BMC Genomics

RESEARCH

Open Access

CGRclust: Chaos Game Representation for twin contrastive clustering of unlabelled DNA sequences

Fatemeh Alipour[1*], Kathleen A. Hill[2] and Lila Kari[1]

Abstract

Background Traditional supervised learning methods applied to DNA sequence taxonomic classification rely on the labor-intensive and time-consuming step of labelling the primary DNA sequences. Additionally, standard DNA classification/clustering methods involve time-intensive multiple sequence alignments, which impacts their applicability to large genomic datasets or distantly related organisms. These limitations indicate a need for robust, efficient, and scalable unsupervised DNA sequence clustering methods that do not depend on sequence labels or alignment.

Results This study proposes CGRclust, a novel combination of unsupervised twin contrastive clustering of Chaos Game Representations (CGR) of DNA sequences, with convolutional neural networks (CNNs). To the best of our knowledge, CGRclust is the first method to use unsupervised learning for image classification (herein applied to two-dimensional CGR images) for clustering datasets of DNA sequences. CGRclust overcomes the limitations of traditional sequence classification methods by leveraging unsupervised twin contrastive learning to detect distinctive sequence patterns, without requiring DNA sequence alignment or biological/taxonomic labels. CGRclust accurately clustered twenty-five diverse datasets, with sequence lengths ranging from 664 bp to 100 kbp, including mitochondrial genomes of fish, fungi, and protists, as well as viral whole genome assemblies and synthetic DNA sequences. Compared with three recent clustering methods for DNA sequences (DeLUCS, *i*DeLUCS, and MeShClust v3.0.), CGRclust is the only method that surpasses 81.70% accuracy across all four taxonomic levels tested for mitochondrial DNA genomes of fish. Moreover, CGRclust also consistently demonstrates superior performance across all the viral genomic datasets. The high clustering accuracy of CGRclust on these twenty-five datasets, which vary significantly in terms of sequence length, number of genomes, number of clusters, and level of taxonomy, demonstrates its robustness, scalability, and versatility.

Conclusion CGRclust is a novel, scalable, alignment-free DNA sequence clustering method that uses CGR images of DNA sequences and CNNs for twin contrastive clustering of unlabelled primary DNA sequences, achieving superior or comparable accuracy and performance over current approaches. CGRclust demonstrated enhanced reliability, by consistently achieving over 80% accuracy in more than 90% of the datasets analyzed. In particular, CGRclust performed especially well in clustering viral DNA datasets, where it consistently outperformed all competing methods.

Keywords Chaos Game Representation (CGR), Taxonomic classification, Alignment-free DNA sequence comparison, Unsupervised learning, DNA sequence clustering, Twin contrastive learning, Convolutional neural network, Data augmentation

*Correspondence:
Fatemeh Alipour
falipour@uwaterloo.ca
Full list of author information is available at the end of the article

Introduction

DNA sequence classification is essential for genomic analyses, contributing to the identification of evolutionary relationships, functional elements, and genetic variants, through the detection of sequence similarity. Conventional methods for classifying DNA sequences typically depend on labor-intensive and expert-mediated labelling of primary DNA sequences to determine sequence origin, function, and type. Furthermore, the stability of genome labels can be questioned, as taxonomic labels are not always definitive due to the absence of a clear taxonomic "ground truth" [1, 2]. Moreover, most traditional DNA sequence classification and clustering methods are alignment-based. The time complexity of DNA sequence alignment [3], coupled with a dependence on additional sequence information such as sequence homology [4], makes these methods unsuitable for analyzing large or evolutionarily divergent genomic datasets. These challenges emphasize the importance of developing robust and flexible alignment-free unsupervised approaches to DNA sequence classification that do not rely on DNA sequence labels, annotation, or alignment.

In 1990, Jeffery introduced Chaos Game Representation (CGR), a method for mapping one-dimensional DNA sequences into two-dimensional space using chaotic dynamics [5, 6]. A CGR maps each DNA sequence to a unique image. The process begins with a unit square whose corners are labelled A, C, G, and T, in a clockwise order starting from the bottom-left corner. The initial point in any CGR plot is the center of this square. To generate the CGR for a specific DNA sequence, the sequence is read from left to right, one nucleotide at a time. For each nucleotide read, a point is plotted midway between the previous point and the corner labelled with that nucleotide. Several studies [4, 7, 8] have demonstrated that CGRs can act as genomic signatures, defined by Karlin and Burge [9] as numerical quantities that can distinguish closely from distantly related organisms based on DNA sequence identity. The distance between CGRs of DNA sequences can be computed using various metrics, e.g., Euclidean distance, and can then be used for alignment-free comparisons and phylogeny construction to demonstrate evolutionary relationships within a group of organisms. Due to these properties, CGR has been considered a milestone in graphical bioinformatics [10, 11].

Frequency CGR (FCGR), a quantified variant of CGR, divides the CGR into smaller squares to calculate and display the frequency of nucleotides within each segment. An FCGR at resolution k creates a $2^k \times 2^k$ numerical matrix which can be presented as a grayscale image wherein pixel intensities represent k-mer frequencies. Consequently, FCGR provides a compressed representation of DNA sequences and facilitates the analysis of distinct genomic signatures across different species. Figure 1 illustrates some examples of FCGRs at resolution $k = 8$ (selected for visualization purposes) of real genomic DNA sequences, side by side with FCGRs of computer-generated DNA sequences. FCGR representations of DNA sequences have been used in many alignment-free genome comparison applications, overcoming the quadratic runtime and scalability problems associated with alignment-based methods [4, 7, 12, 13]. The use of FCGR permits alignment-free genomic sequence comparisons, when used in conjunction with digital signal processing techniques [14, 15] and machine learning methods [16–21].

