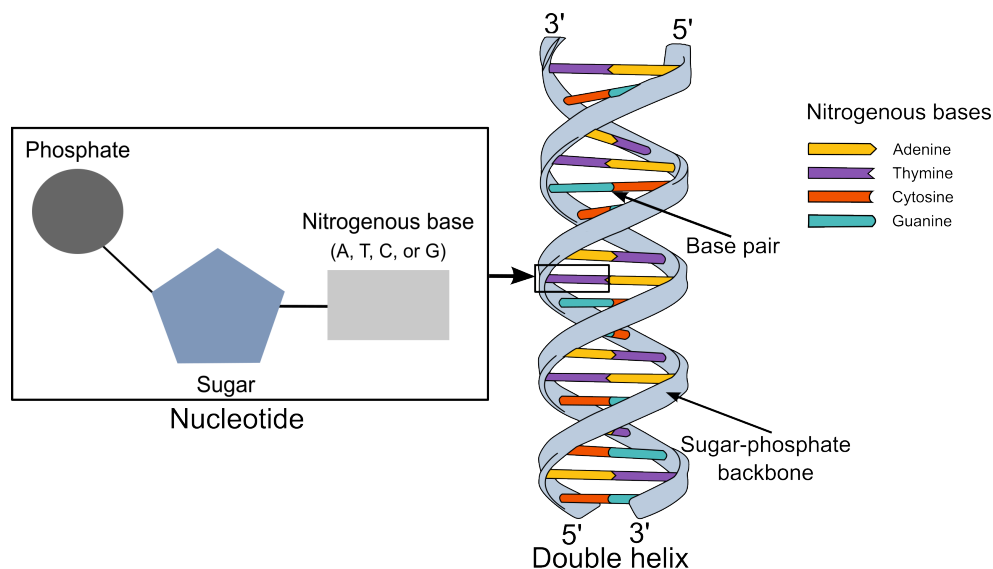


Figure 2.1: Illustration of a DNA double helix with detailed chemical structure, and color-coded labels for the four nitrogenous bases: adenine (A), thymine (T), cytosine (C), and guanine (G). This representation emphasizes the molecular structure and base pairing rules of DNA, and illustrates the orientation of DNA strands, marked by the 5' to 3' and 3' to 5' directions.

directions, one from the 5' end to the 3' end and the other from the 3' end to the 5' end, a configuration known as anti-parallel. The numbers 5' and 3' refer to the carbon atoms in the sugar component of the nucleotide. The 5' end has a phosphate group attached to the fifth carbon atom of the sugar, while the 3' end has a hydroxyl group attached to the third carbon atom of the sugar. To create the double helix structure of DNA, the four nucleotides on opposing strands are linked by hydrogen bonds. In DNA, each nucleotide in one strand pairs with a nucleotide in the opposite strand: adenine pairs with thymine, and cytosine pairs with guanine. Based on this pairing rule, the sequence of one strand determines the sequence of its complementary strand. This opposite orientation is critical for DNA replication and transcription, as it enables enzymes such as DNA polymerase, which synthesizes complementary DNA strands during replication, to function effectively. Additionally, this orientation is vital for the mechanisms of DNA repair, allowing repair enzymes to accurately identify and correct errors [7].

While DNA is the primary genetic material in all living organisms, viruses—which are not considered living organisms—exhibit a diversity in their genetic materials. Some viruses, such as influenza and coronaviruses, rely upon *ribonucleic acid (RNA)*, instead

of DNA. Viral genomes can vary significantly, including RNA or DNA, and can be either single-stranded or double-stranded. In contrast to DNA, RNA contains the sugar ribose in place of deoxyribose and uses the nucleobase uracil (U) instead of thymine (T) [87]. The entire DNA in an organism—or RNA in the case of RNA viruses—forms its *genome*, which is defined as the complete set of genetic instructions essential for building and maintaining that organism. DNA sequences are organized into *chromosomes*, each a single, contiguous sequence that not only carries genetic information but also plays a crucial role in ensuring the accurate distribution of DNA during cell division. An organism’s genome may comprise multiple chromosomes, representing different contiguous sequences that collectively contain the complete set of genetic instructions [302]. The study of genomes—including the structure, annotation, function, inheritance, divergence, and evolution of genetic materials—is known as *genomics*. This field explores how genetic material is arranged, expressed, and regulated within an organism, providing insights into the biological complexities that define genetic identity and variability.

Central to the role of DNA in cellular processes is the *central dogma* of molecular biology, which explains the flow of genetic information within a cell: DNA is *transcribed* into messenger RNA (mRNA), which is then *translated* into proteins. Specifically, a protein is a chain of *amino acids* (there are 20 different amino acids), and an RNA sequence is translated into a protein according to a mapping called the *genetic code*. The genetic code maps each 3-letter DNA/RNA subsequence, called a *codon*, to a specific amino acid [57]. In this process, transfer RNA (tRNA), which carries specific amino acids based on the sequence of the messenger RNA (mRNA), moves to the *ribosome*, a complex molecular machine that synthesizes proteins. The ribosome, composed of ribosomal RNA (rRNA) and proteins, consists of two major subunits: one reads the mRNA sequence while the other catalyzes the assembly of the carried amino acids into protein chains. Ribosomal RNA plays a critical role in the structural and catalytic functions of the ribosome, ensuring the correct alignment and linking of amino acids.

In cells, proteins perform a variety of functions, including acting as enzymes, structural components, and signaling molecules. The synthesis of proteins is directed by *genes*, specific sequences of DNA that serve as the templates for producing proteins through the processes of *transcription* and *translation*. The information flow within a cell, as well as the processes of transcription and translation, are illustrated in Figure 2.2. The regulation of these processes is managed by various genetic elements such as promoters and enhancers, which are specific DNA sub-sequences. Promoters initiate the transcription of genes, while enhancers boost transcription levels, ensuring proteins are synthesized at the right times and in required amounts. As a result, DNA may not only be viewed as a series of instructions but as an active language, directing and influencing cellular functions and processes

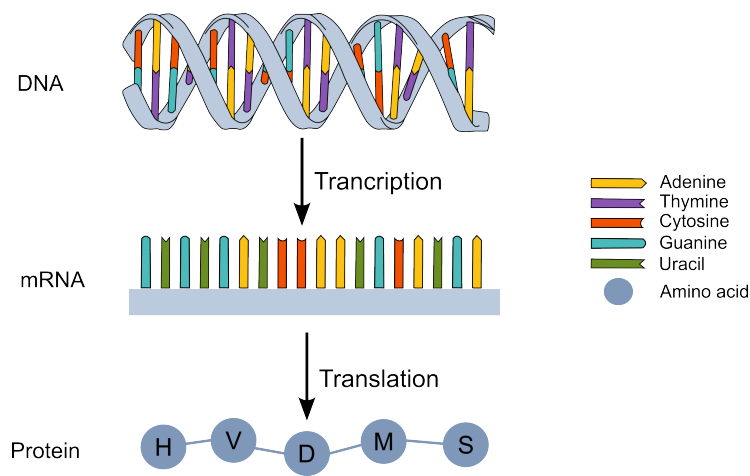


Figure 2.2: The central dogma of molecular biology outlines the flow of genetic information from DNA to RNA to protein. Genetic information, consisting of nucleotide sequences in DNA that encode instructions for building proteins, is transcribed into messenger RNA (mRNA) and then translated into proteins. These proteins are complex molecules that perform various cellular functions.

in the form of proteins.

Within a gene, regions known as *exons* and *introns* play critical roles. Exons, the coding regions, are sequences that are transcribed and translated into proteins, while introns, the non-coding regions, are sequences that are transcribed but removed before mRNA translation and do not translate into proteins. Consequently, not all sequences in the genome are translated into proteins; much of the genome consists of non-coding regions and repetitive sequences, which serve regulatory and structural functions or whose roles are yet to be fully understood.

In the following sections, although our discussion will predominantly focus on DNA, it is important to note that the principles and mechanisms described also apply to RNA, providing a broader understanding of genetic functions across different types of organisms.

2.2 Evolutionary Taxonomy

Evolutionary *taxonomy*, also known as classical taxonomy, is a branch of biological classification that categorizes organisms based on their evolutionary relationships, focusing on genetic relatedness to determine how species are interconnected through evolutionary

history. The origins of taxonomy can be traced back to Aristotle, who is often regarded as the first taxonomist due to his early attempts to classify living organisms based on their characteristics [69]. Evolutionary taxonomy, however, has its roots in the pioneering work of early naturalists such as Carl Linnaeus, who developed a system for naming and classifying organisms [324]. Although modified, this system is still in use today, underlying much of modern taxonomy.”

Evolutionary taxonomists often employ a holistic approach to taxonomy, considering a wide array of characteristics from fossil records, comparative anatomy, embryology, and increasingly, molecular data. This broad base of evidence helps to provide a more comprehensive understanding of the evolutionary pathways and connections among different species [193].

In recent years, the advent of molecular genomics has transformed the landscape of taxonomy. DNA sequencing has provided a new and powerful tool for determining evolutionary relationships with greater precision. As a result, modern evolutionary taxonomy increasingly relies on DNA sequencing data, leading to more robust and accurate classifications. DNA sequencing, allows taxonomists to compare genetic material directly, providing a quantitative method to measure differences and similarities at the molecular level. This approach is particularly useful in cases where morphological characteristics are convergent or exhibit plasticity (ability to adapt to different conditions), leading to potential misclassification [255].

One prominent example of the use of molecular genomics in taxonomy is the use of *mitochondrial DNA (mtDNA)* sequencing. Because of its high mutation rate and maternal inheritance, mtDNA is ideal for studying evolutionary relationships among closely related species. The Barcode of Life project [115] utilizes a short genetic marker in the mitochondrial DNA intending to develop a comprehensive and universal system for species identification. Specifically, it uses a portion of the cytochrome c oxidase I (COI) gene, called a DNA barcode, to identify species. The barcode region of the COI gene typically ranges between 648 to 658 base pairs in most animals. Another example of incorporating molecular genomics in taxonomy is the use of ribosomal RNA (rRNA) gene sequencing, which has been fundamental in reconstructing the phylogenies of life. The 16S ribosomal RNA gene, used primarily for classifying prokaryotes (bacteria and archaea) [138, 133], and the 18S rRNA gene, used for eukaryotes [177, 36], are conserved enough to maintain basic genetic functionality but variable enough to provide evidence of speciation over time.

Molecular techniques have led to a revised understanding of numerous organisms and a deeper understanding of the tree of life [135, 192, 132]. The integration of molecular genetics into taxonomy is critical for fields such as conservation biology and ecology where

precise taxonomic resolutions can have significant practical implications. Furthermore, in medicine, understanding the precise taxonomy of pathogens enhances disease diagnosis and treatment strategies, ultimately influencing public health outcomes [100].

2.3 DNA Representations

To unravel the complex functions and mechanisms of biological systems, understanding the complex language of DNA is crucial. For the effective application of computational techniques to analyze and infer new biological hypotheses from DNA sequences, it is essential to transform the raw sequences of nucleotides into a format that computational models can effectively process (i.e., numerical values). This transformation is facilitated by *DNA representations*, which convert the sequence of nucleotides into numerical formats that retain essential biological information while making the data suitable for computational analysis. This process is particularly critical when analyzing extensive genomes, such as the human genome, which contains approximately 3 billion base pairs [214] across 23 pairs of chromosomes.

DNA representations translate DNA sequences into discrete numerical representations while attempting to capture the intrinsic properties of DNA sequences, including sequence patterns and structural similarities. These numerical representations enable models to perform exceptionally well in applications such as sequence classification, function prediction (inferring the biological role of genes or proteins), and evolutionary studies. DNA representation is a crucial step in applying advanced machine-learning techniques to genomic data, ultimately enhancing our comprehensive understanding of complex biological data [310]. In the following subsections, we will explore some of the most popular methods for DNA embeddings, each providing unique advantages for the processing and analysis of genomic sequences.

2.3.1 Ordinal Encoding

Ordinal encoding, also referred to as sequential encoding, converts DNA sequences into numerical values by assigning each nucleotide (A, T, G, C) a specific numerical value; for instance, A=0.25, T=0.5, G=0.75, and C=1 [310]. Several variations in numerical assignments for nucleotides have been introduced and explored in numerous studies [236, 58, 51]. This type of encoding is particularly useful for computational analyses, where a simple numerical representation of the DNA sequence is needed. However, it is important to note

that while ordinal encoding provides a straightforward numeric representation, it may not always capture the complex biological relationships between nucleotides. While the encoding assigns numerical values to nucleotides, some of these values are arbitrary, creating an artificial ordinal relationship—where the order of values suggests a hierarchy that may not reflect true biological significance—whereas others are grounded in the biological or chemical properties of the nucleotides [236]. One limitation of this encoding approach is that it fails to inherently reduce the dimensionality of the data, meaning it does not simplify the number of variables or features represented. This becomes particularly problematic for long sequences as it can lead to increased computational complexity and potential information overload. Additionally, since machine learning algorithms typically require inputs of uniform size, the variable length of encoded DNA sequences poses challenges in model training and analysis, necessitating techniques such as truncating or padding sequences with “0”s to achieve consistent vector lengths [26, 236].

2.3.2 One-hot Encoding

One-hot encoding is a straightforward yet powerful method to represent DNA sequences. In this method, each nucleotide is represented as a distinct binary vector of four bits with one of them being hot (i.e., 1) while others being 0 [51]. Typically A is represented as [0, 0, 0, 1], T as [0, 0, 1, 0], G as [0, 1, 0, 0], and C as [1, 0, 0, 0] in this DNA encoding [310]. Alternatively, a more efficient two-digit binary encoding can also be employed, using patterns such as 00, 01, 10, 11, which are frequently used in biocomputing. These one-hot encoded vectors can either be concatenated or turned into two-dimensional arrays. The simplicity of one-hot encoding makes it especially useful for initial machine learning models where the interpretability of input features is crucial.

While one-hot encoding is simple to implement and understand, it does not capture the sequential or contextual relationships between nucleotides, potentially limiting its effectiveness for more complex biological analyses. Each nucleotide is treated independently, without regard to its neighbors, which can be a significant drawback when sequence context plays a critical role in biological functions. Similarly to ordinal encoding, the issue of high dimensionality for longer DNA sequences and variable-length encoding persists with this method and can pose challenges in the application of machine learning techniques.

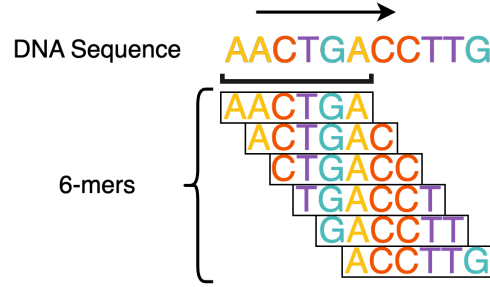


Figure 2.3: Visualization of the k -mer counting method for a DNA sequence. A sliding window of size $k = 6$ moves along a sequence of length $L = 11$, generating $11 - 6 + 1 = 6$ overlapping 6-mers. This figure shows each unique position that the k -mer window can occupy from the start to the end of the sequence.

2.3.3 Frequency of k -mers

A k -mer is a substring of length k in a DNA or RNA sequence. If we view DNA as the language of life, k -mers can be considered words of length k within this language. Analyzing the frequency of all possible overlapping k -mers of DNA sequences, in the form of k -mer frequency vectors, is a popular technique in bioinformatics used to analyze and compare the sequences. The process of creating the k -mer profile of a DNA sequence, as shown in Figure 2.3 involves sliding a window of size k along the sequence. In a sequence of length L , the total number of overlapping k -mers that can be generated by sliding the window across the sequence is $L - k + 1$, assuming $L \geq k$. Additionally, the number of possible different k -mer combinations in a DNA sequence (size of DNA “vocabulary”), which comprises four nucleotides, is 4^k . Counting the occurrences of each unique k -mer in input DNA sequences can be used as feature vectors.

One of the primary advantages of using a k -mer frequency vector as a DNA representation is its simplicity and the detailed insights it can provide into the sequence’s composition. This DNA embedding converts DNA sequences of different lengths into a uniform length (4^k) numerical vector, making them suitable as input for machine learning algorithms such as classification and clustering of DNA sequences. The use of small values of k , typically ranging from 3 to 6, is particularly advantageous as it limits the representation space to manageable levels, avoiding the exponential growth observed in larger 4^k spaces. This approach allows for a detailed yet computationally efficient analysis of frequent short subsequences, facilitating faster processing and simpler pattern recognition in genomic data. While there is no universally “optimal” k value for k -mer analysis, the sequence length significantly influences the optimal k value. The relationship $k = \log_4(|s|)$, where $|s|$ denotes

the DNA sequence length, provides a theoretical minimum for k [25, 64]. This k value represents the k -mer size at which, assuming a uniform distribution of k -mers, each possible k -mer would appear exactly once. Notably, in random sequences, the entropy of k -mer frequencies is maximized at this k -mer size [64]. The choice of k necessitates lower values for shorter genomic sequences, such as mitochondrial DNA and viral genes/genomes, where $k = [6, 9]$ is often adequate [201, 202, 10]. Conversely, higher k values (e.g. $k \geq 20$) are required for larger nuclear genome fragments [266, 112]. Furthermore, the selection of k may vary depending on the specific taxonomic level under study, with larger values generally more suitable for analyzing relationships between closely related species [64]. The k -mer frequency vectors are used in several applications, such as genome assembly [272, 42, 208], metagenomic studies [72, 242, 306, 30], phylogenetic analysis [304, 92, 292] and many others [59, 140, 62]. They are particularly valuable in identifying *genomic signatures* that predict evolutionary relationships. Genomic signatures, defined by Karlin and Burge [147] are numerical quantities that can distinguish closely from distantly related organisms based on DNA sequence identity.

The choice of k greatly impacts the resolution and effectiveness of the biological analysis. Smaller k values tend to capture more general features of the sequence and require less computational power. In contrast, larger k values can capture more specific and detailed features but exponentially increase the size of the frequency vector and the computational resources needed, especially for large genomic datasets. Choosing the optimal value of k depends on the specific characteristics of the DNA sequences being analyzed, as well as the objectives and limitations of the research [118]. It requires a customized approach to ensure the analysis is both accurate and efficient.

Given the computational challenges associated with large k -mer vectors, approximating the k -mer frequency becomes a practical necessity in many studies [188, 196]. Approximation methods, such as probabilistic counting [321] or subsampling techniques [253], provide a manageable way to estimate k -mer distributions without fully enumerating all possible k -mers. Approximation methods for k -mer counting can be significantly faster and require substantially less memory than exact k -mer counting algorithms. However, a notable drawback of approximation over exact k -mer frequency counting is the potential for reduced accuracy, as these methods can introduce sampling biases and error margins that might affect the quality of the genomic analysis.

2.3.4 Chaos Game Representation

Introduced by Barnsley in 1988, the *Chaos Game* algorithm generates fractals by iteratively plotting points within a polygon starting from a randomly chosen initial position [21].

Its extension, *Chaos Game Representation (CGR)*, proposed by Jeffery in 1990, is an innovative method of visualizing biological sequences (e.g., DNA/RNA) by mapping them into a fractal space [136]. CGR represents the sequence data in a compact, visual format that displays the arrangement patterns of the nucleotides. As illustrated in Figure 2.4a, a *CGR square* $\mathcal{G}$, is a square with vertices (corners) in the set $V = \{(1, 1), (1, 0), (0, 0), (0, 1)\}$. The corners of the CGR square are labelled as follows: the bottom left corner is labelled by A , the top left corner is labelled by C , the top right corner is labelled by G , and the bottom right corner labelled by T . Formally, the labelling function $l : \Delta \rightarrow V$ is defined as $l(A) = (0, 0)$, $l(C) = (0, 1)$, $l(G) = (1, 1)$ and $l(T) = (1, 0)$, where Δ is the set $\{A, C, G, T\}$ corresponding to the four different nucleotides.

A CGR X_s of a sequence $s \in \Delta^n$ of length n , is the set of points $X_s = \{p_0, p_1, \dots, p_n\} \in \mathbb{R}^2$, situated inside the CGR square, whose coordinates are defined recursively by $p_0 = (x_0, y_0) = (0, 0)$ and, for all $1 \leq i \leq n$,

$$p_i = \frac{p_{i-1} + l(s_i)}{2} \quad (2.1)$$

Informally, the plotting begins from the center of the CGR square. The first nucleotide in the sequence is plotted halfway between the center and the vertex corresponding to that nucleotide. Each subsequent nucleotide is plotted halfway between the previous point and the vertex representing the current nucleotide. This iterative process generates a pattern of points within the square, reflecting the order and frequency of nucleotides in the DNA sequence. For example if $s = ATGC$, then $X_s = \{p_0, p_1, p_2, p_3, p_4\}$ is the set of points shown in Figure 2.4b. The points plotted in the CGR represent nucleotide occurrences in the sequence and the last plotted point of a CGR could in theory (at infinite resolution) be used to recover the original DNA sequence [97, 39, 11].

Frequency CGR (FCGR) of resolution k , a quantified variant of CGR, is a $2^k \times 2^k$ grayscale image in which the intensity of each pixel directly corresponds to the frequency of its corresponding k -mer. FCGR is a compressed representation of DNA sequences with the compression degree indicated by the resolution k . It is obtained by dividing the CGR image into smaller, equally-sized squares and counting the number of points (or the frequency) within each square. Since FCGR smoothes the data in a grayscale image, rather than plotting each point individually, it is inherently less noisy than the traditional CGR. Similar to CGR, FCGR can identify over- and under-representation of k -mers in DNA sequences and, thereby, can be used to determine the degree of identity between the DNA sequences of different species [35].

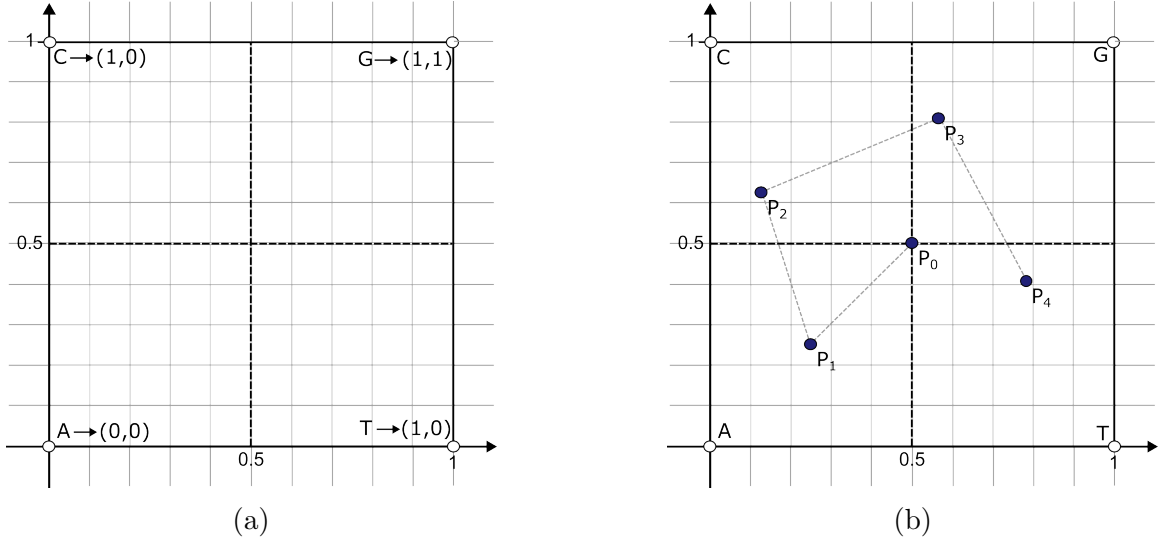


Figure 2.4: **(a)**: The CGR square $\mathcal{G}$. The vertices indicate the labelling of the corners of the CGR square. **(b)**: The CGR image X_{ACGT} , of the sequence $s = ACGT$, consists of the points p_0, p_1, p_2, p_3, p_4 illustrated in the figure.

Formally, FCGR of a sequence $s \in \Delta^n$ with resolution k with $n \geq k$, is a matrix in $\mathbb{R}^{2^k \times 2^k}$ derived from X_s , the CGR image of s . Its (i, j) th entry f_{ij} is defined as:

$$f_{ij} = \text{Number of points of } X_s \text{ in cell } (i, j) \quad (2.2)$$

where cell (i, j) is the (i, j) th subsquare, starting from the bottom left, of the CGR square if we subdivide the square into $2^k \times 2^k$ equal size subsquares.

It is important to note that each $2^k \times 2^k$ cell (i, j) in the FCGR corresponds to one of the 4^k k -mers in the k -mer frequency vector. Specifically, the frequency f_{ij} of pixels in the CGR image X_s of sequence s falling into cell (i, j) equals the frequency of a specific k -mer in the sequence s . As a result, the k -mer frequency vector can be seen as a flattened version of the FCGR for a specific DNA sequence. As with k -mer frequency vectors, increasing k in the FCGR enhances resolution, leading to a more detailed representation at the expense of exponential growth in data complexity; more information is captured with higher values of k , but this also increases the computational overhead. Figure 2.5 illustrates some examples of FCGRs at a resolution of $k = 8$ (selected for visualization purposes) of real genomic DNA sequences, displayed alongside FCGRs of computer-generated DNA sequences.

Similar to k -mer frequency vectors, CGRs/FCGRs of DNA sequences can be considered as a genomic signature [147]. Consequently, the distance between CGRs/FCGRs of DNA

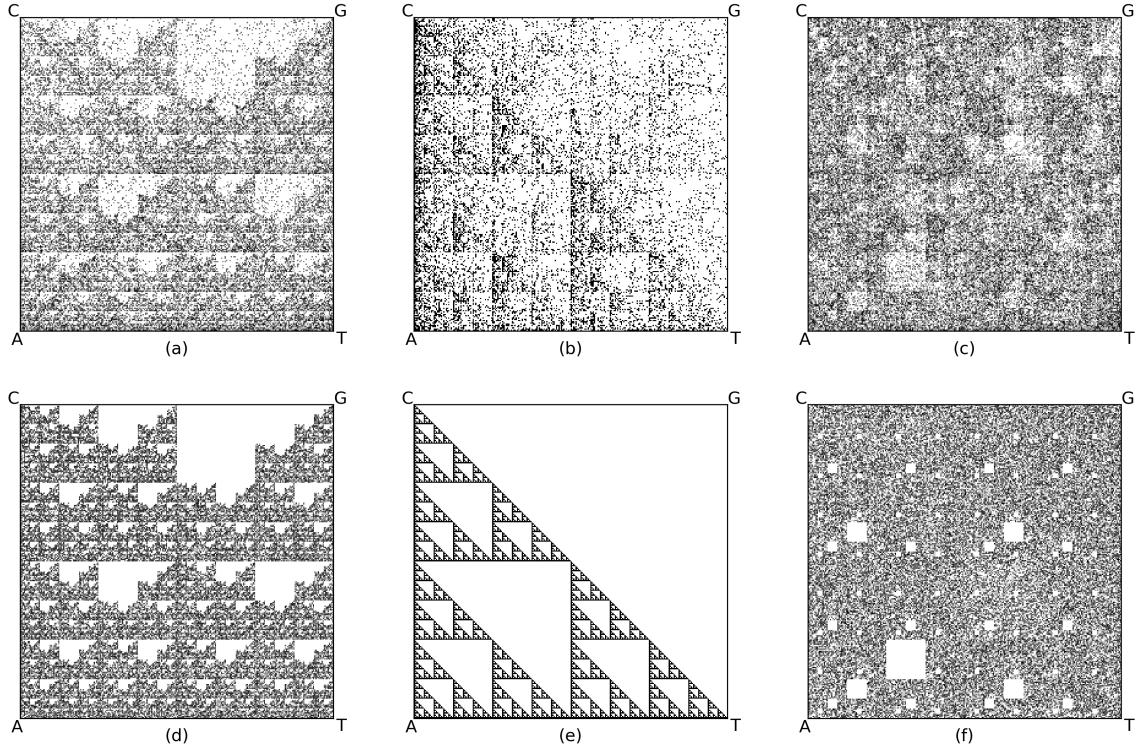


Figure 2.5: Frequency Chaos Game Representation (FCGR) at resolution $k = 8$ (for visualization purposes) of **(a)** human beta globin region on chromosome 11 of length 73,308 bp (Accession ID: U01317.1); **(b)** complete genome of *Homo sapiens* isolate LI-T1 mitochondrion of length 16,566 bp (Accession ID: KX228192.1); **(c)** *Escherichia coli* plasmid of JE86-ST05 DNA with length 114,953 (Accession ID: AP022816.1); Computer-generated “random” DNA sequences of length 100,000 bp avoiding substrings: **(d)** “CG”, **(e)** “G”, **(f)** “CTA.”

sequences acts as a metric for similarity, enabling comparisons and the construction of phylogenies that elucidate evolutionary relationships among organisms [333, 184]. FCGR visualizations are particularly valuable in genome comparison and classification [247, 283, 105], phylogenetic analysis [11, 225, 184, 264], and genomic functionality analysis [329].

2.3.5 Subword Tokenization Methods

Subword tokenization is a technique in Natural Language Processing (NLP) that involves dividing words into smaller units called subwords, or tokens. This approach is employed in NLP tasks that require managing large vocabularies and complex word structures. The underlying principle in subword tokenization techniques is that commonly occurring words should be included in the vocabulary, while less frequent words are broken down into more commonly appearing subwords. It is important to note that, these tokenization algorithms are data-driven and do not necessarily correspond to linguistically meaningful units.

Due to the structural parallels between linguistic texts and genetic sequences, subword tokenization methods have been recently applied in the field of genomics [330, 71, 322, 86]. In genomics, the vast and complex “vocabulary” of genetic information, composed of sequences of nucleotides (A, C, G, T), mirrors the challenges found in natural language processing. As subword tokenization methods break down rare or complex words into frequently occurring subwords, in genomics, it can decompose uncommon genetic sequences into more common subsequences. This allows computational models to manage large amounts of genetic data efficiently and enhances their capability to generalize from known sequences to novel or mutated sequences they have not been explicitly trained on. Moreover, this approach supports the efficient handling of sequence alignment and comparison, which is a crucial step in various genomic analyses, such as identifying evolutionary relationships or genetic variations. One key advantage of subword tokenization over methods like k -mer frequency vectors and CGR is that it is not confined to a fixed resolution k . Therefore, with significant potential for advancements, subword tokenization serves as a bridge between established NLP techniques and genomic research.

DNABERT-2 [330] was the first study to adapt the subword tokenization framework SentencePiece [159] for DNA sequences, setting the groundwork for training a DNA foundation model—a comprehensive, pre-trained model serving as a base for various genomic analysis tasks. The authors also employed Byte-Pair Encoding (BPE) [265], which is a subword tokenization method that starts with the smallest units of text (nucleotides in this case) and iteratively combines the most frequently occurring pairs to form larger tokens. This process continues until a set vocabulary size is reached, effectively capturing common patterns and reducing the overall complexity of the input data. The BPE algorithm

was also used in GENA-LM [86] and DeepGene [322], as the initial step of the proposed pipelines for a spectrum of DNA analysis tasks. Additionally, Dotan et al. (2024) compared three NLP subword tokenization methods: BPE, WordPiece [262], and Unigram [158], in different genomic and proteomic tasks, revealing that no single method was universally optimal, highlighting the importance of selecting appropriate tokenizers for specific biological tasks [71].

2.3.6 Additional Methods for DNA Representation

While this section has focused on specific DNA representation methods such as ordinal encoding, one-hot encoding, frequency of k -mers, chaos game representation, and subword tokenization, several other approaches also play significant roles in genomic analyses. This includes methods like the Voss binary representation, which simplifies DNA sequences into four binary indicator sequences each representing one nucleotide (A, C, G, or T). In this DNA representation, a “1” indicates the presence and a “0” the absence of a nucleotide at a specific position within the sequence [296, 162]. Another method of DNA representation is the nearest dissimilar nucleotide approach, where a DNA sequence is expressed as four numerical sequences representing the nearest dissimilar (ND) distances [3]. The ND distance is related to the repetition structure of single nucleotides and measures the number of nucleotides between a given nucleotide and the next occurrence of a different nucleotide in the sequence.

Additional methods for representing DNA sequences in two dimensions include the 2D rectangular walk, where each nucleotide maps to a direction and unit distance on a 2D grid [93]. Other 2D DNA representation methods are detailed in [209, 175, 325]. These graphical approaches offer insights into the local and global characteristics of sequences, providing unique perspectives not as readily accessible through other methods [210].

2.4 Sequence Alignment and Comparison Techniques

Sequence alignment and comparison are one of the fundamental procedures in bioinformatics, having the purpose of identifying regions of similarity that may point to functional, structural, or evolutionary relationships among biological sequences (DNA, RNA, protein) [249]. These techniques are crucial for various applications, such as phylogenetic analysis, gene identification, drug discovery, and comparative genomics. In sequence alignment, biological sequences are organized to optimize alignments by maximizing matches

and minimizing mismatches and *indels* (insertion or deletion of one or more nucleotides). This alignment process utilizes distances based on a predefined scoring system as a measure of dissimilarity between sequences, which can serve as an estimate of their evolutionary relationship. Pairwise alignment is the process of aligning two sequences, whereas multiple sequence alignment involves the simultaneous alignment of three or more sequences. Global alignment and local alignment are two frequently used pairwise sequence alignment algorithms which we will discuss in more detail in the following subsection.

2.4.1 Global Alignment

Global alignment is a computational method that aligns the full lengths of two input sequences—DNA, RNA, or proteins—to produce an alignment that optimizes the overall similarity. This method systematically maximizes matches and minimizes mismatches and gaps, providing an output that includes the aligned sequences along with their similarity score. Global alignment is typically employed when the sequences under comparison are expected to exhibit similarity consistently along their entire lengths and are approximately equal in size. This method offers robust means for identifying evolutionary relationships, conserved regions, or functional similarities among sequences [254].

The Needleman-Wunsch algorithm [211] is a fundamental algorithm for global alignment, utilizing dynamic programming to evaluate different alignment options and assign scores accordingly, ultimately identifying the optimal alignment with the highest cumulative score as a measure of similarity. This algorithm starts by creating a scoring matrix whose dimensions are the lengths of the two sequences to be aligned. Each cell in this matrix holds a score that represents the best alignment achievable between the segments of the sequences up to that specific point. To populate the matrix, the algorithm considers three potential actions for each pair of nucleotides: aligning the nucleotides (resulting in a match or mismatch), inserting a gap in the first sequence, or inserting a gap in the second sequence. It assigns scores to each of these actions based on predefined rules: positive scores for matches and negative scores for mismatches and gaps.

The matrix is filled iteratively from the top-left to the bottom-right, with each cell's score calculated by comparing the total scores of the three possible preceding cells (directly above, left, and diagonally above-left) and selecting the highest score among them. The optimal alignment is determined by tracing back from the bottom-right cell to the top-left cell of the matrix, following the path that resulted in the highest score. This traceback reveals the alignment of the sequences that maximizes the overall similarity. The final alignment output includes the aligned sequences along with a similarity score, which quantifies their overall correspondence and potential evolutionary relationship.

In terms of computational requirements, the Needleman-Wunsch algorithm has a time complexity of $O(m \times n)$, where m and n are the lengths of the two sequences being aligned. This quadratic time complexity is because each cell in the $m \times n$ matrix must be calculated by considering the values of three adjacent cells (directly above, left, and diagonally above-left), making the process computationally intensive, especially for long sequences [280]. This characteristic highlights a potential challenge in applying the algorithm to very long sequences, leading to significant demands on computational resources. However, for shorter sequences or where maximum accuracy is required, the precision of the Needleman-Wunsch algorithm makes it particularly valuable.

2.4.2 Local Alignment

Local alignment is a computational method used to align segments of two sequences to identify regions of high similarity (“high-scoring segments”), regardless of the overall sequence similarity. This approach is especially useful when the sequences under comparison contain one or several significantly similar regions mixed with dissimilar segments, or when the sequences vary significantly in length. Local alignment is effective at uncovering evolutionary events such as functionally important motifs, regions of significance, and conserved functional units across different DNA sequences by focusing on the most highly matching sequence segments [254].

The Smith-Waterman algorithm [273] is a standard tool for performing local alignments, and it employs dynamic programming to compute the optimal local alignment with the highest scoring similarity. This algorithm initializes a scoring matrix with dimensions that correspond to the lengths of the two sequences to be aligned. Each cell in the matrix represents the highest score that can be obtained by aligning the sequences up to that point, starting anywhere within either sequence. As the matrix is populated, the algorithm considers three potential scenarios at each step: extending a current alignment with a match or mismatch, starting a new alignment (representing a local optimum), or introducing a gap in either sequence. Scores are assigned based on a set of predefined rules for matches, mismatches, and gaps (e.g., a match might be scored as +1, a mismatch as -1, and a gap as -2), ensuring that only the highest scoring local alignments are considered. The primary difference from the Needleman-Wunsch algorithm is that in the Smith-Waterman algorithm, any negative scores in the matrix are reset to zero. The traceback process begins at the matrix cell with the highest score and continues until a cell with a score of zero is reached, which determines the highest scoring local alignment [280].

By some optimization in the Smith-Waterman algorithm introduced in [99], this algorithm has a time complexity of $O(m \times n)$, similar to the Needleman-Wunsch algorithm,

where m and n are the lengths of the two sequences being aligned. Due to this quadratic time complexity, aligning very long sequences can be challenging as it requires significant computing resources. Despite these demands, the algorithm’s ability to pinpoint precise local similarities makes it a valuable tool for detailed and specific comparative studies of sequences, particularly when accuracy in identifying functional or evolutionary relationships is a priority.

2.4.3 Multiple Sequence Alignment

Multiple Sequence Alignment (MSA) involves the simultaneous alignment of three or more sequences. This technique builds on pairwise alignment by including additional sequences to reveal conserved regions useful for understanding evolutionary relationships among the sequences. MSA is particularly useful for phylogenetic analysis, comparing sequences from different species, or for identifying shared structural and functional motifs [270]. For a set of n sequences, the naive approach of MSA involves creating an n -dimensional matrix, analogous to the one used in standard pairwise sequence alignment. As n grows, the search space expands exponentially, also heavily influenced by the length of the sequences. In terms of computational complexity, this method requires $O(L^n)$ time to compute, where L is the length of the longest sequence. Finding the global optimum for n sequences using this method is considered an NP-complete problem [300]. Due to this constraint, exact solutions to MSA are typically possible only for small numbers of relatively short sequences.

Due to the NP-complete nature of the problem, many practical MSA tools use heuristic or approximation methods to find a solution that is good enough in a reasonable amount of time, rather than the absolute best possible alignment. These methods fall into two main categories: progressive and iterative methods. Progressive methods, such as ClustalW [49, 285] and T-Coffee [213], build alignments step-by-step by first aligning pairs of sequences and then adding others sequentially. On the other hand, iterative methods like MUSCLE [74] and MAFFT [149] continuously refine alignments by repeatedly aligning subsets of sequences and adjusting based on initial results. These algorithms generally aim to reduce the time complexity through various algorithmic strategies, but precise time complexities for these methods can vary significantly based on the specific algorithms and optimizations they employ [228].

2.4.4 BLAST (Basic Local Alignment Search Tool)

Basic Local Alignment Search Tool (BLAST) [13] is a widely used sequence alignment algorithm. BLAST is designed to enable rapid searching of sequence databases for sequences that resemble the query sequence above a certain threshold of similarity. This tool uses a heuristic method that emphasizes speed over blueguaranteed alignment optimality, making it possible to process large volumes of sequence data efficiently. BLAST identifies regions of local similarity between sequences and calculates the statistical significance of matches, uncovering functional and evolutionary relationships between sequences.

The time complexity of BLAST is significantly lower than that of comprehensive alignment algorithms such as the Smith-Waterman algorithm due to BLAST’s heuristic nature. BLAST achieves this by first identifying short matches between the query sequence and database sequences, which are called “hits”. These hits are then used as seeds to find longer segments of similarity, or high-scoring segment pairs (HSPs), without having to align every possible sequence combination comprehensively. Typically, BLAST can operate in time complexity approximating $O(nm)$, where n is the length of the query sequence and m is the total length of the database sequences [44]. This makes BLAST particularly suitable for searching large databases. In practical applications, BLAST is used extensively in molecular biology for tasks such as identifying species, gene discovery, phylogenetic analysis, and functional analysis [206]. Its ability to handle vast databases efficiently and its implementation of an intuitive output format that ranks matches by statistical significance has made it a valuable tool in many genomic studies.

However, BLAST does have some limitations due to its heuristic approach, which trades optimality for speed. Due to its focus on high-probability regions based on initial hits, it might miss alignments or significant matches that a more exhaustive alignment algorithm would capture [228]. Additionally, BLAST’s performance can be dependent on the choice of parameters such as the scoring matrix or the size of the word for initiating a query, which may not be optimal for all types of sequences or all biological queries. BLAST may have trouble identifying very distantly related sequences due to their low sequence similarity as it may miss subtle functional relationships between sequences [31]. Consequently, while BLAST is highly effective for many standard searches, it may not always provide the most comprehensive understanding of complex genetic relationships or detect less obvious homologies that require more sensitive or finely-tuned analysis tools.

2.4.5 Sequence Comparison Techniques Beyond Traditional Alignment

With the advancement of genomic research, it has become increasingly evident that traditional alignment methods that had been used for decades cannot fully capture the complexity of genomic data. Traditional techniques such as global, local, and multiple sequence alignments play crucial roles in identifying sequence similarities and differences across different biological sequences. Alternative approaches, known as *alignment-free* sequence analysis techniques, offer several benefits over traditional alignment-based methods, particularly for certain applications. These methods are significantly faster and require less memory, which makes them particularly useful when dealing with extensive next-generation sequencing datasets [276]. This efficiency is primarily due to bypassing the computationally intensive dynamic programming required for traditional alignment.

Two primary categories of alignment-free methods are word-based methods (also known as k -mer based methods) and information theory-based methods [333]. Word-based techniques employ mathematical models to measure the frequency of short subsequences (k -mers) and calculate distances directly from these frequencies, thus obviating the need for sequence alignment. A key assumption here is that similar sequences will exhibit similar k -mer frequency profiles, and the difference between the k -mer frequency profiles of sequences can serve as a metric for assessing sequence dissimilarity [295, 304, 140]. On the other hand, information theory-based methods evaluate the shared information between two biological sequences using principles such as complexity and entropy [294]. For instance, Kolmogorov complexity can be approximated using compression algorithms [171], while Shannon entropy measures the unpredictability of encountering specific elements within a sequence [54]. Beyond these main groups, other alignment-free approaches include methods based on the length of common substrings [289, 168, 311], graphical representation methods [11, 247, 35], and signal processing techniques [236, 26]. For an in-depth examination of alignment-free methods, including a comprehensive comparison between various alignment-free algorithms and their applications in genomics, the reader is referred to [332, 243].

In general, alignment-free approaches demonstrate greater robustness against sequence rearrangements and horizontal gene transfers—issues that often challenge alignment-based methods [333]. They are also adept at analyzing sequences with low identity levels, a scenario where traditional methods frequently struggle [96]. Crucially, alignment-free methods do not depend on evolutionary model assumptions, such as substitution matrices or gap penalties, avoiding potential biases introduced by these models [23]. Their application spans a wide range of scenarios, from rapid screening of sequence databases to detailed

analysis of metagenomic data. While traditional alignment-based methods continue to be essential for detailed residue-level comparisons, alignment-free approaches offer a viable alternative for managing large-scale data, navigating complex evolutionary scenarios [269], and exploring relationships among sequences with minimal similarity.

2.5 Supervised Machine Learning

Supervised machine learning, commonly referred to as supervised learning, is a fundamental type of machine learning where models are trained on labelled datasets. This method involves learning a function $f : X \rightarrow Y$ from provided training examples, where each example is a pair consisting of an input object $x \in X$ and a desired output value $y \in Y$. By learning patterns from the training data, the model is able to predict the output associated with new inputs.

The process begins with the splitting of the dataset into training and testing sets. The model learns by adjusting its parameters to reduce the difference between its predictions and the actual outputs on the training set. This adjustment is often achieved through an optimization process, typically using a method such as gradient descent. The objective is to minimize a loss function L , which quantifies the prediction errors made by the model:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \text{loss}(y_i, f(x_i; \theta))$$

where, θ represents the model parameters, n is the number of samples in training set, y_i is the true output (ground truth), x_i is the input for sample i , and loss is a function measuring the error in prediction. Some of the most commonly used loss functions in supervised learning are mean squared error (MSE), cross-entropy loss, and hinge loss [301].

There are two main types of supervised machine learning tasks: classification and regression, each serving a distinct purpose and utilizing different methodologies. Classification tasks are used for predicting discrete labels, such as the taxonomic classification of a DNA sequence [203, 275]. On the other hand, regression tasks involve predicting continuous values, such as predicting gene expression levels based on different genetic and environmental factors [17, 180]. Given the primary focus on genomics classification problems within this document, the subsequent subsection will explore some common algorithms applied to supervised classification.

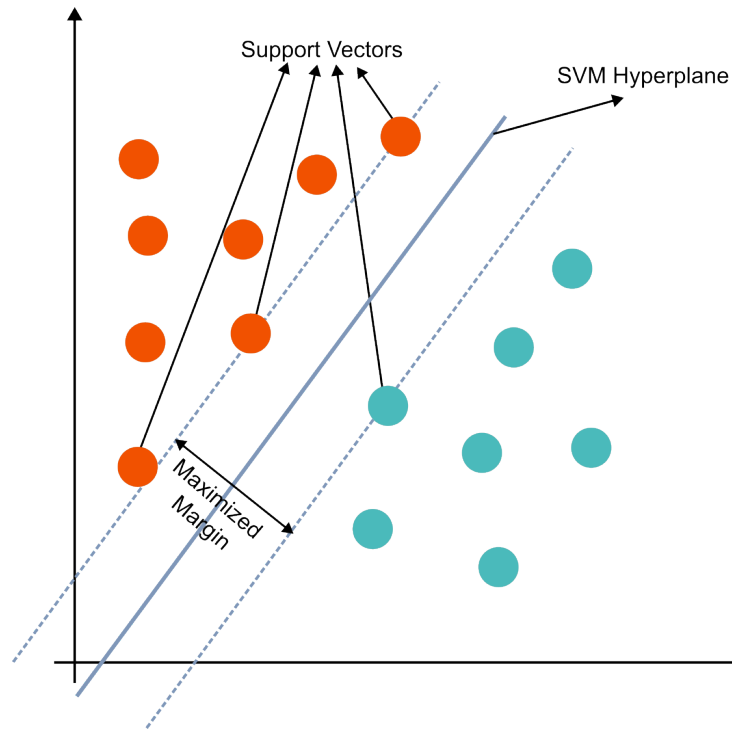


Figure 2.6: Illustration of a linear SVM classifier separating two classes with maximized margin. The support vectors that determine the hyperplane’s position are highlighted.

2.5.1 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a class of supervised learning methods used for classification, regression, and outlier detection tasks [55]. The main goal of an SVM model is to find the best line or plane (called a hyperplane) that separates data into classes with the greatest possible margin, which means the space between the closest data points of different classes is maximized. This clear separation helps with making accurate predictions. The most important data points in determining this separation are known as support vectors. Support Vectors are the closest points to the hyperplane and have a direct influence on its position and orientation. Figure 2.6 shows the effective separation of the two classes by establishing a hyperplane that maximizes the margin between the support vectors.

The process of separating data with a hyperplane works well when the data is linearly separable, meaning it can be divided by a straight line. However, when data is more complex and cannot be separated by a straight line, SVM uses a technique known as the kernel trick [260]. This technique transforms the original data into a higher-dimensional

space where a hyperplane can effectively separate the classes, making SVM a powerful tool for handling various types of data. A kernel function is a computational tool that enables this transformation by implicitly mapping data to a higher-dimensional space without directly computing the coordinates in that space, facilitating the separation of data points that are not linearly separable in the original space. Different types of kernel functions, such as linear, polynomial, radial basis function (RBF), or sigmoid, are employed based on the characteristics of the dataset and the specific requirements of the problem [259].

