RESEARCH Open Access

PCVR: a pre-trained contextualized visual representation for DNA sequence classification

Jiarui Zhou[1], Hui Wu[2], Kang Du[3], Wengang Zhou[2*], Cong-Zhao Zhou[3] and Houqiang Li[2]

*Correspondence: zhwg@ustc.edu.cn

[1] School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei 230026, Anhui Province, China
[2] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, Anhui Province, China
[3] Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230026, Anhui Province, China

Abstract

Background: The classification of DNA sequences is pivotal in bioinformatics, essentially for genetic information analysis. Traditional alignment-based tools tend to have slow speed and low recall. Machine learning methods learn implicit patterns from data with encoding techniques such as *k*-mer counting and ordinal encoding, which fail to handle long sequences or sacrifice structural and sequential information. Frequency chaos game representation (FCGR) converts DNA sequences of arbitrary lengths into fixed-size images, breaking free from the constraints of sequence length while preserving more sequential information than other representations. However, existing works merely consider local information, ignoring long-range dependencies and global contextual information within FCGR image.

Results: We propose PCVR, a Pre-trained Contextualized Visual Representation for DNA sequence classification. PCVR encodes FCGR with a vision transformer into contextualized features containing more global information. To meet the substantial data requirements of the training of vision transformer and learn more robust features, we pre-train the encoder with a masked autoencoder. Pre-trained PCVR exhibits impressive performance on three datasets even with only unsupervised learning. After fine-tuning, PCVR outperforms existing methods on superkingdom and phylum levels. Additionally, our ablation studies confirm the contribution of the vision transformer encoder and masked autoencoder pre-training to performance improvement.

Conclusions: PCVR significantly improves DNA sequence classification accuracy and shows strong potential for new species discovery due to its effective capture of global information and robustness. Codes for PCVR are available at https://github.com/jiaruizhou/PCVR.

Keywords: Bioinformatics, DNA sequence, Deep learning, Pre-training, Contextualized representation, Classification

Introduction

DNA sequence classification is of indispensable importance in understanding and identifying genetic differences among different species, thereby promoting the research and conservation of biodiversity. High-throughput sequencing technologies have generated vast amounts of genomic data. However, in metagenomic studies, many sequences are lost during contig assembly and binning, with only a small fraction being accurately aligned and classified by existing tools. Effective DNA sequence classification requires high precision, recall, and computational efficiency. In addressing the challenges of sequence classification and analysis in metagenomic research, various tools and methods are widely developed, primarily including alignment-based methods [1, 2] and machine learning (ML) methods [3–5].

Early methods identify the taxonomic relationships among sequences by aligning unknown DNA sequences against references in database [6–9]. However, these methods are susceptible to the quality of the database [10]. On the one hand, existing databases cover only a fraction of known species, making it challenging to identify homologous sequences for many DNA sequences. On the other hand, alignment-based tools often exhibit low computational efficiency or recall rates [11, 12]. Therefore, ML are introduced into DNA sequence classification for its capability of learning latent patterns from data without requiring references in database.

Considering the superior capability in complex pattern recognition of Deep Learning (DL) compared to ML [13, 14], DL methods are adopted more and more frequently for DNA sequence classification. These methods typically rely on DNA sequence representations, e.g., k-mers [15], one-hot encoding [16, 17], and Word2Vec [18], as network inputs. However, these representations either fail to preserve inherent structural information or are restricted to fixed-length sequences. In contrast, Frequency Chaos Game Representation (FCGR) enables handling sequences of arbitrary lengths by constructing frequency profiles that encapsulate statistical properties and patterns, offering a more effective DNA sequence representation. Current research [19–21] primarily employs convolutional neural networks (CNNs) to process FCGR. While CNNs encodes local information to learn image-specific patterns, they often neglect global information, which is crucial for understanding sequence characteristics and functionalities [22]. Due to their limited local receptive fields, CNN-based methods struggle to capture global dependencies and fully leverage FCGR's potential features.