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) [22]. In a study by Rizzo et al. [16], 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. [23] 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 [24], achieving 96.29% accuracy utilizing a ResNet50 neural network [25] and outperforming Covidex [26], a random forest-based clade assignment tool. Hammad et al. [27] 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 FCGR for the unsupervised clustering of DNA sequences in large, diverse datasets (up to 9,027 genomes) across different taxonomic levels and genetic distances [28, 29]. However, in both DeLUCS [28] and *i*DeLUCS [29], 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 [30], used the mean-shift algorithm for generating pairwise identity scores without alignment. MeShClust v3.0 is built on its predecessors, MeShClust v1.0 [31] (a DNA clustering method) and *Identity* [32] (a sequence

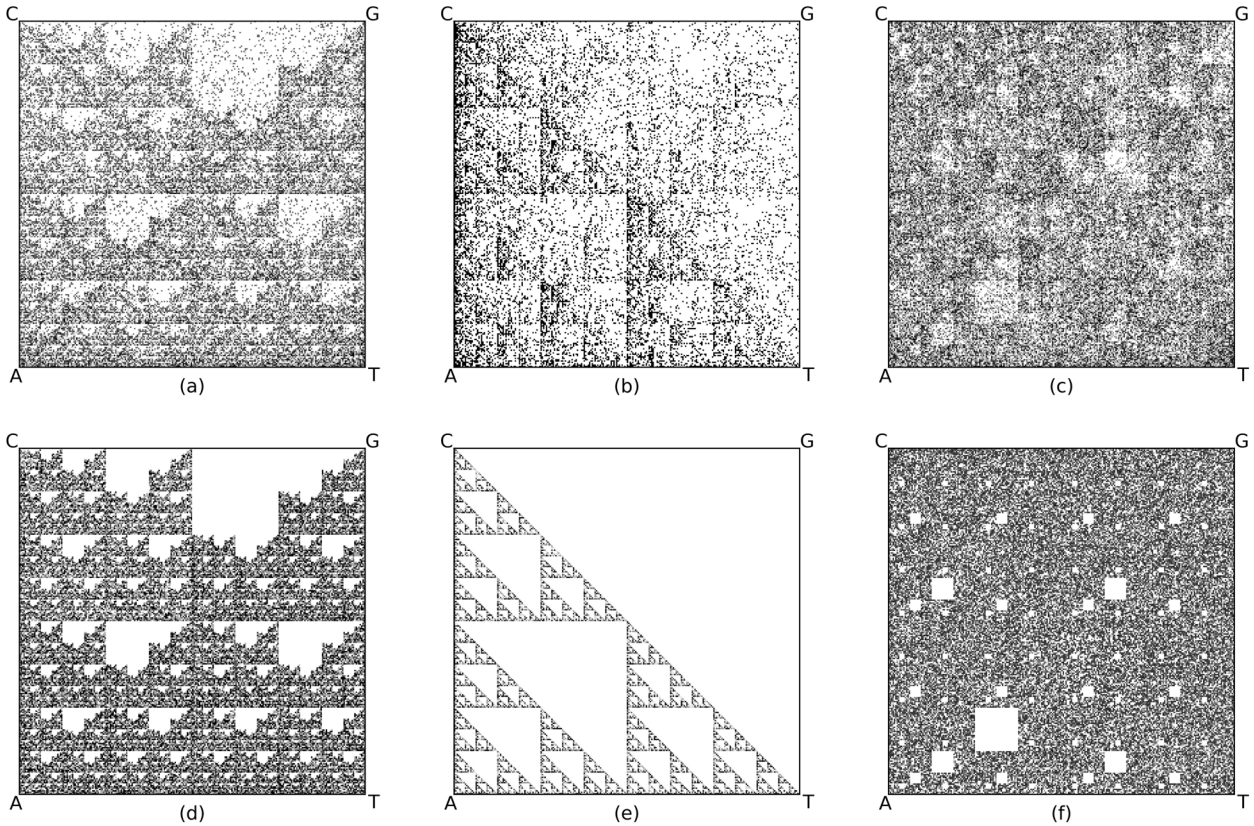

Fig. 1 Frequency Chaos Game Representation (FCGR) at resolution $k = 8$ (for visualization purposes) of **a** human beta globin region on chromosome 11 of length 73,308 bp (Accession ID: U01317.1); **b** complete genome of *Homo sapiens* isolate LI-T1 mitochondrion of length 16,566 bp (Accession ID: 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"

alignment identity score predictor), and can efficiently cluster both long sequences, up to 3.7 million base-pairs, and large datasets containing up to a million sequences. MeShClust v3.0 was tested on twenty-seven datasets, including twenty-two synthetic datasets and five real biological datasets, such as the human microbiome and maize transposons. In spite of this progress, DeLUCS, *i*DeLUCS, and MeShClust v3.0 underperform in clustering astrovirus sequences when compared to *K*-means++ [33], even though they were previously validated on other viral datasets. These limitations highlight the need for the development of more robust approaches that can effectively manage the complexities of genetic diversity of a wide range of genomic datasets.

This paper 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 this study employs *twin contrastive learning* (TCL) [34, 35], 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 [28], *i*DeLUCS [29], and MeShClust v3.0 [30]. 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 [29], along with synthetic DNA sequences [30]. 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 paper 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.

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. Chaos Game Representation (CGR), the graphical representation of DNA sequences used in this paper, is then defined, together with its quantified variant FCGR. Next, a description of the data augmentation strategies used for this graphical representation (generation of *mimic sequences*) is presented, serving as the initial component of CGRclust's pipeline. Afterwards, 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.