In genomics, SVMs have found extensive application across various classification tasks, such as classifying types of cancer based on gene expression profiles [119, 125] or categorizing DNA sequences by biological functions [124, 233]. SVMs can effectively manage non-linear relationships in genomic data using various kernel functions and are particularly well-suited for high-dimensional data with small sample sizes, which are common in genomic datasets [326]. Despite their accuracy and robustness in high-dimensional spaces, SVMs have certain computational demands such as intensive requirements for hyperparameter tuning and kernel selection, necessitating careful preprocessing of data and feature scaling to optimize performance [125].

2.5.2 Artificial Neural Networks

Artificial neural networks are a type of machine learning model inspired by the biological neural networks found in brains and complex nervous systems. They are primarily employed for tasks such as classification, regression, and pattern recognition. A neural network consists of interconnected layers of nodes, known as *neurons*, which process data in a manner similar to the neural networks of the brain. Each neuron computes its output y using the formula:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right) \quad (2.3)$$

where x_i represents the input values, w_i denotes the weights assigned to these inputs, b is the bias, f is an *activation function* that introduces non-linear properties to the model, and n represents the number of inputs to the neuron.

As shown in Figure 2.7, the architecture of a neural network typically includes an *input layer*, one or more *hidden layers*, and an *output layer*. Neurons in each layer receive inputs, which are the outputs from the preceding layer, compute a weighted sum followed by a non-linear activation function, as described in Equation 2.3 and forward the output to the

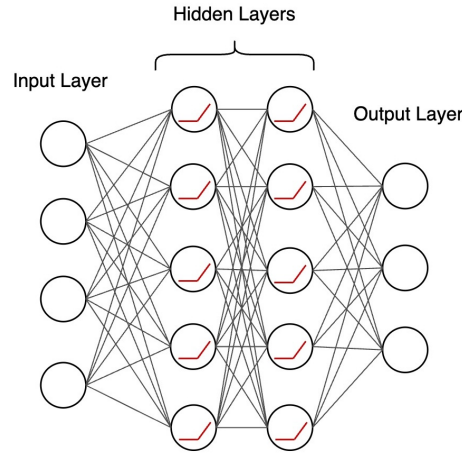


Figure 2.7: Architecture of a neural network, showing the input layer, two hidden layers, and the output layer. Each circle represents a neuron, which processes input data using a weighted sum and a non-linear activation function, passing the results to the next layer. A neuron receives input data, which it multiplies by specific weights—parameters that adjust during training to optimize the network’s performance. These weighted inputs are then summed together and passed through an activation function, a non-linear transformation that introduces non-linearity to the model, enabling it to learn and perform more complex tasks.

subsequent layer. The learning process involves adjusting the weights and biases of the neurons (w_i and b) to minimize the error between the predicted output y and the actual target values (known as *ground truth*) by optimizing a *loss function*. This adjustment is achieved through *backpropagation*, where gradients of a loss function are calculated and used to update the weights, thereby reducing loss in subsequent iterations [4].

Deep Neural Networks (DNNs), an advanced version of basic neural networks, feature multiple hidden layers (anywhere from two to hundreds of hidden layers depending on the specific application and complexity of the tasks the network is designed to perform) that add levels of abstraction and complexity to the model’s learning capabilities [98]. By capturing complex patterns in large datasets, DNNs frequently surpass other machine learning models in tasks requiring detailed feature recognition, such as image classification, natural language processing, and speech recognition. Their extensive depth and learning capabilities make them particularly suitable for processing and analyzing large amounts of complex data.

In genomics, neural networks have been used to tackle many types of problems, such as

predicting gene function [17, 78, 258], classifying different cell types [317, 164], and analyzing epigenetic data such as DNA methylation or histone modifications to uncover patterns associated with gene regulation [212, 328]. As a result of deep learning advancements, these models are capable of capturing subtle patterns in genomic sequences that might be missed by simpler, linear models.

Despite their versatility, neural networks also come with challenges. They require large amounts of data and computational resources for training, and their ‘black box’ nature makes it difficult to interpret exactly how decisions are made, which is a critical drawback where understanding the decision-making process is crucial [189].

Neural networks are constantly evolving, introducing new architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), Autoencoders, and Generative Adversarial Networks (GANs). Each of these architectures is tailored to efficiently manage particular data types and tasks, allowing them to be applied to a variety of real-world problems [257].

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of neural network that are particularly effective for processing data with a grid-topology, such as images. In order to capture spatial hierarchies and patterns such as edges in an image, CNNs employ a mathematical operation known as convolution, which involves applying learnable filters (kernels) to the input image. These filters, small matrices of weights, are trained to detect specific features within the image, such as edges, textures, or shapes. As the filter slides over the image, a process known as striding, it applies a dot product between the filter and the local region of the input, producing a feature map. This feature map represents the presence and intensity of the detected feature at different locations across the input image, effectively identifying how different objects and elements are arranged. The CNN’s ability to capture the arrangement of objects, makes it extremely efficient for tasks that require recognition and classification of objects within images. In addition to visual applications, this characteristic is leveraged in any data that can be represented as a grid, such as audio spectrograms [117, 320] and chaos game representation of genomic data [247, 252, 16]

As illustrated in Figure 2.8, the architecture of a CNN typically consists of one or several layers that transform the input through convolutions, non-linear activation functions (e.g. ReLU), and pooling operations (e.g. max pooling). Each convolutional layer applies multiple filters to the input and passes the result to the next layer. This process allows the network to build a complex understanding of the input image as information flows

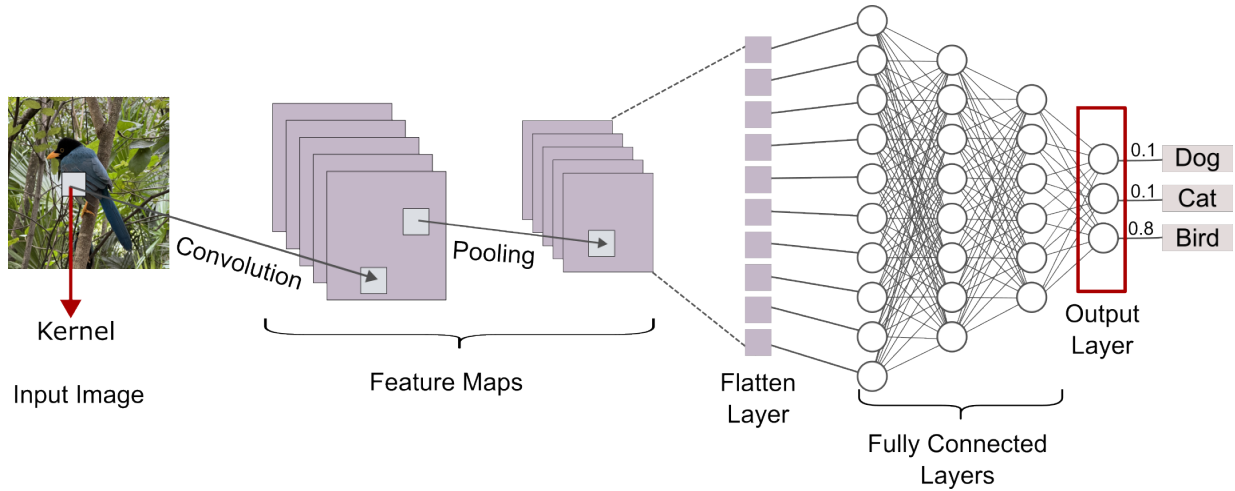


Figure 2.8: Architecture of a convolutional neural network (CNN) for classifying images. Starting from the input image of a bird, the network applies convolutional layers to extract key visual features. The convolutional layer utilizes a kernel (or filter) to scan over the image, applying mathematical operations at each position to extract fundamental features such as edges and textures. These features are reduced in dimension through pooling layers, then flattened into a vector form suitable for processing by fully connected layers. The final classification output is determined by the output layer, which here predicts the input as a “Bird” with a probability of 0.8, compared to lower probabilities for “Dog” and “Cat”. This visualization highlights the sequential flow and decision-making process within a CNN.

through the architecture. After the convolutional and pooling layers, the extracted features are flattened and passed through one or more fully connected layers for classification or regression tasks [293, 166].

The training process of CNNs involves backpropagation and gradient descent. However, training CNNs from scratch requires substantial computational resources and a large amount of labelled data, which might not be feasible for all applications. To address this challenge, pre-trained CNN architectures, such as VGG [271], ResNet [114], and Inception [281], offer a practical solution. These models have been previously trained on large datasets such as ImageNet [66], encompassing millions of images across thousands of categories. By using these pre-trained models, one can leverage the learned features and apply them to new tasks with relatively small datasets through a process known as *fine-tuning*. In fine-tuning, the pre-trained network is slightly adjusted and retrained to adapt its understanding to specific, perhaps more narrow, tasks. This approach, known

as *transfer learning* significantly reduces the resources and time required for training while still maintaining high accuracy and performance.

2.6 Unsupervised Machine Learning

Unsupervised machine learning, commonly referred to as unsupervised learning, is a type of machine learning where models are developed to identify patterns and relationships in datasets without any explicitly provided ground truth labels. Unlike supervised learning, unsupervised learning does not use output labels or correct answers during training. Instead, it focuses on discovering the underlying structure of the data, which makes it ideal for exploratory data analysis, clustering, anomaly detection, and dimensionality reduction.

One of the primary methods in unsupervised learning is clustering, where the goal is to group a set of objects such that objects in the same group (or cluster) are more similar to each other than to those in other groups. Traditional clustering methods such as *K*-Means [187], Gaussian Mixture Models (GMMs) [246, 5], hierarchical clustering [5], and DBSCAN [79] are effective in grouping data based on straightforward and predefined metrics (e.g. distance or density). However, these clustering methods can struggle with complex patterns and high-dimensional data, where the relationships between data points are not easily defined by simple metrics.

On the other hand, deep clustering leverages deep learning models to perform clustering tasks, which can be more effective in identifying complex patterns and processing large volumes of high-dimensional data. These deep learning models can learn non-linear transformations and features that traditional methods cannot, providing a more sophisticated understanding and effective grouping of the data. Unlike supervised neural networks that use loss functions such as cross-entropy to minimize the error between the predicted and actual labels [98], deep clustering utilizes loss functions that encourage the discovery of data structures without any labels. These functions may include variations of contrastive loss [103], which aim to minimize the distances between embeddings of similar items while maximizing distances between those of dissimilar items. In this context, an “embedding” refers to a vector representation of an item in a high-dimensional space, where similar items are positioned closer together, and dissimilar items are further apart. Alternatively, some functions utilize autoencoder-based losses [217, 284], which focus on minimizing the reconstruction error. This error measures the difference between the original input and its reconstruction produced by the autoencoder, reflecting the model’s ability to capture and recreate the essential features of the input data. This allows deep clustering to operate effectively in scenarios where no labels are available or are expensive to obtain.

2.6.1 K -Means

K -Means clustering [187] is a widely used method in unsupervised learning, valued for its straightforward methodology and effectiveness in grouping unlabelled data into predefined K clusters based on similarity measures. The algorithm initiates by selecting initial centroids randomly from the dataset. These centroids, which represent the centers of the future clusters, play a critical role in the clustering outcome, as their initial positions can significantly influence the final results. Next, each data point is assigned to the nearest centroid, typically calculated using Euclidean distance. This step organizes the data into clusters, each grouped around a central point, thereby minimizing the variance within each cluster. Following the assignment, the centroids are recalculated to represent the mean location of all points within the cluster. This process repeats iteratively until the centroids stabilize and no longer relocate significantly. As a result of this convergence, the clusters appear to be as refined as they can be under the given configuration.

While K -Means is highly effective, it does present several challenges. The algorithm requires the number of clusters K to be specified a priori, which may not be straightforward without intrinsic knowledge of the data structure. It is also sensitive to the initial placement of centroids, with potential convergence to suboptimal solutions. Moreover, K -Means assumes that clusters are spherical and of similar size, which may not align with real-world data distributions [5].

To address some of these limitations, variations such as K -Means++ [15] have been developed. K -Means++ improves the initialization phase of the standard K -Means algorithm by carefully choosing the initial centroids with a specific probability distribution, which helps to spread them out within the dataset. This method significantly reduces the chances of falling into suboptimal solutions and often leads to better clustering results. The improved initialization process of K -Means++ makes it more robust and efficient, particularly in avoiding the problems of random initial centroid selection.

In practice, K -Means is employed across many scientific disciplines, including genomics, where it effectively identifies distinct clusters within large datasets. In genomics, K -Means clustering is utilized for tasks such as gene expression analysis [251, 185], classification of different cell types based on their genetic profiles [307, 47], and alignment-free DNA clustering and disease subtype discovery [19, 37, 127, 8]. Despite its challenges, K -Means clustering offers valuable insights that lead to significant advances in genomics, revealing complex genetic patterns and relationships efficiently.

2.6.2 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) [246, 5] is a more sophisticated clustering approach, which assumes that the data is generated from a mixture of several Gaussian distributions, each with its own mean and covariance. Instead of assigning each data point to a single cluster definitively, GMMs compute the probability of each point belonging to each Gaussian distribution. This soft clustering approach allows GMMs to handle overlapping clusters and model complex cluster shapes, making them more flexible compared to the rigid spherical clusters assumed by K -Means. The Expectation-Maximization (EM) algorithm is usually used to estimate the GMM parameters. This iterative process begins with an initial guess of the parameters and alternates between two steps: the Expectation (E) step, which updates the probabilities of cluster assignments based on current parameter estimates, and the Maximization (M) step, which recalculates the parameters to maximize the likelihood of the data given these new assignments.

One of the primary advantages of GMMs over K -Means is their ability to model clusters of different shapes and sizes due to the flexibility in covariance structures. This property allows them to perform well on a diverse set of data distributions that are not well-suited to the assumptions of K -Means. However, GMMs also present challenges, including sensitivity to initial parameter estimates and the potential for convergence to local optima. Moreover, similar to K -Means, the number of clusters still needs to be predefined in GMM. In genomics, GMMs have been applied successfully in several applications, such as the analysis of gene expression data [83, 181], predicting genomic protein-coding regions [6], and population stratification estimation from genomics data [34]. Despite the complexity and computational demands, the robustness of GMMs in handling the complex structure of high-dimensional data makes them a valuable statistical clustering tool for bioinformatics.

2.6.3 Hierarchical Clustering

Hierarchical clustering is a powerful unsupervised learning method that builds a hierarchy of clusters. Unlike partitioning methods such as K -Means or GMMs, hierarchical clustering does not require the number of clusters to be predefined, making it suitable for exploratory data analysis [143, 5]. This clustering method can be categorized into two types: agglomerative (bottom-up) and divisive (top-down). Agglomerative clustering starts with each data point as a separate cluster and merges them step by step into larger clusters, based on a distance matrix and a linkage criterion such as single linkage, complete linkage, or average linkage. The process continues until all points are merged into a single cluster, or a termination condition is met. On the other hand, divisive clustering begins with all

data points in a single cluster that is iteratively split into smaller ones. The split in divisive clustering is determined by analyzing the data structure and performing a split of the cluster based on a specified criterion, such as the longest distance between points or the highest between-cluster dissimilarity.

One of the major advantages of hierarchical clustering is the ability to visualize the clustering process through a dendrogram, a tree-like diagram that shows the arrangement of the clusters produced during the hierarchical clustering. This visualization helps with understanding the data structure at different levels of granularity and is particularly useful in genomics for understanding genetic relationships and variations [61]. However, hierarchical clustering can be computationally intensive, especially for large datasets, due to the need to calculate and update the distance matrix as clusters are merged or divided. The choice of linkage criteria also significantly affects the results, and inappropriate choices can lead to clusters that do not reflect the underlying data structure [151]. In bioinformatics, hierarchical clustering has been effectively applied in numerous studies, such as analyzing gene expression data [316, 73, 130], grouping proteins based on similarity [305, 231], and classifying diseases based on genetic profiles [232].

2.6.4 Contrastive Clustering

Contrastive learning is a powerful technique that derives its value from the ability to learn informative representations by comparing how similar or different pairs of examples are, rather than relying solely on raw data or labelled examples [129]. This method helps models understand the underlying structures of the data by pulling similar instances closer and pushing dissimilar ones farther apart in the representation space. A *positive pair* consists of an original sequence and an “augmented” version of that sequence, which is generated so as to be similar to the original or related to it in a meaningful way. In this context, a *negative pair* is any other pair of sequences in the dataset. This method has gained significant attention due to its ability to learn powerful and meaningful representations that are useful for a variety of downstream tasks such as clustering, visualization, and anomaly detection.

The core idea of contrastive learning is to encode data points in such a way that similar (or positive) pairs are brought closer together in the representation space, while dissimilar (or negative) pairs are pushed apart. This is typically achieved through a contrastive loss function, such as the contrastive loss [52], triplet loss [261] or the noise contrastive estimation (NCE) [102]. These loss functions are designed to form a representation space in which the distance between representations of similar items is minimized and the distance

between those of dissimilar items is maximized in a self-supervised manner. One of the popular methods in contrastive learning is the use of Siamese networks [156], which involve twin networks that share weights and process two distinct inputs simultaneously, comparing their outputs to learn discriminative features based on the similarity or difference of the inputs. The outputs of these networks are then compared using a contrastive loss. This setup is powerful for learning from unlabelled data as it only requires a notion of which data points should be similar or different, rather than exact labels.

Contrastive learning has been successfully employed in domains like image processing, natural language processing, and speech recognition. In the field of genomics, contrastive learning has been applied in single-cell multimodal analysis [312], RNA isoform representation learning [89], and motif occupancy identification [178]. Despite its strengths, contrastive learning faces challenges such as the need for careful selection of positive and negative pairs and the potential for instability in training due to the dynamics of learning embeddings. Furthermore, the performance of contrastive learning can heavily depend on the size and diversity of the dataset. Overall, contrastive learning represents a promising direction in unsupervised learning, offering a robust mechanism for learning from unlabelled data and discovering underlying data structures that traditional methods might miss.

2.6.5 Dimensionality Reduction

Another type of unsupervised learning is dimensionality reduction, an important statistical technique, which often analyzes large numbers of samples and/or variables in fields such as genomics. Dimensionality reduction techniques effectively streamline large, high-dimensional datasets without relying on any annotations. This method transforms high-dimensional data into a lower-dimensional space, facilitating more efficient computations and helping to avoid the ‘curse of dimensionality’ [22]. The curse of dimensionality describes different phenomena that occur when analyzing and organizing data in high-dimensional spaces that do not appear in low-dimensional settings. For example, in high dimensions, the volume of the space increases exponentially, causing data points to become isolated and making distance measures less effective, as most points are roughly the same distance from each other. These phenomena can significantly hinder the performance of machine learning models by leading to overfitting and making the data sparse, which complicates the task of identifying meaningful patterns [291]. Additionally, these techniques are capable of helping to overcome the obstacles associated with visualizing and interpreting large-scale data.

In the field of genomics, researchers work with datasets where the number of variables (such as the size of k -mer frequency vectors) can far exceed the number of samples. This disparity makes it challenging to uncover meaningful patterns or perform accurate predictions without overfitting. Furthermore, the high dimensionality of these datasets introduces significant challenges in data visualization. Dimensionality reduction techniques, such as Principal Component Analysis (PCA) [227, 144], t-Distributed Stochastic Neighbor Embedding (t-SNE) [290], and Linear Discriminant Analysis (LDA) [85, 194], are employed to address these challenges. Dimensionality reduction techniques are essential tools in genomic data analysis, enabling researchers to uncover hidden patterns, visualize complex relationships, and gain insights into various biological phenomena.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a fundamental technique in dimensionality reduction, originally conceptualized by Karl Pearson in 1901 [227]. PCA transforms a dataset with potentially correlated variables into a set of linearly uncorrelated variables called principal components. This process involves calculating the eigenvalues and eigenvectors of the dataset's covariance matrix, aiming to ensure that the first principal component captures the greatest variance possible from the original dataset [144].

Mathematically, in a data matrix $\mathbf{X}$, each row represents a different observation and each column represents a different variable. The covariance matrix $\mathbf{C}$ of this data matrix is given by:

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$$

where n is the number of data points. The principal components are then the eigenvectors $\mathbf{v}$ of $\mathbf{C}$, ordered by decreasing eigenvalues λ :

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

These eigenvectors, which are the output of PCA, are the new variables that represent the directions of maximum variance in the data. Each corresponding eigenvalue indicates the amount of variance carried along its respective eigenvector, effectively quantifying the contribution of each direction to the overall variability in the dataset.

By emphasizing variation and bringing out strong patterns in a dataset, PCA helps with identifying directions of maximum variance, thus simplifying the complexity of large datasets without substantial information loss. PCA is a valuable tool in preprocessing data for other machine learning algorithms, enhancing their performance by reducing the

dimensions they need to process. It also aids in filtering noise from datasets by reconstructing the original data excluding the last few principal components, which often carry the least variance and are most susceptible to noise [88].

2.7 Machine Learning Evaluation Metrics

In machine learning, the selection of appropriate evaluation metrics is crucial for assessing the performance of models for different tasks, including classification, regression, and clustering. Effective metrics offer valuable insights into a model’s performance, helping to identify necessary improvements and adaptations. For clustering evaluation, metrics can be categorized into internal and external metrics. External metrics, such as clustering accuracy, Adjusted Rand Index, and Normalized Mutual Information, compare the clustering results to a pre-defined ground truth, assessing the performance based on known classifications. On the other hand, internal metrics, such as the Silhouette Coefficient, evaluate the clustering outcome based on the data itself, focusing on aspects like cohesion and separation within clusters. In the subsequent subsections, we will explore several key metrics frequently used in both classification and clustering tasks. Note, the term ‘metric’ in this thesis is used informally to refer to evaluation measures and is not intended to convey the formal mathematical notion of ‘metric’ (a distance function associated with a metric space, which satisfies the properties of reflexivity, non-negativity, symmetry, and triangle inequality).

2.7.1 Classification Accuracy

Classification accuracy is the primary metric used to calculate the overall effectiveness of a classification model. It quantifies the proportion of total predictions that the model correctly classifies. The mathematical formula for accuracy is given by:

$$\text{Accuracy}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{1}(\hat{y}_i = y_i)$$

In this formula, n represents the number of samples in the dataset, y_i is the actual ground truth label of the i -th sample, and $\hat{y}_i$ is the predicted label. The function $\mathbf{1}(x)$ is an indicator function that returns 1 if $\hat{y}_i = y_i$ and 0 otherwise. An accuracy value of 1 indicates a perfect match, where all samples are correctly assigned, while an accuracy

value of 0 indicates that no samples were assigned correctly. Higher accuracy values are indicative of better alignment with the ground truth.

Despite its intuitive nature and simplicity, accuracy can be a reliable measure of classification model performance when used with balanced datasets. In cases of class imbalance, however, accuracy alone may not effectively reflect a model’s performance, as it could disproportionately represent the prevalence of the majority class. In such scenarios, it is essential to either balance the dataset or utilize other evaluation metrics to achieve a more accurate assessment of the model.

2.7.2 Clustering Accuracy

The clustering algorithm does not provide any relationship between the class labels and the predicted cluster labels. The primary challenge in these tasks is to identify the optimal mapping between these labels. The accuracy for clustering is mathematically defined as follows:

$$\text{Accuracy}(y, \hat{y}) = \max_{\sigma \in S} \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{1}(\sigma(\hat{y}_i) = y_i)$$

Here, vector y represents the true class labels, vector $\hat{y}$ denotes the predicted cluster labels, y_i represents the true class label for sample i , $\hat{y}_i$ denotes the predicted cluster label for sample i , n is the total number of data points, σ refers to a permutation of cluster labels, and $\mathbf{1}$ is the indicator function that returns 1 if the condition is true and 0 otherwise. S represents the set of all possible permutations of cluster labels. This formula considers all possible label assignments to find the one that maximizes the number of correct matches [314].

However, considering that there are $K!$ possible permutations of cluster labels (where K is the number of clusters), directly applying this formula becomes computationally inefficient, especially for a large number of clusters. To efficiently compute this metric, the Hungarian algorithm [160] is employed, which optimizes the matching process to run in $O(K^3)$ time. The Hungarian algorithm achieves this by systematically reducing the cost matrix, a matrix where each element represents the cost of assigning one element (e.g., a predicted label) to another (e.g., a true label). It achieves this by first reducing the costs in the matrix to create zeros. Then, it finds the minimum number of horizontal or vertical lines (either rows or columns) needed to cover all these zeros. This step is crucial as it ensures that all zero-cost assignments are considered, allowing for the best possible match between predicted and true labels. This efficiency is crucial in making the computation of

clustering accuracy feasible even for large datasets with a significant number of clusters. The modified clustering accuracy formula, which employs the Hungarian algorithm to determine the optimal matching, is as follows:

$$\text{Accuracy}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(\pi(\hat{y}_i) = y_i)$$

In this formula, vector y represents the true class labels, vector $\hat{y}$ denotes the predicted cluster labels, y_i represents the true class label for sample i , $\hat{y}_i$ denotes the predicted cluster label for sample i , n is the total number of samples, and $\mathbf{1}$ is the indicator function. π represents the permutation (or assignment) of predicted labels to true labels that minimizes the mismatch, determined by the Hungarian algorithm.

2.7.3 Normalized Mutual Information (NMI)

Normalized Mutual Information (NMI) is a statistical measure derived from the concepts of information theory, specifically used to assess the quality of clustering by quantifying the amount of shared information between two clusterings. NMI quantifies the similarity between two distinct clustering outcomes: the ground truth labels and the labels predicted by a clustering algorithm. This metric evaluates how closely the algorithm's results align with the true labels. The formula for NMI is given by:

$$\text{NMI}(X, Y) = \frac{2 \times I(X, Y)}{H(X) + H(Y)}$$

where $I(X, Y)$ denotes the mutual information between the random variables X and Y , and $H(X)$ and $H(Y)$ are the entropies of X and Y , respectively.

The mutual information $I(X, Y)$ quantifies the information shared by X and Y , measuring how much knowing one of these variables reduces uncertainty about the other. Entropy, on the other hand, measures the uncertainty inherent in a variable, where $H(X)$ and $H(Y)$ represent the entropies of the true and predicted clusterings, respectively [278].

One of the key features of NMI is its normalization, which scales the mutual information value to a range between 0 and 1, where 0 indicates no mutual information (independent cluster assignments) and 1 denotes perfect correlation between the clusterings. The independence of NMI from the absolute cluster sizes, together with its robustness and interpretability, makes it a powerful metric for evaluating the performance of clustering algorithms. By comparing the NMI derived from the true labels and the labels predicted by

a clustering algorithm, we can quantitatively assess how well the algorithm has identified the true underlying clusters.

2.7.4 Adjusted Rand Index (ARI)

The Adjusted Rand Index (ARI) is a widely used measure for evaluating the similarity between two data clusterings (the true labels and the predicted labels obtained from a clustering algorithm). ARI is an extension of the original Rand Index [235], with adjustment for the expected similarity of all pairings under random clustering, providing a more precise metric for comparing clustering outcomes [126].

The Rand Index is calculated as follows:

$$\text{Rand Index} = \frac{a + b}{\binom{n}{2}}$$

where a is the number of pairs of elements that are in the same cluster in both clusterings and b is the number of pairs of elements that are in different clusters in both clusterings. $\binom{n}{2}$ denotes the total number of possible pairs in the dataset.

To account for the possibility that clusters might be formed by chance, the ARI is computed using the formula:

$$\text{ARI} = \frac{\text{RI} - \text{Expected RI}}{\text{Max RI} - \text{Expected RI}}$$

Here, RI represents the Rand Index, Expected RI is the expected value of the Rand Index under random cluster assignments, and Max RI is the maximum value the Rand Index could take. This adjustment ensures that the ARI has a value of 0 when the Rand Index is equivalent to what would be expected by random assignment, and a value of 1 for perfect agreement between the cluster assignments. ARI is bounded between -1 and 1, providing a clear differentiation of performance levels. In the case of negative ARI values, there is significant misalignment between cluster assignments. This indicates that the agreement between the cluster assignments is worse than what would be expected by random assignment.

2.7.5 Silhouette Coefficient

The Silhouette Coefficient is an important metric used to assess the quality of clustering algorithms. It measures the degree of separation between clusters, evaluating how well

each object lies within its cluster compared to other clusters. The Silhouette Coefficient for a single sample is calculated using the following formula:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

In this formula, $a(i)$ represents the average distance from the i -th sample to all other points in the same cluster, which reflects cohesion, and $b(i)$ is the minimum average distance from the i -th sample to points in a different cluster, indicating separation. The Silhouette Coefficient for the dataset is the average of all individual sample scores.

The value of the Silhouette Coefficient ranges from -1 to 1. A value close to 1 indicates that the sample is distant from the neighbouring clusters. A value of 0 implies that the sample is on or very close to the decision boundary of two neighbouring clusters, and negative values suggest that those samples were assigned to the wrong cluster [250]. To assess the overall clustering performance across a dataset, the Silhouette Coefficients of all samples are averaged. This aggregated score reflects the overall separation and cohesion of the clusters: higher average scores suggest better-defined clusters, while lower or negative scores may indicate overlapping or poorly defined clusters.

The Silhouette Coefficient is particularly useful in scenarios where ground truth labels are not available, making it a valuable tool for evaluating unsupervised clustering algorithms. However, it's important to note that the Silhouette Coefficient tends to be higher for convex clusters, and may not be as effective for density-based clusters with non-convex shapes.

2.8 Recent Applications and Relevant Technologies

Advancements in genomic research, especially in the classification and clustering of DNA sequences, are the result of many new applications, each solving specific problems in analyzing and understanding complex biological data. This section reviews some of these latest applications and relevant studies, which will be used for comparison of proposed methods in the following chapters.

2.8.1 Machine Learning with Digital Signal Processing (ML-DSP)

Machine Learning with Digital Signal Processing (ML-DSP) combines machine learning with digital signal processing for fast and accurate alignment-free classification of DNA

sequences [236]. ML-DSP turns DNA sequences into numerical formats that are suitable for signal processing. The study [236] examined thirteen different DNA numerical representations, which can be categorized into fixed mappings without biological significance, mappings based on chemical properties of nucleotides, and mappings deduced from the doublets or codons [162, 26]. The authors found that the “Purine/Pyrimidine” (PP) (encoding T and C as “1”, and G, A as “-1”), “Just-A” (encoding A as “1”, and G, C, T as “0”), and “Real” (encoding T as “1.5”, C as “0.5”, A as “1.5”, and G as “0.5”) representations were the most effective for classifying mtDNA datasets.

ML-DSP quantifies differences between DNA sequences by calculating the Pearson Correlation Coefficient between the magnitude spectra obtained from their discrete Fourier transforms (DFTs). DFT converts the DNA sequences into frequency distributions, allowing ML-DSP to compare and quantify the sequence distance based on these spectral differences. It uses the Pearson Correlation Coefficient (PCC) to compare the magnitude spectra, creating a pairwise distance matrix that shows the relationships between all sequences in the dataset. ML-DSP uses several supervised machine learning algorithms, such as Linear Discriminant, Linear SVM, Quadratic SVM, Fine KNN, Subspace Discriminant, and Subspace KNN, for classification. These algorithms are trained using the distance matrix and the known taxonomic labels from a training set. Once trained, ML-DSP can classify new DNA sequences by comparing their spectral data to those in the training set. The results show that ML-DSP outperforms traditional alignment-based and other alignment-free methods in terms of speed and accuracy, achieving on average 97% accuracy on mtDNA at different levels of taxonomy, 100% accuracy on subtyping dengue virus genomes, and 95.5% accuracy on bacterial genomes at phylum level.

2.8.2 Viral Genome Deep Classifier (VGDC)

VGDC (Viral Genome Deep Classifier) is a viral subtyping tool that employs a deep CNN [80] to detect specific combinations of nucleotides that distinguish between different virus subtypes. VGDC replaces nucleotide symbols with their corresponding ASCII codes and extends the length of each genome to match the longest genome in the dataset. The CNN is trained on a dataset of viral genomes using the categorical cross-entropy loss function. This loss function measures the discrepancy between the true labels and the predicted probabilities, penalizing predictions that differ from the actual labels [301].

The authors tested VGDC on five viral datasets (Dengue, Hepatitis B, Hepatitis C, HIV-1, and Influenza A), comparing its performance to existing tools such as CASTOR [241] and COMET [279]. Results indicate that VGDC is a robust tool capable of classifying

different viral families, outperforming competitors in HIV-1 and influenza A subtyping. Notably, VGDC demonstrated high accuracy especially for complex datasets with numerous subtypes and recombinants, as well as better scalability for large training datasets.

2.8.3 MeShClust v3.0

MeShClust v3.0 [95] is the latest version of a software tool designed for unsupervised clustering of DNA sequences. It builds on the strengths of earlier versions—MeShClust v1.0 [131] (a DNA clustering method) and *Identity* [96] (a sequence alignment identity score predictor). MeShClust v3.0 uses the mean shift algorithm [53], a clustering method that identifies and analyzes the most dense regions of data points in a feature space. It iteratively shifts data points towards the peak of the density function, effectively grouping them into clusters based on their proximity to these peaks.

Traditional tools such as CD-HIT [172] and UCLUST [75] depend on the Needleman-Wunsch global alignment algorithm to determine how similar sequences are. CD-HIT and UCLUST can be slow and inefficient for very long DNA sequences, such as those from bacterial genomes. In MeShClust v3.0, input sequences are first converted into k -mer frequency vectors, which serve as initial one-member cluster centers. The mean shift clustering algorithm then iteratively updates these centers by computing new means from sequences whose identity scores exceed a *threshold*, merges centers with similar scores, and replaces each cluster’s mean with the sequence that has the highest identity score. This process is repeated until the algorithm converges, optimizing cluster formation based on sequence similarity, ensuring that related sequences are grouped together while dissimilar sequences are placed into different clusters. This process enables the efficient clustering of very long DNA sequences, such as those in a bacterial dataset where the average length of each sequence is 3.7 million base pairs, without requiring traditional sequence alignment.

MeShClust v3.0 enhances clustering by automatically selecting the optimal similarity score *threshold* based on the data, a notable advancement over CD-HIT and UCLUST, which necessitate manual threshold setting often without specific guidelines. Although MeShClust v3.0 has been validated on a variety of datasets—including twenty-two synthetic and five real biological datasets like the human microbiome and maize transposons, demonstrating its ability to efficiently process up to a million sequences—the effectiveness of these automatically chosen thresholds can vary widely. This variability stems from the unique characteristics of each genomic dataset, meaning that different datasets might require different thresholds for optimal clustering, complicating the application of a single automatic method across diverse datasets.

Chapter 3

DeLUCS: Deep Learning for Unsupervised Clustering of DNA Sequences

The content in this chapter is an adaptation of the paper “DeLUCS: Deep learning for unsupervised clustering of DNA sequences” [201]. This chapter begins with Section 3.1, which lays an introduction to the foundational concepts and the significance of unsupervised learning of DNA sequences, setting the stage for a detailed exploration of the DeLUCS methodology.

Following the introduction, Section 3.2 discusses related work in the field, providing a critical review of both supervised and unsupervised machine learning approaches that have been previously applied to genomic data. This review highlights the advancements in the field and identifies gaps that DeLUCS methodology aims to address.

In Section 3.3, we discuss the materials and methods employed in DeLUCS, detailing the technical aspects of the methodology including the Invariant Information Clustering (IIC), data augmentation strategies for DNA sequences (generation of *mimic sequences*), the architecture of the artificial neural network used in DeLUCS pipeline, and the evaluation setup that demonstrate the effectiveness of the proposed approach. This section also covers the implementation details and the datasets utilized throughout this study.

The results obtained from applying the DeLUCS methodology, in comparison with other state-of-the-art methods, are presented in Section 3.4. This is followed by Section 3.5, which discusses the implications of the findings, the strengths and limitations of the proposed DeLUCS methodology, and its potential impact on the field.

The chapter concludes with Section 3.6, summarizing the key contributions and findings of the DeLUCS study and suggesting directions for future research in this area.

3.1 Introduction

Traditional DNA sequence classification algorithms rely on large amounts of labour-intensive and human expert-mediated annotating of primary DNA sequences, informing origin and function. Moreover, some of these genome annotations are not always stable, given inaccuracies and temporary assignments due to limited information, knowledge, or characterization, in some cases. Also, since there is no taxonomic “ground truth,” taxonomic labels can be subject to dispute (see, e.g., [14, 226, 274]). In addition, as methods for determining phylogeny, evolutionary relationships, and taxonomy evolved from physical to molecular characteristics, this sometimes resulted in a series of changes in taxonomic assignments. An instance of this phenomenon is the microbial taxonomy, which recently underwent drastic changes through the Genome Taxonomy Database (GTDB) in an effort to ensure standardized and evolutionary consistent classification [224, 46, 223].

The applicability of existing classification algorithms is limited by their intrinsic reliance on DNA annotations, and on the “correctness” of existing sequence labels. For example, alignment-based methods crucially rely on DNA annotations indicating the gene name and genomic position. Similarly, supervised machine learning algorithms rely on the training data having stable taxonomic labels, since they carry forward any current misclassifications into erroneous future sequence classifications. To avoid these limitations, and given the ease of extensive sequence acquisition, there is a need for highly accurate unsupervised machine learning approaches to sequence classification that are not dependent on sequence annotations.

In this chapter, we propose a novel **Deep Learning** method for the **Unsupervised Clustering of DNA Sequences** (DeLUCS), that is independent of sequence labels or annotations, and thus is not vulnerable to their inaccuracies, fluctuations, or absence. DeLUCS is, to the best of our knowledge, the first *highly-effective/light-preparation* DNA sequence clustering method, in that it achieves high accuracies while using only a minimum of data preparation and information. Indeed, the only information external to the primary DNA sequence that is used by DeLUCS is the implicit requirement that all sequences be of the same type (nuclear DNA, mtDNA, plastid, chloroplast) and that the selection of the dataset be based on some taxonomic criteria. Importantly, DeLUCS does not need any DNA annotations, does not require sequence homology or similarity in sequence lengths,

and does not use any true taxonomic labels or sequence identifiers during the learning process.

In fact, DeLUCS only uses Frequency Chaos Game Representations with resolution k (FCGR) of primary DNA sequences to find their cluster assignments, in an unsupervised manner. (For the methodologies discussed in this chapter, $k = 6$ was empirically selected as achieving the optimal balance between accuracy and time complexity.) The *post hoc* step of evaluating the performance of DeLUCS uses, by necessity, true taxonomic labels to quantify the accuracy of the computed clusters. Since the output of clustering methods such as DeLUCS only consists of numerical labels rather than meaningful taxonomic labels, the Hungarian algorithm [160] is implemented to find the optimal mapping between these numerical cluster labels and the actual taxonomic labels. Based on this mapping, the resulting taxonomic cluster labels are then used to calculate the method’s clustering accuracy (see Section 2.7.2 for more details). Note that, for readability purposes, we will hereafter use true taxonomic labels for clusters, even though DeLUCS only outputs numerical cluster labels.

DeLUCS compensates for the absence of information external to the primary DNA sequence by leveraging the capability of deep learning to discover patterns (genomic signatures) in unlabelled raw primary DNA sequence data. DeLUCS is alignment-free and learns clusters that match true taxonomic groups for large and diverse datasets, with high accuracy: 2,500 vertebrate complete mitochondrial genomes at multiple taxonomic levels, with accuracy ranging from 79% to 100%; 3,200 randomly selected bacterial genome segments, with an average length of 400 kbp, into bacterial families, with accuracy of 77% (inter-phylum) and 90% (intra-phylum); several datasets of viral gene sequences and of full viral genomes, averaging 1,300 sequences each, into virus subtypes, with accuracy of 99%, and 100% respectively. To the best of our knowledge, these are the largest real datasets classified to date, in clustering studies of genomic data: The biggest dataset analyzed in this work totals over 1 billion bp of data, a full order of magnitude bigger than previous studies [19, 131, 37, 123, 197, 6, 8]. In addition, all but the viral gene dataset would be impossible to classify with alignment-based methods, due either to the prohibitive time cost of multiple sequence alignment or to the lack of sequence homology.

A direct comparison shows that DeLUCS significantly outperforms two classic algorithms for clustering unlabelled datasets (K -Means++ and Gaussian Mixture Models), sometimes by as much as 47%. We also note that, for the majority of the computational tests, DeLUCS accuracy is also comparable to, and sometimes higher than, that of a supervised machine learning algorithm with the same architecture.

DeLUCS is a fully automated method that determines cluster assignments for its input

sequences, independent of any homology or same-length assumptions, and oblivious to sequence taxonomic labels. DeLUCS can thus be used for successful *ab initio* clustering of datasets that were previously unclassifiable by alignment-based methods, as well as datasets with uncertain or fluctuating taxonomy, where supervised machine learning methods are biased by their reliance on current taxonomic labels.

In summary, DeLUCS is the first effective alignment-free method that uses deep learning for unsupervised clustering of unlabelled raw DNA sequences. The main contributions of this work are:

- DeLUCS clusters large and diverse datasets, such as complete mitochondrial genomes at several taxonomic levels; randomly selected bacterial genome segments into families; viral genes and viral full genomes into virus subtypes. To date, these are the largest real datasets of genomic data to be clustered, with our largest experiment comprising over 1 billion bp of data, a full order of magnitude larger than previous studies [19, 131, 37, 123, 197, 6, 8].
- DeLUCS achieves clustering accuracies of 77% to 100%, in each case significantly higher than classic alignment-free clustering methods (K -Means++ and GMM), with double-digit improvements in most cases. For the majority of the computational tests, DeLUCS clustering accuracies are also comparable to, or higher than, classification accuracies of a supervised machine learning algorithm with the same architecture.
- DeLUCS is a highly-effective/light-preparation method for unsupervised clustering of DNA sequences. Its high clustering accuracies are a result of combining the novel concept of *mimic sequences* with the invariant information learning framework and a majority voting scheme. It is termed light-preparation because it does not require sequence homology, sequence-length similarity, or any taxonomic labels/identifiers during the learning process.

3.2 Related Work

The time-complexity limitations of alignment-based methods, [300], in addition to their reliance on extraneous sequence information such as sequence homology, have motivated the development of numerous alignment-free methodologies [333, 332]. Of these, methods based on k -mer counts have been among the fastest and the most widely used [332]. In parallel to alignment-free approaches, machine learning methods have emerged as promising alternatives for solving classification problems both in genomics and biomedicine [299].

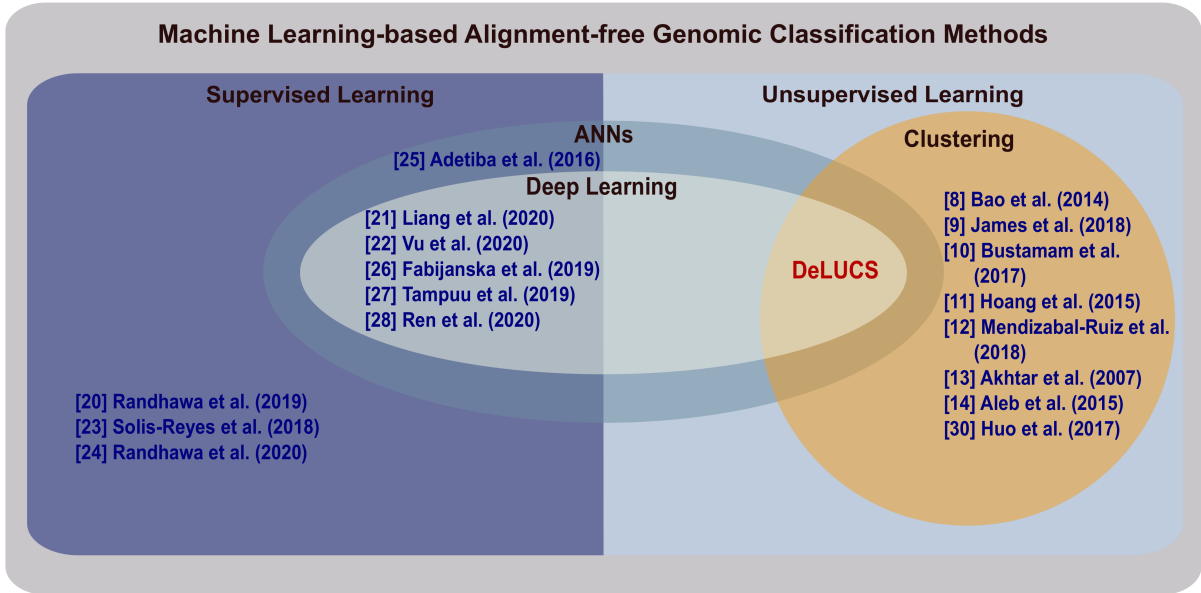


Figure 3.1: Machine learning-based alignment-free methods for classification/clustering of DNA sequences. DeLUCS is the first method to use deep learning for accurate unsupervised clustering of unlabelled DNA sequences. The novel use of deep learning in this context significantly boosts the clustering accuracy, compared to two other unsupervised machine learning clustering methods (K -Means++ and GMM).

Figure 3.1 illustrates a summary of methods that combine alignment-free approaches with machine learning for genomic classification/clustering tasks. (The difference between classification and clustering is that, while in classification methods the cluster labels are given *a priori*, in clustering methods the clusters are “discovered” by the method.)

3.2.1 Supervised Machine Learning Approaches

Among supervised learning algorithms, Artificial Neural Networks (ANNs) have proven to be the most effective, with ANN architectures with several layers of neurons (“deep learning”) being the top performers [315].

In the context of genome classification, alignment-free methods that employ supervised machine learning have been shown to outperform alignment-based methods in the construction of high-quality whole-genome phylogenies [236], profiling of microbial communities [174], and DNA barcoding at the species level [298]. In recent years, alignment-free

methods have successfully applied supervised machine learning techniques to obtain accurate classification of HIV subtypes, [275], as well as accurate and early classification of the SARS-CoV-2 virus (COVID-19 virus) [238]. The increasing success of machine learning, and in particular deep learning, techniques is partly due to the introduction of suitable numerical representations for DNA sequences and the ability of the methods to find patterns in these representations (see [275, 2], respectively [174]). Other classification tasks in genomics such as taxonomic classification [80], and the identification of viral sequences among human samples from raw metagenomic segments [282, 244] have also been explored from the deep learning perspective.

One limitation of supervised deep learning is that the performance of the ANNs is heavily dependent on the number of labelled sequences that are available during training. This can become a limiting factor, even though raw sequencing data can now be obtained quickly and inexpensively, [198]. The reason for this is the intermediate process that lies between obtaining a raw DNA sequence and uploading that sequence into a public sequence repository, namely the “invisible” work that goes into assigning a taxonomic label and attaching biological annotations. This is a laborious, expensive, and time-consuming multistep process, comprising *ad hoc* wet lab experiments and protocols that cannot be automated due to the human expertise required. Another limitation of supervised learning is its sensitivity to perturbations in classification, since any present misclassifications in the training set are “learned” and propagated into future classification errors.

To overcome these limitations, one can attempt to use unsupervised learning, which operates with unlabelled sequences and compensates for the absence of labels by inferring identity-relevant patterns from unlabelled training data. Moreover, unsupervised learning does not perpetuate existing labelling errors, as the algorithms are oblivious to labels. It can correctly classify sequences of a type never seen during any previous training, by assigning the sequences to dynamically defined new clusters.

3.2.2 Unsupervised Machine Learning Approaches

Unlike supervised learning, in unsupervised learning training samples are unlabelled, i.e., the cluster label associated with each DNA sequence is not available (or is ignored) during training. In general, 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. The effort made so far in the development of unsupervised alignment-free clustering algorithms for genomic sequences has been mainly focused on using generic clustering algorithms such

as K -Means or Gaussian Mixture Models (GMM) for different numerical representations of DNA sequences. For example, Bao et al., [19] used a representation of DNA sequences based on their word counts and Shannon entropy, whereby each sequence is represented by a 12-dimensional vector, and the clustering is performed using K -Means with Euclidean distance. James et al., [131] grouped DNA sequences based on four different similarity measures obtained from an alignment-free methodology that used k -mer frequencies and an adaptation of the mean shift algorithm, normally used in the field of image processing. Similar work [37, 127, 8] also builds on the K -Means algorithm and k -mer counts. Another approach is the use of digital signal processing [123, 197, 6], whereby Fourier spectra calculated from a numeric representation of a DNA sequence are used as their quantitative description, and the Euclidean distance is used as a measure of dissimilarity to be employed by either the K -Means or the GMM clustering algorithms.