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 viral whole genomes. Additionally, to facilitate direct comparisons with established methodologies, we incorporated datasets previously analyzed by Millán et al. [29] and by Girgis [30] (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 January 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 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 [36, 37], makes mtDNA data an ideal candidate for initial clustering evaluations.

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 [33] (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 by Solis-Reyes et al. [38] with supervised machine learning methods. Table 1 outlines the details of the Group 2 dataset.

Table 1 Details of the Group 1, 2, and 3 dataset in Tests 1 through 13

Test	Taxonomic Clustering (No. of seq. per cluster)	No. of seq.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)
Group 1: Cypriniformes Full Mitochondrial Genomes					
1	**Order into Suborder**: Cypriniformes into Catostomoidei, Cobitoidei, Cyprinoidei (166 each)	498	15,655	16,610	17,859
2	**Suborder into Family**: Cyprinoidei into Acheilognathidae, Cyprinidae, Danionidae, Gobionidae, Leuciscidae, Xenocyprididae (105 each)	630	15,616	16,620	18,220
3	**Family into Subfamily**: Cyprinidae into Acrossocheilinae, Cyprininae, 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
Group 2: Viral Whole Genomes					
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
Group 3: mtDNA of Insects, Protists, and Fungi (from [29])					
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

Underlined font indicates the cluster with the highest number of sequences, which was subsequently selected for clustering at a lower taxonomic level

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 by Millán et al. [29] (Group 3 dataset: mtDNA of Insects, Protists, and Fungi), as well as 12 synthetic DNA datasets analyzed by Girgis [30] (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 1. Note that, given the observed mixed taxonomic levels used by Millán et al. [29] 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 [30] are presented in Table 2. 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 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 [29] and [30], respectively.

Method overview

The contrastive clustering method proposed in this paper, CGRclust, utilizes a quantified variant of CGR, a graphical encoding of DNA sequences introduced by Jeffrey [5]. This quantified DNA encoding, referred to as *FCGR*, 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 [12]. For a formal definition of CGR and FCGR see Supplementary Material 1. 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 how similar or different pairs of examples are, rather than relying solely on raw data or labelled examples [39]. This approach helps the model understand the underlying structures of the data by pulling

Table 2 Details of the Group 4 dataset in Tests 14 through 25 (synthetic sequences with different length and identity score thresholds from [30])

Test	Dataset	No. of seq.	Min. seq. len. (bp)	Avg. seq. len. (bp)	Max. seq. len. (bp)	No. of clusters	Min. cluster size	Avg. cluster size	Max. cluster size
14	Medium-60	18,210	653	1,365	2,062	100	13	202	398
15	Medium-70	18,731	678	1,359	2,027	100	8	212	398
16	Medium-80	20,939	664	1,425	2,043	100	14	222	398
17	Medium-90	21,266	730	1,340	2,016	100	5	194	400
18	Medium-95	24,039	724	1,446	2,038	100	7	203	396
19	Medium-97	20,772	736	1,358	2,022	100	13	192	390
20	Long-60	20,885	1,393	2,758	4,039	100	7	207	398
21	Long-70	18,558	1,441	2,754	4,062	100	19	224	399
22	Long-80	20,525	1,396	2,639	3,974	100	5	194	398
23	Long-90	22,518	1,489	2,586	3,964	100	5	196	400
24	Long-95	20,222	1,461	2,890	4,049	100	10	206	400
25	Long-97	19,960	1,486	2,715	3,988	100	5	210	398

The prefixes "Medium-" and "Long-" in the dataset names denote the length of the sequences they contain and the numerical values ranging from 60 to 97 in these names represent the identity score, indicating the percentage of similarity between the sequences