Although K -Means (and its improved version K -Means++) is a simple and versatile algorithm, it is dependent on several restrictive assumptions about the dataset, such as the need for manual selection of the parameter K , and the assumption that all clusters have the same size and density. It is also heavily dependent on the selection of initial cluster centroids, meaning that for large datasets, numerous initializations of the centroids are required for convergence to the best solution and, moreover, that convergence is not guaranteed [98]. Although GMM is more flexible in regards to the distribution of the data and does not assume that all clusters are spherical, the initialization of clusters is still challenging, especially in high dimensional data, [24, 5].

A potential solution to these drawbacks could lie in recent developments in the field of computer vision [40, 308, 139], specifically in the concepts at the core of *invariant information clustering* (IIC) [139], one of the successful methods for the clustering of unlabelled images. These methods are effective for visual tasks and, as such, are not directly applicable to genomic data. In this chapter, we propose the use of Frequency Chaos Game Representations (FCGR) of DNA sequences and the novel notion of *mimic sequences* as a method for DNA data augmentation, to leverage the idea behind IIC. In our approach, FCGR pairs of sequences and their *mimics* are generated and used as input for a *de novo* simple but general Artificial Neural Network (ANN) architecture, specifically designed for the purpose of DNA sequence clustering. Finally, majority voting over several independently trained ANN copies is used, to obtain the accurate cluster assignment of each DNA sequence.

3.3 Materials and Methods

In this section, we first give an overview of our method and the computational pipeline of DeLUCS. We then describe the core concepts of invariant information clustering, and detail how these concepts are adapted to DNA sequence clustering, by introducing the notion of *mimic sequences*. This is followed by a description of the architecture of the neural networks employed, the evaluation setup used for assessing the performance of DeLUCS, and all of the implementation details. Finally, we give a description of all the datasets used in this chapter.

3.3.1 Method Overview

DeLUCS employs a graphical representation of DNA sequences introduced by Jeffrey in [136], called Chaos Game Representation (CGR). In the DeLUCS pipeline, we use a quantified version of CGR, called Frequency CGR with resolution k , and denoted by FCGR. The FCGR of a DNA sequence is a two-dimensional unit square image, with the intensity of each pixel representing the frequency of a particular k -mer in the sequence [68]. FCGR is a compressed representation of the original DNA sequence, with the degree of compression indicated by the resolution k (see Section 2.3.4 for a comprehensive description of CGR and FCGR). All computational experiments in this study use $k = 6$, which was empirically assessed as achieving the best balance between accuracy and time complexity.

Unlike some quantization techniques [215, 163], DeLUCS does not need an intermediate step of supervised learning to produce a compressed representation of data, retaining only the information needed for the correct classification of the feature vector and the cluster assignment. This is because FCGR already is a compressed representation, by virtue of storing only the counts of all k -mers in the sequence for a given value of k . These k -mer counts contain the intrinsic, taxonomically relevant, information used for the unsupervised learning in DeLUCS.

The general pipeline of DeLUCS, illustrated in Fig 4.1, consists of three main steps:

1. For each DNA sequence in the dataset several artificial *mimic sequences* (augmented DNA sequences) are constructed, and considered to belong to the same cluster. These mimic sequences are generated using a probabilistic model based on transversions and transitions. The k -mer counts for both the original sequence and its mimic sequences are then computed, to produce their respective FCGRs. In this study, $k = 6$ was empirically assessed as achieving the best balance between high accuracy and speed.

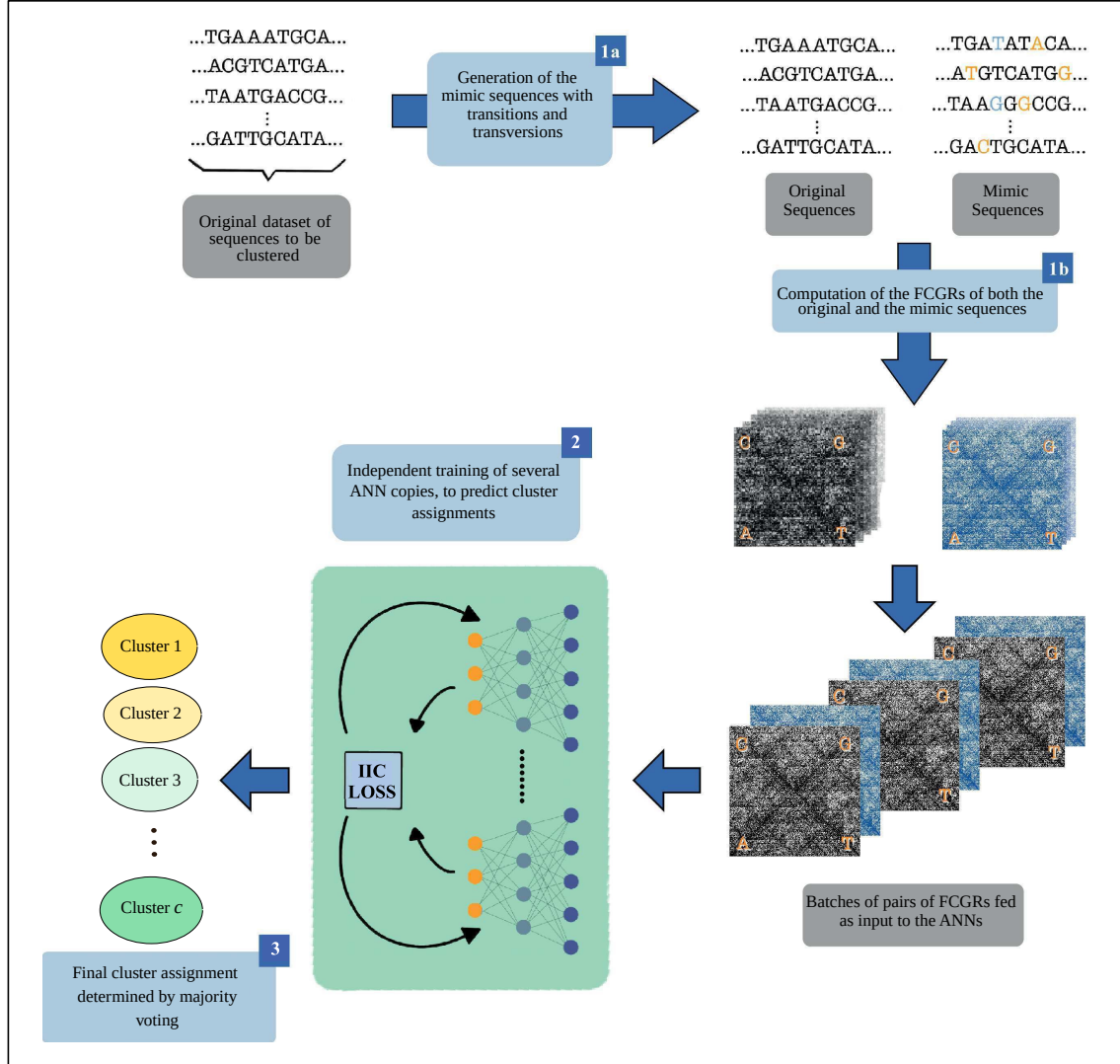


Figure 3.2: General DeLUCS pipeline. The input consists of the original DNA sequences to be clustered. (1a): Artificial *mimic sequences* are generated from the original sequences, by using a probabilistic model based on transitions and transversions. (1b): FCGRs of all original (black) sequences, and of all *mimic* (blue) sequences are computed, and data pairs of the form “FCGR of DNA sequence, FCGR of one of its mimics” are divided into batches for the training process. (2): Several copies of the ANN are trained independently, with the loss function being the negative mutual information between the network predictions for a sequence and that of its mimic. (3): Majority voting is used to obtain the final cluster assignment for each sequence.

2. Pairs consisting of the FCGR of the original DNA sequence and the FCGR of one of its mimic sequences are then used to train several copies of an Artificial Neural Network (ANN) independently, by maximizing the mutual information between the network predictions for the members of each pair.
3. As the training process of the ANNs is a randomized algorithm that produces different outcomes with high variance, a majority voting scheme over the outcomes of the ANNs in Step 2 is used to determine the final cluster assignment for each sequence.

To evaluate the quality of the clusters, an additional step is performed, independent of DeLUCS. This step first utilizes the Hungarian algorithm to determine the optimal correspondence between the cluster assignments learned by DeLUCS and the true taxonomic cluster labels.

3.3.2 Invariant Information Clustering (IIC)

Steps 1 and 2 in the DeLUCS pipeline build upon the underlying concepts of IIC, [139], which leverages some information theory notions described in this subsection. Given a discrete random variable X that takes values $x \in \mathcal{X}$ and has probability mass function $p(x) = P(X = x)$, the entropy $H(X)$ is a measure of the average uncertainty in the random variable and is defined by:

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \quad (3.1)$$

$H(X)$ also represents the average number of bits required to describe the random variable X .

Given a second random variable $\tilde{X}$ that takes values $\tilde{x} \in \tilde{\mathcal{X}}$, we can also define the conditional entropy $H(X|\tilde{X})$, for a pair sampled from a joint probability distribution $p(x, \tilde{x}) = P(X = x, \tilde{X} = \tilde{x})$, as the entropy of a random variable X conditional on having some knowledge about the variable $\tilde{X}$. The reduction in the uncertainty of X introduced by the additional knowledge provided by $\tilde{X}$ is called *mutual information* and it is defined by:

$$I(X, \tilde{X}) = H(X) - H(X|\tilde{X}) = \sum_{x, \tilde{x}} p(x, \tilde{x}) \log \frac{p(x, \tilde{x})}{p(x)p(\tilde{x})} \quad (3.2)$$

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

The information bottleneck principle [287, 288], which is part of the information-theoretic approach to clustering, suggests that clusters only need to capture relevant information. In order to filter out irrelevant information, IIC aims to learn only from paired data, i.e., from pairs of samples $(\mathbf{x}, \tilde{\mathbf{x}}) \in \mathcal{X} \times \tilde{\mathcal{X}}$ taken from a joint probability distribution $p(\mathbf{x}, \tilde{\mathbf{x}})$. If, for each pair, $\tilde{\mathbf{x}}$ is an augmented version of $\mathbf{x}$, it is possible to find a mapping Φ that encodes what is common between $\mathbf{x}$ and $\tilde{\mathbf{x}}$, while dropping all the irrelevant information. If such a mapping Φ is found, the image $\mathcal{Y} = \Phi(\mathcal{X})$ becomes a compressed representation of the original space $\mathcal{X}$.

To find the best candidate for Φ , one way is to make $\Phi(\mathbf{x})$ represent a random variable, and then maximize the predictability of sample $\mathbf{x}$ from sample $\tilde{\mathbf{x}}$ and vice versa, that is, find a mapping $\Phi(\mathbf{x})$ that maximizes $I(\Phi(\mathbf{x}), \Phi(\tilde{\mathbf{x}}))$ – the mutual information between the encoded variables – over all $\mathbf{x} \in \mathcal{X}$.

This idea suggests that Φ can be calculated using a deep neural network with *softmax* as the output layer. For a dataset with an expected number of c clusters, $c \in \mathbb{N}$, the output space will be $\mathcal{Y} = [0, 1]^c$, where for each sample $\mathbf{x}$ we have that $\Phi(\mathbf{x})$ represents the distribution of a discrete random variable over the c clusters. The mutual information can be modified with the introduction of a hyper-parameter $\lambda \in \mathbb{R}$ that weighs the contribution of the entropy term in Eq (3.2). However, instead of maximizing the weighted mutual information, we use a numerical optimizer to minimize its opposite (mathematically, the negative weighted mutual information) during the training process of the ANN. Hence, the loss function to be minimized becomes:

$$\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}) = -\lambda \cdot H(\Phi(\mathbf{x})) + H(\Phi(\mathbf{x}) \mid \Phi(\tilde{\mathbf{x}})). \quad (3.3)$$

In Eq (3.3), the entropy term $H(\Phi(\mathbf{x}))$ measures the amount of randomness present at the output of the network, and it is desirable for that value to be as large as possible, in order to prevent the architecture from assigning all samples to the same cluster. The conditional entropy term $H(\Phi(\mathbf{x}) \mid \Phi(\tilde{\mathbf{x}}))$ measures the amount of randomness present in the original sample $\mathbf{x}$, given its correspondent $\tilde{\mathbf{x}}$. This conditional entropy should be as small as possible, since the original sample $\mathbf{x}$ should be perfectly predictable from the augmented sample, $\tilde{\mathbf{x}}$.

3.3.3 Generation of Mimic Sequences

The success of the method described in the previous section is fundamentally dependent on the way $\tilde{\mathbf{x}}$ (augmented version of $\mathbf{x}$) is artificially generated from $\mathbf{x}$. In the particular case of our application, where the samples $\mathbf{x}$ are DNA sequences, we refer to the artificially created augmented DNA sequence $\tilde{\mathbf{x}}$ as *mimic sequences* (sometimes called simply *mimics*). In this context, the generation of mimic sequences poses the additional challenge that they should be sufficiently similar to the originals so as not to be assigned to a different cluster.

Given a set $X = \{x_1, \dots, x_n\}$ of n DNA sequences, we construct the set of pairs

$$\{(x_i, x_i^1), (x_i, x_i^2), (x_i, x_i^3), \dots, (x_i, x_i^m) \mid 1 \leq i \leq n\}, \quad (3.4)$$

where $m \geq 3$ is a parameter representing the number of mimic sequences generated for each original sequence x_i , $1 \leq i \leq n$. We use a simple probabilistic model based on DNA substitution mutations (transitions and transversions) to produce different mimic sequences, as follows. Given a sequence x_i and a particular position j in the sequence, we fixed independent transition and transversion probabilities $p_{ts}[j]$ and $p_{tv}[j]$ respectively. Next, we produce the following mimic sequences, probabilistically: x_i^1 with only transitions, x_i^2 with only transversions, and x_i^j 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, but may have to be increased if the number of available sequences per cluster is insufficient to obtain a high clustering accuracy.

The rationale behind using transition and transversion probabilities for DNA data augmentation, and generating mimics is biologically inspired. That being said, we use this method only as a mathematical tool without attributing any biological significance, to create minimally different sequences through randomly distributed base substitutions. In this work, we use probabilities $p_{ts} = 10^{-4}$ and $p_{tv} = 0.5 \times 10^{-4}$, assessed empirically to result in the best clustering accuracies. Although the mutation rates used are biologically inspired, they are not biologically precise given that mutation rates vary regionally, with species [207, 313], and with the estimation method [256]. Lastly, in practice, with no taxonomic label, it is impossible to select species-specific mutation rates.

3.3.4 Artificial Neural Network (ANN) Architecture

The pairs of FCGRs of the original DNA sequences and their mimic sequences are used as inputs, to train several independent copies of an ANN. Since the size of the genomic datasets



Figure 3.3: Architecture of the deep artificial neural network used in DeLUCS pipeline. The input FCGRs are flattened into one-dimensional representations prior to entering the first *Linear* layer. The parameter in each linear layer except the output layer represents the number of neurons. For the *Dropout* layer, the parameter represents the dropout rate. The parameter c (in $c_clusters$) of the output linear layer, represents the expected upper bound of the number of clusters. The *Softmax* layer is used to obtain a probability distribution as the output of the network.

under study is at least an order of magnitude smaller than what is used in computer vision, we noted that the common architectures that have proven effective in the application of deep learning for various visual tasks were not suitable for our datasets. Hence, we designed, *de novo*, a simple but general architecture that is suitable for the clustering of DNA sequences. The complete architecture is presented in Figure 3.3 and it consists 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 dropout rate of 0.5. The output layer *Linear ($c_clusters$)*, where c is a numerical parameter representing the upper bound of the number of clusters, is followed by a *Softmax* activation function.

The network receives as input pairs of FCGRs (two-dimensional representations of DNA sequence composition) and flattens them into one-dimensional representations, which are then fed sequentially to the first *Linear* layer. The inclusion of *ReLU*s is essential for the training process, as they help mitigate the problems of vanishing gradients and other back-propagation errors. The dropout layers prevent the model from over-fitting, which in unsupervised learning comes in the form of degenerate solutions, i.e., all the samples being assigned to the same cluster. Finally, the *Softmax* layer gives as output a c -dimensional vector $\Phi(\mathbf{x}) \in [0, 1]^c$, such that $\Phi_{c_j}(\mathbf{x})$, $1 \leq c_j \leq c$, represents the probability that an input sequence $\mathbf{x}$ belongs to a particular cluster c_j .

Note that this general architecture was designed so as to be successful for the clustering of all the diverse datasets presented in this study. However, the main pipeline of DeLUCS

allows it to be used also with other architectures, including architectures that make use of the two-dimensional nature of the FCGR patterns and are performant for specific types of genomic data (e.g., Convolutional Neural Networks).

3.3.5 Evaluation

Clustering results can be evaluated using both internal and external validation measures. Internal validation methods [250, 182] evaluate clustering algorithms by measuring some of the discovered clusters’ internal properties (e.g., separation, compactness), while external validation methods [278, 235, 314] evaluate clustering algorithms by measuring the similarity of the discovered clustering with the ground truth. We note that many genomic datasets are sparse, incomplete, and subject to sampling bias (more than 86% of existing species on Earth and 91% of species in the oceans have not yet been classified and catalogued [145]). Thus, when taxonomic ground truth is available for an external validation, agreement between discovered clusters and real taxonomic groups is preferable to, and more informative than, internal validation methods.

We include performance comparison results obtained using clustering accuracy, as it uses the optimal mapping between the discovered clusters and the ground truth clusters, and has been used extensively in recent deep unsupervised learning studies [146]. Additionally, we incorporated results from comparisons using two other external evaluation metrics: Normalized Mutual Information (NMI), and the Adjusted Rand Index (ARI). See Section 2.7 for a detailed description of these three evaluation metrics used in this chapter.

Several unsupervised clustering methods exist in the literature, and various deep learning-based clustering tools have been adapted to bioinformatics [146]. However, these methods are domain-specific and are not optimized to operate with DNA sequences, hence they perform poorly when used with DNA datasets. In this chapter, *K*-Means++ and Gaussian Mixture Model (GMM) were selected for comparison with DeLUCS because they are general clustering algorithms at the core of many unsupervised learning frameworks, and have been previously used with DNA sequence datasets [19, 37, 6, 8]. Note that the Hungarian algorithm is used to find the mapping that maximizes clustering accuracy for all the unsupervised clustering methods considered. In all three cases, the use of the true taxonomic labels is for *post hoc* evaluation purposes only, as true labels are never used during the training process.

Lastly, we compare these three unsupervised clustering methods to 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 accuracy of the classification is calculated by first taking 70% of the data for training, and 30% of the data for testing.

3.3.6 Implementation

During the training procedure, all the hyperparameters of the method are fixed and common to all of the tests, and were empirically selected as yielding the best performance. All the flattened FCGR were normalized before being fed into the network, by using the $L1$ norm, i.e., by dividing the values of each k -mer count vector by their sum. This normalization brings all of the inputs of the ANN into the same range of values, which contributes to the reliability of the ANN convergence.

The networks are initialized using the Kaiming method [113], to avoid exponential reduction or magnification of the input magnitudes. 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. We use the Adam optimizer, [154], with a learning rate of 5×10^{-5} , and the networks were trained for 150 epochs with no early stopping conditions. Another vital consideration during training is the selection of the batch size (empirically determined to be 512), because the marginalization that is performed to find the distribution of the output is done over each batch of pairs. If the batch size is not large enough to represent the real distribution of the data, the entropy term in the loss function becomes dominant, leading to sub-optimal solutions. Lastly, we fix the value of the hyperparameter λ to 2.5 (in Eq 3.3).

DeLUCS is fully implemented in Python 3.7, and the source code is publicly available in the GitHub repository <https://github.com/Kari-Genomics-Lab/DeLUCS>. Users may reproduce the results obtained in this chapter, or use their own datasets for the purpose of clustering new sequences. All of the tests were performed on one of the nodes of the Cedar cluster of Compute Canada (2 x Intel E5-2650 v4 Broadwell @ 2.2GHz CPU, 32 GB RAM) with NVIDIA P100 Pascal (12G HBM2 memory).

3.3.7 Datasets

We used three different datasets in this study to confirm the applicability of our method to different types of genomic sequences (mitochondrial genomes, randomly selected bacterial genome segments, viral genes, and viral genomes), and all data was retrieved from publicly available databases. Tables 3.1, 3.2 and 3.3, summarize the dataset details for each of the

11 computational tests performed.

Mitochondrial genomes (Table 3.1): The first dataset consists of complete vertebrate mitochondrial genomes. We used the software Geneious 2020.2.4 ¹ to obtain a list with 13,300 accession numbers. Sequences were downloaded from the National Center for Biotechnology Information (NCBI) on November 16, 2020, after removing all of the sequences shorter than 14,000 bp and longer than 24,500 bp. For this dataset, the goal was to assess the performance of DeLUCS at different taxonomic levels, starting at the sub-phylum level, down to the genus level. To do so, the cluster with the largest number of sequences available at each taxonomic level was selected for the next computational experiment, to ensure that we could reach as deep as possible into the phylogenetic tree.

DeLUCS reaches optimal performance when a dataset consists of balanced clusters, that is, clusters that are similar in size, and when each cluster has more than a minimum number of sequences (herein 20 sequences). These two requirements, together with the number of sequences per cluster available on NCBI, were used to determine the minimum and maximum cluster size for each test.

After determining the minimum cluster size for a test (the size of the smallest cluster larger than 20), the clusters that were smaller than this test minimum were discarded. In addition, sequences that belonged to the parent taxon, but lacked a sub-taxon identifier (cluster label) were excluded.

Some clusters were larger than the test maximum (a number bigger than – but not more than 20% bigger than – the test minimum, to ensure cluster balance). For such clusters, sequences were randomly selected until the test maximum was reached. Note that since the minimum and maximum cluster sizes are test-dependent, the number of sequences in a cluster is also test-dependent (for example, Test 1 uses 500 Actinopterygii sequences, while Test 2 uses only 113). Moreover, for similar reasons, the number of sequences per cluster is usually smaller than the total number of sequences with that cluster label that are available on NCBI.

Bacterial genomic sequences (Table 3.2): The second dataset consists of 3,200 randomly selected genomic segments of bacterial DNA from the eight different families analyzed in [123]. These bacterial families belong to three different phyla: *Spirochaetes* (*Treponemataceae*), *Firmicutes* (*Bacillaceae*, *Clostridiaceae*, and *Staphylococcaceae*), and *Proteobacteria* (*Enterobacteriaceae*, *Rhodobacteriaceae*, *Desulfovibrionaceae*, and *Burkholderiaceae*).

¹<https://www.geneious.com/>

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 (Actinopterygii: 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, Xenocyprididae: 80, Acheilognathidae: 70, Nemacheilidae: 80, 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

Table 3.1: Details of the datasets used in computational tests 1 through 6 (full vertebrate mitochondrial genomes). The choice of computational Tests 1 to 6 (datasets within the sub-phylum Vertebrata) follows a decision-tree approach whereby, at each taxonomic level, one particular cluster is selected for further in-depth analysis. Note that, since the minimum and maximum cluster sizes are different for each test, the number of sequences selected from a given taxon can differ from one test to another. 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 of them belonged to the over-represented Order Cypriniformes (2,171 available sequences). Test 1 and 2 illustrate a different scenario: 500 of the total 7,876 available Actinopterygii sequences were used in Test 1, since this cluster size was sufficient to achieve high accuracy, while 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).

To construct a balanced dataset that captured as much diversity as possible, we considered all of the available species per family, according to GTDB (release 95) [223], and first excluded those species for which none of the sequences had a contig that was of the minimum length (herein, 150 kbp). For these computational tests, the cluster size was selected to be 400 sequences per cluster. This led to the following cases and corresponding experiment design choices:

1. The number of available species in a family is larger than 400

(*Rhodocacteriaceae* and *Burkholderiaceae*)

We selected 400 species at random and, for each of the selected species, we randomly selected one genome. Then we selected a random segment of at most 500 kbp from each genome. If there was only one genome available for a particular species, and the length of its largest contig was between 150 kbp and 500 kbp, the entire contig was selected. The selection strategy was designed to include as many families as possible in the final dataset.

2. The number of available species in a family is less than 400

- A. The total number of genomes in that family is larger than 400

(*Bacillaceae*, *Clostridiaceae*, *Staphylococcaceae*, *Enterobacteriaceae*)

We calculated the median M of the number of genomes per species, for that family, and randomly chose at most M genomes per species ($M = 2$ for *Bacillaceae*, $M = 1$ for *Clostridiaceae*, $M = 7$ for *Staphylococcaceae*, and $M = 2$ *Enterobacteriaceae*). We then selected a random segment of at most 500 kbp from each genome. If there were fewer than M genomes representing a species, and the length of their largest contig was between 150 kbp and 500 kbp, then the entire contig was selected. Since after this selection, more sequences were still required to reach the required 400 sequences per cluster, we chose the rest of the genomes at random from the family, without repetition, and selected a segment of 500 kbp from each such genome.

- B. The total number of genomes in the family is less than 400

(*Treponemataceae* and *Desulfovibrionaceae*)

For every genome, whenever possible, contigs were divided into successive segments of at most 500 kbp, that were added to a “pool”. If the length of a contig was between 150 kbp and 500 kbp, the entire contig was added to the pool. From this pool, 400 segments were selected for inclusion in the final dataset. This

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

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

selection strategy ensured that all of the species were represented and that the shortest segments were selected last.

A description of the final composition of each cluster is presented 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 that included only the segments from genomes in bacterial families from phylum Proteobacteria.

Viral genomes (Table 3.3): The third group of datasets consists of three different sets of viral genome sequences. The minimum and maximum cluster sizes were determined in a manner similar to that of the mitochondrial DNA datasets, i.e., for all the tests, the minimum cluster size was fixed to the size of the smallest cluster available, and the

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

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).

maximum cluster size was fixed manually to a number exceeding the minimum cluster size by no more than 20%, to ensure cluster balance.

To assess the performance of DeLUCS at the gene level, the dataset of Test 9 comprises 949 sequences of segment 6 of the *Influenza A virus* genome (encoding the neuraminidase protein, average length 1,409 bp). The sequences were downloaded from NCBI, [20], as belonging to subtypes H1N1, H2N2, H5N1, H7N3, and H7N9, as per [123]. The dataset of Test 10 comprises 1,633 *Dengue virus* full genome sequences downloaded from NCBI, [109], comprising four different subtypes. Finally, the dataset for Test 11 comprises 1,562 *Hepatitis B virus* (HBV) full genome sequences, downloaded from the Hepatitis Virus Database, [111], comprising six different subtypes. A description of the composition of each dataset is presented in Table 3.3.

3.4 Results

3.4.1 Qualitative and Quantitative DeLUCS Performance Results

We first used a qualitative measure to assess DeLUCS’s ability to group DNA sequences into meaningful clusters within the previously described datasets. Figure 3.4 illustrates

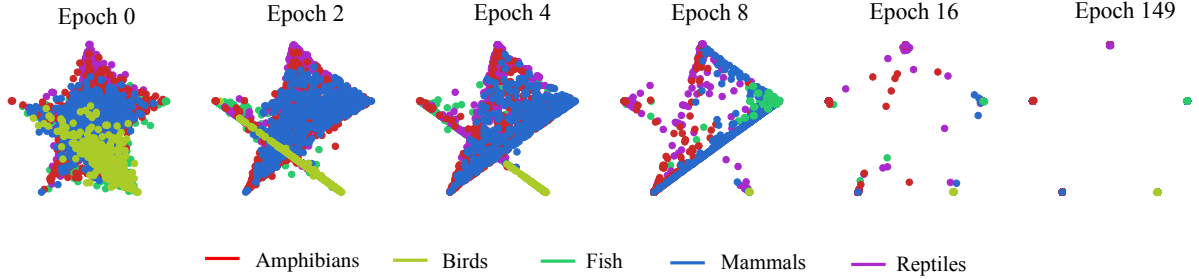


Figure 3.4: Learning process for the clustering of 2,500 vertebrate mtDNA full genomes into $c = 5$ different clusters, each corresponding to one corner of the pentagon. Each point represents a sequence, and its position indicates the probability that it is assigned to different classes. Hence, a point in the center has equal probabilities to be assigned to all 5 vertices/clusters, while a point located in a vertex has probability of 1 of belonging to that vertex/cluster. With successive epochs, the learning progresses until, at epoch 149, all sequences are correctly placed in their respective vertex/cluster. Note that points in the figure overlap if they have the same probability vector.

how, during the training stage, the ANN discovers meaningful clusters as the learning process evolves. The progress of DeLUCS is demonstrated in terms of the number of epochs (one epoch means a single pass of the ANN through the entire dataset, trying to discover new patterns). In Figure 3.4, the number of clusters is $c = 5$ and it corresponds to the number of vertices in the regular polygon, whereby each vertex represents a taxonomic label. The coordinates of each point are calculated using a convex combination of the components of the c -dimensional probability vector (see section Artificial Neural Network (ANN) Architecture), [139]. When the learning process starts, all the sequences are located at the center of the polygon, which means that each sequence is equally likely to be assigned any of the five cluster labels. As the learning process continues, the network starts assigning the sequences to clusters (with similar sequences being grouped together closer to their respective vertex/cluster), with higher and higher probability. Note that if two sequences are assigned the same probability vectors, their corresponding points in Figure 3.4 will overlap.

For the initial quantitative assessment of DeLUCS, we calculated its clustering accuracy. For the vertebrate mtDNA dataset, Table 3.4 shows that, at each taxonomic level, down to the genus level, our unsupervised deep learning algorithm outperforms the other two

Test #	Classification	Number of Mimics	Supervised	Unsupervised		
				GMM	K-Means++	DeLUCS
1	Subphylum	3	99%	72%	81%	93%
2	Class to Subclass	8	98%	92%	96%	100%
3	Subclass to Superorder	3	99%	70%	82%	85%
4	Superorder to Order	8	100%	68%	73%	94%
5	Order to Family	8	87%	66%	77%	79%
6	Family to Genus	8	80%	84%	87%	91%

Table 3.4: Clustering/classification accuracies for the mtDNA dataset in Table 3.1, Tests 1 to 6. For unsupervised learning, reported accuracy values are the average over 10 runs of the algorithm. For supervised learning, the accuracy is that of classifying the test set.

unsupervised clustering methods. Surprisingly, DeLUCS also outperforms a supervised learning algorithm that uses the same architecture in some tests (e.g., in Test 2, Class to Subclass), sometimes by a significant margin (e.g., in Test 6, Family to Genus, by 11%). One potential explanation for DeLUCS outperforming a supervised learning algorithm in scenarios with limited samples per cluster, such as in Tests 2 and 6, is that the supervised methods utilize only 70% of the data for training and perform testing with the remaining 30%. In contrast, unsupervised methods such as DeLUCS leverage the entirety of the available data for both training and testing, potentially enhancing their effectiveness in smaller datasets. Note that, in order to obtain a reasonable accuracy, the number m of mimics had to be increased for the tests/datasets with less than 150 sequences per cluster. The confusion matrices corresponding to the DeLUCS cluster assignments that maximize accuracy at each taxonomic level of the mtDNA dataset (Tests 1-6) are displayed in Figure 3.5.

To test the ability of DeLUCS to cluster bacterial genome segments, and achieve a direct accuracy comparison with other unsupervised learning methods, we first attempted clustering a dataset comprising genome segments, averaging 400,000 bp, from the eight different bacterial families considered in [123] (Table 3.5, Test 7). The DeLUCS clustering accuracy of 77% is 19% to 21% higher than the accuracies of the other two unsupervised learning methods (GMM 58%, and K -Means++ 56%). The relatively low clustering accuracies for all three unsupervised methods may be due in part to this dataset having a very heterogeneous evolutionary composition, with recent changes in the taxonomy of all eight bacterial families resulting during the reclassification and transition from NCBI to GTDB. In particular, *Bacillaceae*/*Staphylococcaceae* and *Clostridiaceae* (formerly all Firmicutes) are now split into two different phyla, and similarly *Enterobacteriaceae*/*Burkholderiaceae*/

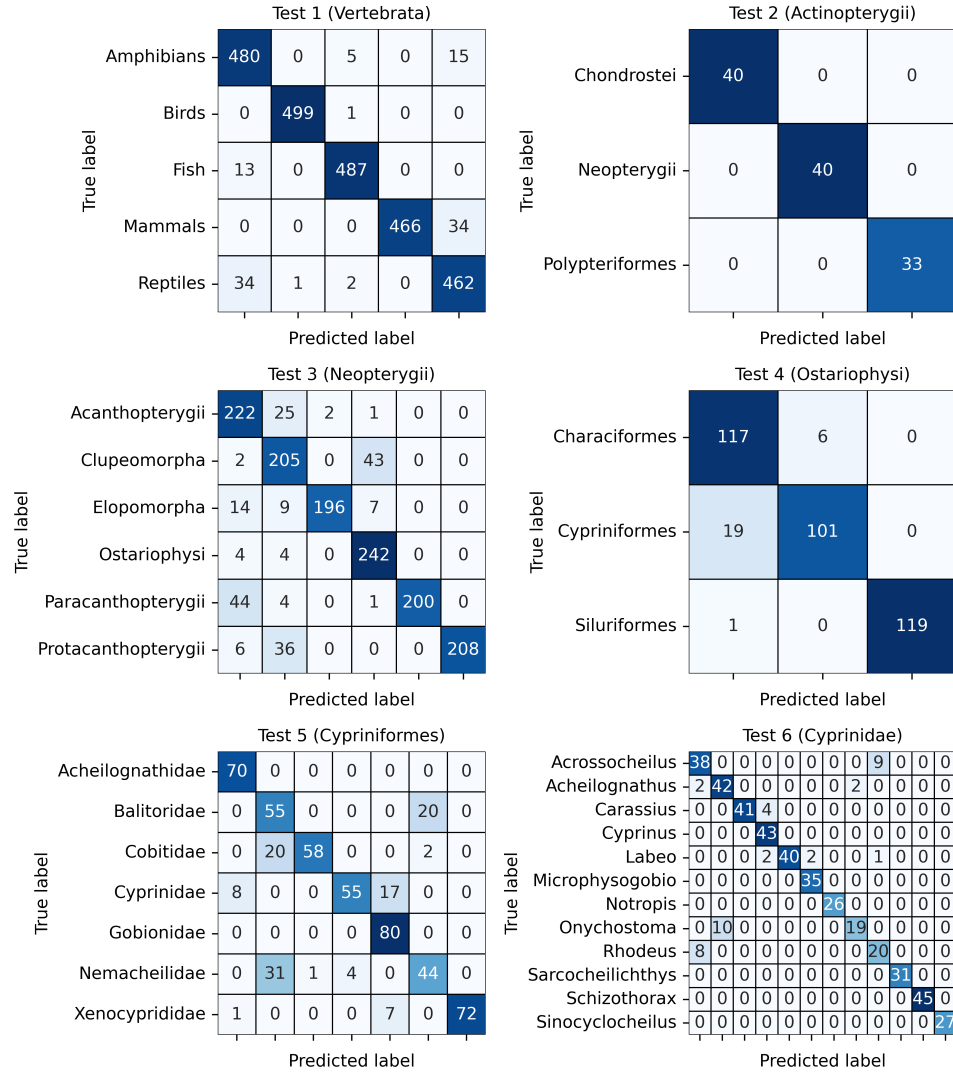


Figure 3.5: Confusion matrices for DeLUCS cluster assignments maximizing accuracy across taxonomic levels in the mtDNA dataset (Tests 1-6). Numeric cluster assignments as predicted labels are omitted for readability. Displayed results are from a single run of computational Tests 1-6.

Test #	Classification	Number of Mimics	Supervised	Unsupervised		
				GMM	K-Means++	DeLUCS
7	Bacteria into families	3	98%	58%	60%	77%
8	Proteobacteria into families	3	99%	50%	43%	90%

Table 3.5: Clustering/classification accuracy for the bacterial datasets in Table 3.2, Test 7 and 8. For unsupervised learning, reported accuracy values are the average over ten runs. For supervised learning, the reported accuracy is that of classifying the test set.

Rhodobacteraceae and *Desulfovibrionaceae* (formerly all Proteobacteria) are now split into two different phyla. The heterogeneity of the dataset makes clustering a challenging task, as the algorithm attempts to determine cluster labels simultaneously for both closely related families and distantly related families. The patterns observed in the confusion matrix (see Figure 3.6) support the hypothesis of heterogeneity in genetic distance between members of the dataset. Indeed, misclusterings between phyla are a minority, and most of the misclusterings occur among families that were previously placed within the same phylum, but are now placed in different phyla.

To verify that the lower clustering accuracy could be partially caused by the heterogeneity of the dataset, we next considered an intra-phylum clustering of a subset of this dataset, comprising only sequences belonging to phylum Proteobacteria (Table 3.5, Test 8). As predicted, the clustering accuracy of DeLUCS shows a significant increase, from 77% to 90%, now outperforming the other two unsupervised learning methods by 40% and 48% respectively. The majority of the misclustered genome segments in Test 8 belong to the family *Desulfovibrionaceae*. This may be partly due to this family having the shortest genome segments: The average *Desulfovibrionaceae* genome segment length is 359,337 bp, significantly shorter than the average genome segment length in Test 7, which is 433,613 bp.

Note that, in both Tests 7 and 8, we used the default value $m = 3$ for the number of mimic sequences. This is because the number of available sequences (genome segments of the required size) per cluster was large, and the clustering accuracy did not increase by increasing the number of mimic sequences.

To test the ability of DeLUCS to cluster closely related sequences, we next attempted clustering of three viral sequence datasets, *Influenza A* virus NA-encoding gene, *Dengue* virus complete genomes, and *Hepatitis B* virus complete genomes into clusters corresponding to virus subtypes. The clustering/classification accuracies presented in Table 3.6 show that all machine learning methods, supervised or unsupervised, perform well, in spite of the fact that the viral sequences are very similar. In Tests 9, 10, 11 the default value



Figure 3.6: Confusion matrices for DeLUCS cluster assignments maximizing accuracy in the bacterial genomic dataset (Tests 7-8). Numeric cluster assignments as predicted labels are omitted for readability. Displayed results are from a single run of computational Tests 7-8.

$m = 3$ for the number of mimic sequences was used, since this was sufficient to obtain near perfect clustering accuracy. The confusion matrices corresponding to the DeLUCS cluster assignments that maximize accuracy of all three viral genomic dataset (Tests 9-11) are displayed in Figure 3.7.

Test #	Classification	Number of Mimics	Supervised	Unsupervised		
				GMM	K-Means++	DeLUCS
9	Influenza A	3	100%	96%	99%	99%
10	Dengue	3	100%	100%	100%	100%
11	HBV	3	100%	100%	100%	100%

Table 3.6: Clustering/classification accuracy for the viral sequences datasets, Table 3.3, Tests 9, 10, and 11. For unsupervised learning, reported accuracy values are the average over ten runs. For supervised learning, the reported accuracy is that of classifying the test set.

Additionally, our analysis incorporates two other external validation measures, Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI), to compare DeLUCS with K -Means++ and GMM. These metrics statistically assess the clustering quality either by comparing the predicted cluster labels against the true labels, where available, or by evaluating the homogeneity and completeness of the clusters formed. Table 3.7 presents a comprehensive comparison of DeLUCS, K -Means++, and GMM across all eleven computational tests, utilizing the average values of NMI and ARI obtained over ten independent runs of the algorithms on each dataset. This ensures the robustness and reliability of our comparative analysis. The results in Table 3.7 confirms that DeLUCS consistently offers more reliable clustering outcomes compared to the other methods. Even in cases where DeLUCS does not lead, such as in Tests 6 (Cyprinidae) and 9 (Influenza-A), the differences in performance are relatively minor. This underscores DeLUCS’s effectiveness in handling diverse and complex datasets, making it a generally dependable choice in the realm of unsupervised learning algorithms, with only slight variations in performance across different contexts.

Together, these computational tests show that, in spite of the absence of any information external to the primary sequences, DeLUCS is capable of learning meaningful clusters from unlabelled, raw, and sometimes non-homologous DNA sequences. Moreover, DeLUCS outperforms classical unsupervised clustering algorithms that use k -mer counts as input features, often by a significant margin.

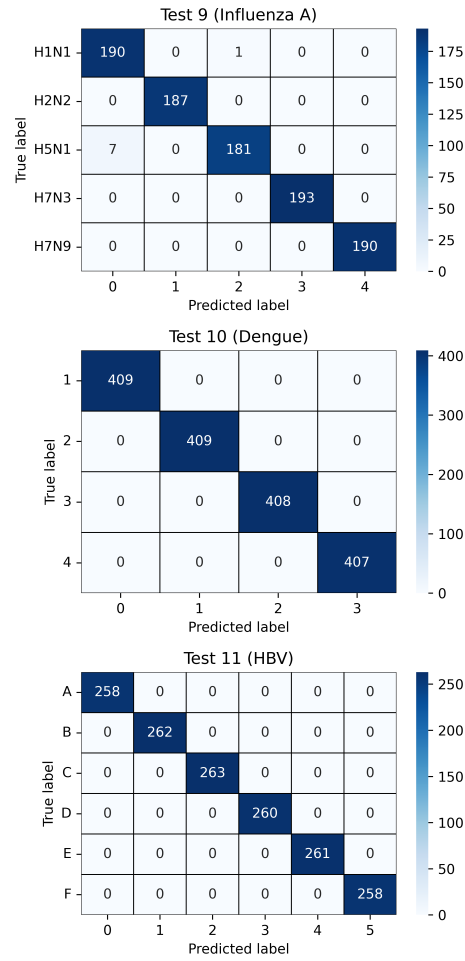


Figure 3.7: Confusion matrices for DeLUCS cluster assignments maximizing accuracy in the viral genomic dataset (Tests 9-11). Numeric cluster assignments as predicted labels are omitted for readability. Displayed results are from a single run of computational Tests 9-11.

Test #	Classification	Method	NMI	ARI
1	Vertebrata	DeLUCS	0.8980	0.9039
		<i>K</i> -Means++	0.5815	0.5155
		GMM	0.6510	0.6330
2	Actinopterygii	DeLUCS	1.0000	1.0000
		<i>K</i> -Means++	0.8617	0.8562
		GMM	0.8818	0.8798
3	Neopterygii	DeLUCS	0.7011	0.6522
		<i>K</i> -Means++	0.6859	0.5915
		GMM	0.5312	0.4408
4	Ostariophysi	DeLUCS	0.7345	0.7528
		<i>K</i> -Means++	0.5620	0.4867
		GMM	0.4765	0.3969
5	Cypriniformes	DeLUCS	0.7270	0.6008
		<i>K</i> -Means++	0.7264	0.6002
		GMM	0.6909	0.5654
6	Cyprinidae	<i>K</i> -Means++	0.9012	0.8366
		DeLUCS	0.8846	0.8211
		GMM	0.8999	0.8273
7	Bacteria	DeLUCS	0.7005	0.6117
		<i>K</i> -Means++	0.5854	0.4453
		GMM	0.6667	0.5492
8	Proteobacteria	DeLUCS	0.7723	0.7521
		<i>K</i> -Means++	0.2318	0.1579
		GMM	0.4147	0.3212
9	Influenza-A	<i>K</i> -Means++	0.9799	0.9844
		DeLUCS	0.9597	0.9680
		GMM	0.9799	0.9844
10	Dengue	DeLUCS	1.0000	1.0000
		<i>K</i> -Means++	0.9999	1.0000
		GMM	1.0000	1.0000
11	HBV	DeLUCS	1.0000	1.0000
		<i>K</i> -Means++	1.0000	1.0000
		GMM	0.8998	0.7729

Table 3.7: Comparison of DeLUCS, *K*-Means++, and GMM across all eleven computational tests, utilizing the average values of the external validation metrics NMI and ARI, calculated over ten independent run of each algorithm on each dataset. Boldface entries indicate top scores for each metric and test.

3.4.2 Methodology Results

Number of mimic sequences

The default number of mimic sequences was chosen to be $m = 3$, to correspond to the three different mechanisms used to generate mimic sequences from the original sequences: transitions, transversions, or combinations of both.

For the datasets in this study it was empirically determined that, if the cluster sizes are similar, and each cluster has more than 150 sequences, then the default value of $m = 3$ results in optimal performances (as is the case in Tests 1, 3, 7, 8, 9, 10 and 11).

For the datasets with less than 150 sequences per cluster, we tested the hypothesis that the clustering accuracy could be improved by generating more artificial mimic sequences per original DNA sequence. Our observations confirmed this hypothesis in Tests 2, 4, 5, and 6, where increasing the number of mimics to $m = 8$ resulted in improved clustering accuracies.

Note that, for the datasets with more than 150 DNA sequences per cluster (e.g., Tests 1, 3, 7, 8), increasing the number of mimics did not always result in increased clustering accuracy. Figure 3.8 illustrates the effect of increasing the number of mimic sequences on clustering accuracy for four different tests (Tests 2, 4 and 6, with fewer than 150 sequences/cluster, and Test 3 with more than 150 sequences/cluster). Note also that Tests 9, 10, 11 had more than 150 sequences per cluster, as well as near perfect clustering accuracies, and thus no increase in the number of mimic sequences was explored.

In general, the optimal number of mimics per sequence may depend on the number of available sequences per cluster, as well as on other particulars of the dataset being analyzed. The values of m mentioned above ($m = 3$, respectively $m = 8$) are suggested, but further optimization may be possible through a hyperparameter search.

Adding Gaussian Noise

We now provide further insight into the performance of our method by showing the relationship between the loss function and the DeLUCS clustering accuracy. The learning curves presented in Fig 3.9 illustrate the optimization process of a single ANN. During each epoch, the entire set of training sequences is utilized for optimizing the network parameters, and we observe an inverse correlation between the clustering accuracy and the unsupervised loss function for each epoch (Fig 3.9, top). The same graph illustrates that the ANN sometimes tends to converge to a suboptimal solution, and this is because the

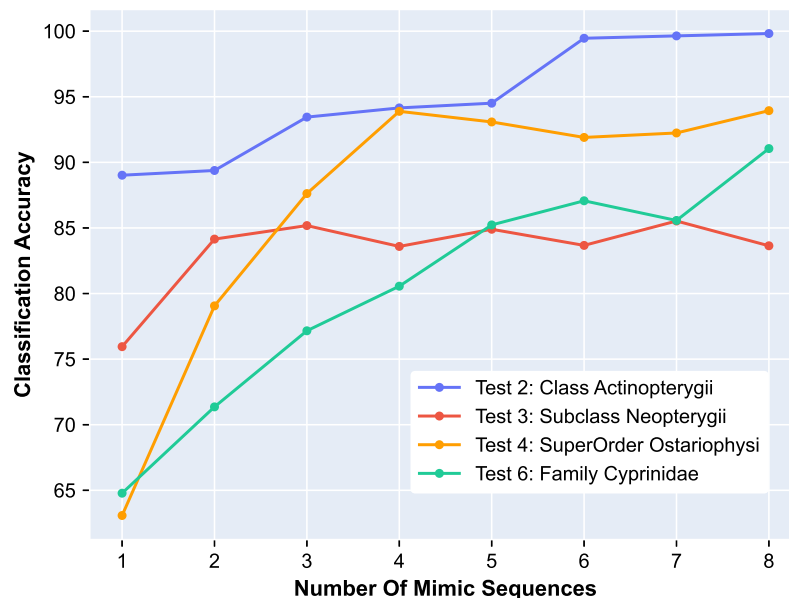


Figure 3.8: The effect of the number of mimic sequences on clustering accuracy. Blue - Class Actinopterygii (Test 2); Red -Subclass Neopterygii (Test 3); Yellow - Superorder Ostariophysi (Test 4); Green - Family Cyprinidae (Test 6). In Tests 2, 4, and 6, with fewer than 150 sequences per cluster, increasing the number of mimics per sequence results in a marked increase in clustering accuracy. In Test 3 (red), where the number of available sequences per cluster is sufficiently high (226 to 250), increasing the number of mimics to more than 3 does not result in an increase in the clustering accuracy.

mutual information might still be high for suboptimal solutions where relatively many related sequences are similarly misplaced (i.e., assigned to the wrong cluster, while still being close to each other within their subgroup).

To prevent the model from converging to suboptimal solutions, we added Gaussian noise to the network parameters every 50 epochs. We confirmed empirically that the introduction of noise is beneficial, as the accuracy increases after the introduction of noise every 50 epochs (Fig 3.9, bottom).

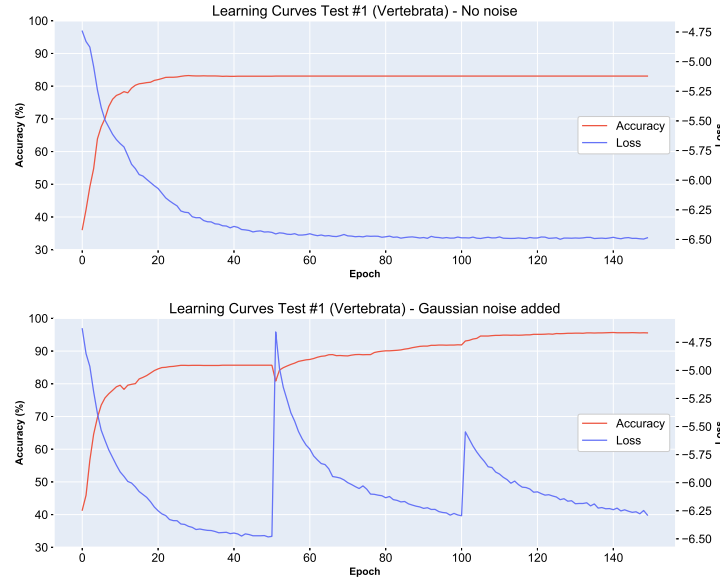


Figure 3.9: Learning curves for the training process of a single ANN in Step 2 of the method, for the dataset in Test 1 (vertebrate mtDNA genomes) with (top), and without (bottom) the addition of Gaussian noise to the parameters of the networks every 50 epochs. Each graph displays the relation between the unsupervised loss function that is being minimized and the overall accuracy of the method. The true labels of the sequences are only used to evaluate the accuracy at each epoch, but they do not influence the learning process. The graph illustrates the positive effect that the periodic addition of Gaussian noise has on the prevention of convergence to suboptimal solutions, and on the clustering accuracy (from $\approx 82\%$ top, to $\approx 96\%$, bottom).

Majority Voting for Variance Mitigation

The training process of an ANN is a randomized algorithm, with the randomness introduced by the initialization of the ANN, and the selection of random batches of training data in each epoch. The random selection of batches is beneficial for the marginalization that is performed to calculate the loss function. However, we also observed a high variance in the outcomes of training several independent copies of the ANN over the same dataset, likely due to the aforementioned randomness. To overcome this challenge, for each dataset, we independently trained ten ANNs, and integrated the obtained clustering results using a majority voting scheme. Fig 3.10 compares the accuracy of six tests that use an ensemble of 10 ANNs, with majority voting, compared to six tests that use a single ANN. Each test

was repeated ten times to compare the variance of the two approaches. As Fig 3.10 shows, majority voting not only reduced the variance of predictions but also improved the overall clustering accuracies.

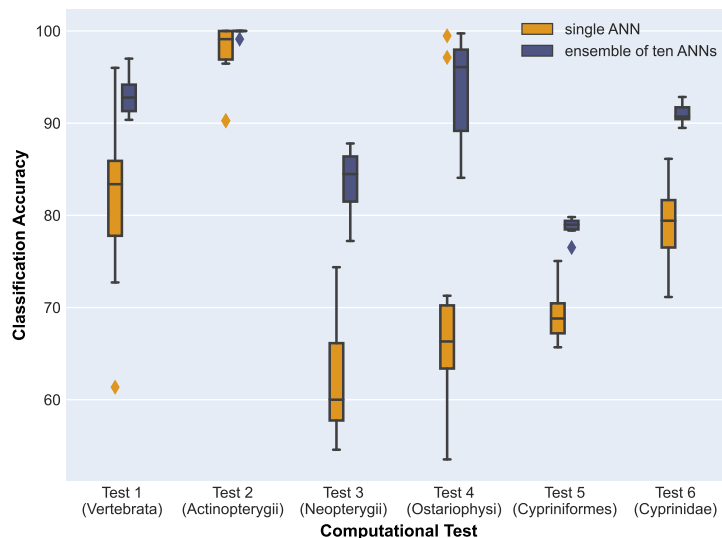


Figure 3.10: Comparison between training a single ANN (yellow), versus training an ensemble of 10 ANNs (black) and using majority voting to decide the final cluster label for each sequence (Tests 1-6). All tests were repeated 10 times. The minimum, first quartile (Q1), median, third quartile (Q3), and maxima of the distributions are shown. The diamonds outside the boxes represent outliers. Note that the result variance for single ANNs is larger than the result variance for the corresponding ensembles of 10 ANNs. In addition, the clustering accuracy is higher for the majority voting, in all cases.

3.5 Discussion

The use of FCGR as a numerical representation of genomic sequences, in combination with invariant information clustering (a deep unsupervised learning method for computer vision), allows DeLUCS to accurately cluster large datasets of genomic sequences. The largest computational experiment in this chapter comprises 3,200 randomly selected bacterial genome segments, totalling more than 1 billion bp, a dataset which is a full order of

magnitude larger than previous studies clustering genomic data [19, 131, 37, 123, 197, 6, 8].

In addition, DeLUCS achieves significantly higher clustering accuracies compared to other unsupervised machine learning clustering methods (K -Means++ and GMM), in comparable time. The running time of the whole pipeline of DeLUCS, considering both training and testing time, is less than 40 minutes for the largest bacterial dataset.

Note that for datasets comprising sequences with minimal homology (Tests 1-6, and 10-11) using alignment-based methods would be near impossible, while for datasets comprising non-homologous sequences (Tests 7, 8) using alignment-based methods would be impossible. Indeed, in the former case (vertebrate mitochondrial genomes, and virus genomes respectively), the time complexity of multiple-sequence alignment is prohibitive. In the latter case (bacterial genome segments averaging 400 kbp), the random sampling of genome segments from bacterial genomes almost guarantees non-homologous regions, thus making alignment impossible even if time complexity were not an issue. Test 9, of the NA-encoding gene of *Influenza A* virus, where an alignment-based method would be possible and probably successful, was included to showcase the applicability of our method to a variety of genomic data.

This study serves as a proof of concept of the ability of unsupervised deep learning to effectively cluster unlabelled raw DNA sequences. DeLUCS is a fully-automated method that could be used for successful clustering of datasets where traditional methods are not applicable, such as: large datasets of dissimilar and variable length raw DNA sequences, DNA sequences with incomplete or missing biological annotations, and DNA sequences for which taxonomic information or other sequence identifiers are insufficient or unavailable. Future work can address some of the current limitations of this approach, as described below.

First, in order to achieve accurate clustering, DeLUCS requires its training dataset to have balanced and well-represented clusters, with at least a minimum number of sequences per cluster. The minimum and maximum cluster size were determined individually for each computational test, based on the number of available sequences and the aforementioned requirements. As a result, many available sequences were not included in the training data, either due to the fact that under-represented clusters were discarded, or due to the fact that the excess sequences from over-represented clusters were not used. This limitation could be addressed by adjusting the loss function to handle unbalanced training databases, that is, datasets where clusters are of different sizes.

Second, we observe that the clustering accuracy of an ANN is heavily dependent on the initialization of the parameters, which is random for each run of the experiment. In other words, the clustering accuracy for a dataset can vary from run to run of an ANN, sometimes

by a large amount. On the other hand, one of the reasons behind DeLUCS’ successful clustering capability lies in this randomness in the parameter initialization. We attempted to address this trade-off between performance and reproducibility by training several copies of the same ANN, and using majority voting to determine the final cluster labels. The overall clustering accuracy stabilized and increased as a result, with the downside being that this approach increased the running time of the training step, since ten ANNs were sequentially trained. More time-efficient solutions, such as training the ANNs in parallel, would lead to a tenfold improvement of the running time.

Third, we observed that for the datasets with fewer than 150 sequences per cluster, an increase in the number of mimics resulted in clustering accuracy improvements (Tests 2, 4, 5, 6). However, this was not the case for some of the datasets with more than 150 sequences per cluster (Tests 1, 3, 7, 8). Some other tests, e.g., Tests 9, 10, 11, resulted in near perfect accuracy from the start and needed no further optimization. For experiments in this chapter, the number of mimic sequences per cluster was empirically determined to be optimal with respect to the cluster size. Further exploration is needed to determine the relationship between cluster size and the number of mimics per sequence, as well as to find other mechanisms to boost clustering accuracy for specific datasets, such as the use of Convolutional Neural Networks, which make full use of the two-dimensional aspect of the FCGR representation.

3.6 Conclusions

In this work we introduce DeLUCS, a novel unsupervised deep learning clustering method for DNA sequences. DeLUCS leverages deep learning to discover identity-relevant patterns in raw, primary DNA sequence data, without requiring homology, biological annotations, or the time-consuming and laborious step of defining taxonomic labels for the training data. DeLUCS obtains high clustering accuracies by the novel fusion of bioinformatics approaches with recent developments in the field of deep learning for computer vision, through the use of Chaos Game Representation of original and mimic sequences as input for invariant information clustering.

This is, to the best of our knowledge, the first effective alignment-free method that utilizes deep ANNs for unsupervised clustering of unlabelled DNA sequences. DeLUCS clusters diverse datasets comprising thousands of homology-free, identifier-free sequences (vertebrate mtDNA full genomes at various taxonomic levels; random segments of bacterial genomes into families; viral genomes into virus subtypes). Following the determination of the optimal mapping between the clusters learned by DeLUCS and true taxonomic groups

(via the Hungarian algorithm), we note that the resulting clustering is of the highest sensitivity at the species into subtypes level (99% to 100%), while in all other tests clustering accuracy is double-digit higher than that of classic unsupervised machine learning clustering methods.

Chapter 4

Machine Learning Methods for Taxonomic Classification of Emerging Astroviruses

This chapter is based on the paper “Leveraging machine learning for taxonomic classification of emerging astroviruses” [10]. It starts with Section 4.1, which introduces the use of machine learning techniques for classifying newly sequenced astrovirus genomes. This introduction also covers previous work in the field, critically reviewing both traditional and machine learning-based methods used for viral classification. It highlights the limitations of existing approaches and emphasizes the necessity for novel methodologies to manage the growing complexity and volume of viral genome data.

Section 4.2 details the materials and methods implemented, including the data collection and preprocessing, the machine learning models used, and the evaluation metrics for assessing model performance. This section provides a comprehensive explanation of the experimental setup and methodologies employed in the study.

Section 4.3 presents the outcomes of applying machine learning models to classify emerging astroviruses, including the elimination of recombinant sequences and the identification of potential subgenera within genera of this viral family. Lastly, Section 4.4 discusses the broader implications of the proposed method, highlighting its effectiveness in classifying emerging astroviruses and its adaptability for various taxonomic challenges. This section also addresses future directions for expanding viral taxonomy using machine learning approaches.

4.1 Introduction

Astroviruses are a genetically diverse virus family notably responsible for the second most common cause of nosocomial diarrhea following rotaviruses [199], as well as substantial economic losses in the poultry industry [148, 169]. Also, astrovirus infection has been associated with encephalitis and meningitis in immunocompromised patients [297] and astrovirus infection has been shown to be present in the brains of some mammals [43]. According to several studies [245, 199, 152], the prevalence of astroviruses among human populations ranges from 2% to 9% at any given time. In developing countries, this percentage can be significantly higher, affecting up to 30% of the population [63]. Infants between 3 and 8 months of age [268, 67, 137, 50, 116], and the elderly [18] along with immunocompromised patients are primarily infected [179, 220, 101]. Outbreaks have been reported for immunocompromised adults [200, 216, 134]. Astrovirus transmission occurs exclusively through the fecal-oral route, [167] with notable interspecies transmission [63]. Astrovirus genetic diversity is linked to the proposed replication via the class III PI3K pathway during autophagy [33] and genetic recombination associated with cross-species transmission through abiotic vectors such as drinking water, sewage, and other contaminated systems [1, 165]. The increasing interspecies transmission enhances the risk of extraintestinal infections in humans as reported in animal populations [234]. Genetic recombination elevates genetic diversity in the context of a concurrent multiplicity of infections for viruses of different genera given cross-species transmissions.

The International Committee on the Taxonomy of Viruses (ICTV) structures the family *Astroviridae* into two genera, Avastrovirus and Mamastrovirus [167], and determines taxonomic classification by defining species intragroups as strains with a minimum amino acid identity of 75% in the open reading frame 2 (ORF2) region. The two genera include many host-associated astroviruses and the number of known animal hosts has reached over 160, spanning 13 classes of organisms. Next-generation sequencing continues to achieve rapid detection of new astroviruses and the identification of new host species [327, 297], with a steadily increasing number of astrovirus genomes on National Center for Biotechnology Information (NCBI) awaiting classification at the genus level.

Urgency and need to refine the taxonomy of family *Astroviridae* is accelerated by reports of astrovirus recombination during concomitant infections [167, 222], including some involving *HAsVs* [297]. Also, avian and mammalian astrovirus species have been found in nonhuman primates [56], and inter-species crossover between humans and various animals, such as felines, cats, pigs, California sea lions, dogs, sheep, and turkeys have been identified [141, 56, 148, 195]. These observations complicate a taxonomic classification based solely on host species and nucleotide sequence identity. Moreover, with the emer-

gence of interspecies transmission, confusion has arisen in classifying astroviruses based on their origins. Due to classification at the species level based on sequence identity, some inter-cluster species of different genera, namely *HAsVs*, show more relatedness than those of the same intra-cluster genus [141]. This has led to a call for standardized methods of classification for family *Astroviridae*[56].

Recent years have seen a rapid growth in the volume of accessible genomic data, due to notable advancements in next-generation sequencing (NGS) technologies and a reduction in sequencing costs [263]. Consequently, there is an increasing demand for computationally efficient and scalable methods to handle large genomic datasets [270, 150]. Earlier attempts to tackle genomic classification/clustering problems can be categorized into two approaches: alignment-based and alignment-free methods. As discussed in previous chapters, the high computational cost and the reliance on sequence homology of alignment-based techniques make alignment-free methods a more suitable choice for addressing the virus classification problem. Consequently, a range of alignment-free classification methods, including ML-DSP [236], VGDC [80], along with others like Kameris [275] and PhaGCN2 [142], have been developed. Additionally, clustering techniques such as DeLUCS (introduced in Section 3), *i*DeLUCS [202], and MeShClust v3.0 [95] have been adapted for viral genomic datasets. Initial studies have shown that these methods are both effective and scalable, offering significant advantages over traditional alignment-based methods such as MUSCLE [74] and CLUSTALW [285].

This chapter presents a novel machine-learning classification method hereafter called the Three-Pronged Classification Method (3PCM) to classify astrovirus sequences that are as-yet unclassified. The method utilizes the primary sequence composition of the entire genome in the form of k -mer frequency vectors, where k is set to 6. In this method, $k = 6$ was empirically found to achieve the best balance between accuracy and computational complexity. An initial optional component was incorporated into 3PCM’s pipeline to detect potential recombinant sequences and exclude them from the analysis. This step aims to prevent any noise caused by inter-species crossover, which could otherwise confound machine-learning models. 3PCM consists of three main components: Prong 1 (a supervised classification method utilizing Quadratic SVM), Prong 2 (an unsupervised clustering technique based on K -Means++); and Prong 3 (the identification of host labels at the class level from relevant literature for the as-yet unclassified viral sequences). In this chapter, taxonomic classification was suggested when all three prongs of 3PCM agreed on a taxonomic label. When Prong 1 and Prong 2 concurred on a classification that differed from Prong 3, a taxonomic classification was suggested, subject to independent confirmation.

The design of 3PCM utilizes genome composition information from astrovirus sequences with known taxonomic labels to classify/cluster astrovirus sequences with mammalian or

avian hosts that are as-yet unclassified. Although the default output of 3PCM is based on the consensus prediction of the three prongs, the individual prongs can be used independently in cases where one or two prongs are not applicable or do not agree. For instance, Prong 1 is not suitable for classifying as-yet unclassified astrovirus sequences with non-mammalian non-avian hosts, due to the absence of ground truth labels which are necessary for training a supervised model. In such a situation, Prong 2 can be used in conjunction with Prong 3 to investigate the classification of the sequences. In this and other scenarios, other analyses such as genome composition analysis can be employed to validate the results.

The main contributions of this chapter are:

- Proposing genus labels (Mamastrovirus or Avastrovirus) for 191 as-yet unclassified astrovirus genome sequences for which the results of Prongs 1, 2, and 3 all agree.
- Suggesting genus labels (Mamastrovirus or Avastrovirus) for 8 additional as-yet unclassified astrovirus genome sequences, for which incompatibility was observed between the taxonomic label proposed by Prong 1 and Prong 2, and the host label provided by Prong 3. This may be due to cross-species transmission, and further investigation is needed to resolve the contrasting labels associated with these sequences.
- Providing evidence supporting the hypothesis of the existence of a human astrovirus subgenus of the genus Mamastrovirus and a goose astrovirus subgenus of the genus Avastrovirus, through the application of the proposed machine learning-based approach, enhanced by a Principal Component Analysis (PCA) of the sequence composition.

Overall, this multipronged machine learning approach provides a fast, reliable, and scalable prediction method of taxonomic labels, able to keep pace with emerging viruses and the exponential increase in the output of modern genome sequencing technologies.

4.2 Materials and Methods

This section details the datasets and methodologies used to classify RNA sequences from the *Astroviridae* family. It begins with a description of the datasets, outlining the source, selection criteria, and organization of the RNA sequences into various datasets for analysis. The methodological pipeline is introduced next, employing a three-pronged classification approach that incorporates both supervised and unsupervised machine learning techniques,

along with host label identification. An optional component of the proposed pipeline is discussed for excluding recombinant sequences to simplify the genetic analysis. The section further elaborates on each prong of the methodology, specifically focusing on the use of machine learning models to predict taxonomic labels based on genomic data.

The section further explores the optimal selection of k -mer sizes for the classification models, evaluating how different sizes affect the accuracy and computational efficiency of the taxonomic classification. The methodology is then tested against a closely related viral family, *Potyviridae*, to validate the robustness and applicability of the classification approach across similar genomic datasets. Finally, the last subsection outlines the computational infrastructure utilized for the analyses, specifying the software and hardware configurations employed in the processing and analysis of the astrovirus genomic datasets.

4.2.1 Datasets

The dataset used in this study consists of RNA sequences from the viral family, *Astroviridae* downloaded from the NCBI database. In the RNA sequence, Ns replaced all sequence characters other than adenine (A), cytosine (C), guanine (G), and uracil (U). The N in an RNA sequence means that any of the four bases could occupy the position in question. All sequences were uploaded to a folder in Genbank. These were then exported as a single multifasta file for further testing.

A total of 1,039 sequences from the family *Astroviridae* were downloaded from the NCBI database on July 27th, 2022. The sequences included in this study were between 5 kbps and 10 kbps in length. The host for each virus sequence was identified from the literature where a publication was available. In the absence of published records, the organism listed in the NCBI database submission was considered the host. We excluded 47 out of 1,039 sequences from our analysis due to the lack of information regarding the host of the virus given collection from sewage, rivers, and streams. Patent sequences were also excluded. Following the removal of these sequences, 992 sequences were used in this study as the primary dataset (Dataset 1, described in Table 4.1). Among the 992 sequences in this dataset, 308 are as-yet unclassified at the genus level. The individual host species were ascribed to their respective class and genus. The final dataset contains astrovirus genomes found in 13 unique host classes and 96 unique host genera.

In addition to Dataset 1, two other subsets of this dataset are used throughout the chapter, as described below. Dataset 2 (described in Table 4.2) comprises the 684 genomes in Dataset 1 that belong to either Avastrovirus or Mamastrovirus genus. Dataset 2 was used

Genus	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Avastrovirus	213	5,130	7,146	7,799
Mamastrovirus	471	5,003	6,395	7,353
Unknown	308	5,030	6,536	8,840
All/Average	992	5,003	6,600	8,840

Table 4.1: Description of Dataset 1 containing 992 viral genomes belonging to the family *Astroviridae*. Avastrovirus and Mamastrovirus are two genera currently designated within family *Astroviridae*. Of the 992 sequences, 308 lack a genus label as they have as-yet not been classified at this taxonomic level.

Genus	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Avastrovirus	213	5,130	7,146	7,799
Mamastrovirus	471	5,003	6,395	7,353
All/Average	684	5,003	6,629	7,799

Table 4.2: Description of Dataset 2, a subset of Dataset 1 consisting of sequences with available ground truth. There are 684 sequences in Dataset 2 which belong to one of the two established genera of this viral family, *Avastrovirus* and *Mamastrovirus*.

both as the main training dataset and as the dataset employed for determining different parameters of the proposed three-pronged classification method.

With a goal to predict genus level labels for the 308 as-yet unclassified sequences, we investigated the current information about these sequences, that is, their hosts. A summary of the host information for 308 as-yet unclassified astrovirus genomes is shown in Table 4.3. NCBI categorizes these sequences as unknown at the genus level of their taxonomy classification. For purposes described in the Results section (Section 4.3), Dataset 3 was created, consisting of 187 astrovirus genomes in Dataset 1 with unknown genus level labels and mammalian hosts, and 42 astrovirus genomes in Dataset 1 with unknown genus level labels and avian hosts, (see Table 4.4).

4.2.2 Overview of the Methodological Pipeline

We herein propose a three-pronged classification method (3PCM) for the taxonomic classification of emergent but as-yet unclassified astrovirus sequences.

Host	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Amphibia	11	5,038	6,746	7,723
Arachnid	1	6,943	6,943	6,943
Aves	42	5,084	6,806	8,417
Actinopterygii	20	5,518	6,748	7,538
Chondrichthyes	4	6,450	6,951	7,168
Bivalvia	4	6,842	7,059	7,436
Insecta	4	5,243	5,476	5,832
Crustacea	5	5,030	5,834	6,984
Magnoliopsida	3	7,755	7,782	7,836
Sarcopterygii	2	6,865	6,927	6,989
Mammalia	187	5,209	6,348	7,426
Cephalaspidomorphi	4	6,555	7,148	7,642
Reptilia	21	5,062	7,188	8,840
All/Average	308	5,030	6,536	8,840

Table 4.3: Distribution of the hosts of the 308 astrovirus genomes in Dataset 1 that are as-yet unclassified at the genus level. Host labels are at the class level.

Genus	Host	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Unknown	Mammalia	187	5,209	6,348	7,426
Unknown	Aves	42	5,084	6,806	8,417
All/Average	-	229	5,084	6,432	8,417

Table 4.4: Description of Dataset 3, consisting of 187 astrovirus genomes in Dataset 1 that are as-yet unclassified (unknown genus) and have a mammalian host, and 42 astrovirus genomes in Dataset 1 that are as-yet unclassified (unknown genus) and have an avian host.

An optional initial component of 3PCM aims to eliminate recombinant sequences from the training and testing datasets, for scenarios where their presence may confound the machine learning process. The main methodological pipeline consists of three prongs, as illustrated in Figure 4.1:

1. **Prong 1 (supervised learning):** training a classification model using the whole genome sequence for astroviruses with known taxonomic classification in the training phase and leveraging the trained predictive model to predict the labels of as-yet unclassified astroviruses in the testing phase.
2. **Prong 2 (unsupervised learning):** training a clustering model using the whole genomes of astroviruses in the training phase and using the trained predictive model to predict taxonomic labels of as-yet unclassified astroviruses in the testing phase. Taxonomic labels are not used in the training phase, therefore, this model is less vulnerable to inaccuracies of current taxonomic labels and classifications.
3. **Prong 3 (identifying host label):** identifying the host label at class level (e.g., Mammalia, Aves, etc.) from which the as-yet unclassified viral sample was obtained, by reviewing the relevant literature.

In the event that the predictions of all three prongs agree, taxonomic labels are proposed for as-yet unclassified sequences. If there is agreement only between Prong 1 and Prong 2, taxonomic labels are proposed with recommendations for further investigation.

4.2.3 An Optional Initial Component of Eliminating Recombinant Sequences

In part, the genetic diversity associated with astroviruses is associated with genetic recombination. The nature of the multiplicity of viral infection possible for astroviruses enables recombination between viruses of different taxa thereby potentially increasing genetic diversity significantly [303]. The resultant recombinant mosaic genome may present challenges to alignment-free (e.g. k -mer based) approaches based solely on the primary genome sequence. To assess the impact of recombinant genomes upon machine learning based classification approaches, the virus dataset was analyzed for identification of candidate recombinants and testing was carried out for the genome datasets with and without the candidate intergeneric recombinants.

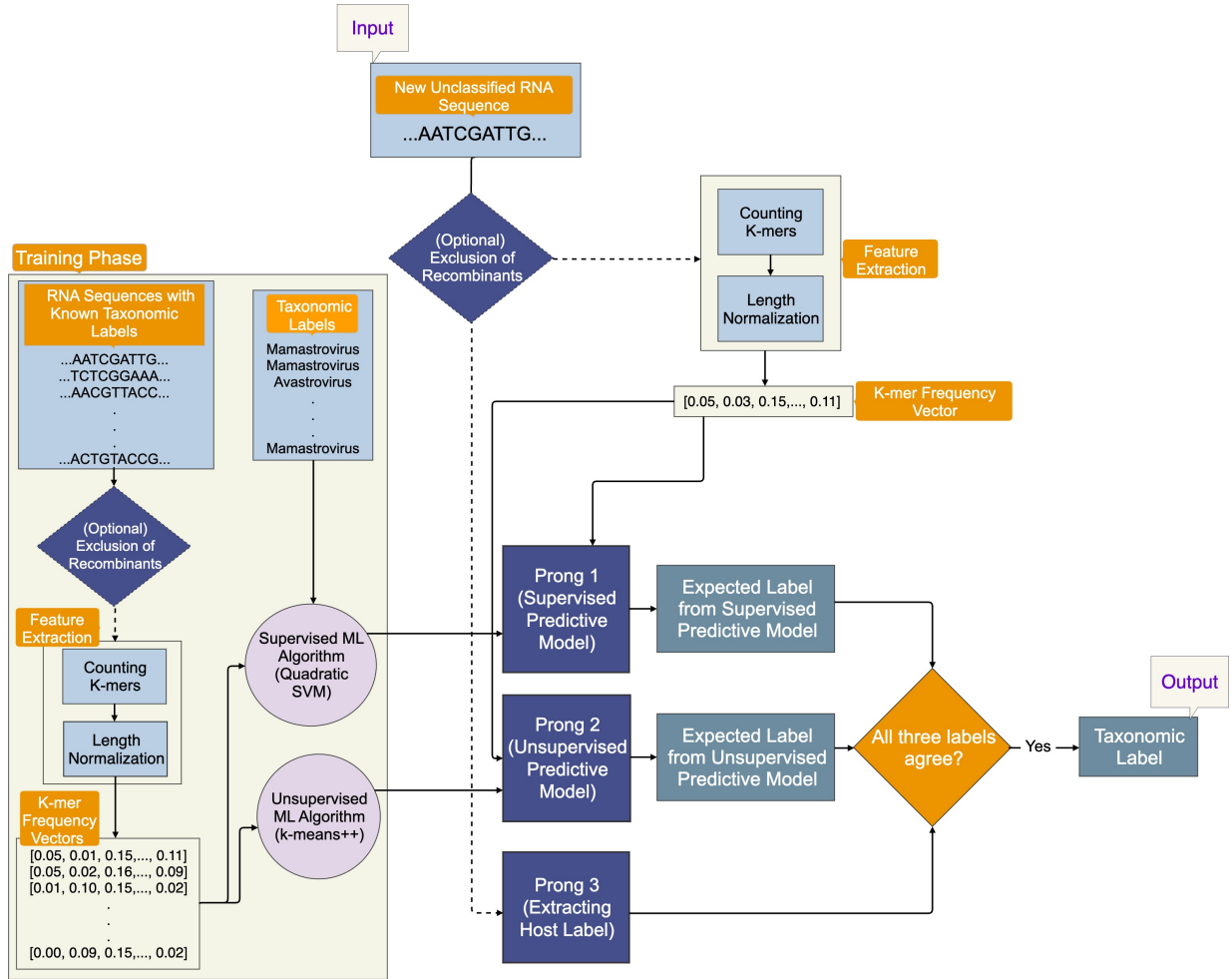


Figure 4.1: An overview of the proposed Three-Pronged Classification Method (3PCM). The input is an as-yet unclassified genomic sequence. As an initial elective step, a component within the pipeline gives the option to eliminate recombinant sequences from the dataset. Prong 1 employs a supervised predictive model trained on genomic sequences with known taxonomic labels. Prong 2 uses an unsupervised predictive model trained on the same genomic sequences, but it does not use their taxonomic labels for training. Prong 3 uses the host label of the input genomic sequence. All three prongs of 3PCM must agree in their prediction, in order to produce a suggested taxonomic label.

The multifasta file containing aligned astrovirus sequences was uploaded to the Recombination Detection Program *RDP4* [190]. To identify potential recombination events, the tools *RDP*, *GENECONV*, *BOOTSCAN*, *MaxChi*, *SiScan*, *CHIMAERA*, and *TOPAL* were run to examine all possible sequence triplets and identify a recombinant sequence, major parent and minor parent. When one or more programs identified recombination signals, *RDP4* utilized BURT as a hidden Markov model to identify where recombination breakpoint positions resided in the genome. *Phylpro*, *EEEP*, and *VisRD* techniques were then leveraged to propose a recombinant sequence. A csv file was exported with potential recombination events and tool results for further analysis. A region count matrix was calculated using *RDP4* to generate a heatmap of detected *Astroviridae* recombinant sequence loci.

Detection of potential triplets of a recombinant, major parent, and minor parent was carried out to identify candidate recombination events. 382 recombination events were identified by at least one of the seven tools employed. We focused on events identified by at least two tools, narrowing potential recombination events to 36. The genus *Sus* was the host involved in the most recombination events, followed by the genus *Homo*. In total when considering major and minor parents, 162 sequences were identified to be involved in recombination events, out of which 54 unique sequences were proposed to be recombinant sequences, equivalent to 5.4% of the population of 992 sequences analyzed. The wide host range of astroviruses, their ability to infect across species, frequent infections, and the chance for recombination could make classification difficult. We can identify potential recombinants and decide whether to include or exclude them from the classification process based on the amount and type of recombination.

4.2.4 Prong 1: Supervised Machine Learning

The first prong makes use of existing and established knowledge regarding astroviruses to train a supervised machine learning model based on the complete genomes of astroviruses with known taxonomic labels (Dataset 2) in the training phase. We then used the trained model to predict the labels of as-yet unclassified astroviruses (Dataset 3) during the testing phase.

The features used in the supervised learning methods in this study were the k -mer frequency vector of each astrovirus genome. The k -mers containing an N (not specifically one of the four bases in RNA) were not included. The value $k = 6$ was empirically found to achieve the best balance between accuracy and computational complexity for the datasets and computational experiments in this chapter. The performance of Prong 1 for values of

k in the range $[1, 9]$ in terms of classification accuracy and running time can be found in Section 4.2.6. To avoid potential effects of sequence length variation (astrovirus genome lengths range between 5,003 bp and 8,840 bp), the feature vectors were normalized to the interval $[0, 1]$ by dividing each vector by the length of the originating RNA sequence.

By utilizing a supervised model trained on the astrovirus sequences with known taxonomic labels, we were able to take advantage of established knowledge about this virus family. The supervised classification algorithms tested in this study are 10-Nearest Neighbours [12], Nearest Centroid Mean [286], Nearest Centroid Median [286], Logistic Regression [191], Linear Support Vector Machines (SVM) [55], SVM with quadratic polynomial kernel (Quadratic SVM), SVM with cubic polynomial kernel (Cubic SVM), SVM with stochastic gradient descent learning and linear kernel function (SGD), Decision Tree [29], Random Forest [28], AdaBoost [91], Gaussian Naive Bayes [45], Linear Discriminant Analysis (LDA) [108], Quadratic Discriminant Analysis (QDA) [108], and Multilayer Perceptron (MLP) [121] [154]. Python library scikit-learn’s implementations of the fifteen aforementioned classifiers [229] were used. Table 4.5 presents the performance results in terms of classification accuracy of these fifteen candidate algorithms as Prong 1 of 3PCM.

Only astrovirus RNA sequences in Dataset 2 with existing labels (described in Table 4.2) were used for training and testing for our initial experiment in order to select the most effective supervised classification algorithm as well as to demonstrate the effectiveness of Prong 1. In order to assess the accuracy of the classifiers, we used Stratified 10-Fold Cross-Validation [229][240]. In Stratified K -Fold Cross-Validation, the data is divided so that each fold has approximately the same proportion of instances of each class as the entire dataset. This is particularly important in the case of imbalanced datasets, where one class may have a much smaller representation than another (as is the case in Dataset 2, see Table 4.2). We conducted 10 independent experiments for each classifier, considering one dataset partition as testing data and nine as training data. We then calculated the average of ten accuracies from each experiment and reported the results in Table 4.5. Based on the results, most algorithms’ predictions match the true label (thirteen of fifteen algorithms achieved accuracy greater than 90%). Quadratic SVM and Cubic SVM were the most accurate algorithms for classifying astrovirus whole genomes by achieving an accuracy of 99.56%. Consequently, Quadratic SVM was selected as the classification algorithm in Prong 1 for the remainder of this chapter.

To further assess the performance of 3PCM’s Prong 1, we conducted a comparative analysis by benchmarking our outcomes against two leading alignment-free machine-learning genome classification methods suitable for viral classification: Machine Learning with Digital Signal Processing (ML-DSP) [236, 237] and the Viral Genome Deep Classifier (VGDC) [80] (for further information on these methods, see Section 2.8). The performance

Classifier	Classification Accuracy
Quadratic SVM	99.56%(SD=0.70%)
Cubic SVM	99.56%(SD=0.70%)
10-Nearest Neighbours	98.98% (SD=0.99%)
AdaBoost	98.98% (SD=0.98%)
Linear SVM	98.97% (SD=0.99%)
SGD	98.53% (SD=2.32%)
Logistic Regression	98.54% (SD=1.20%)
Random Forest	97.80% (SD=1.73%)
Decision Tree	97.80% (SD=1.72%)
Nearest Centroid Mean	97.51% (SD=2.19%)
Gaussian Naive Bayes	97.37%(SD=1.92%)
Nearest Centroid Median	96.64% (SD=2.49%)
LDA	90.04% (SD=11.65%)
Multilayer Perceptron	68.86%(SD=0.53%)
QDA	54.95% (SD=6.30%)

Table 4.5: Comparative classification accuracy of fifteen supervised classifier candidates for astrovirus whole genomes, utilizing stratified 10-fold cross-validation. The values represent average accuracies and standard deviations derived from ten different validation datasets, with Quadratic SVM and Cubic SVM models demonstrating the highest performance.

Classifier	Classification Accuracy
Prong 1 (Quadratic SVM)	99.56%
ML-DSP	99.00%
VGDC	95.88%

Table 4.6: Comparative classification accuracy of 3PCM’s Prong 1 against two state-of-the-art alignment-free machine-learning viral genome classification methods (ML-DSP, VGDC) using 10-fold cross-validation technique. The values in this table are averages over 10 different validation datasets.

comparison between these two methods and the proposed Prong 1 with Quadratic SVM is detailed in Table 4.6, based on experiments using Dataset 2 with 10-fold cross-validation. As seen in Table 4.6, Prong 1 achieves superior classification accuracy compared to both ML-DSP and VGDC by margins of 0.56% and 3.68%, respectively.

4.2.5 Prong 2: Unsupervised Machine Learning

Prong 2 of the proposed classification method is unsupervised clustering, which is agnostic to and independent of taxonomic labels and annotations. Taking into account the possibility that the current classification of viruses based solely on their host may be flawed or incomplete due to limited information, knowledge, or characterization, it was necessary to use an alternate approach that does not rely on current labels. The use of unsupervised clustering alongside Prong 1 (supervised learning) allowed for the flexible use of as-yet unclassified and unannotated astrovirus genomes in the training phase. Approximately one-third of astrovirus sequences are as-yet unclassified (308 out of 992) and cannot be used in supervised models as they lack “ground truth” taxonomic labels. The potential inclusion of these sequences in the clustering model allows for the examination of the hypothesis that the astrovirus family consists of more than two genera (Mamastrovirus and Avastrovirus), which was not possible in Prong 1.

In Prong 2, the same feature vectors as in Prong 1 (k -mer counts) were used. The performance of Prong 2 for different values of k in the range $[1, 9]$ in terms of classification accuracy and time can be found in Section 4.2.6. The value $k = 6$ was empirically found to achieve the best balance between accuracy and computational complexity for the datasets and computational experiments in this chapter. Furthermore, to find the most suitable clustering algorithm, we calculated and normalized the feature vectors and then tested three different clustering algorithms, K-Means++ [15], Gaussian Mixture Model (GMM) [65],

Clustering Algorithm	Clustering Accuracy	NMI [0,1]	ARI [-1,1]	Silhouette Coefficient [0,1]
<i>K-Means++</i>	85.54% (SD=7.37%)	0.42 (SD=0.08)	0.51 (SD=0.17)	0.08 (SD=0.00)
GMM	72.20%(SD=15.23%)	0.29 (SD=0.16)	0.26 (SD=0.27)	0.08 (SD=0.01)
Hierarchical Clustering	78.51%	0.27	0.26	0.07

Table 4.7: Performance of Prong 2 in clustering astrovirus whole genomes using *K-Means++*, GMM, and Hierarchical Clustering algorithms. Average and standard deviation (for clustering algorithms with variability) of clustering accuracy, Normalized Mutual Information (NMI) [0,1], Adjusted Rand Index (ARI) [-1,1], and silhouette coefficient [0,1] of 10 runs are reported as evaluation metrics. Higher values indicate better performance for all evaluation metrics.

and hierarchical clustering [32]. We used the Python library scikit-learn 1.3.0’s implementations of the three candidate clustering algorithms [229]. These algorithms were chosen due to their effectiveness in RNA classification [37, 131, 8, 6, 123, 197, 157].

These three clustering algorithms were compared by calculating the silhouette coefficient [250] as an internal evaluation metric. In addition, we calculated external evaluation metrics such as Normalized Mutual Information (NMI) [278], Adjusted Rand Index (ARI) [235], and clustering accuracy to further compare the five clustering algorithms. See Section 2.7 for details about the four evaluation metrics used in this section. In external evaluation metrics, the results of clustering are compared with some known ground truth or with a reference set of labels. Consequently, we focused on sequences that had already been established as belonging to the Mamastrovirus and Avastrovirus genera of the *Astroviridae* family (Dataset 2 described in Table 4.2) and used this information for calculating external evaluation metrics. The performance results of the clustering of the three clustering algorithm candidates measured in terms of the internal and external evaluation metrics are shown in Table 4.7.

Since *K-Means++* and GMM are non-deterministic algorithms and their outcomes can vary depending on the initialization parameters, we repeated both experiments ten times with different initializations. The results shown in Table 4.7 represent the average of these ten runs and standard deviations. As part of our analysis of the dataset, we tested several linkage methods for hierarchical clustering. Ward’s method provided the most coherent clustering and the highest silhouette coefficient score. Therefore, we have presented only the results of this linkage method in the table. We cut the hierarchical clustering tree at a suitable height in order to form exactly two clusters.

Among the three clustering algorithms, *K-Means++* performs the best in terms of all

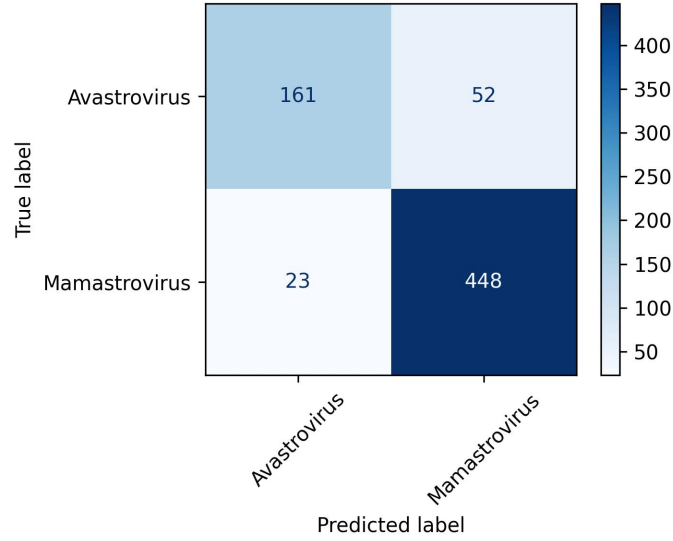


Figure 4.2: Confusion matrix for the clustering of Dataset 2, the astrovirus whole genomes with available taxonomic labels (dataset described in Table 4.2) into the Avastrovirus and Mamastrovirus genera using *K*-Means++ algorithm in Prong 2 of 3PCM.

four evaluation metrics. *K*-Means++ succeeded to achieve a clustering accuracy of 85.54% NMI of 0.42, ARI of 0.51, and silhouette coefficient of 0.08. Consequently, *K*-Means++ was selected as the clustering algorithm in Prong 2 for the remainder of this study. Figure 4.2 displays the confusion matrix obtained from clustering Dataset 2 using the *K*-Means++ algorithm in Prong 2 of 3PCM. According to the figure, 448 out of 471 Mamastroviruses and 161 out of 213 Avastroviruses were clustered correctly. Major misclustering occurred for 52 Avastroviruses that were grouped with the majority of Mamastroviruses. This may be due to the existence of additional genera or subgenera inside the family *Astroviridae*, which will be investigated in the Results section.

To further evaluate the performance of 3PCM’s Prong 2, we conducted a comparative analysis by contrasting our results with three state-of-the-art alignment-free machine-learning genome clustering methods suitable for viral sequences: DeLUCS (introduced in Section 3), its enhanced and interactive version (*i*DeLUCS) [202], and MeShClust v3.0 [95]. DeLUCS and *i*DeLUCS rely on deep learning to uncover patterns (genomic signatures) within raw, unlabeled primary RNA/DNA sequence data, while MeShClust employs a mean-shift algorithm on pairwise alignment-free identity scores. The performance com-

Clustering Algorithm	Clustering Accuracy	NMI [0,1]	ARI [-1,1]	Silhouette Coefficient [0,1]
Prong 2 (<i>K</i> -Means++)	85.54% (SD=7.37%)	0.42 (SD=0.08)	0.51 (SD=0.17)	0.08 (SD=0.00)
DeLUCS	66.40%(SD = 4.00%)	0.17 (SD = 0.05)	0.11 (SD = 0.06)	0.1 (SD = 0.01)
<i>i</i> DeLUCS	66.01%	0.11	0.10	0.04
MeShClust v3.0	65.20%	0.03	-0.03	0.005

Table 4.8: Performance of 3PCM’s Prong 2 against three state-of-the-art alignment-free machine-learning viral genome clustering methods (DeLUCS, *i*DeLUCS, and MeShClust v3.0) for clustering astrovirus whole genomes, with available taxonomic labels at the genera level (Dataset 2). Average and standard deviation (for clustering algorithms with variability) of clustering accuracy, Normalized Mutual Information (NMI) [0,1], Adjusted Rand Index (ARI) [-1,1], and silhouette coefficient [0,1] of 10 runs are reported as evaluation metrics. Higher values indicate better performance for all evaluation metrics.

parison of these three clustering methods and the proposed Prong 2 with *K*-Means++ is presented in Table 4.8, based on experiments utilizing Dataset 2 for both the testing and the training phases.

The computational experiments involving DeLUCS and *i*DeLUCS showed that increasing the mutation rate to $p_{ts} = 10^{-3}$ and $p_{tv} = 0.5 \times 10^{-3}$ (rather than the default values $p_{ts} = 10^{-4}$ and $p_{tv} = 0.5 \times 10^{-4}$) and using 9 *mimic sequences* (rather than the default value of 3) increased the accuracy of astrovirus genome clustering. Due to the variability in the results of *K*-Means++ and DeLUCS, the results reported in Table 4.8 are the average values and standard deviations over ten different runs. In contrast to the other methods, MeShClust v3.0 does not allow for the pre-setting of the number of clusters due to its density-based nature (see Section 2.8.3 for more details). As a result, multiple values were examined for the identity score threshold within the range of [0, 1], and the value of 0.4005 was selected as it was the one resulting in two clusters. Overall, as shown in Table 4.8, Prong 2 outperforms DeLUCS, *i*DeLUCS, and MeShClust in terms of both internal and external evaluation metrics, with a clustering accuracy that is approximately 22% higher than the other three methods.

4.2.6 Selection of *k*-mer Size

Optimizing *k*-mer size is crucial for the performance of 3PCM’s two components: Prong 1, which uses a quadratic SVM, and Prong 2, which employs *K*-Means++. The performance metrics considered here are classification/clustering accuracy and computational running

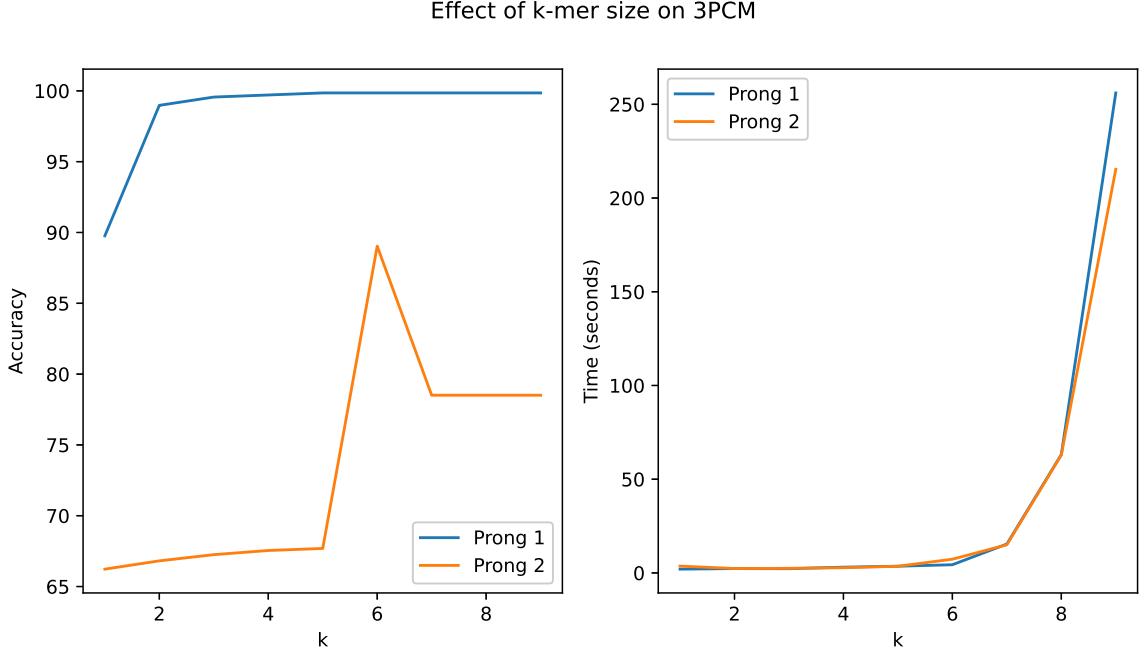


Figure 4.3: Effect of k -mer size in the performance of 3PCM’s Prong 1 and Prong 2 in terms of classification/clustering accuracy (left panel) and running time (right panel).

time, as shown in Figure 4.3. A range of k -mer sizes from 1 to 9 was tested to determine the most effective k for the proposed pipeline.