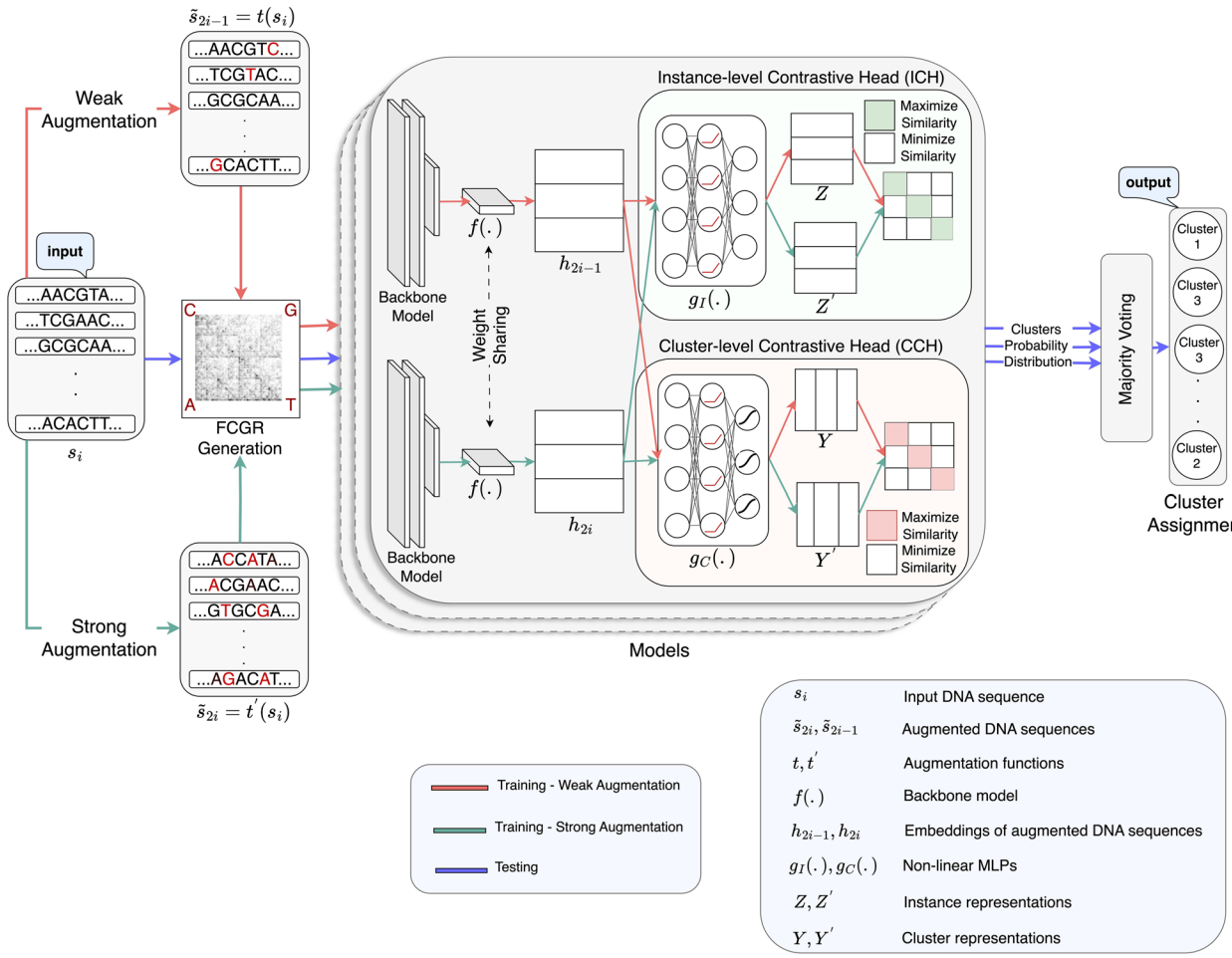

Fig. 2 CGRclust pipeline: Left panel: The process begins with data augmentation to create positive pairs (pairs of *mimic sequences*, pipeline component 1), followed by the generation of FCGR images of these augmented DNA sequences. Middle Panel: The FCGR images are fed into the backbone model (CNN) 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 component 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

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, called *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 this study takes advantage of the concept of *twin contrastive learning* (TCL) [34, 35], a method that simultaneously optimizes two contrastive learning objectives, one at the instance-level and another at the cluster-level, as detailed below.

Figure 2 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 Fig. 2, while the other three components are in the middle panel. Initially, pairs of mimic sequences constructed during 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 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 Fig. 2), which uses the outcomes of five distinct CNN models with different initializations to determine the final cluster assignment for each sequence.

To evaluate the quality of the clusters, an additional step, independent from the previous components, is conducted. This step utilizes the Hungarian algorithm [40], a method that effectively pairs elements from two sets to minimize the overall mismatch, to determine the optimal correspondence between the cluster assignments learned by the CGRclust and the actual taxonomic cluster labels. Subsequently, it evaluates the accuracy of the CGRClust predictions. Note that in unsupervised learning, 'training' comprises parameter optimization using unlabelled data; 'testing' then evaluates the trained model by comparing its output on the same data against the ground truth for evaluation purposes.

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 [34], 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 [41]. Fragmentation, denoted by $frag(len)$, has the length len of the desired fragment as 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. Details on these computational experiments can be found in Supplementary Material 2. The final findings suggest that mutation outperforms fragmentation as a data augmentation function, and its optimal parameters were empirically determined to be $\mu_1 = 10^{-4}$ for the weak augmentation, and $\mu_2 = 10^{-2}$ for the strong augmentation. 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 Backbone model architecture section.

Twin contrastive learning (TCL)

Inspired by [34, 35], during the training phase, the backbone, ICH, and CCH undergo joint optimization based on the following twin contrastive loss function:

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

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}).

The parameter α controls the relative importance of the two components during optimization. To determine its optimal value, we tested different values for this hyperparameter and it was empirically determined 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 different data conditions. For additional details, the reader is referred to Supplementary Material 2.

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 $\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_{2i-1}, \tilde{s}_{2i}, ...\tilde{s}_{2N}$. 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 [42], which includes a computational parameter so-called "temperature parameter" (τ) to scale the contrastive loss, is applied to fine-tune both contrastive mechanisms. A comprehensive hyperparameter optimization of the twin deep clustering model focused on the instance- and cluster-level temperature parameters (τ_I and τ_C) within the ICH and CCH was conducted. Examining different values for each temperature parameter in the range [0.1, 1], it was empirically determined that $\tau_I = 0.1$ and $\tau_C = 1.0$ consistently yield relatively high accuracy across all datasets. This advancement aligns with the hypothesis that a lower τ_I encourages individual instance differentiation, aligning with the ICH's goal, while a higher τ_C enhances group discrimination, mirroring the CCH's objective [43].

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 [34], 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 Supplementary Material 3 and [34].