From the analysis, as Figure 4.3 illustrates, both prongs exhibit a significant variation in performance across different k -mer sizes. For Prong 1, accuracy peaks sharply at $k = 6$, maintaining a high level of performance with increasing k values. In contrast, Prong 2 shows a distinct decline in accuracy beyond $k = 6$. Moreover, the running time for both prongs escalates substantially as k increases, particularly notable after $k = 6$, suggesting a decrease in computational efficiency. Thus, $k = 6$ emerges as the optimal k -mer size, balancing accuracy and computational demands effectively for both prongs of the 3PCM pipeline.

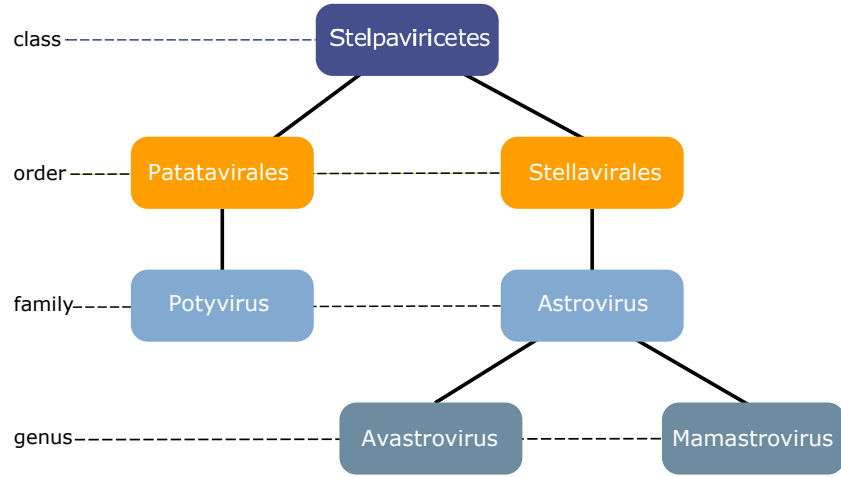


Figure 4.4: Taxonomic classification of class *Stelpaviricetes*.

4.2.7 Astrovirus Near-Neighbour Analysis (Potyvirus)

The proposed methodology was further tested by collecting 1,450 genomes of the closest viral family to astrovirus, namely potyvirus (family *Potyviridae*), within the Patatavirales order. This analysis also incorporated the Stellavirales order, to which astroviruses belong. Both orders fall under the Stelpaviricetes class, highlighting a structured taxonomic relationship crucial for our comparative study. The taxonomic classification of the *Stelpaviricetes* class, which includes both orders and their respective families and genera, is detailed in Figure 4.4.

A total of 1,450 sequences of the Family *Potyviridae* were downloaded from the NCBI database on August 30th, 2022. The sequences were filtered so that they are between 4 kbp and 10 kbp in length. This dataset comprises all 992 astrovirus sequences from Dataset 1 augmented with 1,450 potyvirus sequences and is described in Table 4.9.

Table 4.10 summarizes the results of the two prongs for both Prong 1 (classification accuracy) and Prong 2 (clustering accuracy, NMI, ARI, silhouette coefficient). The experiment of Prong 1 was performed using 10-fold cross-validation, and the experiment of Prong 2 was conducted ten times with varying initialization of the *K*-Means++ model. The exclusion of recombinants from this dataset was not performed due to the rarity of interspecific recombination within potyviruses. Additionally, no recombination is anticipated between families of potyvirus and astrovirus [94]. In Table 4.10, the values represent the average value of the ten runs. By computing all metrics used to evaluate classifi-

Genus	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Astrovirus	992	5,003	6,600	8,840
Potyvirus	1,450	4,005	8,676	9,585
All/Average	2,442	4,005	7,833	9,585

Table 4.9: Description of the dataset comprising 2,422 genomes of viruses belonging to the Class *Stelpaviricetes*. Astrovirus and potyvirus are the only two families within this viral class.

Method	Accuracy	NMI [0, 1]	ARI [-1, 1]	Silhouette Coefficient [0,1]
Prong 1	99.80% (SD=0.29%)	N/A	N/A	N/A
Prong 2	95.33% (SD=0.97%)	0.73 (SD=0.04)	0.82 (SD=0.03)	0.10 (SD=0.00)

Table 4.10: Performance of Prong 1 and Prong 2 applied to the classification/clustering of RNA sequences of the *Stelpaviricetes* class into family *Astroviridae* and family *Potyviridae*. Average and standard deviation of clustering accuracy, Normalized Mutual Information (NMI) [0,1], Adjusted Rand Index (ARI) [-1,1], and silhouette coefficient [0,1] of 10 runs are reported for Prong 2. Prong 3 host labels are considered the ground truth for calculating the external evaluation metrics (NMI, ARI, Classification/Clustering accuracy). Higher values indicate better performance for all evaluation metrics.

cation/clustering performance, we observe that both Prong 1 and Prong 2 can separate potyviruses and astroviruses sequences effectively.

4.2.8 Computational Setup

The laptop used for data collection and recombinant analysis was a Lenovo L-series ThinkPad with an Intel core i5 processor and 32 GB RAM. Datasets consisting of astroviruses and potyvirus in this study were retrieved from the National Center for Biotechnology Information (NCBI). Sequences were downloaded using the application Geneious Prime 2022.1 <https://www.geneious.com/> via the NCBI nucleotide database.

We empirically selected the hyperparameters of different classification/clustering algorithms that yielded the best performance during the training procedure. Both Prong 1 and Prong 2 of 3PCM are implemented in Python 3.10 and the source code, as well as all the datasets used in this study, are publicly available in the GitHub repository

<https://github.com/fatemehalipour/3PCM>. All of the tests were performed on Google Colab Pro environment (2 x Intel(R) Xeon(R) CPU @ 2.20GHz, 32 GB RAM) with NVIDIA A100 GPU.

4.3 Results

In this section, we showcase the outcomes achieved through the pipeline explained in the preceding section, Materials and Methods. First, the results of applying recombinant elimination to astrovirus sequences will be discussed. Following that, we will present the results of the novel classification method applied to as-yet unclassified astroviruses. Lastly, the existence of subgenera within Mamastrovirus and Avastrovirus genera, as suggested by our observations, will be explored.

4.3.1 Identification and Elimination of Recombinant Sequences

The optional component to eliminate recombinants was employed to examine Dataset 1, the primary dataset used in this study. As a result, 54 sequences (5.4% of the dataset) involved in interspecific recombination, associated with 34 recombination events, were identified. Notably, out of these 54 recombinations, 7 were intergeneric. Although the taxonomic classification task performed in this study is at the genus level, the presence of a negligible number (7) of intergeneric recombinations (which would yield noticeable variations in evaluation metrics) led to the decision to eliminate all 54 recombinants.

4.3.2 Classification of Unclassified Astroviruses with Mammal and Avian Hosts

Using 3PCM, we attempted to predict taxonomic classification for as-yet unclassified astrovirus sequences. The predictive models of 3PCM's Prong 1 (supervised) and Prong 2 (unsupervised) were trained on sequences with known taxonomic labels (Mamastrovirus or Avastrovirus), and later used to predict the genus of as-yet unclassified astrovirus sequences. In Prong 3, the class level host labels of the input as-yet unclassified astrovirus sequences were considered. In cases where all three labels agree, those labels were proposed as genus labels for the respective as-yet unclassified astrovirus genomes. When the taxonomic labels predicted by Prong 1 and Prong 2 agree, but they differ from the host label found by Prong 3, they are considered tentative and subject to further investigation.

For this analysis, only sequences with mammalian or avian hosts were investigated, since the two supervised and unsupervised predictive models were trained only on Mamastrovirus and Avastrovirus genomes. The sequences obtained from 11 other hosts were discarded, resulting in Dataset 3 comprising 229 astroviruses with mammalian or avian hosts, as the testing dataset (Table 4.4). Details of the distribution of hosts for the 308 as-yet unclassified astrovirus RNA sequences can be found in Table 4.3.

Figure 4.5 displays the confusion matrices resulting from the use of Prong 1 (supervised) and Prong 2 (unsupervised) for the classification/clustering datasets including (top), respectively excluding (bottom) recombinant sequences. Note that, out of 54 identified recombinant sequences, 41 were present in the training set (Dataset 2), while the remaining 13 belong to the testing set (Dataset 3). Table 4.11 summarizes the results of the evaluation metrics for both Prong 1 (classification accuracy) and Prong 2 (clustering accuracy, NMI, ARI, silhouette coefficient). As seen from Figure 4.5 and Table 4.11, the accuracy of both Prong 1 and Prong 2 increased slightly when the recombinant sequences were removed from the training and test sets (by 1.59% in the case of supervised learning, and by 1.48% in the case of unsupervised clustering). This observation indicates the potential negative impact that recombination events can have on machine learning-based classification/clustering approaches. While the impact may not be highly significant in this analysis, it is crucial to recognize that this may not hold true in all cases. The effect of this elimination process will correlate with the extent to which recombination contributes to noise in the classification. This, in turn, varies based on virus biology, frequency within the dataset, and the taxonomic level of classification.

As seen in Table 4.12, when both the training and test datasets include recombinant sequences, all three prongs agree on 83.41% (191 out of 229) of the sequences in the testing dataset. For an additional 3% (8 out of 229) of the sequences, the Prong 1 and Prong 2 predictions agree, but differ from the Prong 3 prediction. When recombinant sequences are excluded from both the training and testing datasets, all three prongs agree on predictions for 85.65% (185 out of 216) of the sequences in the testing dataset. Similarly, for an additional 3% (6 out of 216) of the sequences, Prong 1 and Prong 2 agree in their predictions, while Prong 3 disagrees. The NCBI accession IDs of the as-yet unclassified 191+8 astrovirus sequences when including recombinants and the as-yet unclassified 185+6 astrovirus sequences when excluding recombinants, together with the taxonomic labels (at the genus level) predicted by 3PCM can be found in Appendix A. The group of 185+6 sequences, where 3PCM led to a genus level classification when excluding recombinants, is a subset of 191+8 sequences for which a classification was proposed using 3PCM when including recombinants, with the exception of one sequence with accession ID *MT138006*, for which the classification prediction was only generated when recombinants were excluded

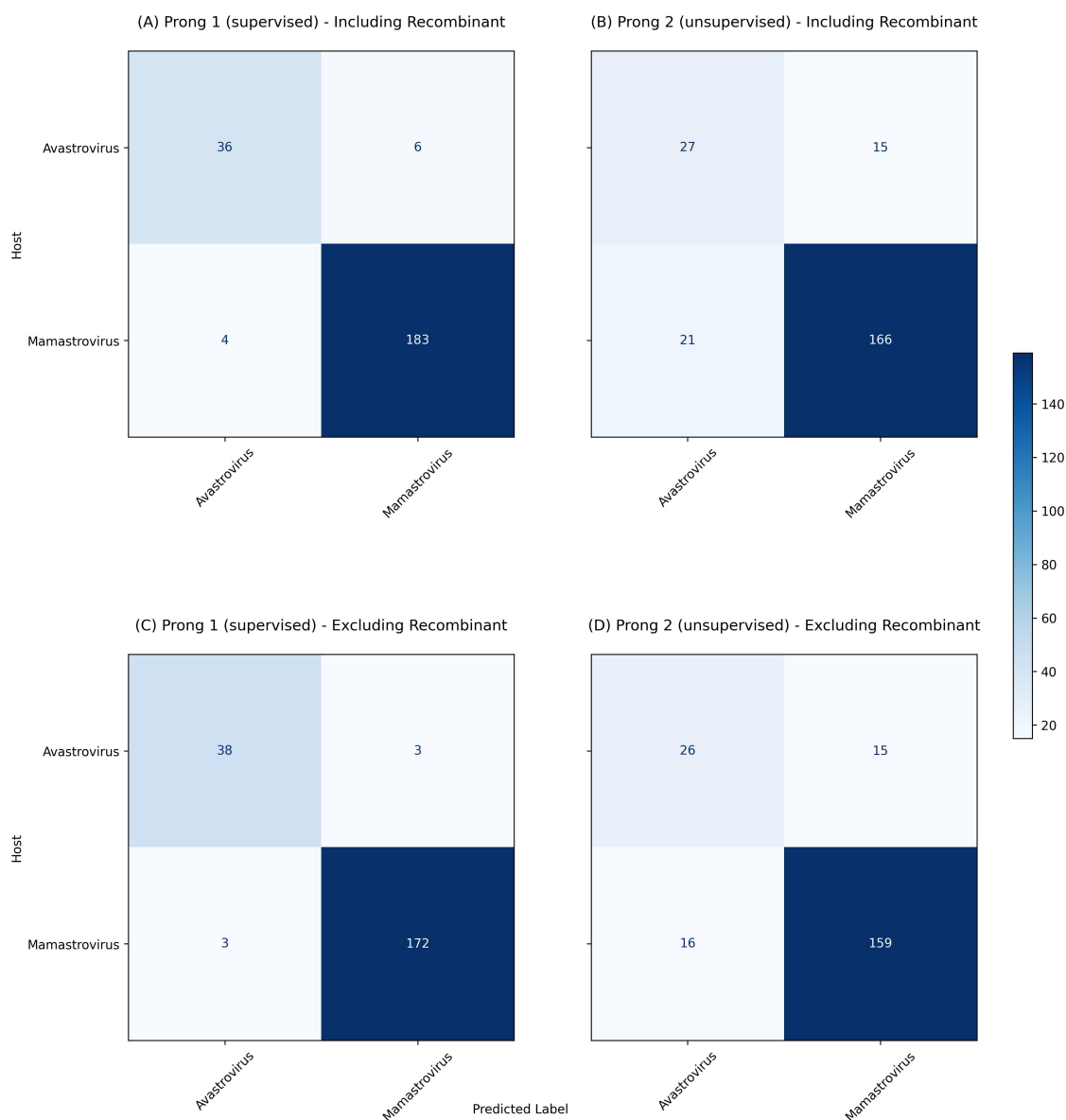


Figure 4.5: Top left: Confusion matrix for the classification of as-yet unclassified astroviruses including recombinants using Prong 1 (supervised). Top right: Confusion matrix for clustering as-yet unclassified astroviruses including recombinants using Prong 2 (unsupervised). Bottom left: Confusion matrix for the classification of as-yet unclassified astroviruses excluding recombinants using Prong 1 (supervised). Bottom right: Confusion matrix for clustering as-yet unclassified astroviruses excluding recombinants using Prong 2 (unsupervised). Prong 3, the host labels, was considered the ground truth for both confusion matrices.

Method	Recombinants	Accuracy	NMI [0, 1]	ARI [-1, 1]	Silhouette Coefficient [0,1]
Supervised (Quadratic SVM)	Included	95.63%	N/A	N/A	N/A
Supervised (Quadratic SVM)	Excluded	97.22%	N/A	N/A	N/A
Unsupervised (<i>K</i> -Means++)	Included	84.15%	0.22	0.39	0.06
Unsupervised (<i>K</i> -Means++)	Excluded	85.63%	0.25	0.43	0.06

Table 4.11: Performance of Prong 1 and Prong 2 applied to the classification/clustering unclassified astroviruses into Mamastroviruses and Avastroviruses when using datasets that include, respectively exclude recombinant sequences. Prong 3 host labels are considered the ground truth for calculating the external evaluation metrics (Normalized Mutual Information (NMI), Adjusted Rand Score (ARI), and Classification/Clustering Accuracy). Higher values indicate better performance for all evaluation metrics. Values within the brackets indicate the range of each evaluation metric.

Including Recombinants				
Prong 1 (supervised)	✓	✓	✓	✗
Prong 2 (unsupervised)	✓	✓	✗	✓
Prong 3 (host)	✓	✗	✓	✓
# Viral Genomes Agreed	191 (229)	8 (229)	28 (229)	2 (229)
Excluding Recombinants				
Prong 1 (supervised)	✓	✓	✓	✗
Prong 2 (unsupervised)	✓	✓	✗	✓
Prong 3 (host)	✓	✗	✓	✓
# Viral Genomes Agreed	185 (216)	6 (216)	25 (216)	0 (216)

Table 4.12: Number of as-yet unclassified viral genomes for which two or all three, of the prongs, agree in their genus label prediction. A checkmark indicates agreement between the prongs. For example, the first column represents the case when agreement of all three prongs occurs, the second column represents the case when Prong 1 and Prong 2 but not Prong 3 agree, etc. The numbers in parentheses represent the total size of the test dataset.

from the analysis.

Although both sets of results (with and without recombinants) are meaningful, we selected as the primary result of this chapter the 191+8 genus label predictions obtained from the analysis that includes recombinants. This decision was mainly influenced by the fact that the majority of the recombination events observed are intragenomic, and are thus not significant at genus level classification/clustering.

For the 191 sequences for which all three prongs agree, 26 are predicted to belong to genus *Avastrovirus*, and the remainder 165 are predicted to belong to genus *Mamastrovirus*. For these 191 sequences, the proposed genus labels are more certain than for the 8 sequences for which Prong 3’s host label disagreed with Prong 1 and Prong 2’s predictions. For the latter, three out of eight sequences were classified as *Avastrovirus* by Prong 1 and Prong 2 despite being obtained from mammals. These three sequences are *JN420353* (a California sea lion astrovirus [170]), *MH933754* (a human astrovirus [318]), and *NC_035758* (a human astrovirus [219]). Regarding the remaining five sequences, obtained from an avian host, the first two prongs predicted that they belonged to the *Mamastrovirus* genus. The five sequences are: *KP663426* [221], *MT138010* [267], *NC_027426* [219], *ON304005* [90], *MK096773* [82]. Further investigation is needed in order to determine the origin, and the spectrum of natural host species, of these eight sequences.

4.3.3 Classification of Unclassified Astroviruses with Hosts Other than Mammals and Avian Species

The genomes of as-yet unclassified astroviruses with hosts other than Mammals and Avians were examined to determine whether they all belong to one of the two genera Mamastrovirus and Avastrovirus or if more than two genera may exist within this family of viruses. Accordingly, the hosts of 288 as-yet unclassified astroviruses presented in Table 4.3 were reviewed. There are fewer than six sequences available for each of the hosts Arachnida, Chondrichthyes, Bivalvia, Insecta, Crustacea, Magnoliopsida, Sarcopterygii, and Cephalaspidomorphi; these few sequences do not capture the variation of the astrovirus genomes contributed by their hosts to the classification and clustering analysis. Although the minimum number of data points per cluster may differ depending on the complexity of the data and the clustering algorithm employed, a minimum of 10-20 data points per cluster is generally recommended to achieve reliable results in supervised and unsupervised machine learning [60]. Consequently, we decided to exclude the sequences belonging to these eight hosts and focus instead on the sequences belonging to the other five Animalia classes (Amphibia, Aves, Actinopterygii, Mammalia, and Reptilia).

In Section 4.3.2, two host classes, Mammalia and Aves, were investigated and we succeeded to label most of these viruses as Mamastroviruses or Avastroviruses. For the purpose of investigating other as-yet unclassified astroviruses, a new dataset was constructed consisting of 875 Mamastrovirus and Avastrovirus genomes labeled at the genus level (Dataset 2, comprising previously labeled mamastrovirus and Avastrovirus genomes, as well as the 191 genomes with genus labels predicted by 3PCM), augmented with 11, 20, and 21 as-yet unclassified astrovirus sequences with hosts belonging to the Amphibia, Actinopterygii, and Reptilia classes, respectively (see Table 4.13). This dataset was created to explore the separation of astroviruses with hosts other than Aves and Mammalia from each other and from Mamastroviruses and Avastroviruses. Prong 1 (supervised) was not applicable to this dataset, due to the absence of known astrovirus genomes with Amphibia, Actinopterygii, and Reptilia labels in the training set. The clustering results obtained by using Prong 2 (unsupervised) showed no clear separation between the as-yet unclassified astroviruses with hosts Amphibia, Actinopterygii, and Reptilia, nor was there any clear separation found between these genomes and Avastroviruses Mamastroviruses.

In the absence of any conclusive clustering results, Principal Component Analysis (PCA) was applied to this dataset, utilizing the first three principal components of the 6-mer counts for each genome, and preserving $\sim 20\%$ of the explained variance. As seen in Figure 4.6, PCA also does not result in an obvious separation of the astrovirus genomes with hosts Amphibia, Actinopterygii, and Reptilia from each other, or from the Mamastrovirus

Genus	Host	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.
Avastrovirus	-	239	5,130	7,113	8,417
Mamastrovirus	-	636	5,003	6,385	7,426
Unknown	Reptilia	21	5,062	7,188	8,840
Unknown	Actinopterygii	20	5,518	6,748	7,538
Unknown	Amphibia	11	5,038	6,746	7,723
All	-	927	5,003	6,599	8,840

Table 4.13: Description of the dataset including astrovirus genomes belonging to Avastrovirus and Mamastrovirus genera (previously taxonomically classified or the classification was proposed in Section 4.3.2) augmented with astrovirus genomes of unknown genus and having one of the Reptilia, Actinopterygii, or Amphibia hosts.

and Avastrovirus sequence clusters. This could potentially be due to a lack of availability of sufficiently many genomes with non-mammalian and non-avian hosts, which can negatively affect the efficacy of machine learning methods. These observations suggest that further data collection and possibly the development of more specialized computational methods may be necessary to achieve effective classification and separation of astrovirus genomes across diverse host species.

4.3.4 A Closer Look at Genera Mamastrovirus and Avastrovirus

From the initial visualization in Figure 4.6, distinct clusters—referred to as Cloud 1 and Cloud 2—were observed, representing specific subsets within the Mamastrovirus and Avastrovirus genera, respectively. Given the mounting evidence for different serotypes, clades, and genotypes associated with unique cross-species transmissions, different rates of evolution and intraspecific recombination for human astroviruses (*HAstV*) [70, 27, 230, 106] and goose astroviruses (*GoAstV*) [81, 331], we further investigated the sequences belonging to these two subgroups. Figure 4.7 displays a visualization of the 6-mer counts of the genomes in Dataset 2 (Mamastrovirus and Avastrovirus genomes with established labels), together with the 191 genomes with genus labels predicted by 3PCM with the use PCA. For visualization purposes, the first three principal components of the 6-mer counts for each genome are utilized, preserving $\sim 21\%$ of the explained variance.

The first step in this analysis was to extract information about the host species. The available 636 Mamastrovirus sequences with a host label at the species level were obtained from 69 different species. Of these, 362 sequences were obtained from the four most

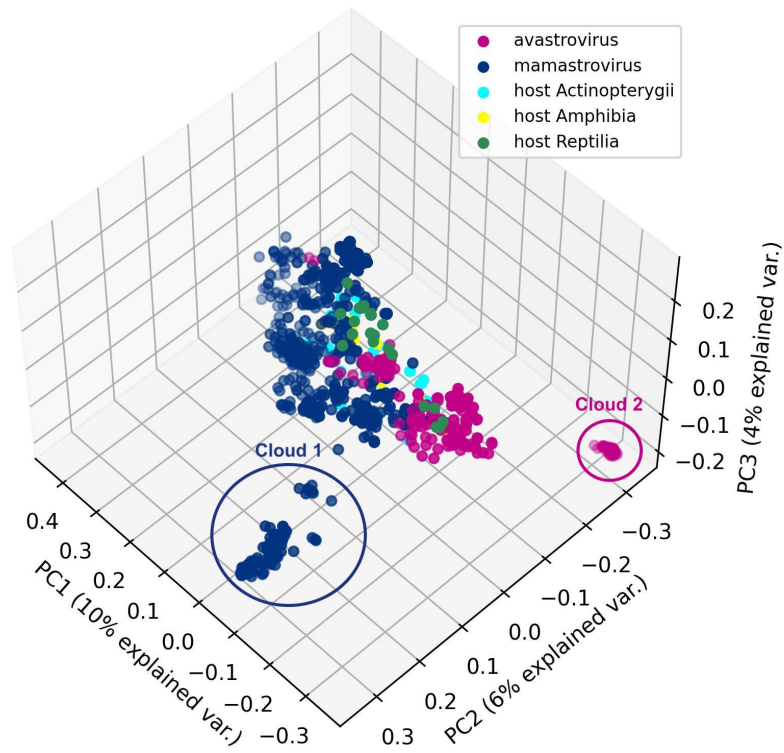


Figure 4.6: A comparison of as-yet unclassified sequences obtained from hosts Amphibia, Actinopterygii, and Reptilia with 875 astrovirus sequences belonging to Mamastrovirus or Avastrovirus. The visualization is based on the first three principal components of the 6-mer counts for the entire genome. In the figure, Cloud 1 represents a subset of Mamastroviruses, while Cloud 2 represents a subset of Avastroviruses.

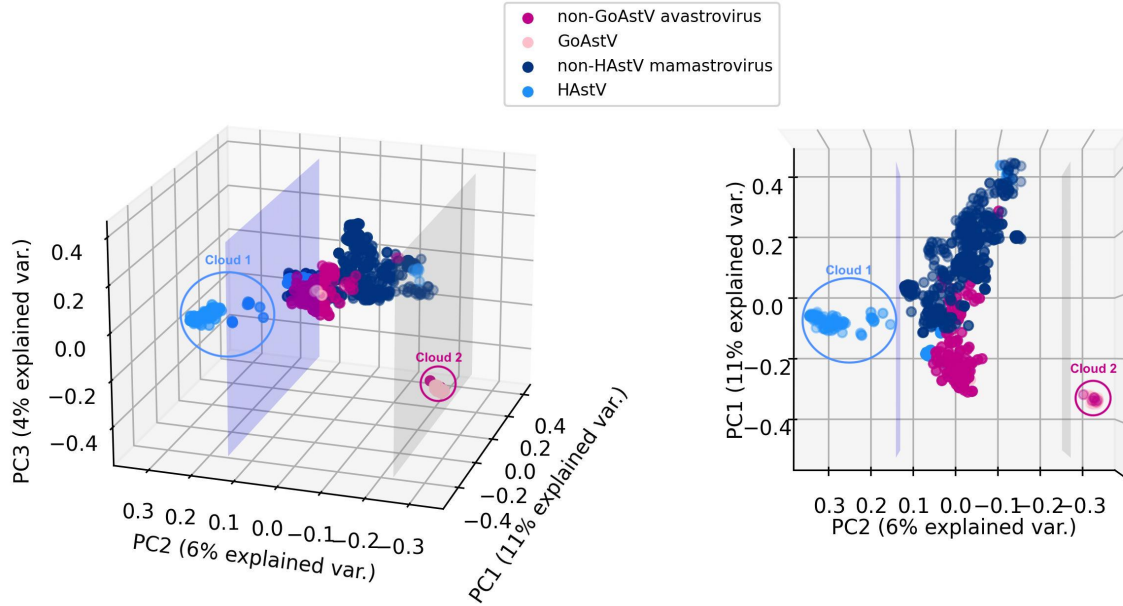


Figure 4.7: Two views of 3D PCA data visualizations of Mamastrovirus and Avastrovirus sequences k -mer frequencies: astrovirus sequences in Dataset 2 (known genus labels), together with the 191 astrovirus genomes with genus labels predicted by 3PCM. For comparison purposes, *HAstV* and *GoAstV* are highlighted with different colors compared to the rest of Mamastroviruses (*Non-HAstV* Mamastroviruses) respectively the rest of the Avastroviruses (*non-GoAstV* Avastroviruses). The lavender plane illustrates the separation between two possible subgenera of Mamastrovirus. The grey plane illustrates the separation between two possible subgenera of Avastrovirus. This visualization is based on the first three principal components of 6-mer counts for the entire genome. In this figure, Clouds 1 and 2 represent possible subgenera of *HAstV* (Cloud 1) within genus Mamastrovirus, and *GoAstV* (Cloud 2) within genus Avastrovirus.

Method	Accuracy	NMI [0, 1]	ARI [-1, 1]	Silhouette Coefficient [0,1]
Supervised (Quadratic SVM)	99.36%	N/A	N/A	N/A
Unsupervised (<i>K</i> -Means++)	80.88%	0.34	0.37	0.07

Table 4.14: Performance of Prong 1 and Prong 2 applied to the classification/clustering of Mamastrovirus sequences into *HAsV* and *Non-HAsV* Mamastrovirus. Prong 3 host labels are considered the ground truth for calculating the external evaluation metrics (Normalized Mutual Information (NMI), Adjusted Rand Score (ARI), and Classification/Clustering Accuracy). Higher values indicate better performance for all evaluation metrics. Values within the brackets indicate the range of each evaluation metric.

representative hosts, 105 from *Sus scrofa*, 44 from *Sus domesticus*, 153 from *Homo sapiens*, and 60 from *Bos taurus*.

As seen in Figure 4.7, a separating plane exists that separates Cloud 1 (111 Mamastrovirus genomes) from the rest of the Mamastrovirus genomes. A closer examination reveals that, while not all 163 *HAsV* (human host) sequences are in Cloud 1, all 111 sequences in Cloud 1 are *HAsV* sequences. This suggests that a *HAsV* subgenus exists within the genus Mamastrovirus. A comparison between Cloud 1 and the human Mamastrovirus sequences analyzed in [230] reveals that all the 91 MAsV-Sp7G3 human astrovirus sequences included in our analysis (not collected from sewage, not collected from an unknown host, etc.) are located in Cloud 1. Moreover, all 13 MAsV-Sp6G2 sequences and all 18 MAsV-Sp6G7 sequences analyzed in [230] are separated from Cloud 1 and located in the main cloud (the sequences located between the two separating planes). This suggests a correspondence between the Cloud 1 sequences and MAsV-Sp7G3 sequences.

To further examine this hypothesis, 3PCM was applied to the genome sequences of the Mamastrovirus genus (the aforementioned 636 Mamastrovirus genome sequences), with the labels being *HAsV* and *Non-HAsV* Mamastrovirus depending on the sequences' host species. The accuracies of applying Prong 1 (supervised) and Prong 2 (unsupervised) to this dataset, computed using Prong 3 (host labels) as the ground truth, are shown in Table 4.14. The high classification accuracy of Prong 1 (99.36%) and unsupervised clustering accuracy of Prong 2 (80.88%), provide additional evidence supporting the hypothesis of the existence of a *HAsV* subgenus of the genus Mamastrovirus. Using the available data, no separation is evident for the other hosts of Mamastroviruses.

When investigating the host species of genus *Avastrovirus*, we analyzed the 239 Avastroviruses with available host species labels, obtained from 28 different species. Of these, 135 sequences were obtained from the two most representative hosts, 64 from Goose and 71 from Chicken (*Gallus gallus*).

As seen in Figure 4.7, a separating plane exists that separates Cloud 2 (66 Avastrovirus genomes) from the rest of the Avastrovirus genomes. Further investigation of Cloud 2 revealed that 59 of its sequences have a *Goose* host (*GoAstV* sequences), amounting to 92% (59 out of 64) of the available *GoAstV* sequences. This suggests that a *GoAstV* subgenus may exist within the genus Avastrovirus. A comparison between the sequences in Cloud 2 and two established genotypes of *GoAstV* (see [331]) reveals that all 51 *GoAstV*-2(G2) goose astrovirus sequences analyzed in [331] are located in Cloud 2, and all 5 *GoAstV*-1(G1) sequences analyzed in [331] are separated from Cloud 2 and located in the main cloud (the sequences located between the two separating planes). This suggests that the observed separation of the Cloud 2 *GoAstV* sequences from the rest corresponds to the aforementioned two genotypes of the *GoAstV* virus analyzed in [331].

To further examine this hypothesis, 3PCM was applied to the genome sequences in this dataset, with the labels being *GoAstV* and *Non-GoAstV* Avastrovirus depending on the sequence’s host species. The classification/clustering accuracies of applying Prong 1 (supervised) and Prong 2 (unsupervised) to this dataset, computed using Prong 3 (host labels) as the ground truth, are shown in Table 4.15. The high classification accuracy of Prong 1 (94.96%), and unsupervised clustering accuracy of Prong 2 (94.98%), provide additional evidence supporting the hypothesis of the existence of a *GoAstV* subgenus of the genus Avastrovirus. According to the available data, no separation is apparent for the other hosts of Avastroviruses.

4.4 Discussion

In this chapter, we introduce the Three-Pronged Classification Method (3PCM), a novel approach that integrates both supervised and unsupervised machine learning paradigms, along with information about the originating species, to classify emerging astroviruses. The main objective of this study was to suggest a classification system for 229 as-yet unclassified astrovirus sequences acquired from avian and mammalian hosts. This approach was taken due to the limited number of available sequences and the lack of definitive information on other hosts. Out of the 229 as-yet unclassified sequences, the three-pronged classification yielded consistent predictions for 191 of them, indicating a very high level of reliability for the proposed classification. Furthermore, among the as-yet unclassified

Method	Accuracy	NMI [0, 1]	ARI [-1, 1]	Silhouette Coefficient [0,1]
Supervised (Quadratic SVM)	94.96%	N/A	N/A	N/A
Unsupervised (<i>K</i> -Means++)	94.98%	0.67	0.80	0.21

Table 4.15: Performance of Prong 1 and Prong 2 applied to the classification/clustering of Avastrovirus sequences into *GoAstV* and *Non-GoAstV* Avastrovirus. Prong 3 host labels are considered the ground truth for calculating the external evaluation metrics (Normalized Mutual Information (NMI), Adjusted Rand Score (ARI), and classification/clustering accuracy). Higher values indicate better performance for all evaluation metrics. Values within the brackets indicate the range of each evaluation metric.

sequences, for 8 additional sequences, the computational predictions of Prong 2 and Prong 2 coincided, but were different from the host information obtained by Prong 3. In light of numerous supporting evidence regarding the possibility of cross-species infection [221], the classification proposed by Prong 1 and Prong 2 takes precedence over Prong 3. With the investigation in literature, we were able to validate the taxonomic classification labels proposed by Prong 1 and Prong 2, confirming the existence of cross-species infection in both Mamastroviruses and Avastroviruses in these sequences.

3PCM’s versatility lies in its ability to employ each of the three prongs independently or in combination, providing a highly adaptable classification method suitable for various taxonomy tasks. The hypothesis of the existence of additional Avastrovirus and Mamastrovirus genera associated with astroviruses from Reptiles, Amphibians, and Actinopterygii hosts was explored using Prong 2 and Prong 3, which are not reliant on sequence labels. The results showed no clear separation between these three groups and Mamastrovirus and Avastroviruses, but the reason for this could be the limited availability of sequences. For a more comprehensive and accurate analysis, the identification of more Astrovirus sequences from hosts beyond avians and mammals is necessary.

Elimination of recombinant sequences is an optional step in the 3PCM pipeline. We identified 54 instances of interspecies recombination in Dataset 1, seven of which were intergeneric. The classification/clustering accuracy of Prongs 1 and 2 increased by 1-2% after the 54 recombinant candidates were removed. Since this study was concerned with genus level classification, of all recombination events only intergeneric recombinations could have an influence on the classification accuracy. As the number of intergeneric recombination events detected in this dataset was low (7 out of 992), the impact of the

removal of recombinant sequences on classification accuracies was expected to be negligible. Nonetheless, the option to include this step in the pipeline is essential, as its impact on classification accuracy may vary, depending on the level of taxonomic classification, dataset size, and the frequency and nature of genetic recombination in the virus genomes being classified. The expert user can decide whether or not to include this option, based on the specific virus biology (including the propensity for recombination and whether it is intra or intergeneric), the frequency of such recombinations in the dataset, and the level of classification (e.g., genus or species).

Since the recognition of *Astroviridae* as a family in 1993, this group of viruses infected over 140 hosts across the globe and is the second largest cause of gastroenteritis in humans. Due to the rapid expansion of infected hosts, frequent inter-species transmission, and genetic recombination, traditional classification based solely on a host may be insufficient. 3PCM is a novel machine-learning classification method that utilizes both virus-host and whole-genome composition. To enhance the effectiveness of 3PCM, an optional component was added to the pipeline that is responsible for eliminating recombinant sequences. Following the classification of the as-yet unclassified astroviruses using Prong 1 (supervised classification) and Prong 2 (unsupervised clustering), the NCBI host labels were used as possible ground truth to classify/cluster astrovirus whole genome sequences, with an accuracy of 95.63%, and 84.15% respectively. From this classification method, we propose 26 avian-host-derived sequences and 165 mammalian host-derived sequences be added to Avastrovirus and Mamastrovirus genera, respectively. A taxonomic classification was also proposed for eight additional as-yet unclassified astrovirus sequences, which are not aligned with the host species of the sequences and may be capable of transmitting across species. As anticipated, the need for a rapid and multipronged approach for astrovirus classification continues to grow (the number of unclassified genome sequences grew from 308 in July 2022 to 429 in September 2023). The 3PCM pipeline is available for ongoing classification of newly added sequences and its power increases with the informative increase in the modeling.

Furthermore, 3PCM was used to investigate the hypothesis of the existence of subgenera *GoAstV* and *HAstV* within Avastrovirus and Mamastrovirus respectively. Using 3PCM for classification/clustering of the genus Mamastrovirus into *HAstV* and *Non-HAstV* Mamastrovirus, accuracies of 99.36%, and 80.88% for Prong 1 and Prong 2 were achieved. Furthermore, the accuracy of 94.96% and 94.98% were achieved when Prong 1 and Prong 2 were used for classification/clustering of the genus Avastrovirus into *GoAstV* and *Non-GoAstV*, respectively. The results of these two experiments were further verified by an investigation of the difference in the genome composition of the subgroups. As a result, we propose that each of these subgroups is a distinct sub-genus.

Chapter 5

CGRclust—Advanced Clustering of DNA Sequences Using Twin Contrastive Learning

The content in this chapter is derived from the paper “CGRclust: Chaos Game Representation for twin contrastive clustering of unlabelled DNA sequences” [9]. This chapter begins with Section 5.1, which sets the stage with an introduction that outlines the innovative adaptation of twin clustering methodology originally designed for computer vision to the challenges of genomic data classification.

Following the introduction, Section 5.2 discusses related work in the field, providing a critical review of existing methods, both supervised and unsupervised, that have been previously applied to genomic data. This review identifies gaps in current methodologies that CGRclust aims to address, emphasizing the advantages of using convolutional neural networks and twin contrastive learning in the clustering of DNA sequences.

Section 5.3 describes the materials and methods implemented in CGRclust, including the selection and preparation of datasets, the architectural details of the neural network models used, and the twin contrastive learning framework. This section provides a comprehensive breakdown of the methods and experimental setups employed in the study.

The results of applying the CGRclust methodology to various genomic datasets are presented in Section 5.4. This section provides qualitative and quantitative evaluations and comparative analyses against established methods such as DeLUCS, which was introduced in Chapter 3.

Section 5.5 discusses the broader implications of the findings, exploring the potential of CGRclust in fields beyond taxonomic clustering, such as RNA analysis and virus-host genomic signature similarity. It also addresses the challenges encountered during this work and the strengths and limitations of the proposed approach. The chapter concludes with Section 5.6, summarizing the key contributions and findings of the CGRclust study.

5.1 Introduction

Building on the insights from Chapter 4, where DeLUCS was found to underperform in clustering astrovirus sequences compared to K -Means++, this chapter introduces CGRclust as an advanced approach to overcome these limitations. While DeLUCS addressed the principal shortcomings of alignment-based methods by utilizing a flattened form of Frequency Chaos Game Representation (FCGR), CGRclust further advances this methodology. It fully exploits the two-dimensional nature of FCGR images through convolutional neural networks (CNNs), and integrates *twin contrastive learning* (TCL) [319, 173] to enhance the clustering process, designed to effectively manage diverse genomic datasets, including astroviruses.

As discussed previously in Section 2.3.4, FCGR transforms one-dimensional DNA sequences into two-dimensional images, serving as effective genomic signatures for distinguishing genetic relationships between DNA sequences [147]. This DNA representation has been identified as effective genomic signatures for distinguishing between closely and distantly related organisms without the need for sequence alignment [147, 333, 184]. Both CGR and FCGR have not only addressed the scalability issues inherent in alignment-based methods [333, 184, 68, 120] but have also facilitated alignment-free genomic comparisons using advanced digital signal processing [122, 176] and machine learning techniques [247, 329, 283, 105, 77, 264].

This chapter presents CGRclust, a DNA sequence clustering method designed to identify discriminative features of DNA sequences, using two-dimensional FCGR images as the input to convolutional neural networks (CNNs), to fully leverage the information in this powerful DNA encoding. The clustering process in CGRclust employs TCL [319, 173], a method proven effective in clustering images and text, which optimizes two contrastive learning objectives simultaneously—one at the instance-level and another at the cluster-level.

CGRclust’s accuracy was evaluated across twenty-five datasets against DeLUCS [201] (introduced in Chapter 3), *i*DeLUCS [202], and MeShClust v3.0 [95]. Its clustering capabilities were tested on 2,688 mtDNA genomes of Cypriniformes, as well as five different viral

genome datasets, including astroviruses, dengue virus, hepatitis C virus, and HIV-1. Furthermore, CGRclust was also assessed using mtDNA genomes from insects, protists, and fungi [202], along with synthetic DNA sequences [95]. All DNA sequences were unlabelled, with their taxonomic labels used solely for post-hoc accuracy evaluation.

In summary, CGRclust is a novel, scalable, alignment-free clustering method that uses FCGR images and CNNs, for twin contrastive clustering of unlabelled primary DNA sequences. The main contributions of this chapter are:

- Being, to best of our knowledge, the first application of twin contrastive learning to the clustering of DNA sequences, without requiring sequence homology, sequence labels, or sequence-length similarity.
- Highly accurate clustering of a current dataset of 2,688 unlabelled fish mtDNA assemblies (order Cypriniformes). Clustering was performed at four different taxonomic levels, and CGRclust consistently achieved accuracy greater than 81.70% at all levels. This was either higher than, or comparable to, clustering accuracies of the other state-of-the-art clustering methods (DeLUCS, *i*DeLUCS, MeShClust v3.0).
- Highly accurate clustering of several current datasets of unlabelled viral whole genomes (Astroviridae family into genera; dengue, HCV, HIV-1 species into virus subtypes), with accuracies ranging from 81.77% to 100% (no classification error), surpassing the other state-of-the-art clustering methods.
- Effective handling of challenging cases, such as unbalanced data, and scenarios with a high number of clusters and a small number of samples per cluster.
- Superior or competitive accuracies compared to state-of-the-art methods on their benchmark datasets of unlabelled DNA sequences, e.g., 73.56% for insect mtDNA, 85.50% for protist mtDNA, and 97.10% for fungi mtDNA. Furthermore, CGRclust consistently exceeded 92.26% accuracy in clustering unlabelled synthetic DNA sequences of different lengths and identities.

5.2 Related Work

FCGR’s ability to convert variable-length sequences into fixed-size dimensions is a key capability for machine learning, especially in DNA classification using convolutional neural networks (CNNs) [166]. In a study by Rizzo et al. [247], CNNs outperformed Support

Vector Machines (SVMs) in classifying FCGR images of bacterial 16S gene sequences for both full-length sequences and 500 bp fragments. Moreover, Safoury et al. [252] achieved an accuracy of 87% with a simple CNN in classifying FCGRs of 660 DNA sequences across eleven genomic datasets. In 2023, Avila et al. effectively classified SARS-CoV-2 DNA sequences into eleven clades using FCGR and CNNs [16], achieving 96.29% accuracy utilizing a ResNet50 neural network [114] and outperforming Covidex [38], a random forest-based clade assignment tool. Hammad et al. [104] introduced a hybrid CGR-based approach for detecting COVID-19, analyzing both whole and partial genome sequences of 7,951 human coronaviruses using AlexNet, Lasso algorithm, and KNN classifier. In spite of the effectiveness of these DNA classification methods, their reliance on labelled data is a significant limitation, which highlights the urgent need for unsupervised algorithms that can perform well without the need of DNA sequence labels.

To address this gap, dense neural networks have been used in conjunction with flattened FCGR for the unsupervised clustering of DNA sequences in large, diverse datasets (up to 9,027 genomes) across different taxonomic levels and genetic distances in DeLUCS, which was introduced in Chapter 3. Building on this foundation, *i*DeLUCS enhanced DeLUCS's clustering outcomes by implementing a novel contrastive loss function and substituting majority voting with a more robust, information theory-based clustering ensemble [202]. However, in both DeLUCS and *i*DeLUCS, the neural networks first flattened each two-dimensional FCGR into a one-dimensional k -mer frequency vector. As a result, the features of two-dimensional FCGR images were not fully exploited in these methods. Another approach to clustering unlabelled DNA sequences, MeShClust v3.0 [95], used the mean-shift algorithm for generating pairwise identity scores without alignment (See Section 2.8.3 for more details about MeShClust v3.0 methodology). In spite of this progress, in Chapter 4 it was shown that DeLUCS, *i*DeLUCS, and MeShClust v3.0 underperform in clustering astrovirus sequences when compared to K -Means++, even though they were previously validated on other viral datasets. These limitations highlight the need for more robust methods like CGRclust that can handle the diverse complexities of genomic data more effectively.

5.3 Materials and Methods

This section starts with a description of the datasets utilized in this study. This is followed by an overview of the proposed computational pipeline for contrastive clustering of DNA sequences in CGRclust. Next, a description of the data augmentation strategies used for FCGR of DNA sequences (generation of *mimic sequences*) is presented, serving as the

initial component of CGRclust’s pipeline. Afterward, the core concept of twin contrastive learning, details about the backbone model, and the majority voting scheme adapted to clustering FCGRs of DNA sequences are described. Lastly, details of implementation and testing are provided.

5.3.1 Datasets

To comprehensively evaluate the performance of CGRclust in clustering DNA sequences, we strategically selected four groups of datasets, comprising diverse genomic data both real and synthetic. The selection rationale was driven by the need to assess the clustering method across different levels of taxonomy with different degrees of relatedness, genomic conservation, and evolutionary dynamics. The Group 1 dataset includes mitochondrial DNA of fish, while the Group 2 dataset includes whole viral genomes. Additionally, to facilitate direct comparisons with established methodologies, we incorporated datasets previously analyzed in [202] and [95] (Group 3 and Group 4 datasets, respectively). In the following, the test labels are integrally linked with datasets and are used for ease of reference when discussing results.

The Group 1 dataset comprised complete mitochondrial DNA (mtDNA) sequences of Cypriniformes (an order of ray-finned fish). This dataset was retrieved from the National Center for Biotechnology Information (NCBI) on Jan 30, 2024, with a filter selecting mtDNA sequences of length between 4 kbp and 25 kbp. Following the removal of “partial” and “unverified” genomes, 2,688 complete mitochondrial genomes of Cypriniformes were collected. At each taxonomic level, the cluster with the highest number of sequences was selected for the lower taxonomic level clustering task. Due to significant variability and imbalance in the number of available sequences across the four taxonomic levels, sequences from clusters with fewer than 50 sequences were discarded. To address the imbalance, in the first three computational tests (Tests 1-3), we established a threshold based on the minimum number of sequences available in a cluster and randomly selected an equivalent number of sequences from the other clusters. Balancing the clusters was not needed in Test 4, as the dataset was already evenly distributed. Table 5.1 summarizes the dataset details for the Group 1 dataset (Cypriniformes mtDNA). The selection of this group of datasets was motivated by the conservative nature of mtDNA, which is predominantly coding and thus provides a stable framework for assessing clustering methodologies at multiple taxonomic levels. The uniformity of high conservation over the mtDNA genome compared to the regional variation in sequence conservation of the nuclear genome, coupled with its wide use in phylogenetic studies [84], makes mtDNA data an ideal candidate for initial clustering evaluations.

Test	Taxonomic clustering (No. of seq. per cluster)	No. of seq.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)
1	Order into Suborder: Cypriniformes into Catostomoidei, Cobitoidei, <u>Cyprinoidei</u> (166 each)	498	15,655	16,610	17,859
2	Suborder into Family: Cyprinoidei into Acheilognathidae, <u>Cyprinidae</u> , Danionidae, Gobionidae, Leuciscidae, Xenocyprididae (105 each)	630	15,616	16,620	18,220
3	Family into Subfamily: Cyprinidae into Acrossocheilinae, <u>Cyprininae</u> , Labeoninae, Poropuntiinae, Schizopygopsinae, Schizothoracinae, Smiliogastrinae, Torinae (56 each)	448	15,609	16,603	17,426
4	Subfamily into Genus: Cyprininae into Carassioides (74), Cyprinus (77), Sinocyclocheilus (62)	213	16,562	16,592	17,426

Table 5.1: Details of the Group 1 dataset in Tests 1 through 4 (Cypriniformes full mitochondrial genomes). The underlined font indicates the cluster with the highest number of sequences, which was subsequently selected for clustering at a lower taxonomic level. The subfamily Cyprininae includes, e.g., the goldfish (*Carassius auratus*).

To further demonstrate the effectiveness of CGRclust, we assessed its performance across five viral whole genome datasets in the Group 2 dataset: an updated version of the virus family Astroviridae genomes analyzed in [10] (Test 5) and its balanced version (Test 6), an updated version of whole genomes of dengue virus (Test 7), hepatitis C virus (HCV) (Test 8), and human immunodeficiency virus 1 (HIV-1) (Test 9) previously classified in [275] with supervised machine learning methods. Table 5.2 outlines the details of the Group 2 dataset. In Test 5, 1,089 complete astrovirus genomes were collected, for taxonomic clustering of the sequences from family to genus level. Test 6 uses a cluster-balanced variant of the astrovirus dataset to address the initial label imbalance, thereby ensuring that the clustering results are not skewed by this disparity. All astrovirus sequences were downloaded from NCBI on April 4, 2024, with a filter selecting genome lengths ranging between 5 kbp and 10 kbp. Furthermore, we addressed the clustering of viral sequences at a lower level of species to subtypes in Tests 7-9. This categorizing, which is called viral subtyping is crucial for understanding intraspecific variation, tracking

epidemiological trends, and developing targeted treatments or vaccines. The dengue virus sequences used in Test 7 were obtained from <https://www.ncbi.nlm.nih.gov/genomes/VirusVariation/Database/nph-select.cgi?taxid=12637> using the query parameters “Nucleotide”, “Full-length sequences only”, and “Collapse identical sequences”, resulting in a dataset of 5,868 sequences. Following cluster balancing, we obtained a dengue dataset comprising 1,628 dengue virus whole genomes spanning four distinct subtypes. The HCV genomes utilized in Test 8 were sourced from the LANL sequence database, accessible at <https://hcv.lanl.gov/components/sequence/HCV/search/searchi.html>, with the query settings “Excluding recombinants”, “Excluding ‘no genotype’”, “Genomic region: complete genome”, and “Excluding problematic”, resulting in 3,612 whole HCV genomes. After removing clusters with less than 100 sequences and balancing the dataset, we obtained 950 full HCV genomes spanning five different subtypes. Finally, the HIV-1 genomes in Test 9 were retrieved from the Los Alamos (LANL) sequence database, accessible at <https://www.hiv.lanl.gov/components/sequence/HIV/search/search.html> with query parameters “virus: HIV-1, genomic region: complete genome, excluding problematic,” which resulted in a dataset comprising 20,525 HIV-1 full genomes. We then removed HIV-1 subtypes with fewer than 100 sequences and balanced the remaining subtypes, thus obtaining a dataset comprising 13,000 HIV-1 whole genome sequences spanning 13 subtypes. The three viral datasets used in Tests 7-9 were downloaded on April 1, 2024. Viral genomes are characterized by higher mutation rates and greater evolutionary diversity compared to the mtDNA, presenting distinct challenges for clustering algorithms. This variability tests the robustness and adaptability of CGRclust under conditions of rapid genomic changes and diverse evolutionary pressures.

Next, we evaluated the performance of CGRclust on three core datasets used in [202] (Group 3 dataset: mtDNA of Insects, Protists, and Fungi), as well as 12 synthetic DNA datasets analyzed in [95] (Group 4 dataset: synthetic sequences). Including these datasets allowed for direct comparisons with existing studies, providing benchmarks against established clustering methods. The Group 3 dataset is described in Table 5.3. Note that, given the observed mixed taxonomic levels used in [202] for clustering the Fungi dataset, and the fact that both subphyla “Pezizomycotina” and “Saccharomycotina” belong to phylum Ascomycota, we divided this clustering task into two parts, Tests 12 and 13. The first task (Test 12) involved clustering kingdom Fungi into phyla “Ascomycota” and “Basidiomycota”, while the second task (Test 13) focused on clustering phylum Ascomycota into subphyla “Pezizomycotina” and “Saccharomycotina”. Details about the Group 4 dataset [95] are presented in Table 5.4. The sequence lengths of six datasets, each beginning with the prefix “Medium-” range between 653 and 2,062 bp, while the other six datasets, prefixed with “Long-”, span from 1,393 to 4,049 bp. The numerical values ranging from 60 to 97

Test	Taxonomic clustering (No. of seq. per cluster)	No. of seq.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)
5	Family into Genus [unbalanced]: Astroviridae into Avastrovirus (363), Mamastrovirus (726)	1,089	5,003	6,653	8,324
6	Family to Genus [balanced]: Astroviridae into Avastrovirus, Mamastrovirus (363 each)	726	5,003	6,787	7,960
7	Species into Subtypes: dengue virus into subtypes 1, 2, 3, 4 (407 each)	1,628	10,161	10,563	10,940
8	Species into Subtypes: hepatitis C virus into subtypes 1, 1a, 1b, 2b, 3a (190 each)	950	7,005	9,059	9,678
9	Species into Subtypes: human immunodeficiency virus type 1 into subtypes 01B, 01_AE , 02_AG, A1, A1C, A1CD, A1D, A6, B, BF1, C, D, G (100 each)	1,300	8,001	8,904	9,839

Table 5.2: Details of the Group 2 dataset in Tests 5 through 9 (viral whole genomes).

in the dataset labels represent the identity score, a measure of designed relatedness determined by the ratio of identical nucleotides in two sequences relative to the alignment length (including gaps). These synthetic sequences, designed with different sequence lengths and identity score thresholds, evaluate the performance of CGRclust under controlled, and different conditions. For further details on Group 3 and Group 4 datasets, the reader is referred to, [202], and [95], respectively.

5.3.2 Method Overview

The contrastive clustering method proposed in this chapter, CGRclust, utilizes FCGR, which represents a DNA sequence at resolution k as a two-dimensional unit square image. In an FCGR, the intensity of each pixel signifies the frequency of a particular k -mer in the input DNA sequence [68]. To capture the positional information (location of points) within FCGR images, a CNN model was integrated into the pipeline. CGRclust enhances the clustering performance by leveraging unsupervised contrastive learning. Contrastive learning is a powerful technique that can learn informative representations by comparing

Test	Taxonomic clustering (No. of seq. per cluster)	No. of seq.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)
10	Class into Order: Insecta into Lepidoptera, Hemiptera, Diptera, Coleoptera, Dictyoptera, Orthoptera, Hymenoptera (650 each)	4,550	14,602	15,897	25,011
11	Kingdom into Phylum: Chromista/Plantae (Protista) into Alveolata, Stramenopiles, Rhodophyta (315 each)	945	5,498	24,697	24,697
12	Kingdom into Phylum: Fungi into Ascomycota, Basidiomycota (335 each)	670	22,528	59,864	99,976
13	Phylum into Subphylum: Ascomycota into Pezizomycotina, Saccharomycotina (535 each)	1,070	20,063	63,388	99,850

Table 5.3: Details of the Group 3 dataset in Tests 10 through 13 (mtDNA of Insects, Protists, and Fungi, from [202]).

how similar or different pairs of examples are, rather than relying solely on raw data or labelled examples [129] (see Section 2.6.4 for more details). This approach helps the model understand the underlying structures of the data by pulling similar instances (elements of a so-called “positive pair”) closer, while pushing dissimilar ones (elements of a so-called “negative pair”) farther apart in the representation space. Here, a *positive pair* is defined as consisting of two “augmented” versions of an input DNA sequence (mimic sequences). Mimic sequences are generated by the algorithm from an original DNA sequence so as to be similar to the original, or related to it in a meaningful way. In this context, a *negative pair* is defined as any other pair of sequences in the dataset. The clustering process in CGRclust takes advantage of the concept of *twin contrastive learning* (TCL) [319, 173], a method that simultaneously optimizes two contrastive learning objectives, one at the instance-level and another at the cluster-level, as detailed below.

Figure 5.1 illustrates an overview of the proposed CGRclust pipeline. The pipeline consists of four main components: 1) data augmentation (generation of mimic sequences) for FCGR positive pair construction, 2) backbone model for projection into a latent feature space, 3) *instance-level* contrastive head (ICH), and 4) *cluster-level* contrastive head (CCH). The first component is shown in the left panel of Figure 5.1, while the other three components are in the middle panel. Initially, pairs of mimic sequences constructed during

Test	Dataset	No. of seq.	Min. seq. len.(bp)	Avg. seq. len.(bp)	Max. seq. len.(bp)	Min. cluster size	Avg. cluster size	Max. cluster size
14	Medium-60	18,210	653	1,365	2,062	13	202	398
15	Medium-70	18,731	678	1,359	2,027	8	212	398
16	Medium-80	20,939	664	1,425	2,043	14	222	398
17	Medium-90	21,266	730	1,340	2,016	5	194	400
18	Medium-95	24,039	724	1,446	2,038	7	203	396
19	Medium-97	20,772	736	1,358	2,022	13	192	390
20	Long-60	20,885	1,393	2,758	4,039	7	207	398
21	Long-70	18,558	1,441	2,754	4,062	19	224	399
22	Long-80	20,525	1,396	2,639	3,974	5	194	398
23	Long-90	22,518	1,489	2,586	3,964	5	196	400
24	Long-95	20,222	1,461	2,890	4,049	10	206	400
25	Long-97	19,960	1,486	2,715	3,988	5	210	398

Table 5.4: Details of the Group 4 dataset in Tests 14 through 25 (synthetic sequences with different length and identity score thresholds from [95]). The prefixes “Medium-” and “Long-” in the dataset names denote the length of the sequences they contain. The numerical values ranging from 60 to 97 represent the identity score, indicating the percentage of similarity between the sequences. The number of clusters for all tests is 100.

the data augmentation phase (pipeline component 1), and assumed to belong to the same cluster, are projected into a latent feature space using CNNs (pipeline component 2). It is important to note that in the training phase, the two mimic sequences constructed from each original sequence were used as members of a positive pair, while the original sequence was used exclusively in the testing phase. Subsequently, the ICH (pipeline component 3) and CCH (pipeline component 4) conduct instance-level and cluster-level contrastive learning. ICH is designed to enhance the similarity of representations of positive pairs in the latent feature space, while making the representations of negative pairs more distinct. On the other hand, CCH’s goal is to effectively separate clusters of data points, ensuring that each cluster is distinctly different from the others.

The two components (ICH and CCH) are simultaneously optimized through twin contrastive learning (TCL) by operating on the row (ICH) and column (CCH) spaces of the feature matrix, respectively. Through this simultaneous optimization, CGRclust enhances the representation’s quality by handling both detailed (in ICH) and broad (in CCH) distinctions in the data, all without relying on pre-defined taxonomic labels. As the training

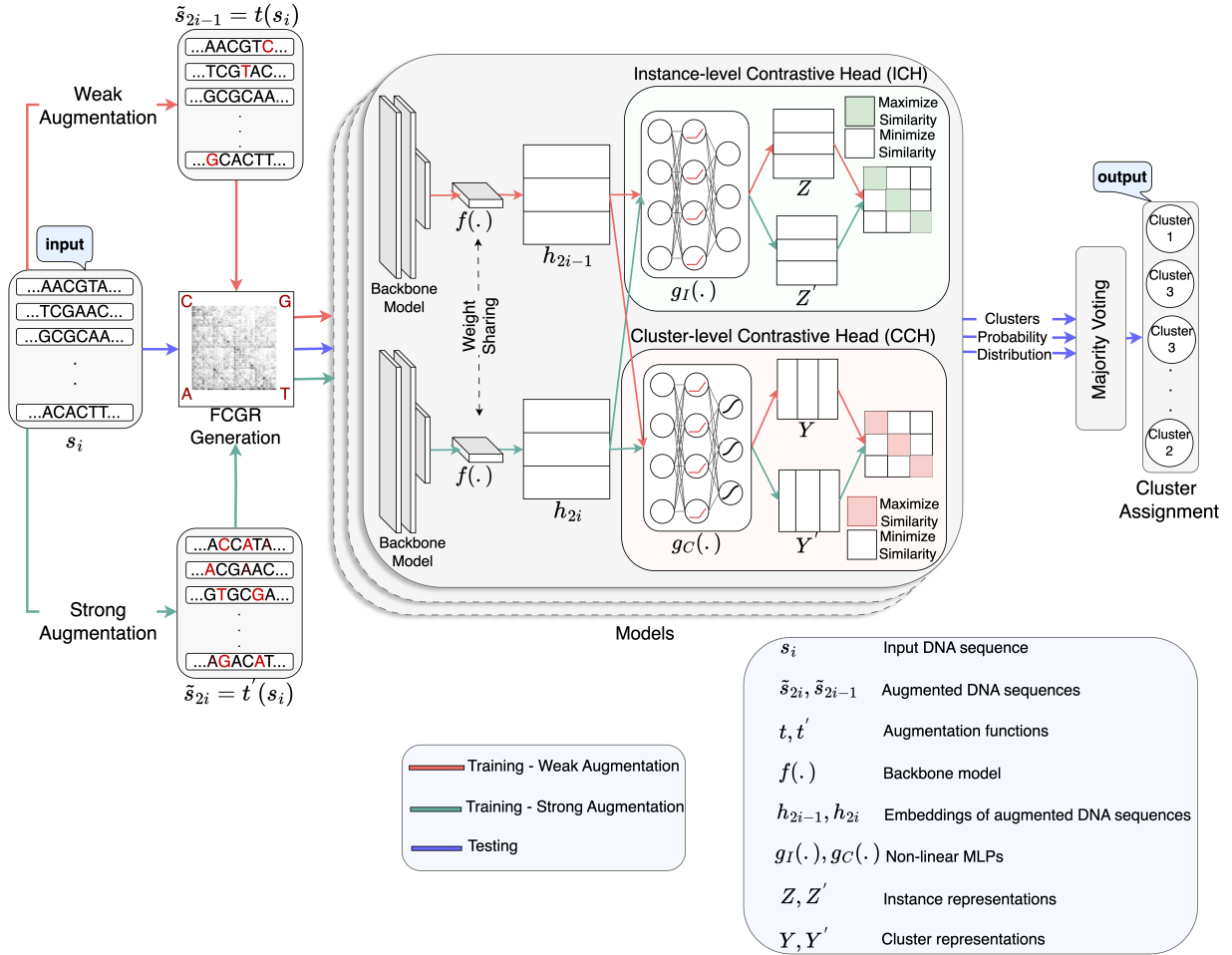


Figure 5.1: Diagram illustrating the overview of the CGRclust pipeline. Left panel: The process begins with data augmentation to create positive pairs (pairs of mimic sequences, pipeline component 1), followed by the generation of FCGR images of these augmented DNA sequences. Middle Panel: The FCGR images are fed into the backbone model (CNN) for embedding into a latent feature space (pipeline component 2). The twin contrastive learning scheme employs an instance-level contrastive head (ICH) and a cluster-level contrastive head (CCH) to perform contrastive learning at both the instance and the cluster levels (pipeline components 3 and 4, respectively). Right panel: To counteract the inherent variance in CNN training outcomes, a majority voting strategy is applied, aggregating results from multiple CNN models with distinct initializations to finalize cluster assignments for each input DNA sequence.

process involves randomized algorithms leading to high variance outcomes depending on the different initializations and random seeds, a majority voting scheme is then employed (right panel of Figure 5.1), which uses the outcomes of five distinct CNN models with different initializations to determine the final cluster assignment for each sequence. The quality of clusters generated by CGRclust is assessed through an independent process using the Hungarian algorithm [160], which aligns the predicted cluster assignments with actual taxonomic labels to evaluate clustering accuracy, as described in Section 2.7.2. Note that in unsupervised learning, ‘training’ comprises parameter optimization using unlabeled data; ‘testing’ then evaluates the trained model by comparing its output on the same data against the ground truth, for evaluation purposes.

5.3.3 DNA Data Augmentation: Mimic Sequences

Data augmentation plays a critical role in contrastive clustering by significantly enhancing the model’s ability to learn invariant representations from limited data. By adding different types of changes to the training data (thereby generating positive pairs), data augmentation helps the model to focus on the key features that define each cluster, avoiding the trap of fitting too closely to random noise or unimportant details. Consequently, CGRclust is based on constructing positive pairs and negative pairs through data augmentations. A pair of positive data points is a pair of mimic sequences, that are considered to be similar or related in some meaningful way (e.g. belonging to the same cluster), while a pair of negative data points is a pair of sequences that are considered to be dissimilar. We adapted a similar approach to [319], and used an effective augmentation strategy by mixing weak and strong transformations as it previously showed superior performance on both image and text data when combined with TCL. For each DNA sequence input s_i , we define transformations t and t' as follows: t and t' are functions from the domain of DNA sequences to the set of augmented DNA sequences, with t applying a set of transformations from an augmentation family T , and t' applying a set of transformations from an augmentation family T' . These transformations are designed to modify the input sequence s_i in distinct ways, generating a positive pair represented as $(\tilde{s}_{2i-1}, \tilde{s}_{2i})$, where $\tilde{s}_{2i-1} = t(s_i)$ and $\tilde{s}_{2i} = t'(s_i)$.

Note that direct image transformations traditionally used in computer vision for data augmentation (image flipping, cropping, or rotation), if applied to CGR/FCGR images, do not correspond to biologically meaningful or minor changes in the original DNA sequence. Indeed, such transformations could result in drastic and non-intuitive sequence changes, since the CGR/FCGR representations depend on the sequence’s nucleotide order and composition. Thus, in CGRclust we opted to modify raw DNA sequences to create

mimic sequences. This approach ensures that any resulting image alterations are meaningful, and mirror potential natural genetic variations in sequence composition.

In CGRclust pipeline, data augmentations were implemented through functions t and t' , belonging to the augmentation families T (weak augmentations) and T' (strong augmentations) respectively. Two types of data augmentation were explored, *mutation* and *fragmentation*. Both mutation and fragmentation of a DNA sequence, when appropriately applied, can alter the sequence while still maintaining patterns within its FCGR that are very similar (but not identical) to the FCGR of the original DNA sequence.

Mutation, denoted by $mutate(\mu)$ has a mutation rate μ as parameter, and performs two types of substitution mutations (transitions and transversions) on the original DNA sequence. The probability of transitions is defined as being μ while the probability of transversions is $0.5 * \mu$, as the mutational hypothesis holds that the transition mutation rates are higher than the transversion rates in practice [186]. Fragmentation, denoted by $frag(len)$, has the length len of the desired fragment as a parameter. Given a DNA sequence of length n as input, fragmentation outputs a random fragment of length len of the input sequence ($len \leq n$).

In each computational experiment, the augmentation functions t and t' can be either mutation or a fragmentation. If the selected augmentation function is mutation, then t is the function $mutate(\mu_1)$ (weak), and t' is the function $mutate(\mu_2)$ (strong), where $\mu_1 < \mu_2$. Similarly, if the selected augmentation function is fragmentation, then the function t is $frag(len_1)$ (weak), while t' is the function $frag(len_2)$ (strong), where $len_2 < len_1$.

To evaluate the impact of different data augmentation strategies on CGRclust, both mutation and fragmentation were explored, each with different values for their respective parameters. We compared these augmentations, looking at different strengths for both t and t' , to see how they affected the clustering accuracy of Test 1 (order Cypriniformes), as shown in Table 5.5. The results highlighted the superiority of mutation as the augmentation technique, with mutation rates set to $\mu = 10^{-4}$ for weak augmentation and $\mu = 10^{-2}$ for strong augmentation, in comparison to alternative methods. Thus, mutation with these parameters was used as the default data augmentation and parameters for all computational experiments in this study.

Given the constructed pairs, a shared backbone $f(\cdot)$ is used to extract features h from the augmented samples (mimic sequences) through $h_{2i-1} = f(X_{\tilde{s}_{2i-1}})$ and $h_{2i} = f(X_{\tilde{s}_{2i}})$. To extract the important features of FCGR images, the backbone model was used to convert the two-dimensional input FCGRs into one-dimensional embeddings. Details about the backbone model used to process FCGR can be found in section 5.3.5 (Backbone model architecture).

Weak augmentation (t)	Strong augmentation (t')	Accuracy
$mutate(\mu = 10^{-4})$	$mutate(\mu = 10^{-3})$	74.50%
$mutate(\mu = 10^{-4})$	$mutate(\mu = 10^{-2})$	94.78%
$mutate(\mu = 10^{-3})$	$mutate(\mu = 10^{-2})$	72.69%
$frag(l = 0.99 \times \text{sequence length})$	$frag(l = 0.8 \times \text{sequence length})$	83.53%
$frag(l = 0.95 \times \text{sequence length})$	$frag(l = 0.8 \times \text{sequence length})$	75.50%
$frag(l = 0.95 \times \text{sequence length})$	$frag(l = 0.7 \times \text{sequence length})$	75.50%
$frag(l = 0.95 \times \text{sequence length})$	$frag(l = 0.6 \times \text{sequence length})$	86.75%
$frag(l = 0.95 \times \text{sequence length})$	$frag(l = 0.5 \times \text{sequence length})$	72.49%
$frag(l = 0.9 \times \text{sequence length})$	$frag(l = 0.8 \times \text{sequence length})$	70.28%

Table 5.5: Impact of various augmentation techniques on clustering accuracy of Test 1 (clustering of order Cypriniformes). This assessment focuses on the effectiveness of the data augmentation technique used for weak and strong augmentation in CGRclust pipeline. The entries in boldface indicate the highest clustering accuracy obtained during the experiments, highlighting the optimal augmentation parameters (i.e. mutation with $\mu = 10^{-4}$ as weak augmentation, mutation with $\mu = 10^{-2}$ as strong augmentation) for this specific dataset.

5.3.4 Twin Contrastive Learning

Twin Contrastive Clustering (TCL) is a machine learning method that utilizes the principles of contrastive learning to enhance clustering performance, especially in tasks involving complex or high-dimensional data, such as images, or text [319, 173]. During the training phase, backbone components, ICH, and CCH undergo joint optimization based on the following twin contrastive loss function:

$$L_{train} = \alpha L_{ins} + (1 - \alpha) L_{clu} \quad (5.1)$$

Here, L_{ins} denotes the instance-level contrastive loss computed via ICH, to increase the similarity between positive pairs and decrease it between negative pairs. Meanwhile, L_{clu} represents the cluster-level contrastive loss, determined through CCH, focusing on refining the pairwise similarities of cluster representations between weak and strong data augmentations. α represents a weighting parameter that balances the contributions of the instance-level contrastive loss (L_{ins}) and the cluster-level contrastive loss (L_{clu}) in the overall training loss (L_{train}).

Instance-level contrastive learning strives to maximize similarities among positive pairs while minimizing those among negative pairs. Optimal clustering would classify instance

pairs within the same class as positive and those across classes as negative. Yet, in the absence of predefined labels, we adapt by forming mimic sequence instance pairs via data augmentations. Given a batch size of N , we subject each DNA sequence, s_i , to two variants of data augmentations, generating $2N$ augmented samples expressed as $\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{2i-1}, \tilde{s}_{2i}, \dots, \tilde{s}_{2N}$.

Before employing ICH and CCH, we map features into two different subspaces using two-layer nonlinear Multilayer Perceptrons (MLPs), symbolized as $g_I(\cdot)$ and $g_C(\cdot)$, respectively. The InfoNCE loss [48] is utilized to optimize pairwise similarities, as defined by the following equation (where we assume that $\tilde{s}_i$ forms a positive pair with $\tilde{s}_j$):

$$l_i = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau_I)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp \text{sim}(z_i, z_k)/\tau_I} \quad (5.2)$$

where $z_i = g_I(X_{s_i})$, the function $\text{sim}(z_i, z_j)$ is pair-wise similarity measured using cosine distance, and τ_I is the instance-level temperature parameter to control the smoothness of the probability distribution. The instance-level contrastive loss is an average of the above loss over the $2N$ samples in the batch:

$$L_{ins} = \frac{1}{2N} \sum_{i=1}^{2N} l_i \quad (5.3)$$

When a sample is mapped into a subspace with dimensions equal to the number of clusters, the i -th component of its feature denotes the likelihood of it belonging to the i -th cluster. If the desired number of target clusters is denoted by C , similar to the instance-level contrastive head, we employ another two-layer MLP $g_C(\cdot)$ to transform the features into a space of dimensionality C . Let $Y = \{y_1, y_3, \dots, y_{2N-1}\} \in \mathbb{R}^{N \times C}$ be the cluster assignment probabilities of the batch (of size $2N$) under the weak augmentation T , and $Y' = \{y_2, y_4, \dots, y_{2N}\} \in \mathbb{R}^{N \times C}$ be the cluster assignment probabilities under the strong augmentation T' , where $y_i = g_C(X_{s_i})$. The columns of Y and Y' correspond to the cluster distributions. We denote the i -th column of Y by $\hat{y}_{2i-1}$ and the i -th column of Y' by $\hat{y}_{2i}$. The InfoNCE loss was once again utilized to optimize pairwise similarities of the representation of cluster i under the weak (and strong) data augmentation, as defined by the following equation:

$$\hat{l}_i = -\log \frac{\exp(\text{sim}(\hat{y}_i, \hat{y}_j))/\tau_C}{\sum_{k=1}^{2C} 1_{k \neq i} \exp \text{sim}(\hat{y}_i, \hat{y}_k)/\tau_C} \quad (5.4)$$

where the y_i and y_j represent the cluster assignment probabilities of different augmentations of the data, function $\text{sim}(\hat{y}_i, \hat{y}_j)$ is the pair-wise similarity measured using cosine

distance, and τ_C is the cluster-level temperature parameter to control the smoothness of the probability distribution. To achieve more balanced clustering outcomes we added cluster entropy to the average of the above cluster-level loss over the C clusters to define the cluster-level contrastive loss:

$$L_{clu} = \frac{1}{2C} \sum_{i=1}^{2C} \hat{l}_i + \sum_i^{2C} (P(\hat{y}_i) \log P(\hat{y}_i)) \quad (5.5)$$

Here, $P(\hat{y}_{2i-1})$ and $P(\hat{y}_{2i})$ represent the probabilities of a sequence being assigned to cluster i under weak and strong augmentation conditions, respectively.

While a confidence-based boosting strategy, which involves iterative adjustments to the learning process based on model prediction confidence, yielded a slight enhancement in the clustering outcomes of [319], no significant improvement was observed for FCGR clustering. Therefore, we opted against incorporating this step to maintain pipeline simplicity and efficiency. For additional information about TCL see [319].

Effectiveness of Two Contrastive Heads and Weight Parameter in Training Loss

In exploring the optimal setting for the weight parameter α in the training loss function $L_{train} = \alpha L_{ins} + (1 - \alpha) L_{clu}$, we tested eleven different values ranging from 0 to 1. Figure 5.2 shows the experimental results, spanning four distinct datasets in Tests 1-4 (Group 1 dataset), indicating that the value of 0.7 for α consistently delivered either the highest or close to the highest accuracy. Furthermore, it was observed that values within the range of 0.5 to 0.8 generally yielded superior outcomes, suggesting a robust zone of performance for α across varying data conditions. This finding emphasizes the critical role of α in balancing the contributions of instance-level and cluster-level loss components to achieve optimal training results.

Optimization of Temperature Parameters

A thorough hyperparameter optimization for the twin deep clustering model, emphasizing the instance- and cluster-level temperature parameters (τ_I and τ_C) in ICH and CCH, was crucial for enhancing clustering accuracy. Figure 5.3 depicts a 3D visualization of this optimization, examining ten distinct values for each temperature parameter in the range $[0.1, 1]$, revealing a detailed accuracy landscape within the parameter spectrum for four datasets in Tests 1-4 (Group 1 dataset). This analysis, illustrated using four 3D surface plots, reveals that while the optimal values of these two hyperparameters vary slightly across four

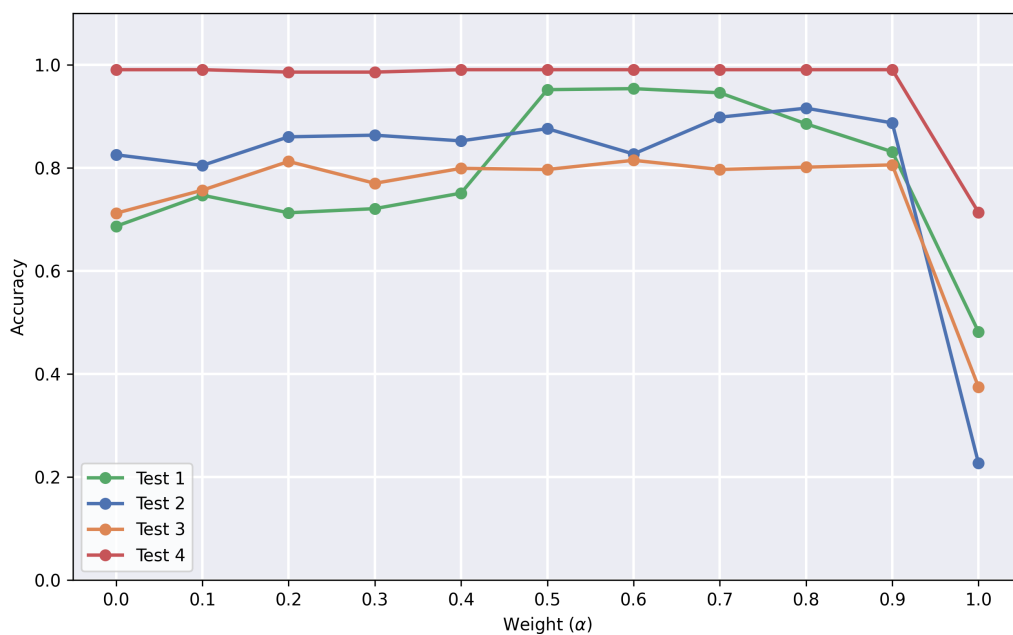


Figure 5.2: Experimental results illustrating the effect of the weight parameter α in the training loss function across the Group 1 dataset. Testing of eleven values from 0 to 1 revealed that values between 0.5 and 0.8 generally delivered superior results. The default $\alpha = 0.7$ consistently yielded the highest or near-highest accuracy in Tests 1-4.

datasets, choosing $\tau_I = 0.1$ and $\tau_C = 1.0$ consistently yields relatively high accuracy across all datasets, comparable to the highest accuracy observed in each dataset. This detailed adjustment of parameters notably improved the clustering of mtDNA sequences over standard unsupervised methods at different taxonomic levels. This advancement aligns with the hypothesis that a lower τ_I encourages individual instance differentiation, aligning with the ICH’s aim, while a higher τ_C enhances group discrimination, mirroring the CCH’s objective [161].

5.3.5 Backbone Model Architecture

The augmented (mimic) DNA sequence pairs of FCGRs ($X_{\tilde{s}_{2i-1}}, X_{\tilde{s}_{2i}}$) serve as inputs for training multiple independent instances of a backbone model, ICH, and CCH. Given that the genomic datasets we are working with are notably smaller in scale compared to those typically encountered in computer vision, we found that common architectures such as *ResNet34* and *ResNet50*, which have demonstrated efficacy in various visual tasks, were not well-suited as backbone models for genomic datasets. Therefore, we opted for a simpler yet versatile architecture that is better suited for clustering FCGRs of DNA sequences. The backbone model architecture, as shown in Figure 5.4, is composed of a single convolutional block featuring two convolutional layers. Each convolutional layer employs a kernel size of 7, a stride of 2, and a padding of 1. Following each convolutional layer is a Rectified Linear Unit (ReLU) activation function and a batch normalization layer for data normalization prior to being passed to the subsequent layer. Subsequently, the output of the final batch normalization layer undergoes max pooling with a kernel size of 2 to downsample the data across its spatial dimension by selecting the maximum value within each 2x2 window. Lastly, to transform the multidimensional input into a one-dimensional embedding, a flattening layer is applied, followed by a linear layer configured to match the desired output dimension.

5.3.6 Majority Voting Scheme

The integration of ensemble learning, particularly through majority voting, has significantly improved the accuracy of genomic sequence classification, as demonstrated in Chapter 3. Majority voting, or hard voting, which was used in DeLUCS pipeline, relies on the most frequent prediction across models, while soft voting considers the probability distributions of outcomes, often yielding higher precision. To optimize the performance of CGRclust, we employed five instances of the backbone model along with instance- and cluster-level

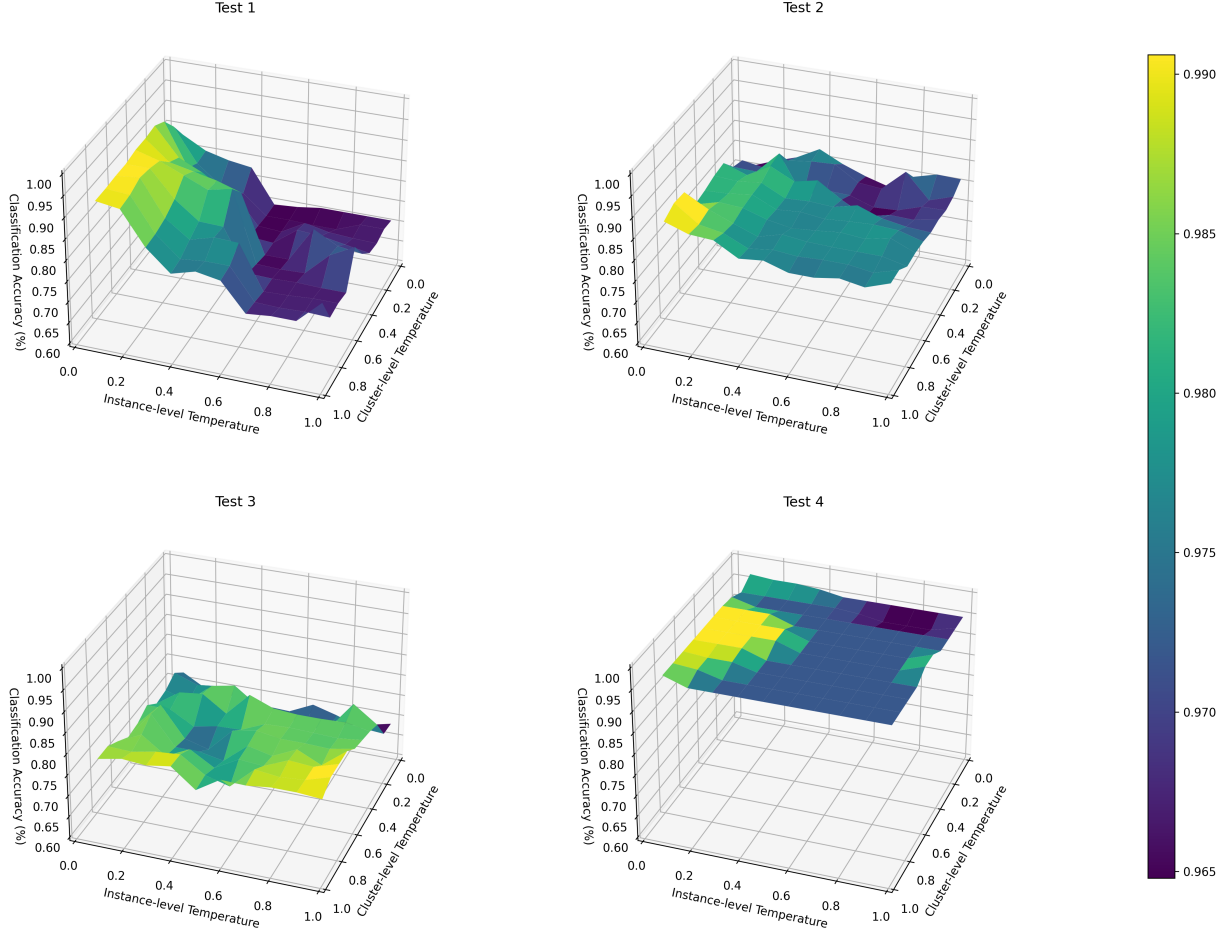


Figure 5.3: 3D visualization of hyperparameter optimization for the twin deep clustering model in Tests 1-4, focusing on τ_I and τ_C parameters. This figure illustrates the exhaustive search across a parameter range of $[0.1, 1]$, presenting the accuracy landscape and emphasizing the optimal configurations for small values of τ_I (e.g. 0.1) and larger values of τ_C (e.g. 1.0), which considerably improve mitochondrial DNA sequence clustering at different taxonomic levels.

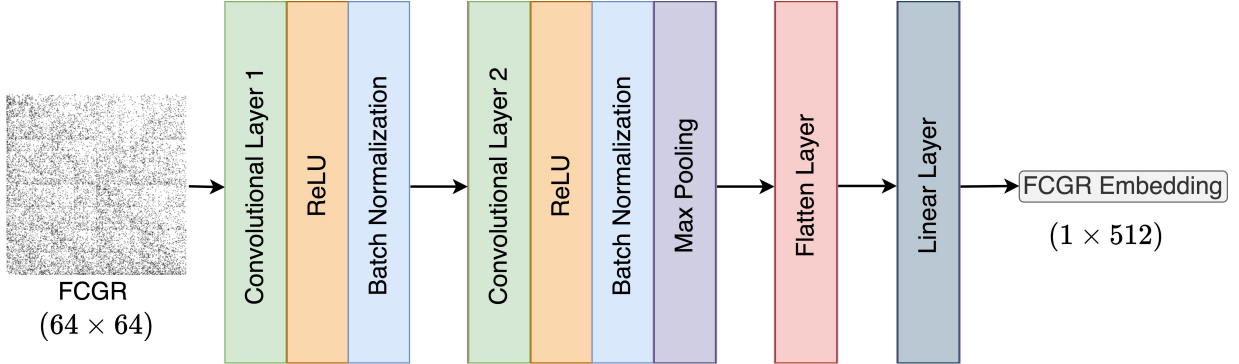


Figure 5.4: Architecture of backbone model designed for clustering FCGR images of DNA sequences. The architecture of the backbone model comprises two convolutional layers, each with a kernel size of 7, stride of 2, and padding of 1. Following each convolutional layer, a Rectified Linear Unit (ReLU) is applied to introduce non-linearity, followed by a batch normalization layer to maintain numerical stability. Next, a max pooling layer with a kernel size of 2 efficiently reduces the spatial dimensions of the feature maps. A flattening layer to transform the multidimensional feature maps into a one-dimensional vector. This is followed by a linear layer, adjusting the output dimension to the desired configuration.

contrastive heads. Each model copy was initialized randomly with distinct random seeds. Both soft and hard voting applied to CGRclust reduce variance due to random initialization and enhance model convergence thereby boosting the robustness and reliability of clustering predictions. Figure 5.5 shows the impact of majority voting on clustering the Group 1 dataset. Although both voting methods enhanced CGRclust’s performance, soft voting showed a slightly higher improvement. Consequently, we adopted soft voting as our default method. This approach integrates classifiers’ certainty levels into the final prediction, thus yielding more reliable and potentially more accurate results.

5.3.7 Confidence Interval for Clustering Accuracy

In the evaluation of machine learning models, it is crucial to consider the potential limitations such as limited data, assumptions of independence, and sampling biases [239, 277]. A confidence interval provides a statistical estimate of the uncertainty surrounding a model’s reported accuracy, offering insights into its expected performance on unseen data. Commonly, a 95% confidence interval is employed, providing a range that is expected to contain the true accuracy of the model in general settings.

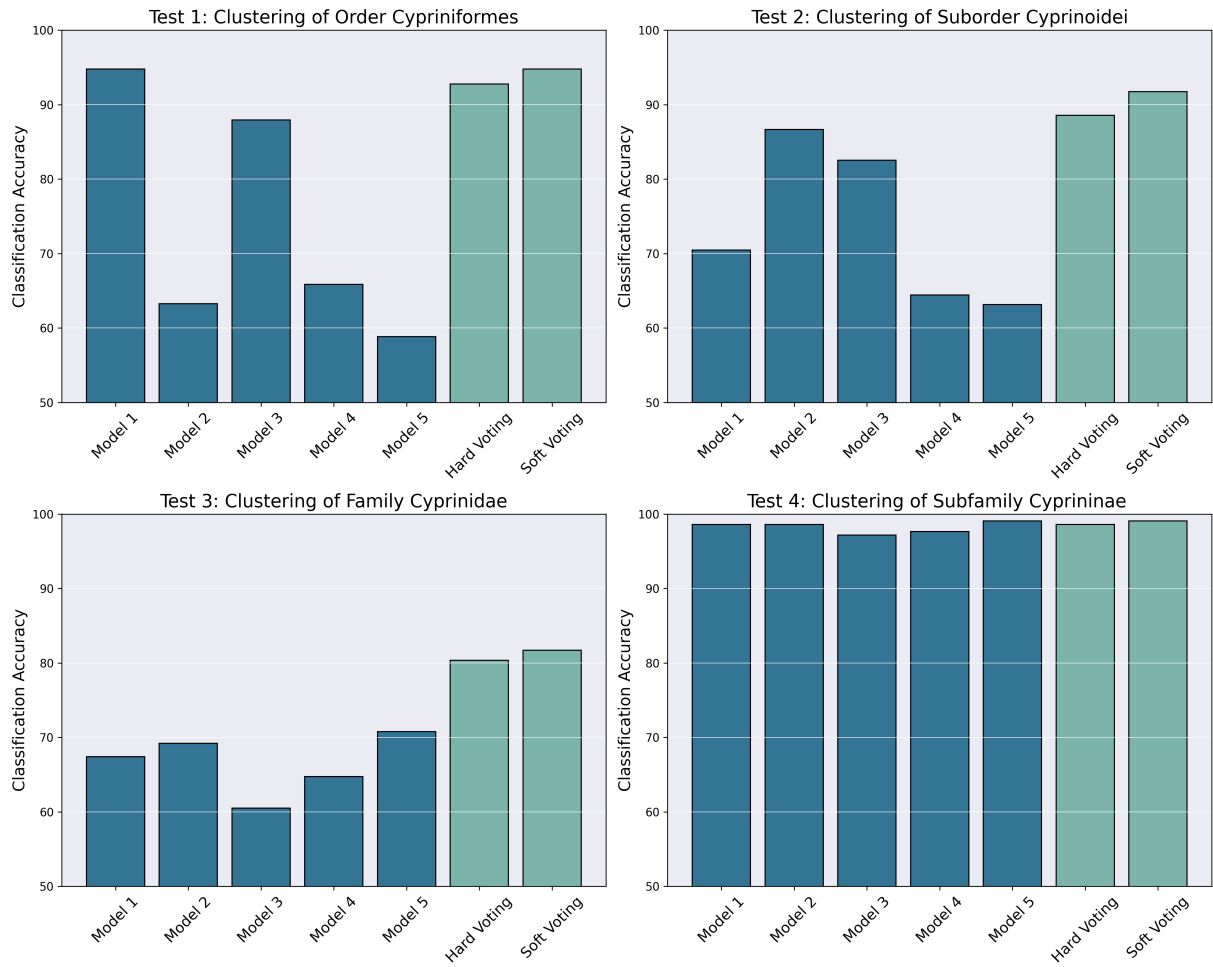


Figure 5.5: Comparative analysis of clustering accuracy across four datasets using hard and soft majority voting. The bar plot illustrates the significant improvement in clustering accuracy when employing a voting scheme within the framework of twin contrastive clustering models, highlighting the enhanced accuracy in genomic sequence classification.

The confidence interval for clustering accuracy is computed using the formula:

$$CI = ACC \pm z \cdot \sqrt{\frac{ACC(1 - ACC)}{n}} \quad (5.6)$$

where ACC is the accuracy observed on the test set, n is the number of samples in the test set, and z is the z -value of 1.96 for a 95% confidence level [239]. This calculation is particularly advantageous in deep learning contexts like CGRclust, where models undergo extensive training over substantial computational resources. By providing a confidence interval, we can more accurately assess the reliability and robustness of the model’s performance, beyond the specific conditions of the trained dataset.

5.3.8 Experimental Settings and Implementation

Throughout the training process, all CGRclust’s hyperparameters remained constant and consistent across all tests, having been empirically chosen to achieve optimal performance. We used the complexCGR library [41] to transform DNA sequences into their FCGR representations. We empirically chose $k = 6$ for the resolution of FCGR after evaluating k values ranging from 6 to 8. This selection offered an optimal trade-off between computational efficiency and accuracy. Prior to input into the network, all FCGRs underwent normalization. This process involved first standardizing each FCGR matrix’s value by the min-max normalization to scale the features to the range of $[0, 1]$, thus mitigating the impact of sequence length on pixel intensity. Subsequently, the FCGR matrices were normalized by Z-score normalization to scale features so that they have the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. This normalization enhanced the stability and convergence of the model.

We utilized the Adam optimizer [154] with an initial learning rate set to $7e-5$ and a weight decay of $1e-4$ to jointly optimize both contrastive heads and the backbone model. In our observations, the implementation of the scheduler did not yield significant improvements. Furthermore, the selection of batch size, empirically set at 512, is a critical factor during training. This importance stems from the batch-wise operation of the unsupervised learning process, which is essential for determining the output distribution. Inadequate batch sizes may fail to accurately represent the true data distribution, resulting in the dominance of the entropy term in the loss function and potentially leading to suboptimal solutions. The dimensionality of ICH was determined empirically to be 128, aiming to preserve discriminative information within the data. The dimensionality of CCH was determined by the target cluster number.

CGRclust’s pipeline is fully implemented in Python 3.10, and the source code is publicly available in the GitHub repository <https://github.com/fatemehalipour/CGRclust>. All tests with CGRclust and DeLUCS were conducted on a node within the Béluga cluster at Compute Canada, which features dual Intel Gold 6148 Skylake CPUs @ 2.4 GHz, 186 GB RAM, and an NVIDIA Tesla V100 SXM2 GPU with 16 GB of memory. Following [202] authors’ recommendation, *i*DeLUCS was executed on Google Colab using an NVIDIA Tesla T4 GPU with 16 GB of memory.

For benchmarking CGRclust’s performance against state-of-the-art methods in DNA sequence clustering, we chose three alignment-free clustering methods noted for their effectiveness in clustering a variety of genomic datasets: DeLUCS (introduced in Chapter 3), *i*DeLUCS [202], and MeShClust v3.0 [95]. For both DeLUCS and *i*DeLUCS, we applied the default hyperparameters, and the accuracies presented in the results section are based on these settings.

Optimal Threshold Values across Datasets for MeShClust v3.0

In our analysis of MeShClust v3.0, a density-based clustering method that does not pre-define the number of clusters, we faced challenges with the automatic selection of identity thresholds. These automatic thresholds often resulted in discrepancies between expected and actual cluster counts. To mitigate this issue, we manually explored various identity score thresholds within the range of $[0, 1]$ for each dataset. This manual adjustment allowed us to pinpoint an optimal threshold that aligned with the desired number of clusters for each specific clustering test.

This approach enabled the selection of an optimal threshold for each clustering test, aligning with the desired cluster count. The optimal threshold values for each of the thirteen datasets tested are detailed in Table 5.6, demonstrating the adjustments needed to achieve accurate clustering results. As shown in the first four rows of Table 5.6, progressing to lower taxonomic levels where sequences become more similar necessitated increasing the identity score threshold to achieve the correct number of clusters.

5.4 Results

5.4.1 Qualitative Performance of Twin Contrastive Learning

A qualitative analysis was first employed to assess the effectiveness of instance-level and cluster-level TCL, as implemented in CGRclust for clustering mtDNA sequences in Test 1.

Test	Dataset	Threshold
1	Cypriniformes	0.670
2	Cyprinoidei	0.695
3	Cyprinidae	0.814
4	Cyprininae	0.850
5	Astroviridae-unbalanced	0.437
6	Astroviridae-balanced	0.400
7	Dengue	0.750
8	HCV	0.750
9	HIV-1	0.824
10	Insecta	0.600
11	Protista	0.350
12	Fungi-phylum	0.300
13	Fungi-subphylum	0.285

Table 5.6: Optimal threshold values for identity scores using MeShClust v3.0 across thirteen real datasets. The identity threshold is a parameter that specifies the minimum similarity score required for sequences to be grouped together in a cluster.