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 *ResNet*34 and *ResNet*50, 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

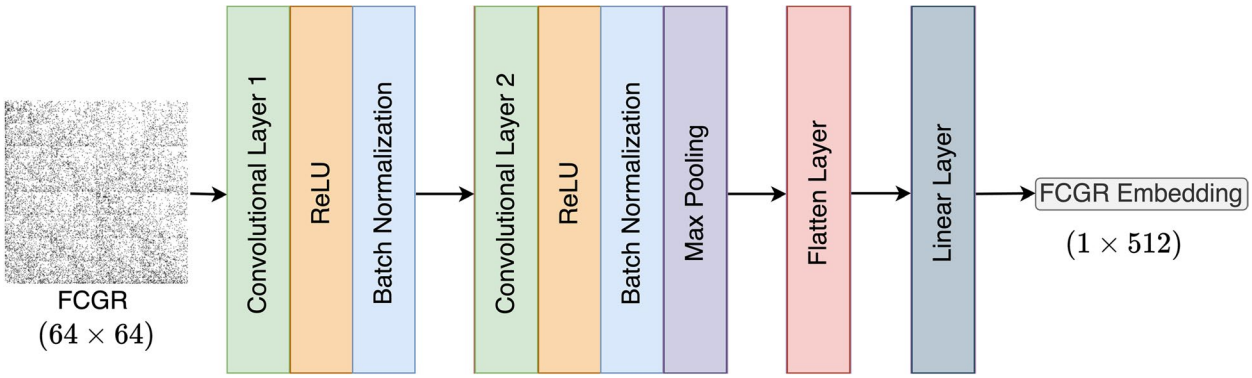

Fig. 3 Architecture of backbone model designed for clustering FCGR images of DNA sequences. The architecture of the backbone model comprises two convolutional layers, each with a kernel size of 7, stride of 2, and padding of 1. Following each convolutional layer, a Rectified Linear Unit (ReLU) is applied 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

architecture that is better suited for clustering FCGRs of DNA sequences. The backbone model architecture, as shown in Fig. 3, 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 2×2 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.

Majority voting scheme

The integration of ensemble learning, particularly through majority voting, has significantly improved the accuracy of genomic sequence classification, as demonstrated by Millán et al. [28, 29]. Majority voting, or hard voting, 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 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. Supplementary Material 2 discusses 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.

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 [44] 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 FCGR raw matrices 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 [45] with an initial learning rate set to 7×10^{-5} and a weight decay of 10^{-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.

For benchmarking CGRclust's performance against state-of-the-art methods in DNA sequence clustering, we chose three recent alignment-free clustering methods noted for their effectiveness in clustering a variety of genomic datasets: DeLUCS [28], *i*DeLUCS [29], and MeShClust v3.0 [30]. For both DeLUCS and *i*DeLUCS, we applied the default hyperparameters, and the accuracies presented in the Results section are based on these settings. MeShClust v3.0, a density-based clustering tool, inherently does not allow the pre-definition of cluster numbers. Consequently, besides the automatic selection of identity thresholds-which often leads to a discrepancy between the expected and actual cluster counts-we tested several identity score thresholds to select an optimal value that resulted in the desired number of clusters for each dataset. The optimal threshold values for each of the thirteen real datasets tested are detailed in Supplementary Material 4.

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 [29] authors' recommendation, *i*DeLUCS was executed on Google Colab using an NVIDIA Tesla T4 GPU with 16 GB of memory.

Results

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. The dynamic learning process during the training phase is shown in Fig. 4, illustrating how the model develops discriminative representations and accurately determines cluster assignments. This progression is documented across epochs and displayed at five timestamps. In Fig. 4, 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.

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 [28], *i*DeLUCS [29], and MeShClust v3.0 [30] (with both

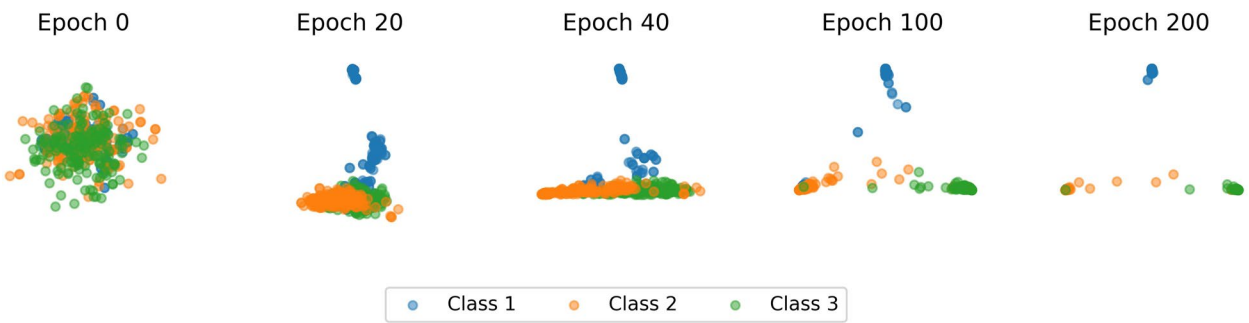

Fig. 4 CGRclust's evolution of clustering 498 Cypriniformes mitochondrial DNA sequences into three distinct clusters in Test 1. Each data point represents a DNA sequence, and its colour indicates its suborder label, and its position indicates the likelihood of assignment to different clusters (corners). 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

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 3 and Figure S5.1 present a summary of the clustering accuracies for the Group 1 dataset described in Table 1 (Cypriniformes mtDNA) across Tests 1–4. The reader is referred to Supplementary Material 5 for the confidence intervals of the CGRclust clustering accuracies of all clustering tests. The accuracies of CGRclust were achieved using the default hyperparameters over 150 epochs. As Table 3, and Table S5.1 in Supplementary Material 5 show, 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 a large margin (up to 75.08%).

Table 4 and Figure S5.1 summarize the accuracies of clustering the five viral datasets in the Group 2 dataset described in Table 1 (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%) without any errors. For the HCV

Table 3 CGRclust performance of clustering the Group 1 dataset (Cypriniformes mtDNA) described in Table 1

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

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). Each row highlights the highest accuracy (within the confidence interval of CGRClust) in bold

The reader is referred to Table S5.1 in Supplementary Material 5 for the confidence intervals of CGRclust clustering accuracies across Tests 1–4

Table 4 CGRclust performance of clustering Group 2 dataset (viral whole genomes) described in Table 1

Test	Taxonomic clustering	CGRclust	DeLUCS	*i*DeLUCS	MeShClust-manual	MeShClust-auto
5	Astroviridae-unbalanced	**84.94%**	69.88%	69.88%	59.69%	70.43%
6	Astroviridae-balanced	**88.84%**	**88.84%**	69.97%	77.27%	76.72%
7	dengue virus	**100%**	**100%**	96.99%	**100%**	52.08%
8	hepatitis C virus	**85.79%**	84.63%	76.84%	81.05%	80.52%
9	human immunodeficiency virus 1	**81.77%**	71.53%	39.38%	32.77%	7.69%

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). Each row highlights the highest accuracy (within the confidence interval of CGRClust) in bold. 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

The reader is referred to Table S5.1 in Supplementary Material 5 for the confidence intervals of CGRclust clustering accuracies across Tests 5–9

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 and Figure S5.1 display the clustering accuracies for Tests 10–13 in the Group 3 dataset (mtDNA of Insects, Protists, and Fungi) from the study [29], detailed in Table 1. 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 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, 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 the Twin contrastive learning (TCL) section.

In the comparison of clustering methods presented in Table 5 and Figure S5.1, *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 automatic 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).

Finally, for a direct comparison with MeShClust v3.0, Table 6 and Figure S5.1 summarize the accuracies of clustering Group 4 dataset (the twelve synthetic datasets

Table 6 CGRclust performance of clustering Group 4 dataset (the synthetic datasets from MeShClust v3.0 [30]), described in Table 2

Test	Dataset	CGRclust	DeLUCS	*i*DeLUCS	MeShClust-auto
14	Medium-60	92.26%	94.97%	91.77%	**99.7%**
15	Medium-70	93.39%	98.36%	94.40%	**99.8%**
16	Medium-80	94.61%	99.42%	97.58%	**99.8%**
17	Medium-90	95.23%	98.76%	97.60%	**99.9%**
18	Medium-95	96.57%	99.73%	99.55%	**100%**
19	Medium-97	95.51%	98.44%	98.57%	**100%**
20	Long-60	93.31%	97.36%	94.13%	**99.8%**
21	Long-70	92.82%	**98.00%**	95.40%	93.41%
22	Long-80	96.29%	99.12%	97.03%	**99.8%**
23	Long-90	94.08%	99.42%	99.13%	**100%**
24	Long-95	94.20%	99.37%	99.67%	**100%**
25	Long-97	94.83%	99.13%	99.11%	**100%**

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. Each row highlights the highest accuracy (within the confidence interval of CGRclust) in bold

The reader is referred to Table S5.1 in Supplementary Material 5 for the confidence intervals of CGRclust clustering accuracies across Tests 14–25

Table 5 CGRclust performance of clustering Group 3 dataset (mtDNA of Insects, Protists, and Fungi) described in Table 1

Test	Taxonomic clustering	CGRclust	CGRclust-adjusted α	DeLUCS	*i*DeLUCS	MeShClust-manual	MeShClust-auto
10	Insects	70.53%	73.56%	78.30%	**83.82%**	47.50%	21.90%
11	Protists	62.22%	85.50%	**88.10%**	80.00%	71.85%	74.92%
12	Fungi (Phylum)	56.72%	56.87%	**69.85%**	50.29%	50.74%	35.67%
13	Fungi (Subphylum)	97.10%	**97.38%**	**97.94%**	59.72%	75.14%	42.52%

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). Each row highlights the highest accuracy (within the confidence interval of CGRclust) in bold

The reader is referred to Table S5.1 in Supplementary Material 5 for the confidence intervals of CGRclust clustering accuracies across Tests 10–13

from [30] and described in Table 2), 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.

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 duration for the twenty-five datasets varied, 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. Notably, as CGRclust converts variable-length DNA sequences into fixed-size FCGRs, the training time remains relatively unaffected by sequence length. For detailed information regarding the total training time across all datasets, the reader is referred to Supplementary Material 6.

Discussion

This study 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 study 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 [29] and MeShClust v3.0 [30]). 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 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 our method 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, this method 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 [46], 'andi' [47], and phylonium [48] exist. As detailed in Supplementary Material 7, 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 [48], 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.

Conclusions

This study introduces 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.

Abbreviations

CCH	Cluster-level Contrastive Head
CGR	Chaos Game Representation
CNN	Convolutional Neural Network
FCGR	Frequency Chaos Game Representation
HCV	hepatitis C virus
HIV	human immunodeficiency virus
ICH	Instance-level Contrastive Head
KNN	K-Nearest Neighbor
mtDNA	Mitochondrial DNA
NCBI	National Center for Biotechnology Information
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
TCL	Twin Contrastive Learning

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-024-11135-y.

Supplementary Material 1.

Supplementary Material 2.

Supplementary Material 3.

Supplementary Material 4.

Supplementary Material 5.

Supplementary Material 6.

Supplementary Material 7.

Acknowledgements

We thank Dr. R. Greg Thorn for his guidance on fungi taxonomy, Matheus Sanita Lima for guidance on protist taxonomy, Joseph Butler for proofreading the manuscript, and Pablo Millan Arias for his assistance with experiments with iDeLUCS.

Authors' contributions

F.A., and L.K. conceived the study and wrote the manuscript. F.A. designed and performed the experiments. F.A., L.K., and K.A.H. conducted the data analysis and edited the manuscript, with K.A.H. contributing biological expertise. All authors read and approved the final manuscript.