The dynamic learning process during the training phase is shown in Figure 5.6, illustrating how the model develops discriminative representations and accurately determines cluster assignments. This progression is documented across epochs and displayed at five time-stamps. In Figure 5.6, the total number of clusters is established at three, corresponding to the points of a triangle, where each point signifies a taxonomic cluster. The placement of each point is derived from its three-dimensional probability vector, and different colors indicate the three ground truth taxonomic labels in Test 1. At the beginning, sequences are located at the triangle’s center, reflecting an equal chance of being assigned to any of the three clusters. As training proceeds, the model increasingly assigns sequences to appropriate clusters, moving similar sequences closer to their respective vertex/cluster with greater probability. Notably, sequences that are assigned the same probability vectors will have their points overlap.

5.4.2 Quantitative Performance Analysis and Comparison with Other Methods

In this section we analyze the performance of CGRclust and compare it with three other established clustering methods for DNA sequences, DeLUCS, *i*DeLUCS [202], and MeSh-

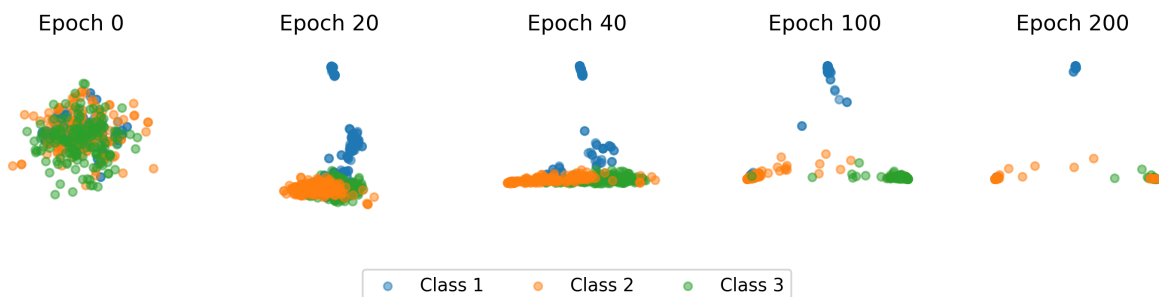


Figure 5.6: CGRclust’s evolution of clustering 498 Cypriniformes mitochondrial DNA sequences into three distinct clusters in Test 1. Each data point represents a DNA sequence, and its colour indicates its suborder label, and its position indicates the likelihood of assignment to different clusters (corners). A point at the center of the triangle has an equal probability of being assigned to any of the three clusters, while a point at a corner indicates a definitive association, with probability 1, to that specific corner/cluster. Note that any overlap of colors in the last epoch corresponds to instances of misclustering, where sequences have not been correctly assigned to the ground truth cluster.

Clust v3.0 [95] (with both manual and automatic selection of the identity score threshold). Note that the ground truth labels are used post-hoc and for evaluation purposes only, and they were not utilized during the clustering process.

Table 5.7 presents a summary of the clustering accuracies for the Group 1 dataset described in Table 5.1 (Cypriniformes mtDNA) across Tests 1-4. Using formula 5.6, the confidence intervals of the CGRclust clustering accuracies are shown in Table 5.7. The accuracies of CGRclust were achieved using the default hyperparameters over 150 epochs. As Table 5.7, shows, CGRclust consistently achieves comparable (within the confidence interval), or the highest accuracy across all four taxonomic levels. Specifically, CGRclust outperforms DeLUCS by 3.21% to 12.95% across different tests. In contrast to the generally superior performance of CGRclust, *i*DeLUCS shows competitive results in certain scenarios. Specifically, it achieves the highest accuracy among all methods at the suborder to family level (92.06%), comparable with CGRclust (within the confidence interval). This indicates that *i*DeLUCS has particular strengths in clustering mtDNA datasets at some specific taxonomic levels. However, at other taxonomic levels, *i*DeLUCS’s performance generally is lower than both CGRclust and DeLUCS, suggesting that its clustering efficacy may vary depending on the nature and extent of sequence variation at a particular taxonomic level, and the characteristics of the dataset being analyzed. Lastly, CGRclust consistently outperforms both the manual and automated versions of MeShClust v3.0, by

Test	Taxonomic clustering	CGRclust	DeLUCS	<i>i</i> DeLUCS	MeShClust-manual	MeShClust-auto
1	Order into suborder	94.78 $\pm$ 1.95%	91.57%	64.05%	34.34%	33.94%
2	Suborder into family	91.75 $\pm$ 2.15%	78.25%	92.06%	20.16%	16.67%
3	Family into subfamily	81.70 $\pm$ 3.58%	68.75%	61.61%	29.91%	12.5%
4	Subfamily into genus	99.06 $\pm$ 1.30%	97.18%	99.53%	59.15%	36.15%

Table 5.7: CGRclust performance of clustering the Group 1 dataset (Cypriniformes mtDNA) as described in Table 5.1. CGRclust’s accuracy is compared with DeLUCS, *i*DeLUCS, and MeShClust v3.0 (with both a manual and the automatic selection of identity score threshold). Confidence intervals, indicating the statistical range of accuracy, are shown for CGRclust.

a large margin (up to 75.08%).

Table 5.8 summarizes the accuracies of clustering the five viral datasets in the Group 2 dataset described in Table 5.2 (viral whole genomes), across Tests 5-9. For the clustering of the astrovirus genomes (Tests 5 and 6), the clustering is at the family to genus level, while for the dengue virus, HCV, and HIV-1 genomes the clustering is performed from the species to the virus subtype level. CGRclust consistently outperforms the other three clustering methods, demonstrating its robustness and accuracy in the context of virus mutagenesis and evolution. In Test 5, using an unbalanced astrovirus dataset, CGRclust surpasses DeLUCS and *i*DeLUCS by 15.06%, and outperforming MeShClust-manual by 25.25%. The results demonstrate CGRclust’s superior performance in challenging clustering tasks, e.g., characterized by dataset imbalance, a condition where other methods—DeLUCS, *i*DeLUCS, and MeShClust v3.0—had a poor performance. In Test 6, which featured a cluster-balanced astrovirus dataset, the accuracy of both CGRclust and DeLUCS improved, while the accuracy of *i*DeLUCS remained relatively unchanged. In the dengue virus genomes dataset (Test 7), CGRclust, along with DeLUCS and MeShClust-manual among the compared methods, achieved perfect accuracy (100%). For the HCV genome dataset (Test 8), CGRclust achieved an accuracy of 85.79%, surpassing all compared methods by a margin of 1.16% to 8.95%. In the HIV-1 genomes dataset (Test 9), CGRclust achieves an accuracy that is 10.24% higher than DeLUCS and significantly surpasses both *i*DeLUCS and MeShClust-manual by 42.39% and 48.1%, respectively.

Table 5.9 displays the clustering accuracies for Tests 10-13 in the Group 3 dataset (mtDNA of Insects, Protists, and Fungi) from the study [202], detailed in Table 5.3. Due to the complexities and specific characteristics of datasets in the Group 3 dataset, we observed an enhancement in CGRclust performance when the hyperparameter α was increased from

Test	Taxonomic clustering	CGRclust	DeLUCS	<i>i</i> DeLUCS	MeShClust-manual	MeShClust-auto
5	Astroviridae-unbalanced	$84.94 \pm 2.12\%$	69.88%	69.88%	59.69%	70.43%
6	Astroviridae-balanced	$88.84 \pm 2.29\%$	88.84%	69.97%	77.27%	76.72%
7	dengue virus	$100.00 \pm 0.00\%$	100%	96.99%	100%	52.08%
8	hepatitis C virus	$85.79 \pm 2.22\%$	84.63%	76.84%	81.05%	80.52%
9	human immunodeficiency virus 1	$81.77 \pm 2.10\%$	71.53%	39.38%	32.77%	7.69%

Table 5.8: CGRclust performance of clustering Group 2 dataset (viral whole genomes) described in Table 5.2. CGRclust’s accuracy is compared with DeLUCS, *i*DeLUCS, and MeShClust v3.0 (with both a manual and the automatic selection of identity score threshold). In Tests 5 and 6, the clustering task is at the family to the genus level, whereas Tests 7-9 involve clustering at the virus species to subtype level. Confidence intervals, indicating the statistical range of accuracy, are shown for CGRclust.

its default value of 0.7 to 0.8, along with a greater emphasis on the instance-level contrastive head. This modification is evidenced in the third and fourth columns of Table 5.9, which display improvements in accuracy due to these adjustments. Generally, the change in the hyperparameter α led to increased accuracy across this group of datasets, with the most notable improvement seen in the Protist dataset in Test 11, where accuracy rose by 23.28%, almost bridging the gap with DeLUCS and surpassing *i*DeLUCS. However, in other datasets, this adjustment yielded minimal changes. This suggests that, in order to achieve optimal clustering outcomes, dataset-specific parameter optimization may be necessary to optimize different hyperparameters, including α . Further details on hyperparameter adjustment of α can be found in Section 2.5 (Twin contrastive learning (TCL)).

In the comparison of clustering methods presented in Table 5.9, *i*DeLUCS exhibits superior performance over other methods in the Insects mtDNA dataset of Test 10. However, both DeLUCS and CGRclust demonstrate higher accuracies in the other three tests. Specifically, in Test 11 (Protists mtDNA), the accuracies of DeLUCS and CGRclust are superior to *i*DeLUCS by 8.10% and 5.50%, respectively. Furthermore, in Tests 12 and 13, both DeLUCS and CGRclust achieved higher accuracy in the Fungi classification at phylum and subphylum levels in comparison to *i*DeLUCS and MeShClust v3.0. The manual and automatic versions of MeShClust generally display lower accuracies, with the auto-

matic version particularly underperforming the manual selection of identity threshold in three out of four datasets. It is important to note that these datasets pose significant clustering challenges due to variations in within-cluster similarities and different sequence lengths, which complicate the clustering process. While CGRclust did not always secure the top clustering accuracy across these datasets compared to other methods, the adjusted version of CGRclust demonstrated comparable clustering performance in the Insects (Test 10) and Protists (Test 11) datasets, as well as the Fungi dataset at the subphylum level (Test 13).

Test	Taxonomic clustering	CGRclust	CGRclust-adjusted α	DeLUCS	<i>i</i> DeLUCS	MeShClust-manual	MeShClust-auto
10	Insects	70.53%	$73.56 \pm 1.28\%$	78.30%	83.82%	47.50%	21.90%
11	Protists	62.22%	$85.50 \pm 2.24\%$	88.10%	80.00%	71.85%	74.92%
12	Fungi (Phylum)	56.72%	$56.87 \pm 3.75\%$	69.85%	50.29%	50.74%	35.67%
13	Fungi (Subphylum)	97.10%	$97.10 \pm 0.96\%$	97.94%	59.72%	75.14%	42.52%

Table 5.9: CGRclust performance of clustering Group 3 dataset (mtDNA of Insects, Protists, and Fungi) described in Table 5.3. CGRclust (with and without an adjusted α hyperparameter) accuracy is compared with DeLUCS, *i*DeLUCS, and MeShClust v3.0 (with both a manual, and the automatic selection of identity score threshold). Confidence intervals, indicating the statistical range of accuracy, are shown for CGRclust.

Finally, for a direct comparison with MeShClust v3.0, Table 5.10 summarizes the accuracies of clustering Group 4 dataset (the twelve synthetic datasets from [95] and described in Table 5.4), for all methods. In the Group 4 dataset, the terms “Medium-” and “Long-” in the dataset names indicate the sequence lengths. The numerical values ranging from 60 to 97 in the dataset names represent the identity score, a measure of sequence similarity. As this identity score increases, the sequences within a cluster become more similar, and this typically leads to enhanced performance of the clustering method. From the table, it is evident that CGRclust maintains a consistently high clustering accuracy, above 90%, across both “Medium” and “Long” dataset categories. Although it does not always achieve the highest accuracy compared to the other methods, CGRclust’s performance is relatively close to DeLUCS and *i*DeLUCS.

Test	Dataset	CGRclust	DeLUCS	<i>i</i> DeLUCS	MeShClust-auto
14	Medium-60	92.26 $\pm$ 0.39%	94.97%	91.77%	99.7%
15	Medium-70	93.39 $\pm$ 0.36%	98.36%	94.40%	99.8%
16	Medium-80	94.61 $\pm$ 0.31%	99.42%	97.58%	99.8%
17	Medium-90	95.23 $\pm$ 0.29%	98.76%	97.60%	99.9%
18	Medium-95	96.57 $\pm$ 0.23%	99.73%	99.55%	100%
19	Medium-97	95.51 $\pm$ 0.28%	98.44%	98.57%	100%
20	Long-60	93.31 $\pm$ 0.34%	97.36%	94.13%	99.8%
21	Long-70	92.82 $\pm$ 0.37%	98.00%	95.40%	93.41%
22	Long-80	96.29 $\pm$ 0.26%	99.12%	97.03%	99.8%
23	Long-90	94.08 $\pm$ 0.30%	99.42%	99.13%	100%
24	Long-95	94.20 $\pm$ 0.32%	99.37%	99.67%	100%
25	Long-97	94.83 $\pm$ 0.31%	99.13%	99.11%	100%

Table 5.10: CGRclust performance of clustering Group 4 dataset (the synthetic datasets from MeShClust v3.0 [95]), described in Table 5.4. CGRclust accuracy is compared with DeLUCS, *i*DeLUCS, MeShClust v3.0. The numerical values in the dataset names (in the range [60-97]) denote an identity score threshold signifying that every sequence within a cluster falls within this threshold distance from the cluster center. Confidence intervals, indicating the statistical range of accuracy, are shown for CGRclust.

5.4.3 Summative Observations

Overall, CGRclust exhibits versatility and robustness, consistently achieving high accuracy across twenty-five diverse datasets. CGRclust proved resilient to variations in dataset size, sequence length, and similarity, effectively handling the challenges posed by different genome types and taxonomic levels. Additionally, its performance in challenging scenarios, such as unbalanced datasets (e.g., Test 5), showcased its robust performance under different conditions. Its consistent performance highlights its superior clustering capabilities and scalability compared to other established methods like DeLUCS, *i*DeLUCS, and MeShClust v3.0. for DNA clustering.

The training times for CGRclust were recorded for each clustering test, as detailed in Table 5.11. As the results show, the time required for each clustering task is directly correlated with the number of genome sequences in the dataset, indicating that increased dataset size increases computational complexity and extends clustering duration—an expected behavior in clustering algorithms. The length of the DNA sequences, however, influences the training time minimally, impacting only the pre-processing phase. In the pre-processing

phase, genome sequences are transformed into Frequency Chaos Game Representations (FCGRs), converting them into 64×64 matrices when $k = 6$. This transformation ensures uniformity in data handling, as all sequences, regardless of their original lengths, are represented by matrices of the same dimensions. Consequently, after this conversion to FCGRs, the clustering algorithm processes datasets of these fixed-sized matrices, making the original sequence lengths irrelevant in terms of computational load during the clustering stage. As a result of this fixed-size representation, the clustering algorithm is focused on analyzing patterns within FCGRs rather than managing different sequence lengths.

5.5 Discussion

This chapter explored the novel application of twin contrastive clustering of DNA sequences using Chaos Game Representation (CGR) to the field of bioinformatics, particularly to the unsupervised clustering of DNA sequences. The findings from this chapter provide a new perspective on the potential for unsupervised clustering methods, originally designed for computer vision, to achieve high accuracy in DNA classification/clustering tasks, traditionally dominated by supervised learning.

Implementing this methodology required developing a robust algorithm capable of handling diverse genomic data types, ensuring consistent performance across different datasets, including fish mitochondrial genomes (Cypriniformes order) at four taxonomic levels, as well as five different viral genomic datasets at genus or virus subtype levels. CGRclust achieved a high accuracy even when used with an unbalanced dataset in Test 5 (the accuracy of CGRclust was 85%, while the accuracies of the other methods were 15% to 34% lower), demonstrating its effectiveness in managing uneven data distributions. To ensure comprehensive evaluation and demonstrate the algorithm’s versatility, we expanded our dataset selection to include datasets previously analyzed by other studies (i.e. *i*DeLUCS [202] and MeShClust v3.0 [95]). This inclusion allowed us to perform direct comparisons and validate the effectiveness of CGRclust across diverse genomic datasets. CGRclust successfully clustered all twenty-five tested datasets, which varied in length from 664 bp to approximately 100 kbp, covering a diverse range of cluster counts and sequence numbers. One of the primary challenges was optimizing the contrastive learning process to improve both the efficiency and accuracy of the clustering results. An effective pipeline that integrates data augmentation (generation of the mimic sequences), feature extraction, and twin contrastive learning mechanisms successfully addressed this issue. It is important to note that, although this study focused on DNA sequences in the clustering experiments, CGRclust could also be applied to RNA analysis. This is due to the fact that both DNA

Test	Dataset	Number of Sequences	Time (seconds)
1	Cypriniformes	498	551.240
2	Cyprinoidei	630	634.810
3	Cyprinidae	448	476.283
4	Cyprininae	213	413.320
5	Astroviridae-unbalanced	1,089	843.984
6	Astroviridae-balanced	726	712.654
7	Dengue	1,628	1146.037
8	HCV	950	720.595
9	HIV-1	1300	961.421
10	Insecta	4,550	2255.074
11	Protista	945	725.403
12	Fungi-phylum	670	662.347
13	Fungi-subphylum	1,070	859.819
14	Medium-60	18,210	7973.674
15	Medium-70	18,731	8170.432
16	Medium-80	20,939	9167.305
17	Medium-90	21,266	9275.432
18	Medium-95	24,039	8764.424
19	Medium-97	20,772	8731.778
20	Long-60	20,885	8941.603
21	Long-70	18,558	8023.114
22	Long-80	20,525	8886.094
23	Long-90	22,518	9592.560
24	Long-95	20,222	8764.240
25	Long-97	19,960	8731.778

Table 5.11: CGRclust training time across twenty-five clustering tests, with the shortest being 413 seconds (almost 7 minutes) in Test 4, and the longest being 10,371 seconds (almost 3 hours) in Test 18, dependent on the sequence count.

and RNA are sequences made up of four “letters,” that can each act as the label of one of the four corners of a CGR square.

The applicability of CGRclust has been primarily evaluated using the datasets mentioned, but further extensive validation across a wider range of DNA clustering tasks is necessary. This includes testing on DNA sequences longer than 100 kb, with a higher number of genome sequences per cluster, and a greater number of clusters, to confirm its general applicability. Beyond taxonomic clustering, CGRclust could also be explored in other contexts such as exploring the impact of extreme environments on genomic signatures, and virus-host genomic signature similarity.

Additionally, while CGRclust is more time-efficient compared to alignment-based methods and comparable to other clustering methods evaluated, it can still be time-consuming, especially when applied to large datasets. This limitation, which comes from the substantial batch sizes required for effective contrastive learning, could limit CGRclust’s practicality in settings where rapid processing of genomic data is required. For the purpose of rapidly estimating evolutionary distances for closely related sequences without relying on labelled data, other tools such as Mash [218], ‘andi’ [110], and phylonium [155] exist. As detailed in Appendix B, our experiments confirmed that phylonium performs efficiently on datasets used in Tests 1, 2, 3, and 4 (mtDNA of Cypriniformes), and Test 9 (HIV-1 genomes), generating evolutionary distance matrices in under a minute. However, for the remaining twenty datasets, characterized by more heterogeneous sequences, phylonium aborted the task and generated matrices with NaN values. This demonstrates that CGRclust is applicable to a wider range of datasets than phylonium, as it effectively clusters datasets containing dissimilar and non-alignable sequences that cannot be classified by tools optimized for closely related sequences. Note, we selected phylonium for our comparative evaluation, as this method showed superior accuracy compared to Mash and ‘andi’ in generating evolutionary distance matrices despite its slower performance [155], which aligns with our focus on accuracy for genomic analyses. Ultimately, the determination of the optimal clustering tool has to be guided by the specifics of the application.

Another limitation of CGRClust is finding a set of hyperparameters that is universally effective across different types of tests, which has proven to be challenging and may indeed be impossible given the diversity in genomic data and clustering objectives. In other words, each type of dataset may require individual finetuning of the model’s hyperparameters in order to achieve optimal accuracy, and this can significantly increase the complexity and duration of the initial set-up.

In light of these limitations, future work should focus on optimizing the computational efficiency of the method, exploring its scalability across diverse genomic datasets, and

developing adaptive hyperparameter tuning mechanisms that can respond dynamically to the characteristics of the data being processed.

5.6 Conclusions

This chapter introduced CGRclust, a novel twin contrastive clustering algorithm for the taxonomic clustering of unlabelled DNA sequences. CGRclust utilizes unsupervised machine learning to identify relevant and discriminative patterns in unlabelled, primary DNA sequence data, without relying on homology, sequence alignment, or any biological and taxonomic labelling. CGRclust achieves high clustering accuracies by combining the visual Chaos Game Representation of DNA sequences, with recent advancements in unsupervised learning for computer vision, namely twin contrastive learning and convolutional neural networks. It successfully clusters different datasets including full mitochondrial DNA genomes from fish, fungi, protists, and viral whole genomes across different taxonomic levels from phyla to intraspecific subtypes. Remarkably, CGRclust obtained high accuracy when encountering cluster imbalance in a dataset, showcasing its robustness with uneven data distributions. CGRclust achieves higher or comparable clustering accuracies compared with state-of-the-art existing unsupervised machine learning clustering methods, across all datasets tested. Notably, in 11 out of 13 real datasets, CGRclust achieved accuracy greater than 80%. In comparison, the DeLUCS algorithm surpassed this accuracy threshold in 7 out of 13 tests, *i*DeLUCS in only 5 tests, and MeShClust v3.0 only once. This demonstrates that CGRclust’s performance is more consistently reliable than other methods. In particular, CGRclust performed especially well on viral datasets, where it consistently achieved the highest accuracies.

Chapter 6

Summary and Future Directions

The primary motivation for this dissertation stems from the significant challenges in taxonomic classification and clustering of DNA sequences, particularly due to the labor-intensive and time-consuming nature of traditional methods and the vast diversity of newly discovered species. These traditional methods, often reliant on extensive manual labeling and multiple sequence alignments, prove inadequate for handling large genomic datasets or analyzing distantly related organisms, emphasizing the need for innovative unsupervised clustering techniques. This work aimed to develop deep learning methodologies that transition from traditional classification systems towards a more scalable and flexible approach that directly learns from genomic data in an unsupervised manner. These methods are designed to be independent of time-consuming sequence alignment, thereby enhancing their applicability in handling the vast diversity and volume of genomic sequences.

This dissertation introduces DeLUCS, a deep learning-based framework employing contrastive learning for the unsupervised clustering of DNA sequences. Specifically designed to address the complexities of genomic datasets, DeLUCS utilizes data augmentation—a crucial aspect of contrastive learning—to generate artificial *mimic sequences* using a probabilistic model. Mimic sequences enable the model to develop more robust and generalizable representations of DNA, enhancing its learning capabilities. Experimental results demonstrate that DeLUCS effectively clusters DNA sequences without reliance on sequence homology, variations in sequence length, or the lack or instability of sequence annotations and taxonomic identifiers. This showcases its substantial ability to manage large and genetically diverse datasets, including mtDNA, bacterial, and viral sequences. Additionally, DeLUCS incorporates Invariant Information Clustering (IIC), a technique that has received significant attention in the field of computer vision for its effectiveness in extracting robust features from complex image data structures. The IIC loss function, a crucial compo-

nent of the DeLUCS pipeline, optimizes mutual information between cluster assignments of a sequence and its corresponding mimic sequences, ensuring that the model captures meaningful biological patterns and dependencies despite the variability inherent in genomic sequences.

However, transitioning from DeLUCS’s strengths to its limitations, it is essential to consider certain challenges that may impact the future scalability and practical implementation of DeLUCS. First, the reliance on data augmentation through mimic sequences, while beneficial for learning robust features, can introduce significant computational overhead in the training process. This overhead primarily arises from the computational demands of generating and processing large quantities of augmented data, which may limit scalability and efficiency, particularly in resource-constrained environments. Second, while DeLUCS demonstrates versatility, its performance can vary significantly across different types of genomic data, which may require careful tuning of parameters and adjustments in the contrastive learning setup to achieve optimal results. Additionally, DeLUCS’s effectiveness relies on the assumption of dataset cluster balance, which is often not the case in real-world scenarios, particularly when labels are unavailable. Without access to labels, determining the balance of the data becomes challenging, potentially leading to skewed results and inferior model performance. Furthermore, while DeLUCS primarily employs flattened FCGR for clustering, this format may not capture all the complex genomic information available. Employing two-dimensional FCGR can fully leverage the detail in this powerful DNA representation, potentially leading to a deeper and more comprehensive understanding of these sequences.

Next, a notable part of this research focused on the taxonomic classification of emerging astroviruses. A robust, multi-pronged approach, 3PCM, was developed that combines supervised and unsupervised machine learning techniques in addition to traditionally used host information. A critical component of 3PCM is the initial step of eliminating recombinant sequences, which enhances the accuracy of the classification process by reducing the confounding effects of recombination. This step ensures that only stable, non-recombinant sequences are used, leading to more reliable taxonomic predictions. 3PCM enhances the classification accuracy of astroviruses, contributing significantly to our understanding of this viral family evolution and interspecies transmission. While 3PCM was specifically designed for the classification of astroviruses, the framework is versatile and can be adapted for the classification of other viral families, offering a scalable and flexible solution for viral taxonomic studies. The methodology provides a fast, reliable, and scalable prediction method of taxonomic labels, able to keep pace with emerging viruses and the exponential increase in the output of modern genome sequencing technologies. Despite its strengths, 3PCM faces challenges, particularly in classifying astrovirus sequences from hosts other

than avian and mammalian species, due to the limited availability of data from these groups. Nevertheless, the continued development and refinement of scalable and powerful viral classification methods such as 3PCM remain critical in facilitating the identification and classification of emerging viruses, thereby aiding in disease management and effective prevention strategies.

While DeLUCS provides a robust framework for clustering various genomic sequences, its application in specific contexts, particularly in the clustering of emerging astroviruses as explored in the 3PCM approach, revealed limitations. For instance, the simpler K -Means clustering algorithm was found to be more effective in handling the distinct characteristics and distribution patterns of astrovirus sequences, offering both higher accuracy and greater computational efficiency due to its simplicity. The challenges encountered with DeLUCS, particularly its complex computational requirements and sensitivity to data distribution, highlighted the necessity for a more adaptable and robust method for clustering DNA sequences. These challenges prompted the development of CGRclust, a novel DNA clustering approach utilizing twin contrastive learning (TCL). TCL optimizes two contrastive learning objectives simultaneously—one at the instance-level and another at the cluster-level, enhancing the robustness and accuracy of the model. Originally designed for computer vision, TCL has been successfully applied to the clustering of images and text data. Motivated by this success, we decided to explore its potential in genomics to address the identified limitations of DeLUCS.

Similar to DeLUCS, CGRclust uses contrastive learning and advanced probabilistic data augmentation techniques to identify relevant and discriminative patterns in unlabelled, primary DNA sequence data. By incorporating FCGR into its pipeline, CGRclust fully leverages the two-dimensional information intrinsic to this DNA representation. This use of a spatial matrix format significantly enhances the model’s ability to detect subtle patterns and correlations that may be obscured when using one-dimensional data formats. This represents a substantial advancement over DeLUCS, offering a more nuanced analysis and understanding of genomic complexities. CGRclust successfully clustered different datasets including mtDNA from fish, fungi, protists, and viral whole genomes across different taxonomic levels from phyla to intraspecific subtypes. Notably, it was also successful in achieving high accuracy in clustering astrovirus sequences, demonstrating its adaptability and effectiveness across diverse viral families. Remarkably, CGRclust obtained high accuracy even when encountering cluster imbalance in a dataset, showcasing its robustness with uneven data distributions. However, the applicability of CGRclust has been primarily evaluated using specific datasets, and extensive validation across a broader range of DNA clustering tasks is necessary to confirm its general applicability. Additionally, while CGRclust is time-efficient compared to alignment-based methods, it can still be time-

consuming when applied to large datasets, potentially limiting its practicality in settings where rapid genomic data processing is required. These challenges highlight the need for ongoing refinement to enhance CGRclust’s efficiency and versatility in various genomic contexts. CGRclust sets a new standard in genomic data analysis through unsupervised learning, addressing the shortcomings observed in DeLUCS while expanding the scope of applications to more complex genomic datasets.

The development and implementation of these methods have advanced the field of bioinformatics, particularly within the realm of unsupervised learning. These methodologies enhance the precision and efficiency of genomic data analysis and can open new avenues for research in biological data interpretation, significantly impacting our understanding of genetic diversity and evolutionary biology. Moving forward, it is crucial to enhance the scalability of these models to manage increasing volumes of genomic data, improve their robustness against dataset variability, and expand their applicability to other genomic areas such as metagenomics and pathogen identification. Additionally, exploring the potential of transfer learning from well-established models in fields like computer vision could provide breakthroughs, particularly in viewing FCGRs as images. Another promising avenue is the adoption of more sophisticated DNA representations, moving beyond CGR and k -mer frequency vectors to include advanced subword tokenization methods, which could offer more comprehensive insights into genomic complexities. Bridging the gap between theoretical models and practical applications is essential to ensure these advanced machine learning techniques are effectively used in real-world scenarios, including emergent virus classification with implications for public health.

In summary, this dissertation has addressed critical challenges in the classification and clustering of DNA sequences, significantly advancing the field of bioinformatics through innovative machine learning techniques. By introducing novel methodologies, this work establishes a robust foundation for future genomic studies and opens new avenues for practical applications in areas such as taxonomic categorization and virus subtyping. Ultimately, this dissertation enhances DNA analysis with advanced machine learning techniques, setting a new standard in computational bioinformatics. It provides scalable and accurate strategies that operate effectively without reliance on sequence alignment and annotations, aligning with the rapid growth in modern genome sequencing technologies.

References

- [1] F. X. Abad, R. M. Pintó, C. Villena, R. Gajardo, and A. Bosch. Astrovirus survival in drinking water. *Applied and Environmental Microbiology*, 63(8):3119–3122, 1997.
- [2] Emmanuel Adetiba and Oludayo O. Olugbara. Classification of eukaryotic organisms through cepstral analysis of mitochondrial DNA. In *Image and Signal Processing*, pages 243–252. Springer, 2016.
- [3] Vera Afreixo, Carlos A. C. Bastos, Armando J. Pinho, Sara P. Garcia, and Paulo J. S. G. Ferreira. Genome analysis with distance to the nearest dissimilar nucleotide. *Journal of Theoretical Biology*, 275(1):52–58, 2011.
- [4] Charu C. Aggarwal. *Neural networks and deep learning: A textbook*. Springer, 2018.
- [5] Charu C. Aggarwal and Chandan K. Reddy. *Data clustering: Algorithms and applications*. CRC Press, 1st edition, 2014.
- [6] 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. IEEE, 2007.
- [7] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular biology of the cell*. Garland Science, 4th edition, 2002.
- [8] 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. IEEE, 2015.
- [9] Fatemeh Alipour, Kathleen A. Hill, and Lila Kari. CGRclust: Chaos game representation for twin contrastive clustering of unlabelled DNA sequences. *BMC genomics*, 25:1214, 2024.

- [10] Fatemeh Alipour, Connor Holmes, Yang Young Lu, Kathleen A. Hill, and Lila Kari. Leveraging machine learning for taxonomic classification of emerging astroviruses. *Frontiers in Molecular Biosciences*, 10:1305506, 2024.
- [11] Jonas S. Almeida, Joao A. Carrico, Antonio Maretzek, Peter A. Noble, and Madilyn Fletcher. Analysis of genomic sequences by chaos game representation. *Bioinformatics*, 17(5):429–437, 2001.
- [12] Naomi S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [13] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
- [14] W. L. Applequist. A brief review of recent controversies in the taxonomy and nomenclature of *Sambucus nigra* sensu lato. *Acta Horticulturae*, (1061):25–33, 2015.
- [15] David Arthur and Sergei Vassilvitskii. *K*-Means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [16] 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, 2023.
- [17] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, 2021.
- [18] Igor V. Babkin, Artem Y. Tikunov, Elena V. Zhirakovskaia, Sergei V. Netesov, and Nina V. Tikunova. High evolutionary rate of human astrovirus. *Infection, Genetics and Evolution*, 12(2):435–442, 2012.
- [19] Junpeng Bao, Ruiyu Yuan, and Zhe Bao. An improved alignment-free model for DNA sequence similarity metric. *BMC Bioinformatics*, 15:321, 2014.
- [20] 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.

- [21] Michael F. Barnsley. *Fractals everywhere: New edition*. Academic Press, 1988.
- [22] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [23] Guillaume Bernard, Cheong Xin Chan, and Mark A. Ragan. Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer. *Scientific Reports*, 6(1):28970, 2016.
- [24] Christopher M. Bishop. *Pattern recognition and machine learning (information science and statistics)*. Springer-Verlag, 2006.
- [25] Vincenzo Bonnici and Vincenzo Manca. Informational laws of genome structures. *Scientific Reports*, 6(1):28840, 2016.
- [26] 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):e110954, 2014.
- [27] Albert Bosch, Rosa M. Pintó, and Susana Guix. Human astroviruses. *Clinical Microbiology Reviews*, 27(4):1048–1074, 2014.
- [28] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [29] Leo Breiman, Jerome Friedman, and Charles J. Stone. *Classification and regression trees*. Chapman and Hall, Abingdon, UK, 1984.
- [30] Florian P. Breitwieser, Daniel N. Baker, and Steven L. Salzberg. KrakenUniq: Confident and fast metagenomics classification using unique k -mer counts. *Genome Biology*, 19:1–10, 2018.
- [31] Steven E. Brenner, Cyrus Chothia, and Tim J. P. Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences*, 95(11):6073–6078, 1998.
- [32] Cecil C. Bridges Jr. Hierarchical cluster analysis. *Psychological Reports*, 18(3):851–854, 1966.
- [33] Theresa Bub, Virginia Hargest, Shaoyuan Tan, Maria Smith, Ana Vazquez-Pagan, Tim Flerlage, Pamela Brigleb, Victoria Meliopoulos, Brett Lindenbach, Harish N. Ramanathan, Valerie Cortez, Jeremy Chase Crawford, and Stacey Schultz-Cherry. Astrovirus replication is dependent on induction of double-membrane vesicles through a PI3K-dependent, LC3-independent pathway. *Journal of Virology*, 97(9), 2023.

- [34] Arif Budiarto, Bharuno Mahesworo, Alam Ahmad Hidayat, Ika Nurlaila, and Bens Pardamean. Gaussian mixture model implementation for population stratification estimation from genomics data. *Procedia Computer Science*, 179:202–210, 2021.
- [35] Pradeep Kumar Burma, Alok Raj, Jayant K. Deb, and Samir K. Brahmachari. Genome analysis: A new approach for visualization of sequence organization in genomes. *Journal of Biosciences*, 17:395–411, 1992.
- [36] Helen Y. Buse, Jingrang Lu, Ian T. Struewing, and Nicholas J. Ashbolt. Eukaryotic diversity in premise drinking water using 18S rDNA sequencing: Implications for health risks. *Environmental Science and Pollution Research*, 20(9):6351–6366, 2013.
- [37] A. Bustamam, H. Tasman, N. Yuniarti, Frisca, and I. Mursidah. Application of K -Means clustering algorithm in grouping the DNA sequences of hepatitis B virus (HBV). *AIP Conference Proceedings*, 1862(1):030134, 2017.
- [38] Marco Cacciabue, Pablo Aguilera, María Inés Gismondi, and Oscar Taboga. Covidex: An ultrafast and accurate tool for SARS-CoV-2 subtyping. *Infection, Genetics and Evolution*, 99:105261, 2022.
- [39] Neil J. Calkin, Eunice Y. S. Chan, and Robert M. Corless. *Computational discovery on Jupyter*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2023.
- [40] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149. Springer, 2018.
- [41] J. Avila Cartes. Complex CGR. <https://github.com/AlgoLab/complexCGR>, 2024. Accessed: 24 March 2024.
- [42] Antonio Bernardo Carvalho, Eduardo G Dupim, and Gabriel Goldstein. Improved assembly of noisy long reads by k -mer validation. *Genome Research*, 26(12):1710–1720, 2016.
- [43] Su-Beom Chae, Chang-Gi Jeong, Jun-Soo Park, Eun-Jee Na, and Jae-Ku Oem. Detection and genetic characterization of astroviruses in brain tissues of wild raccoon dogs. *Viruses*, 15(7):1488, 2023.
- [44] Alexander Chan. An analysis of pairwise sequence alignment algorithm complexities: Needleman-Wunsch, Smith-Waterman, FASTA, BLAST and gapped BLAST. Stanford University, 2004.

- [45] Tony F. Chan, Gene Howard Golub, and Randall J. LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. In *COMPSTAT 1982, 5th Symposium held at Toulouse 1982*, pages 30–41. Springer, 1982.
- [46] 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.
- [47] Liang Chen, Weinan Wang, Yuyao Zhai, and Minghua Deng. Deep soft K -Means clustering with self-training for single-cell RNA sequence data. *NAR Genomics and Bioinformatics*, 2(2):lqaa039, 2020.
- [48] 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, pages 1597–1607. PMLR, 2020.
- [49] Ramu Chenna, Hideaki Sugawara, Tadashi Koike, Rodrigo Lopez, Toby J. Gibson, Desmond G. Higgins, and Julie D. Thompson. Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Research*, 31(13):3497–3500, 2003.
- [50] Preeti Chhabra, Daniel C. Payne, Peter G. Szilagyi, Kathryn M. Edwards, Mary Allen Staat, S. Hannah Shirley, Mary Wikswo, W. Allan Nix, Xiaoyan Lu, Umesh D. Parashar, and Jan Vinjé. Etiology of viral gastroenteritis in children <5 years of age in the United States, 2008–2009. *Journal of Infectious Diseases*, 208(5):790–800, 2013.
- [51] Allen Chieng Hoon Choong and Nung Kion Lee. Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method. In *2017 International Conference on Computer and Drone Applications (ICoNDA)*, pages 60–65. IEEE, 2017.
- [52] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, pages 539–546. IEEE, 2005.
- [53] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

- [54] Matteo Comin and Morris Antonello. Fast entropic profiler: An information theoretic approach for the discovery of patterns in genomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(3):500–509, 2014.
- [55] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [56] Valerie Cortez, Victoria A. Meliopoulos, Erik A. Karlsson, Virginia Hargest, Cydney Johnson, and Stacey Schultz-Cherry. Astrovirus biology and pathogenesis. *Annual Review of Virology*, 4(1):327–348, 2017.
- [57] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.
- [58] Paul Dan Cristea. Conversion of nucleotide sequences into genomic signals. *Journal of Cellular and Molecular Medicine*, 6(2):279–303, 2002.
- [59] 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.
- [60] Edwin S. Dalmaijer, Camilla L. Nord, and Duncan E. Astle. Statistical power for cluster analysis. *BMC Bioinformatics*, 23(1):1–28, 2022.
- [61] Kwabena Darkwa, Paterne Agre, Bunmi Olasanmi, Kohtaro Iseki, Ryo Matsumoto, Adrian Powell, Guillaume Bauchet, David De Koeyer, Satoru Muranaka, Patrick Adebola, Robert Asiedu, Ryohei Terauchi, and Asrat Asfaw. Comparative assessment of genetic diversity matrices and clustering methods in white Guinea yam (*Dioscorea rotundata*) based on morphological and molecular markers. *Scientific Reports*, 10(1):13191, 2020.
- [62] Bernardo P. de Almeida, Hugo Dalla-Torre, Guillaume Richard, Christopher Blum, Lorenz Hexemer, Maxence Gélard, Priyanka Pandey, Stefan Laurent, Alexandre Lat-erre, Maren Lang, Uğur Şahin, Karim Beguir, and Thomas Pierrot. SegmentNT: Annotating the genome at single-nucleotide resolution with DNA foundation models. *bioRxiv*, 2024.
- [63] Paola De Benedictis, Stacey Schultz-Cherry, Andrew Burnham, and Giovanni Cat-toli. Astrovirus infections in humans and animals – Molecular biology, genetic diver-sity, and interspecies transmissions. *Infection, Genetics and Evolution*, 11(7):1529–1544, 2011.

- [64] Rebeca De la Fuente, Wladimiro Díaz-Villanueva, Vicente Arnau, and Andrés Moya. Genomic signature in evolutionary biology: A review. *Biology*, 12(2):322, 2023.
- [65] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.
- [66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.
- [67] Penelope H. Dennehy, Sara M. Nelson, Sara Spangenberg, Jacqueline S. Noel, Stephan S. Monroe, and Roger I. Glass. A prospective case-control study of the role of astrovirus in acute diarrhea among hospitalized young children. *The Journal of Infectious Diseases*, 184(1):10–15, 2001.
- [68] 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.
- [69] Clécio Danilo Dias-da Silva. Aristotle’s and the classification of living beings: Historical review and contributions to science teaching. *Unisanta BioScience*, 7(4):366–378, 2018.
- [70] Celeste Donato and Dhanasekaran Vijaykrishna. The broad host range and genetic diversity of mammalian and avian astroviruses. *Viruses*, 9(5):102, 2017.
- [71] Edo Dotan, Gal Jaschek, Tal Pupko, and Yonatan Belinkov. Effect of tokenization on transformers for biological sequences. *Bioinformatics*, 40(4):btac196, 2024.
- [72] Veronika B. Dubinkina, Dmitry S. Ischenko, Vladimir I. Ulyantsev, Alexander V. Tyakht, and Dmitry G. Alexeev. Assessment of k -mer spectrum applicability for metagenomic dissimilarity analysis. *BMC Bioinformatics*, 17:1–11, 2016.
- [73] Patrik D’haeseleer. How does gene expression clustering work? *Nature Biotechnology*, 23(12):1499–1501, 2005.
- [74] Robert C. Edgar. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

- [75] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.
- [76] Raphael Eisenhofer and Laura Susan Weyrich. Assessing alignment-based taxonomic classification of ancient microbial DNA. *PeerJ*, 7:e6594, 2019.
- [77] Mohamed Emam, Amna Ali, Eman Abdelrazik, Mustafa Elattar, and Mohamed El-Hadidi. Detection of mammalian coding sequences using a hybrid approach of chaos game representation and machine learning. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2949–2951. IEEE, 2020.
- [78] Gökçen Eraslan, Žiga Avsec, Julien Gagneur, and Fabian J. Theis. Deep learning: New computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403, 2019.
- [79] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.
- [80] Anna Fabijańska and Szymon Grabowski. Viral genome deep classifier. *IEEE Access*, 7:81297–81307, 2019.
- [81] Zhiguo Fei, Anqi Jiao, Minli Xu, Jiaqiang Wu, Yu Wang, Jiang Yu, Lu Lu, Wanchun Jiang, Gaungwei Zhu, Wenbo Sun, Zhi Chen, Yuyu Zhang, Sufang Ren, Fei Liu, and Lin Zhang. Genetic diversity and evolution of goose astrovirus in the east of China. *Transboundary and Emerging Diseases*, 69(5):e2059–e2072, 2022.
- [82] Izaskun Fernández-Correa, Daniel A. Truchado, Esperanza Gomez-Lucia, Ana Doménech, Javier Pérez-Tris, Jonas Schmidt-Chanasit, Daniel Cadar, and Laura Benítez. A novel group of avian astroviruses from Neotropical passerine birds broaden the diversity and host range of *Astroviridae*. *Scientific Reports*, 9(1):9513, 2019.
- [83] Stephen P. Ficklin, Leland J. Dunwoodie, William L. Poehlman, Christopher Watson, Kimberly E. Roche, and F. Alex Feltus. Discovering condition-specific gene co-expression patterns using Gaussian mixture models: A cancer case study. *Scientific Reports*, 7(1):8617, 2017.
- [84] Ewa Filip, Tomasz Strzała, Edyta Stepień, and Danuta Cembrowska-Lech. Universal mtDNA fragment for Cervidae barcoding species identification using phylogeny and

- preliminary analysis of machine learning approach. *Scientific Reports*, 13(1):9133, 2023.
- [85] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.
 - [86] Veniamin Fishman, Yuri Kuratov, Maxim Petrov, Aleksei Shmelev, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. GENA-LM: A family of open-source foundational models for long DNA sequences. *bioRxiv*, 2023.
 - [87] Jane Flint, Vincent R. Racaniello, Glenn F. Rall, Theodora Hatzioannou, and Anna Marie Skalka. *Principles of virology, Volume 1: Molecular biology*. John Wiley & Sons, 5th edition, 2020.
 - [88] Johannes Forkman, Julie Josse, and Hans-Peter Piepho. Hypothesis tests for principal component analysis when variables are standardized. *Journal of Agricultural, Biological and Environmental Statistics*, 24:289–308, 2019.
 - [89] Philip Fradkin, Ruian Shi, Bo Wang, Brendan Frey, and Leo J. Lee. Splicing up your predictions with RNA contrastive learning. *arXiv preprint arxiv:2310.08738*, 2023.
 - [90] Rebecca K. French, Antoine Filion, Chris N. Niebuhr, and Edward C. Holmes. Meta-transcriptomic comparison of viromes in endemic and introduced passerines in New Zealand. *Viruses*, 14(7):1364, 2022.
 - [91] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and Systems Sciences*, 55(1):119–139, 1997.
 - [92] Gihan Gamage, Nadeeshan Gimhana, Indika Perera, Shanaka Bandara, Thilina Pathirana, Anuradha Wickramarachchi, and Vijini Mallawaarachchi. Phylogenetic tree construction using k -mer forest-based distance calculation. *International Journal of Online and Biomedical Engineering (iJOE)*, 16(7):4–20, 2020.
 - [93] Michael A. Gates. A simple way to look at DNA. *Journal of Theoretical Biology*, 119(3):319–328, 1986.
 - [94] Adrian J. Gibbs, Mohammad Hajizadeh, Kazusato Ohshima, and Roger A. C. Jones. The potyviruses: An evolutionary synthesis is emerging. *Viruses*, 12(2):132, 2020.

- [95] 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.
- [96] Hani Z. Girgis, Benjamin T. James, and Brian B. Luczak. Identity: Rapid alignment-free prediction of sequence alignment identity scores using self-supervised general linear models. *NAR Genomics and Bioinformatics*, 3(1):lqab001, 2021.
- [97] Nick Goldman. Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. *Nucleic Acids Research*, 21(10):2487–2491, 1993.
- [98] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.
- [99] Osamu Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162(3):705–708, 1982.
- [100] Sara Green, Annamaria Carusi, and Klaus Hoeyer. Plastic diagnostics: The remaking of disease and evidence in personalized medicine. *Social Science & Medicine*, 304, 2022.
- [101] Gary S. Grohmann, Roger I. Glass, Helio G. Pereira, Stephan S. Monroe, Allen W. Hightower, Rainer Weber, and Ralph T. Bryan. Enteric viruses and diarrhea in HIV-infected patients. *New England Journal of Medicine*, 329(1):14–20, 1993.
- [102] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.
- [103] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [104] Muhammed S. Hammad, Vidan F. Ghoneim, Mai S. Mabrouk, and Walid I. Al-Atabany. A hybrid deep learning approach for COVID-19 detection based on genomic image processing techniques. *Scientific Reports*, 13(1):4003, 2023.
- [105] Guo-Sheng Han, Qi Li, and Ying Li. Comparative analysis and prediction of nucleosome positioning using integrative feature representation and machine learning algorithms. *BMC Bioinformatics*, 22(6):1–23, 2021.