Funding

The authors declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by Natural Science and Engineering Research Council of Canada Grants RGPIN-2023-05256 to K.A.H. and RGPIN-2023-03663 to L.K. This research was enabled in part by support provided by Compute Canada RPP (Research Platforms Portals),https://www.computecanada.ca/, Grant 616 to K.A.H. and L.K. The funders had no role in the preparation of the manuscript.

Data availability
The datasets generated and/or analyzed during the current study are all available in public repositories, and the links can be found in section 2.1 (Datasets) or associated literature. The CGRclust method developed for this study, along with all datasets used are available at https://github.com/fatemehalipour/CGRclust.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Author details
[1]School of Computer Science, University of Waterloo, Waterloo, Canada.
[2]Department of Biology, University of Western Ontario, London, Canada.

Received: 1 July 2024 Accepted: 6 December 2024
Published online: 18 December 2024

References

1. Applequist W. A brief review of recent controversies in the taxonomy and nomenclature of *Sambucus nigra sensu lato*. In: I International Symposium on Elderberry. 2013. pp. 25–33. https://doi.org/10.17660/ActaHortic.2015.1061.1.
2. Lovich JE, Hart KM. Taxonomy: A history of controversy and uncertainty. Ecol Conserv Diamond-Backed Terrapin. 2018;37–50.
3. Wang L, Jiang T. On the complexity of multiple sequence alignment. J Comput Biol. 1994;1(4):337–48. https://doi.org/10.1089/cmb.1994.1.337.
4. Zielezinski A, Vinga S, Almeida J, Karlowski WM. Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol. 2017;18:1–17. https://doi.org/10.1186/s13059-017-1319-7.
5. Jeffrey HJ. Chaos game representation of gene structure. Nucleic Acids Res. 1990;18(8):2163–70. https://doi.org/10.1093/nar/18.8.2163.
6. Barnsley MF. Fractals Everywhere. New. New York: Academic Press; 1988.
7. Löchel HF, Heider D. Chaos game representation and its applications in bioinformatics. Comput Struct Biotechnol J. 2021;19:6263–71. https://doi.org/10.1016/j.csbj.2021.11.008.
8. Karamichalis R, Kari L, Konstantinidis S, Kopecki S, Solis-Reyes S. Additive methods for genomic signatures. BMC Bioinformatics. 2016;17:1–18. https://doi.org/10.1186/s12859-016-1157-8.
9. Karlin S, Burge C. Dinucleotide relative abundance extremes: a genomic signature. Trends Genet. 1995;11(7):283–90. https://doi.org/10.1016/s0168-9525(00)89076-9.
10. Randić M, Novič M, Plavšić D. Milestones in graphical bioinformatics. Int J Quantum Chem. 2013;113(22):2413–46. https://doi.org/10.1002/qua.24479.
11. Kari L, Hill KA, Sayem AS, Karamichalis R, Bryans N, Davis K, et al. Mapping the space of genomic signatures. PLOS One. 2015;10(5):e0119815. https://doi.org/10.1371/journal.pone.0119815.
12. Deschavanne PJ, Giron A, Vilain J, Fagot G, Fertil B. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. Mol Biol Evol. 1999;16(10):1391–9. https://doi.org/10.1093/oxfordjournals.molbev.a026048.
13. Hill KA, Schisler NJ, Singh SM. Chaos game representation of coding regions of human globin genes and alcohol dehydrogenase genes of phylogenetically divergent species. J Mol Evol. 1992;35:261–9. https://doi.org/10.1007/BF00178602.
14. Hoang T, Yin C, Yau SST. Numerical encoding of DNA sequences by chaos game representation with application in similarity comparison. Genomics. 2016;108(3–4):134–42. https://doi.org/10.1016/j.ygeno.2016.08.002.
15. Lichtblau D. Alignment-free genomic sequence comparison using FCGR and signal processing. BMC Bioinformatics. 2019;20:1–17. https://doi.org/10.1186/s12859-019-3330-3.
16. Rizzo R, Fiannaca A, La Rosa M, Urso A. 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. 2016. pp. 222–8. https://doi.org/10.1145/2983468.2983489.
17. Zhou Q, Qi S, Ren C. Gene essentiality prediction based on chaos game representation and spiking neural networks. Chaos Solitons Fractals. 2021;144:110649. https://doi.org/10.1016/j.chaos.2021.110649.
18. Tanchotsrinon W, Lursinsap C, Poovorawan Y. A high performance prediction of HPV genotypes by chaos game representation and singular value decomposition. BMC Bioinformatics. 2015;16:1–13. https://doi.org/10.1186/s12859-015-0493-4.
19. Han GS, Li Q, Li Y. Comparative analysis and prediction of nucleosome positioning using integrative feature representation and machine learning algorithms. BMC Bioinformatics. 2021;22(6):1–23. https://doi.org/10.1186/s12859-021-04006-w.
20. Emam M, Ali A, Abdelrazik E, Elattar M, El-Hadidi M. 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). IEEE; 2020. pp. 2949–51. https://doi.org/10.1109/BIBM49941.2020.9313497.
21. Sengupta DC, Hill MD, Benton KR, Banerjee HN. Similarity studies of corona viruses through chaos game representation. Comput Mol Biosci. 2020;10(3):61. https://doi.org/10.4236/cmb.2020.103004.
22. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1989;1(4):541–51. https://doi.org/10.1162/neco.1989.1.4.541.
23. Safoury S, Hussein W. Enriched DNA strands classification using CGR images and convolutional neural network. In: Proceedings of the 2019 8th International Conference on Bioinformatics and Biomedical Science. 2019. pp. 87–92. https://doi.org/10.1145/3369166.3369176.
24. Avila Cartes J, Anand S, Ciccolella S, Bonizzoni P, Della Vedova G. Accurate and fast clade assignment via deep learning and frequency chaos game representation. GigaScience. 2023;12:giac119. https://doi.org/10.1093/gigascience/giac119.
25. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. pp. 770–8. https://doi.org/10.1109/CVPR.2016.90.
26. Cacciabue M, Aguilera P, Gismondi MI, Taboga O. Covidex: An ultra-fast and accurate tool for SARS-CoV-2 subtyping. Infect Genet Evol. 2022;99:105261. https://doi.org/10.1016/j.meegid.2022.105261.
27. Hammad MS, Ghoneim VF, Mabrouk MS, Al-Atabany WI. A hybrid deep learning approach for COVID-19 detection based on genomic image processing techniques. Scientific Reports. 2023;13(1):4003. https://doi.org/10.1038/s41598-023-30941-0.
28. Millán Arias P, Alipour F, Hill KA, Kari L. DeLUCS: deep learning for unsupervised classification of DNA sequences. PLOS One. 2022;17(1):e0261531. https://doi.org/10.3389/fmolb.2023.1305506.
29. Millán Arias P, Hill KA, Kari L. iDeLUCS: a deep learning interactive tool for alignment-free clustering of DNA sequences. Bioinformatics. 2023;39(9):btad508. https://doi.org/10.1093/bioinformatics/btad508.
30. Girgis HZ. MeShClust v3. 0: high-quality clustering of DNA sequences using the mean shift algorithm and alignment-free identity scores. BMC Genomics. 2022;23(1):423. https://doi.org/10.1186/s12864-022-08619-0.
31. James BT, Luczak BB, Girgis HZ. MeShClust: an intelligent tool for clustering DNA sequences. Nucleic Acids Res. 2018;46(14):e83–e83. https://doi.org/10.1093/nar/gky315.
32. Girgis HZ, James BT, Luczak BB. Identity: rapid alignment-free prediction of sequence alignment identity scores using self-supervised general linear models. NAR Genomics Bioinforma. 2021;3(1):lqab001. https://doi.org/10.1093/nargab/lqab001.
33. Alipour F, Holmes C, Lu YY, Hill KA, Kari L. Leveraging machine learning for taxonomic classification of emerging astroviruses. Front Mol Biosci. 2024;10:1305506. https://doi.org/10.3389/fmolb.2023.1305506.
34. Yunfan L, Mouxing Y, Dezhong P, Taihao L, Jiantao H, Xi P. Twin Contrastive Learning for Online Clustering. Int J Comput Vis. 2022;130:2205–21. https://doi.org/10.1007/s11263-022-01639-z.

35. Li Y, Hu P, Liu Z, Peng D, Zhou JT, Peng X. Contrastive clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35; 2021. pp. 8547–55. https://doi.org/10.1609/aaai.v35i10.17037.
36. Filip E, Strzała T, Stępień E, Cembrowska-Lech D. Universal mtDNA fragment for Cervidae barcoding species identification using phylogeny and preliminary analysis of machine learning approach. Sci Rep. 2023;13(1):9133. https://doi.org/10.1038/s41598-023-35637-z.
37. Yang X, E GX, Yang BG, Liu CL, Guo Y, Gong Y, et al. Genetic diversity and phylogeny pattern across Chongqing (China) chicken populations using mtDNA D-loop sequences. Russ J Genet. 2022;58(8):1007–16. https://doi.org/10.1134/S1022795422080117.
38. Solis-Reyes S, Avino M, Poon A, Kari L. An open-source *k*-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes. PLOS One. 2018;13(11):e0206409. https://doi.org/10.1371/journal.pone.0206409.
39. Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F. A survey on contrastive self-supervised learning. Technologies. 2020;9(1):2. https://doi.org/10.3390/technologies9010002.
40. Kuhn HW. The Hungarian method for the assignment problem. Nav Res Logist Q. 1995;2(1–2):83–97. https://doi.org/10.1002/nav.3800020109.
41. Lyons DM, Lauring AS. Evidence for the selective basis of transition-to-transversion substitution bias in two RNA viruses. Mol Biol Evol. 2017;34(12):3205–15. https://doi.org/10.1093/molbev/msx251.
42. Chen T, Kornblith S, Norouzi M, Hinton G. A Simple Framework for Contrastive Learning of Visual Representations. In: III HD, Singh A, editors. Proceedings of the 37th International Conference on Machine Learning. vol. 119 of Proceedings of Machine Learning Research. San Diego: PMLR; 2020. pp. 1597–607.
43. Kukleva A, Böhle M, Schiele B, Kuehne H, Rupprecht C. Temperature schedules for self-supervised contrastive methods on long-tail data. arXiv preprint arXiv:230313664. 2023. https://doi.org/10.48550/arXiv.2303.13664.
44. Cartes JA. Complex CGR. 2024. https://github.com/AlgoLab/complexCGR. Accessed 24 Mar 2024.
45. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014. https://doi.org/10.48550/arXiv.1412.6980.
46. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, et al. Mash: Fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016;17:132. https://doi.org/10.1186/s13059-016-0997-x.
47. Haubold B, Klötzl F, Pfaffelhuber P. andi: Fast and accurate estimation of evolutionary distances between closely related genomes. Bioinformatics. 2014;31(8):1169–75. https://doi.org/10.1093/bioinformatics/btu815.
48. Klötzl F, Haubold B. Phylonium: Fast estimation of evolutionary distances from large samples of similar genomes. Bioinformatics. 2020;36(7):2040–6.

Publisher's Note