- [106] Virginia Hargest, Amy E. Davis, Shaoyuan Tan, Valerie Cortez, and Stacey Schultz-Cherry. Human astroviruses: A tale of two strains. *Viruses*, 13(3):376, 2021.
- [107] Mubashir Hassan, Faryal Mehwish Awan, Anam Naz, Enrique J deAndrés Galiana, Oscar Alvarez, Ana Cernea, Lucas Fernández-Brillet, Juan Luis Fernández-Martínez, and Andrzej Kloczkowski. Innovations in genomics and big data analytics for personalized medicine and health care: A review. *International Journal of Molecular Sciences*, 23(9):4645, 2022.
- [108] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*, volume 2. Springer, New York, 2009.
- [109] 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.
- [110] Bernhard Haubold, Fabian Klötzl, and Peter Pfaffelhuber. andi: Fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, 31(8):1169–1175, 2014.
- [111] 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.
- [112] Cheng He, Guifang Lin, Hairong Wei, Haibao Tang, Frank F. White, Barbara Valent, and Sanzhen Liu. Factorial estimating assembly base errors using k -mer abundance difference (KAD) between short reads and genome assembled sequences. *NAR Genomics and Bioinformatics*, 2(3):lqaa075, 2020.
- [113] 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.
- [114] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

- [115] 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.
- [116] John E. Herrmann, David N. Taylor, Peter Echeverri, and Neil R. Blacklow. Astroviruses as a cause of gastroenteritis in children. *New England Journal of Medicine*, 324(25):1757–1760, 1991.
- [117] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, et al. CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135. IEEE, 2017.
- [118] Uljana Hesse. *K*-mer-based genome size estimation in theory and practice. In *Plant Cytogenetics and Cyto-genomics: Methods and Protocols*, pages 79–113. Humana, 2023.
- [119] Hussein Hijazi and Christina Chan. A classification framework applied to cancer gene expression profiles. *Journal of Healthcare Engineering*, 4(2):255–283, 2013.
- [120] Kathleen A. Hill, Nicholas J. Schisler, and Shiva M. Singh. Chaos game representation of coding regions of human globin genes and alcohol dehydrogenase genes of phylogenetically divergent species. *Journal of Molecular Evolution*, 35:261–269, 1992.
- [121] Geoffrey E. Hinton. Connectionist learning procedures. In *Machine Learning*, pages 555–610. Elsevier, 1990.
- [122] Tung Hoang, Changchuan Yin, and Stephen S.-T. Yau. Numerical encoding of DNA sequences by chaos game representation with application in similarity comparison. *Genomics*, 108(3-4):134–142, 2016.
- [123] 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.
- [124] Jialu Hu, Jingru Wang, Jianan Lin, Tianwei Liu, Yuanke Zhong, Jie Liu, Yan Zheng, Yiqun Gao, Junhao He, and Xuequn Shang. MD-SVM: A novel SVM-based algorithm for the motif discovery of transcription factor binding sites. *BMC Bioinformatics*, 20(S7), 2019.

- [125] Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics & Proteomics*, 15(1):41–51, 2018.
- [126] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
- [127] 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.
- [128] National Human Genome Research Institute. Deoxyribonucleic Acid (DNA) Fact Sheet, <https://www.genome.gov/about-genomics/fact-sheets/Deoxyribonucleic-Acid-Fact-Sheet>. Accessed on: 2024-07-02.
- [129] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.
- [130] Ismail Jamail and Ahmed Moussa. Current state-of-the-art of clustering methods for gene expression data with RNA-Seq. In *Applications of Pattern Recognition*. IntechOpen, 2020.
- [131] 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.
- [132] Timothy Y. James, Frank Kauff, Conrad L. Schoch, P. Brandon Matheny, Valérie Hofstetter, Cymon J. Cox, Gail Celio, Cécile Gueidan, Emily Fraker, Jolanta Miadlikowska, et al. Reconstructing the early evolution of Fungi using a six-gene phylogeny. *Nature*, 443(7113):818–822, 2006.
- [133] J. Michael Janda and Sharon L. Abbott. 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls. *Journal of Clinical Microbiology*, 45(9):2761–2764, 2007.
- [134] Anna A. Jarchow-Macdonald, Shona Halley, Daniel Chandler, Rory Gunson, Samantha J. Shepherd, and Benjamin J. Parcell. First report of an astrovirus type 5 gastroenteritis outbreak in a residential elderly care home identified by sequencing. *Journal of Clinical Virology*, 73:115–119, 2015.

- [135] Erich D. Jarvis, Siavash Mirarab, Andre J. Aberer, Bo Li, Peter Houde, Cai Li, Simon Y. W. Ho, Brant C. Faircloth, Benoit Nabholz, Jason T. Howard, et al. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.
- [136] H. Joel Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170, 1990.
- [137] Hye Sook Jeong, Ah Yong Jeong, and Doo-Sung Cheon. Epidemiology of astrovirus infection in children. *Korean Journal of Pediatrics*, 55(3):77–82, 2012.
- [138] Jinuk Jeong, Kyeongeui Yun, Seyoung Mun, Won-Hyong Chung, Song-Yi Choi, Young-do Nam, Mi Young Lim, Chang Pyo Hong, ChanHyeok Park, Yong Ju Ahn, and Kyudong Han. The effect of taxonomic classification by full-length 16S rRNA sequencing with a synthetic long-read technology. *Scientific Reports*, 11(1):1727, 2021.
- [139] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874. IEEE, 2019.
- [140] 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.
- [141] B. Jiang, S. S. Monroe, E. V. Koonin, S. E. Stine, and R. I. Glass. RNA sequence of astrovirus: Distinctive genomic organization and a putative retrovirus-like ribosomal frameshifting signal that directs the viral replicase synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 90(22):10539–10543, 1993.
- [142] Jing-Zhe Jiang, Wen-Guang Yuan, Jiayu Shang, Ying-Hui Shi, Li-Ling Yang, Min Liu, Peng Zhu, Tao Jin, Yanni Sun, and Li-Hong Yuan. Virus classification for viral genomic fragments using PhaGCN2. *Briefings in Bioinformatics*, 24(1):bbac505, 2022.
- [143] Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [144] Ian T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2nd edition, 2002.

- [145] Lila Kari, Kathleen A. Hill, Abu S. Sayem, Rallis Karamichalis, Nathaniel Bryans, Katelyn Davis, and Nikesh S. Dattani. Mapping the space of genomic signatures. *PLOS ONE*, 10(5):e0119815, 2015.
- [146] Md 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, 02 2021.
- [147] Samuel Karlin and Chris Burge. Dinucleotide relative abundance extremes: A genomic signature. *Trends in Genetics*, 11(7):283–290, 1995.
- [148] Erik A. Karlsson, Christopher T. Small, Pamela Freiden, Mm Feeroz, Frederick A. Matsen, Sorn San, M. Kamrul Hasan, David Wang, Lisa Jones-Engel, and Stacey Schultz-Cherry. Non-human primates harbor diverse mammalian and avian astroviruses including those associated with human infections. *PLOS Pathogens*, 11(11):e1005225, 2015.
- [149] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, 30(14):3059–3066, 2002.
- [150] Kenneth Katz, Oleg Shutov, Richard Lapoint, Michael Kimelman, J. Rodney Brister, and Christopher O’Sullivan. The Sequence Read Archive: A decade more of explosive growth. *Nucleic Acids Research*, 50(D1):D387–D390, 2022.
- [151] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data: An introduction to cluster analysis*. John Wiley Sons, New York, 1990.
- [152] Adama Mamby Keita, Sanogo Doh, Samba O. Sow, Helen Powell, Richard Omere, M. Jahangir Hossain, Billy Ogwel, John B. Ochieng, Joquina Chiquita M. Jones, Syed M. A. Zaman, Alex O. Awuor, Jane Juma, Dilruba Nasrin, et al. Prevalence, clinical severity, and seasonality of adenovirus 40/41, astrovirus, sapovirus, and rotavirus among young children with moderate-to-severe diarrhea: Results from the vaccine impact on diarrhea in Africa (VIDA) study. *Clinical Infectious Diseases*, 76(Supplement 1):S123–S131, 2023.
- [153] Carsten Kemena and Cedric Notredame. Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics*, 25(19):2455–2465, 2009.

- [154] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [155] Fabian Klötzl and Bernhard Haubold. Phylonium: Fast estimation of evolutionary distances from large samples of similar genomes. *Bioinformatics*, 36(7):2040–2046, 2020.
- [156] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37. ACM, 2015.
- [157] Alexander Kraskov, Harald Stögbauer, Ralph G. Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *Europhysics Letters*, 70(2):278, 2005.
- [158] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.
- [159] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.
- [160] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- [161] Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. *arXiv preprint arXiv:2303.13664*, 2023.
- [162] 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.
- [163] Svetlana Lazebnik and Maxim Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1294–1309, 2009.
- [164] Huy Le, Beverly Peng, Janelle Uy, Daniel Carrillo, Yun Zhang, Brian D. Aevertmann, and Richard H. Scheuermann. Machine learning for cell type classification from single nucleus RNA sequencing data. *PLOS ONE*, 17(9):e0275070, 2022.

- [165] Pierre Le Cann, Sandrine Ranarijaona, Serge Monpoeho, Françoise Le Guyader, and Virginie Ferré. Quantification of human astroviruses in sewage using real-time RT-PCR. *Research in Microbiology*, 155(1):11–15, 2004.
- [166] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [167] Elliot J. Lefkowitz, Donald M. Dempsey, Robert Curtis Hendrickson, Richard J. Orton, Stuart G. Siddell, and Donald B. Smith. Virus taxonomy: The database of the International Committee on Taxonomy of Viruses (ICTV). *Nucleic Acids Research*, 46(D1):D708–D717, 2018.
- [168] Chris-Andre Leimeister and Burkhard Morgenstern. Kmacs: The k -mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics*, 30(14):2000–2008, 2014.
- [169] Haiqin Li, Chunhe Wan, Zhangzhang Wang, Jia Tan, Meifang Tan, Yanbing Zeng, Jiangnan Huang, Yu Huang, Qi Su, Zhaofeng Kang, and Xiaoquan Guo. Rapid diagnosis of duck Tembusu virus and goose astrovirus with TaqMan-based duplex real-time PCR. *Frontiers in Microbiology*, 14:1146241, 2023.
- [170] Linlin Li, Tongling Shan, Chunlin Wang, Colette Côté, John Kolman, David Onions, Frances M.D. Gulland, and Eric Delwart. The fecal viral flora of California sea lions. *Journal of Virology*, 85(19):9909–9917, 2011.
- [171] Ming Li and Paul M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 3rd edition, 2019.
- [172] 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.
- [173] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8547–8555, 2021.
- [174] 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.

- [175] Bo Liao and Tian-Ming Wang. New 2D graphical representation of DNA sequences. *Journal of Computational Chemistry*, 25(11):1364–1368, 2004.
- [176] Daniel Lichtblau. Alignment-free genomic sequence comparison using FCGR and signal processing. *BMC Bioinformatics*, 20:1–17, 2019.
- [177] Alle A. Y. Lie, Zhenfeng Liu, Sarah K. Hu, Adriane C. Jones, Diane Y. Kim, Peter D. Countway, Linda A. Amaral-Zettler, S. Craig Cary, Evelyn B. Sherr, Barry F. Sherr, Rebecca J. Gast, and David A. Caron. Investigating microbial eukaryotic diversity from a global census: Insights from a comparison of pyrotag and full-length sequences of 18S rRNA genes. *Applied and Environmental Microbiology*, 80(14):4363–4373, 2014.
- [178] Ken Lin, Xiongwen Quan, Wenya Yin, and Han Zhang. A contrastive learning pre-training method for motif occupancy identification. *International Journal of Molecular Sciences*, 23(9):4699, 2022.
- [179] Mary B. Liste, Ivelisse Natera, José A. Suarez, Flor H. Pujol, Ferdinando Liprandi, and Juan E. Ludert. Enteric virus infections and diarrhea in healthy and human immunodeficiency virus-infected children. *Journal of Clinical Microbiology*, 38(8):2873–2877, 2000.
- [180] Shuai Liu, Mengye Lu, Hanshuang Li, and Yongchun Zuo. Prediction of gene expression patterns with generalized linear regression model. *Frontiers in Genetics*, 10, 2019.
- [181] Ta-Chun Liu, Peter N. Kalugin, Jennifer L. Wilding, and Walter F. Bodmer. GMM-chi: Gene expression clustering using Gaussian mixture modeling. *BMC Bioinformatics*, 23(1):457, 2022.
- [182] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, Junjie Wu, and Sen Wu. Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, 43(3):982–994, 2013.
- [183] 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.
- [184] Hannah Franziska Löchel and Dominik Heider. Chaos game representation and its applications in bioinformatics. *Computational and Structural Biotechnology Journal*, 19:6263–6271, 2021.

- [185] Yi Lu, Shiyong Lu, Farshad Fotouhi, Youping Deng, and Susan J. Brown. Incremental genetic K -Means algorithm and its application in gene expression data analysis. *BMC Bioinformatics*, 5:1–10, 2004.
- [186] Daniel M. Lyons and Adam S. Luring. Evidence for the selective basis of transition-to-transversion substitution bias in two RNA viruses. *Molecular Biology and Evolution*, 34(12):3205–3215, 2017.
- [187] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, 1967.
- [188] Swati C. Manekar and Shailesh R. Sathe. Estimating the k -mer coverage frequencies in genomic datasets: A comparative assessment of the state-of-the-art. *Current Genomics*, 20(1):2–15, 2019.
- [189] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.
- [190] Darren P. Martin, Ben Murrell, Michael Golden, Arjun Khoosal, and Brejnev Muhire. RDP4: Detection and analysis of recombination patterns in virus genomes. *Virus Evolution*, 1(1):vev003, 2015.
- [191] Peter McCullagh and John A. Nelder. *Generalized linear models*, volume 37. CRC Press, New York, 2nd edition, 1989.
- [192] Catherine S. McFadden, Marymegan Daly, Mercer R. Brugler, Paulyn Cartwright, Allen G. Collins, Michael N. Dawson, Daphne G. Fautin, Scott C. France, Dennis M. Opreko, Estefania Rodriguez, Sandra L. Romano, and Joel L. Stake. The phylum Cnidaria: A review of phylogenetic patterns and diversity 300 years after Linnaeus. *Zootaxa*, 1668:127–182, 2007.
- [193] Bill McKelvey. *Organizational systematics: Taxonomy, evolution, classification*. University of California Press, 1st edition, 1982.
- [194] Geoffrey J. McLachlan. *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 2004.
- [195] Victoria A. Meliopoulos, Ghazi Kayali, Andrew Burnham, Christine M. Oshansky, Paul G. Thomas, Gregory C. Gray, Melinda A. Beck, and Stacey Schultz-Cherry. Detection of antibodies against turkey astrovirus in humans. *PLOS ONE*, 9(5):e96934, 2014.

- [196] Páll Melsted and Bjarni V. Halldórsson. KmerStream: Streaming algorithms for k -mer abundance estimation. *Bioinformatics*, 30(24):3541–3547, 2014.
- [197] 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.
- [198] Michael L. Metzker. Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.
- [199] Caroline T. Meyer, Irma K. Bauer, Martin Antonio, Mitchell Adeyemi, Debasish Saha, Joseph O. Oundo, John B. Ochieng, Richard Omoro, O. Colin Stine, David Wang, and Lori R. Holtz. Prevalence of classic, MLB-clade and VA-clade astroviruses in Kenya and The Gambia. *Virology Journal*, 12(1):78, 2015.
- [200] K. Midthun, H. B. Greenberg, J. B. Kurtz, G. W. Gary, F. Y. Lin, and A. Z. Kapikian. Characterization and seroepidemiology of a type 5 astrovirus associated with an outbreak of gastroenteritis in Marin County, California. *Journal of Clinical Microbiology*, 31(4):955–962, 1993.
- [201] Pablo 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.
- [202] Pablo 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.
- [203] Florian Mock, Fleming Kretschmer, 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), 2022.
- [204] Alfonso Monaco, Ester Pantaleo, Nicola Amoroso, Antonio Lacalamita, Claudio Lo Giudice, Adriano Fonzino, Bruno Fosso, Ernesto Picardi, Sabina Tangaro, Graziano Pesole, and Roberto Bellotti. A primer on machine learning techniques for genomic applications. *Computational and Structural Biotechnology Journal*, 19:4345–4359, 2021.

- [205] 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):e1001127, 2011.
- [206] David W. Mount. *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, NY, 2nd edition, 2004.
- [207] 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, 2008.
- [208] Niranjana Nagarajan and Mihai Pop. Sequence assembly demystified. *Nature Reviews Genetics*, 14(3):157–167, 2013.
- [209] Ashesh Nandy. A new graphical representation and analysis of DNA sequence structure: I. Methodology and application to globin genes. *Current Science*, pages 309–314, 1994.
- [210] Ashesh Nandy, Marissa Harle, and Subhash C. Basak. Mathematical descriptors of DNA sequences: development and applications. *General Papers ARKIVOC*, 9(2006):211–238, 2006.
- [211] 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.
- [212] Peng Ni, Neng Huang, Zhi Zhang, De-Peng Wang, Fan Liang, Yu Miao, Chuan-Le Xiao, Feng Luo, and Jianxin Wang. DeepSignal: Detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics*, 35(22):4586–4595, 2019.
- [213] Cédric Notredame, Desmond G. Higgins, and Jaap Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217, 2000.
- [214] Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V. Bzikadze, Alla Mikheenko, Mitchell R. Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, et al. The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022.
- [215] K.L. Oehler and R.M. Gray. Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):461–473, 1995.

- [216] Isao Oishi, Kenji Yamazaki, Tatsuo Kimoto, Yoshiichi Minekawa, Etsuko Utagawa, Shudo Yamazaki, Sakae Inouye, Gary S. Grohmann, Stephan S. Monroe, Sarah E. Stine, Cesar Carcamo, Tamie Ando, and Roger I. Glass. A large outbreak of acute gastroenteritis associated with astrovirus among students and teachers in Osaka, Japan. *Journal of Infectious Diseases*, 170(2):439–443, 1994.
- [217] Xavier Olive, Luis Basora, Benoit Viry, and Richard Alligier. Deep trajectory clustering with autoencoders. In *ICRAT 2020, 9th International Conference for Research in Air Transportation*. HAL, 2020.
- [218] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. Mash: Fast genome and metagenome distance estimation using MinHash. *Genome Biology*, 17:132, 2016.
- [219] Gregory S. Orf, Ana Olivo, Barbara Harris, Sonja L. Weiss, Asmeeta Achari, Guixia Yu, Scot Federman, Dora Mbanya, Linda James, Samuel Mampunza, Charles Y. Chiu, Mary A. Rodgers, Gavin A. Cloherty, and Michael G. Berg. Metagenomic detection of divergent insect-and bat-associated viruses in plasma from two African individuals enrolled in blood-borne surveillance. *Viruses*, 15(4):1022, 2023.
- [220] E. A. Palombo and R. F. Bishop. Annual incidence, serotype distribution, and genetic diversity of human astrovirus isolates from hospitalized children in Melbourne, Australia. *Journal of Clinical Microbiology*, 34(7):1750–1753, 1996.
- [221] Péter Pankovics, Ákos Boros, Tamás Kiss, Eric Delwart, and Gábor Reuter. Detection of a mammalian-like astrovirus in bird, European roller (*Coracias garrulus*). *Infection, Genetics and Evolution*, 34:114–121, 2015.
- [222] Mary Pantin-Jackwood, Daniel Todd, and Matthew D. Koci. Avian Astroviruses. In *Astrovirus Research*, pages 151–180. Springer, New York, NY, 2012.
- [223] 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.
- [224] Donovan H. Parks, Maria Chuvochina, David W. Waite, Christian Rinke, Adam Skarszewski, 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.

- [225] Tirthankar Paul, Seppo Vainio, and Juha Roning. Detection of intra-family coronavirus genome sequences through graphical representation and artificial neural network. *Expert Systems With Applications*, 194:116559, 2022.
- [226] Gregory B. Pauly, David Hillis, and David Cannatella. Taxonomic Freedom and the Role of Official Lists of Species Names. *Herpetologica*, 65(2):115–128, 2009.
- [227] Karl F.R.S. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- [228] William R. Pearson. An introduction to sequence similarity (“homology”) searching. *Current Protocols in Bioinformatics*, 42(1):3–1, 2013.
- [229] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [230] Lester J. Perez, Kenn Forberg, Gavin A. Cloherty, and Michael G. Berg. Temporal and coevolutionary analyses reveal the events driving the emergence and circulation of human mamastroviruses. *Emerging Microbes & Infections*, 12(1):2217942, 2023.
- [231] Natalia A. Petushkova, Mikhail A. Pyatnitskiy, Vladislav A. Rudenko, Olesya V. Larina, Oxana P. Trifonova, Julya S. Kisrieva, Natalia F. Samenkova, Galina P. Kuznetsova, Irina I. Karuzina, and Andrey V. Lisitsa. Applying of hierarchical clustering to analysis of protein patterns in the human cancer-associated liver. *PLOS ONE*, 9(8):e103950, 2014.
- [232] Thiptanawat Phongwattana, Worrawat Engchuan, and Jonathan H. Chan. Clustering-based multi-class classification of complex disease. In *2015 7th International Conference on Knowledge and Smart Technology (KST)*, pages 25–29. IEEE, 2015.
- [233] Ying Qian, Yu Zhang, Bingyu Guo, Shasha Ye, Yuzhu Wu, and Jiongmin Zhang. An improved promoter recognition model using convolutional neural network. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2018.

- [234] Muhammad I. Qureshi, Brian M. Worthington, Yongmei Liu, William Y-M. Cheung, Shuo Su, Zuoyi Zheng, Lifeng Li, Tommy T-Y. Lam, Yi Guan, and Huachen Zhu. Discovery of novel Mamastroviruses in Bactrian camels and dromedaries reveals complex recombination history. *Virus Evolution*, 9(1):veac125, 2023.
- [235] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [236] 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.
- [237] Gurjit S. Randhawa, Kathleen A. Hill, and Lila Kari. MLDSP-GUI: An alignment-free standalone tool with an interactive graphical user interface for DNA sequence comparison and analysis. *Bioinformatics*, 36(7):2258–2259, 2020.
- [238] 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.
- [239] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.
- [240] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of Database Systems*, 5:532–538, 2009.
- [241] Mohamed Amine Remita, Ahmed Halioui, Abou Abdallah Malick Diouara, Bruno Daigle, Golrokh Kiani, and Abdoulaye Baniré Diallo. A machine learning approach for viral genome classification. *BMC Bioinformatics*, 18:1–11, 2017.
- [242] Jie Ren, Nathan A. Ahlgren, Yang Young Lu, Jed A. Fuhrman, and Fengzhu Sun. VirFinder: A novel k -mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, 5:1–20, 2017.
- [243] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, Ying Wang, Gesine Reinert, and Fengzhu Sun. Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1(1):93–114, 2018.
- [244] Jie Ren, Kai Song, Chao Deng, Nathan A. Ahlgren, Jed A. Fuhrman, Yi Li, Xiaohui Xie, Ryan Poplin, and Fengzhu Sun. Identifying viruses from metagenomic data using deep learning. *Quantitative Biology*, 8(1):64–77, 2020.

- [245] Hugo Reis Resque, Veridiana Munford, Juliana Galera Castilho, Herika Schmich, Thabata Alessandra Ramos Caruzo, and Maria Lucia RÁCz. Molecular characterization of astrovirus in stool samples from children in São Paulo, Brazil. *Memórias do Instituto Oswaldo Cruz*, 102(8):969–974, 2007.
- [246] Douglas A. Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, 741:659–663, 2009.
- [247] Riccardo Rizzo, Antonino Fiannaca, Massimo La Rosa, and Alfonso Urso. Classification experiments of DNA sequences by using a deep neural network and chaos game representation. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, pages 222–228. ACM, 2016.
- [248] Tony Robinson, Jim Harkin, and Priyank Shukla. Hardware acceleration of genomics data analysis: challenges and opportunities. *Bioinformatics*, 37(13):1785–1795, 2021.
- [249] Michael S. Rosenberg. *Sequence alignment: Methods, models, concepts, and strategies*. University of California Press, 1st edition, 2009.
- [250] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
- [251] Heba Saadeh, Reem Q. Al Fayez, and Basima Elshqeirat. Application of *K*-Means clustering to identify similar gene expression patterns during erythroid development. *International Journal of Machine Learning and Computing*, 10(3):452–457, 2020.
- [252] Sarah Safoury and Walid Hussein. Enriched DNA strands classification using CGR images and convolutional neural network. In *Proceedings of the 2019 8th International Conference on Bioinformatics and Biomedical Science*, pages 87–92. ACM, 2019.
- [253] Kristoffer Sahlin. Effective sequence similarity detection with strobemers. *Genome Research*, 31(11):2080–2094, 2021.
- [254] Rashid Saif, Sadia Nadeem, Alishba Khaliq, Saeeda Zia, and Ali Iftekhhar. Mathematical understanding of sequence alignment and phylogenetic algorithms: A comprehensive review of computation of different methods. *Advancements in Life Sciences*, 9(4):401–411, 2023.
- [255] Jorge Ruben Sánchez-González and Alfredo G. Nicieza. Phenotypic convergence of artificially reared and wild trout is mediated by shape plasticity. *Ecology and Evolution*, 7(15):5922–5929, 2017.

- [256] Cristina Santos, Rafael Montiel, Blanca Sierra, Conceicao 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.
- [257] Iqbal H. Sarker. Deep Learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6), 2021.
- [258] Alexander Sasse, Bernard Ng, Anna E. Spiro, Shinya Tasaki, David A. Bennett, Christopher Gaiteri, Philip L. De Jager, Maria Chikina, and Sara Mostafavi. Benchmarking of deep neural networks for predicting personal gene expression from DNA sequence highlights shortcomings. *Nature Genetics*, 55(12):2060–2064, 2023.
- [259] Bernhard Schölkopf. The kernel trick for distances. *Advances in Neural Information Processing Systems*, 13, 2000.
- [260] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2001.
- [261] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823. IEEE, 2015.
- [262] Mike Schuster and Kaisuke Nakajima. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.
- [263] Isabel Schwende and Tuan D. Pham. Pattern recognition and probabilistic measures in alignment-free sequence analysis. *Briefings in Bioinformatics*, 15(3):354–368, 2014.
- [264] Dipendra C. Sengupta, Matthew D. Hill, Kevin R. Benton, and Hirendra N. Banerjee. Similarity studies of corona viruses through chaos game representation. *Computational Molecular Bioscience*, 10(3):61, 2020.
- [265] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [266] Ariya Shajii, Deniz Yorukoglu, Yun William Yu, and Bonnie Berger. Fast genotyping of known SNPs through approximate k -mer matching. *Bioinformatics*, 32(17):i538–i544, 2016.

- [267] Tongling Shan, Shixing Yang, Haoning Wang, Hao Wang, Ju Zhang, Ga Gong, Yuqing Xiao, Jie Yang, Xiaolong Wang, Juan Lu, Min Zhao, Zijun Yang, Xiang Lu, Ziyuan Dai, Yumin He, Xu Chen, Rui Zhou, Yuxin Yao, Ning Kong, Jian Zeng, Kalim Ullah, Xiaochun Wang, Quan Shen, Xutao Deng, Jianmin Zhang, Eric Delwart, Guangzhi Tong, and Wen Zhang. Virome in the cloaca of wild and breeding birds revealed a diversity of significant viruses. *Microbiome*, 10(1):1–21, 2022.
- [268] Sunita Shastri, Anne Martin Doane, Jaime Gonzales, Usha Upadhyayula, and Dorsey M. Bass. Prevalence of astroviruses in a children’s hospital. *Journal of Clinical Microbiology*, 36(9):2571–2574, 1998.
- [269] Muhammad Arslan Shaukat, Thanh Thi Nguyen, Edbert B. Hsu, Samuel Yang, and Asim Bhatti. Comparative study of encoded and alignment-based methods for virus taxonomy classification. *Scientific Reports*, 13(1):18662, 2023.
- [270] Fabian Sievers, Geoffrey J. Barton, and Desmond G. Higgins. Multiple sequence alignments. *Bioinformatics*, 227(1376):227–250, 2020.
- [271] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [272] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and Inanç Birol. ABySS: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.
- [273] Temple F. Smith, Michael S. Waterman, et al. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.
- [274] Vincent S. Smith. DNA barcoding: Perspectives from a “Partnerships for Enhancing Expertise in Taxonomy” (PEET) debate. *Systematic Biology*, 54(5):841–844, 2005.
- [275] Stephen Solis-Reyes, Mariano Avino, Art Poon, and Lila Kari. An open-source k -mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes. *PLOS ONE*, 13(11):e0206409, 2018.
- [276] Kai Song, Jie Ren, Gesine Reinert, Minghua Deng, Michael S. Waterman, and Fengzhu Sun. New developments of alignment-free sequence comparison: Measures, statistics and next-generation sequencing. *Briefings in Bioinformatics*, 15(3):343–353, 2014.

- [277] Peter Steinbach, Felicita Gernhardt, Mahnoor Tanveer, Steve Schmerler, and Sebastian Starke. Machine learning state-of-the-art with uncertainties. *arXiv*, 2022.
- [278] Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
- [279] Daniel Struck, Glenn Lawyer, Anne-Marie Ternes, Jean-Claude Schmit, and Danielle Perez Bercoff. COMET: Adaptive context-based modeling for ultrafast HIV-1 subtype identification. *Nucleic Acids Research*, 42(18):e144–e144, 2014.
- [280] Wing-Kin Sung. *Algorithms in bioinformatics: A practical introduction*. Chapman and Hall/CRC, 2010.
- [281] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [282] 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.
- [283] Watcharaporn Tanchotsrinon, Chidchanok Lursinsap, and Yong Poovorawan. A high performance prediction of HPV genotypes by chaos game representation and singular value decomposition. *BMC Bioinformatics*, 16:1–13, 2015.
- [284] Neda Tavakoli, Sima Siami-Namini, Mahdi Adl Khanghah, Fahimeh Mirza Soltani, and Akbar Siami Namin. An autoencoder-based deep learning approach for clustering time series data. *SN Applied Sciences*, 2:1–25, 2020.
- [285] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.
- [286] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunk centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

- [287] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.
- [288] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.
- [289] Igor Ulitsky, David Burstein, Tamir Tuller, and Benny Chor. The average common substring approach to phylogenomic reconstruction. *Journal of Computational Biology*, 13(2):336–350, 2006.
- [290] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.
- [291] Laurens Van Der Maaten, Eric O. Postma, H. Jaap Van Den Herik, et al. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(66-71):13, 2009.
- [292] Julia Van Etten, Timothy G. Stephens, and Debashish Bhattacharya. A k -mer-based approach for phylogenetic classification of taxa in environmental genomic data. *Systematic Biology*, 72(5):1101–1118, 2023.
- [293] Ragav Venkatesan and Baoxin Li. *Convolutional neural networks in visual computing: A concise guide*. CRC Press, 2017.
- [294] Susana Vinga. Information theory applications for biological sequence analysis. *Briefings in Bioinformatics*, 15(3):376–389, 2014.
- [295] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison—a review. *Bioinformatics*, 19(4):513–523, 2003.
- [296] Richard F. Voss. Evolution of long-range fractal correlations and $1/f$ noise in DNA base sequences. *Physical Review Letters*, 68(25):3805–3808, 1992.
- [297] Diem-Lan Vu, Albert Bosch, Rosa Pintó, and Susana Guix. Epidemiology of classic and novel human astrovirus: Gastroenteritis and beyond. *Viruses*, 9(2):33, 2017.
- [298] Duong Vu, Marizeth Groenewald, and Gerard Verkley. Convolutional neural networks improve fungal classification. *Scientific Reports*, 10(12628), 2020.
- [299] Michael Wainberg, Daniele Merico, Andrew Delong, and Brendan J. Frey. Deep learning in biomedicine. *Nature Biotechnology*, 36(9):829–838, 2018.

- [300] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.
- [301] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26, 2020.
- [302] James Watson, Tania Baker, Stephen Bell, Alexander Gann, Michael Levine, and Richard Losick. *Molecular biology of the gene*. Pearson, Boston, 7th edition, 2013.
- [303] Hongyu Wei, Kattareeya Kumthip, Pattara Khamrin, Arpaporn Yodmeeklin, Nutthawadee Jampanil, Phitchakorn Phengma, Zhenfeng Xie, Nuthapong Ukarapol, Hiroshi Ushijima, and Niwat Maneekarn. Triple intergenotype recombination of human astrovirus 5, human astrovirus 8, and human astrovirus 1 in the open reading frame 1a, open reading frame 1b, and open reading frame 2 regions of the human astrovirus genome. *Microbiology Spectrum*, 11(3), 2023.
- [304] Jia Wen, Raymond H.F. Chan, Shek-Chung Yau, Rong L. He, and Stephen S.T. Yau. k -mer natural vector and its application to the phylogenetic analysis of genetic sequences. *Gene*, 546(1):25–34, 2014.
- [305] Zachary Winder, Tiffany L. Sudduth, David Fardo, Qiang Cheng, Larry B. Goldstein, Peter T. Nelson, Frederick A. Schmitt, Gregory A. Jicha, and Donna M. Wilcock. Hierarchical clustering analyses of plasma proteins in subjects with cardiovascular risk factors identify informative subsets based on differential levels of angiogenic and inflammatory biomarkers. *Frontiers in Neuroscience*, 14:84, 2020.
- [306] Derrick E. Wood and Steven L. Salzberg. Kraken: Ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15:1–12, 2014.
- [307] Fang-Xiang Wu. Genetic weighted k -means algorithm for clustering large-scale gene expression data. *BMC Bioinformatics*, 9:1–10, 2008.
- [308] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, volume 48, pages 478–487. PMLR, 2016.
- [309] Aimin Yang, Wei Zhang, Jiahao Wang, Ke Yang, Yang Han, and Limin Zhang. Review on the application of machine learning algorithms in the sequence data mining of DNA. *Frontiers in Bioengineering and Biotechnology*, 8:1032, 2020.

- [310] Aimin Yang, Wei Zhang, Jiahao Wang, Ke Yang, Yang Han, and Limin Zhang. Review on the application of machine learning algorithms in the sequence data mining of DNA. *Frontiers in Bioengineering and Biotechnology*, 8, 2020.
- [311] Lianping Yang, Xiangde Zhang, Haoyue Fu, and Chenhui Yang. An estimator for local analysis of genome based on the minimal absent word. *Journal of Theoretical Biology*, 395:23–30, 2016.
- [312] Meng Yang, Yueyuxiao Yang, Chenxi Xie, Ming Ni, Jian Liu, Huanming Yang, Feng Mu, and Jian Wang. Contrastive learning enables rapid mapping to multimodal single-cell atlas of multimillion scale. *Nature Machine Intelligence*, 4:696–709, 2022.
- [313] X Yang, G-X E, B-G Yang, C-L Liu, Y Guo, Y Gong, B-E Chen, D-P Zhang, and M-H Li. Genetic diversity and phylogeny pattern across Chongqing (China) chicken populations using mtDNA D-loop sequences. *Russian Journal of Genetics*, 58(8):1007–1016, 2022.
- [314] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10):2761–2773, 2010.
- [315] Yoshua Bengio Yann LeCun and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [316] Huang Chun-Hsi Yin, Longde and Jun Ni. Clustering of gene expression data: Performance and similarity analysis. *BMC Bioinformatics*, 7(Suppliment 4), 2006.
- [317] Qingyang Yin and Liang Chen. CellTICS: An explainable neural network for cell-type identification and interpretation based on single-cell RNA-seq data. *Briefings in Bioinformatics*, 25(1), 2024.
- [318] Claude Kwe Yinda, Emiel Vanhulle, Nádia Conceição-Neto, Leen Beller, Ward De-boutte, Chenyan Shi, Stephen M. Ghogomu, Piet Maes, Marc Van Ranst, and Jelle Matthijnssens. Gut virome analysis of Cameroonians reveals high diversity of enteric viruses, including potential interspecies transmitted viruses. *mSphere*, 4(1), 2019.
- [319] Li Yunfan, Yang Mouxing, Peng Dezhong, Li Taihao, Huang Jiantao, and Peng Xi. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130:2205–2221, 2022.

- [320] Boyang Zhang, Jared Leitner, and Samuel Thornton. Audio recognition using mel spectrograms and convolution neural networks. *Semantic Scholar Corpus ID:237274283*, 2019.
- [321] Qingpeng Zhang, Jason Pell, Rosangela Canino-Koning, Adina Chuang Howe, and C. Titus Brown. These are not the k -mers you are looking for: Efficient online k -mer counting using a probabilistic data structure. *PLOS ONE*, 9(7), 2014.
- [322] Xiang Zhang, Mingjie Yang, Xunhang Yin, Yining Qian, and Fei Sun. DeepGene: An efficient foundation model for genomics based on pan-genome graph transformer. *bioRxiv*, 2024.
- [323] Yifei Zhang and Alexander V. Alekseyenko. Phylogenic inference using alignment-free methods for applications in microbial community surveys using 16s rRNA gene. *PLOS ONE*, 12(11), 2017.
- [324] Zhi-Qiang Zhang and William A. Shear. Linnaeus tercentenary and invertebrate taxonomy: An introduction. *Zootaxa*, 1668(1):7–10, 2007.
- [325] Zhujin Zhang, Tao Song, Xiangxiang Zeng, Yunyun Niu, Yun Jiang, Linqiang Pan, and Yunming Ye. Colorsquare: A colorful square visualization of DNA sequences. *MATCH Communications in Mathematical and in Computer Chemistry*, 68:621–637, 2012.
- [326] Wei Zhao, Xueshuang Lai, Dengying Liu, Zhenyang Zhang, Peipei Ma, Qishan Wang, Zhe Zhang, and Yuchun Pan. Applications of support vector machine in genomic prediction in pig and maize populations. *Frontiers in Genetics*, 11, 2020.
- [327] Yi Zheng, Shan Gao, Chellappan Padmanabhan, Rugang Li, Marco Galvez, Dina Gutierrez, Segundo Fuentes, Kai-Shu Ling, Jan Kreuze, and Zhangjun Fei. Virus-Detect: An automated pipeline for efficient virus discovery using deep sequencing of small RNAs. *Virology*, 500:130–138, 2017.
- [328] Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12:931–934, 2015.
- [329] Qian Zhou, Saibing Qi, and Cong Ren. Gene essentiality prediction based on chaos game representation and spiking neural networks. *Chaos, Solitons & Fractals*, 144:110649, 2021.

- [330] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V. Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genome. *arXiv Preprint arXiv:2306.15006*, 2023.
- [331] Qinghe Zhu and Dongbo Sun. Goose astrovirus in China: A comprehensive review. *Viruses*, 14(8):1759, 2022.
- [332] Andrzej Zielezinski, Hani Z. Girgis, Guillaume Bernard, Chris-Andre Leimeister, Kujin Tang, Thomas Dencker, Anna Katharina Lau, Sophie Röhlhling, Jae Jin Choi, Michael S. Waterman, Matteo Comin, Sung-Hou Kim, Susana Vinga, Jonas S. Almeida, Cheong Xin Chan, Benjamin T. James, Fengzhu Sun, Burkhard Morgenstern, and Wojciech M. Karlowski. Benchmarking of alignment-free sequence comparison methods. *Genome Biology*, 20:144, 2019.
- [333] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, 18:186, 2017.

APPENDICES

Appendix A

Proposed Classification for As-yet Unclassified Astroviruses

The following tables list the accession IDs of as-yet unclassified astroviruses for which a candidate genus assignment has been made by 3PCM. A classification label was suggested at the genus level for those sequences where all three prongs or only Prong 1 and Prong 2 are in agreement. The first section presents the results with the inclusion of candidate recombinant sequences in the dataset and the second section presents the results for the dataset with the exclusion of candidate recombinant sequences from the dataset. The teal color in the following tables represents accession IDs that are not in common between two sets of results derived from analyses that include and exclude recombinant sequences.

A.1 3PCM's Proposed Classification with Inclusion of Candidate Recombinants

MK096774	MN920664	MN920670	MT137997	MT138004	MT138016	MN920669
MK096775	MN920665	MT137991	MT137999	MT138008	MT138014	MN920663
MK096776	MN920666	MT137992	MT138000	MT138009	MT138002	MT137994
MN920667	MT138013	MT138001	MT137993	MK450332		

Table A.1: Accession IDs of the proposed Avastroviruses when all three prongs agreed on the classification. The teal color highlights accession IDs that differ between analyses with and without recombinant sequences.

JN420353	MH933754	NC_035758
----------	----------	-----------

Table A.2: The Accession IDs of the proposed Avastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.

AF141381	KU318319	LC549662	MN503236	MZ546174	OM140654	MN503235
FJ375759	KU318320	LC577870	MN503237	MZ603072	OM451104	MZ357117
L13745	KU318321	LC577871	MN503238	MZ603074	OM451105	NC_035471
JA816485	KX266901	LC577872	MN780842	MZ682112	OM451106	LC201619
JF327666	KX266902	MF175073	MN832787	MZ779126	OM451107	KU318317
JF713711	KX266903	MF175075	MN837531	MZ819166	OM451109	MZ218054
JN420351	KX266904	MG660832	MN837532	MZ819167	OM451110	LC201618
JN420352	KX266905	MG693175	MN837533	MZ819168	OM451111	KU318316
JN420353	KX266906	OM480543	MT152345	MZ819169	OM451112	MZ218053
JN420354	KX266907	MG921619	MT267478	MZ819170	OM451113	LC201617
JN420355	KX266908	MH933752	MT470220	MZ819171	OM451114	KU318315
JN420356	KX907127	MW863310	MT499771	MZ819172	OM451115	OK107512
JN420357	KX907128	MH933756	MT499772	NC_001943	OM451116	KY271945
JN420358	KX907129	MH933757	MT549856	NC_018702	OM451117	KM035759
JN420359	KX907130	MH933758	MT549857	NC_019027	OM451118	NC_037655
JX544743	KX907131	MH933759	MT642595	NC_023636	OM451210	MK404648
JX544744	KX907132	MK387176	MT734809	NC_024498	OM451211	KJ571486
JX556693	KX907133	MK395165	MT766313	NC_032423	OM480521	NC_035758
JX857869	KX907134	MK395166	MW082586	NC_032426	OM480533	KC692365
OM104033	LC201620	OK107515	MN087316	KU318318	OK107514	MK521913
ON571622	OK107513	MK521912	ON482290	KY271946	MZ005893	LC201616
OM890931	MW897147	LC201615	OM480545	MW897146	KF233994	OM480534
KC609001	KX907135	MK404645	MW249010	NC_032484	MZ357116	MH933753
MK404647	MK404649	MK404646	MG693176			

Table A.3: Accession IDs of the proposed Mamastroviruses when all three prongs agreed on the classification. The teal color highlights accession IDs that differ between analyses with and without recombinant sequences.

KP663426	MT138010	NC_027426	ON304005	MK096773
----------	----------	-----------	----------	----------

Table A.4: The Accession IDs of the proposed Mamastroviruses when there was agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3. The teal color highlights accession IDs that differ between analyses with and without recombinant sequences.

A.2 3PCM's Proposed Classification with Exclusion of Candidate Recombinants

MK096774	MN920664	MN920670	MT137997	MT138004	MT138016	MN920669
MK096775	MN920665	MT137991	MT137999	MT138008	MT138014	MN920663
MK096776	MN920666	MT137992	MT138000	MT138009	MT138002	MT137994
MN920667	MT138013	MT138001	MT137993	MT138006		

Table A.5: Accession IDs of the proposed Avastroviruses when all three prongs agreed on the classification. The teal color highlights accession IDs that differ between analyses with and without recombinant sequences.

JN420353	MH933754	NC_035758
----------	----------	-----------

Table A.6: The Accession IDs of the proposed Avastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.

AF141381	KU318319	LC549662	MN503236	MZ546174	OM140654	MN503235
FJ375759	KU318320	LC577870	MN503237	MZ603072	OM451104	MZ357117
HUANSSPS	KU318321	LC577871	MN503238	MZ603074	OM451105	NC_035471
JA816485	KX266901	LC577872	MN780842	MZ682112	OM451106	LC201619
JF327666	KX266902	MF175073	MN832787	MZ779126	OM451107	KU318317
JF713711	KX266903	MF175075	MN837531	MZ819166	OM451109	MZ218054
JN420351	KX266904	MG660832	MN837532	MZ819167	OM451110	LC201618
JN420352	KX266905	MG693175	MN837533	MZ819168	OM451111	KU318316
JN420353	KX266906	OM480543	MT152345	MZ819169	OM451112	MZ218053
JN420354	KX266907	MG921619	MT267478	MZ819170	OM451113	LC201617
JN420355	KX266908	MH933752	MT470220	MZ819171	OM451114	KU318315
JN420356	KX907127	MW863310	MT499771	MZ819172	OM451115	OK107512
JN420357	KX907128	MH933756	MT499772	NC_001943	OM451116	KY271945
JN420358	KX907129	MH933757	MT549856	NC_018702	OM451117	KM035759
JN420359	KX907130	MH933758	MT549857	NC_019027	OM451148	NC_037655
JX544743	KX907131	MH933759	MT642595	NC_023636	OM451210	MK404648
JX544744	KX907132	MK387176	MT734809	NC_024498	OM451211	KJ571486
JX556693	KX907133	MK395165	MT766313	NC_032423	OM480521	NC_035758
JX857869	KX907134	MK395166	MW082586	NC_032426	OM480533	KC692365
OM104033	LC201620	OK107515	MN087316	KU318318	OK107514	MK521913
ON571622	OK107513	MK521912	ON482290	KY271946	MZ005893	LC201616
OM890931	MW897147	LC201615	OM480545	MW897146	KF233994	OM480534
KC609001	KX907135	MK404645	MW249010	NC_032484		

Table A.7: Accession IDs of the proposed Mamastroviruses when all three prongs agreed on the classification.

KP663426	MT138010	NC_027426
----------	----------	-----------

Table A.8: The Accession IDs of the proposed Mamastroviruses when there was an agreement between Prong 1 and Prong 2 on the classification, but they disagreed with Prong 3.

Appendix B

Comparative Evaluation of CGRclust with Phylonium

Phylonium is a program created to quickly estimate evolutionary distances between closely related genomes, useful for applications such as tracking disease outbreaks using whole-genome sequencing. It achieves speed by indexing a single reference sequence and piling all other sequences onto it, creating an approximate multiple-sequence alignment from which the distances are calculated. Phylonium is particularly useful in genomic epidemiology, where rapid and accurate distance estimations are crucial [155].

To compare CGRclust with phylonium, we also tested phylonium with all twenty-five datasets used in the CGRclust study. Phylonium was very fast, generating the evolutionary distance matrix in under a minute for datasets in Groups 1 and 2. As the selection of an appropriate reference can significantly affect the accuracy of phylonium, and Given the variety in our datasets and the complexity of choosing a reference for each, we utilized phylonium’s `-2pass` option. This option allows phylonium to select a central sequence as the reference after an initial run, which generally works well.

After generating the evolutionary distance matrix, we employed Ward’s method to construct phylogenetic trees, minimizing the variance within each cluster. We then cut each tree at heights corresponding to the expected number of clusters. Table B.1 presents the clustering accuracies of CGRclust compared to phylonium for the five datasets, out of twenty-five, where phylonium successfully generated distance matrices.

For the remaining 20 tests, phylonium generated matrices with NaN values, preventing us from generating trees and conducting clustering. These datasets included more heterogeneous sequences, demonstrating a limitation of phylonium, namely that it is primarily

Table B.1: Comparison of clustering accuracies between CGRclust and phylonium for five out of twenty-five datasets where phylonium successfully generated distance matrices.

Test	Dataset	CGRclust Accuracy	Phylonium Accuracy
1	Cypriniformes (Order to Suborder)	94.78%	90.76%
2	Cypriniformes (Suborder to Family)	91.75%	71.75%
3	Cypriniformes (Family to Subfamily)	81.70%	72.77%
4	Cypriniformes (Subfamily to Genus)	99.06%	99.53%
9	Human Immunodeficiency Virus 1	81.77%	64.15%

effective for closely related samples. As noted by Klötzl et al. [155], “when phylonium is applied to divergent sequences: it cannot find any anchors and hence cannot estimate the distance. This restriction to closely related sequences makes phylonium suitable for applications like genomic epidemiology, but not as a general tool for estimating phylogenetic distances.”