To comprehensively understand patterns in DNA sequences, the model should capture local and global information simultaneously. Transformer [23] is a neural network architecture based on the self-attention mechanism. It excels at efficiently processing sequences and capturing interdependencies among positions within the sequence, thereby enhancing the modeling of long-range dependencies. In the domain of computer vision (CV), to address limitations inherent in CNNs when processing images, Vision Transformer (ViT) [24] is proposed to learn more generalized image features. Such enhancement augments the generalization capabilities of the model across diverse tasks and datasets, exhibiting superior performance on multiple benchmarks [25]. Benefiting from the self-attention mechanism in ViT, each patch in the FCGR images is able to attend to patches of all other positions, enabling contextualized and richer representations of DNA patterns.

Although ViT outperforms CNNs in modeling global information, it requires more training data to achieve comparable generalization capacity due to fewer image-specific inductive biases [24]. In the fields of natural language processing (NLP) and CV, self-supervised pre-training is commonly employed to address the lack of data in the training of transformer and ViT. Specifically, numerous supervision signals are generated from large-scale unlabeled texts and images with tasks like Masked Language

Modeling (MLM) [26] and Masked Autoencoder (MAE) [27]. These pre-training methodologies diminish reliance on labeled data, and aid in the acquisition of more universal, generalized, and robust feature representations, thereby enhancing the performance of the model. Limited annotated data in the area of DNA sequence classification hinders sufficient training of ViT. Inspired by self-supervised approaches in NLP and CV, we employ MAE to pre-train the ViT encoder, reducing its dependence on labeled data while learning robust FCGR features.

In summary, we propose a Pre-trained Contextualized Visual Representation (PCVR) for DNA sequence classification. PCVR enhances DNA sequence representations by capturing long-range dependencies and global context through a self-attention-based ViT encoder. To fully exploit the encoding capability of ViT and learn more robust feature representations, we employ MAE to pre-train the model. Specifically, DNA sequences are first converted into FCGR images. Then, we use MAE self-supervised pre-training, where randomly masked image patches are reconstructed by the model to learn semantic representations of FCGRs. Notably, no labeled data is required in the pre-training. Subsequently, we fine-tune the ViT encoder with a hierarchical classification head on labeled data, yielding a model capable of fine-grained classification of DNA sequences.

We evaluate PCVR on three datasets and observe that it outperforms existing methods across all of these datasets on superkingdom and phylum levels. In comparison to state-of-the-art methods, our model exhibits achieves statistically significant improvements on datasets whose samples in the test set do not have identical genus samples in the reference database. To be specific, PCVR achieves a improvement of 5.93% at the superkingdom level and 8.96% at the phylum level on the distantly related dataset. It indicates outstanding generalization capabilities of our model, promising significant applications in the discovery of new species. On both closely related and final datasets, PCVR improves the macro average precision to over 98% and 96% at the superkingdom and phylum levels. These results highlight the exceptional capability of ViT in processing global information and the ability of MAE pre-training to enhance the robustness of the features even across diverse domains. Ablation experiments on ViT and fine-tuning the head substantiates the rationality of combining FCGR representation with the visual encoder model. Prospectively, transforming DNA sequence information into FCGR can serve as a convincing DNA sequence embedding approach for DL models. The employment of self-supervised pre-training enables the adaptations to various downstream tasks, e.g., identification of promoters and enhancers.

To summarize, the main contributions of our work are as follows:

1. It is the first approach to introduce ViT to extract contextualized visual representation of DNA sequence for classification, capturing global contextual information and long-range dependencies in genomic sequence.
2. We leverage MAE pre-training to fully exploit the representation potential of ViT architecture, effectively capture structural features and significantly enhance the robustness of the model.
3. Extensive experiments on multiple datasets demonstrate impressive performance of PCVR, indicating superiority of ViT and MAE pre-training in DNA sequence clas-

sification. The effectiveness of key components in PCVR are also verified by our ablation studies.

Background

DNA sequence classification

Traditionally, BLAST [6] employs a local alignment strategy to identify local similarities and search similar sequences in biological databases. MetaBat2 [9] leverage gene abundance and other gene features to achieve classification and clustering on the microbial phylum level [28]. To improve computational efficiency, MMseq2 [7] divides the input sequences into different clusters and then performs a comparison within each cluster. MMseqs2 taxonomy [29] specializes in sequence classification and annotation and uses pre-trained classifiers to enhance its classification capabilities. Minimap2 [8] employs a split alignment strategy, utilizing a hash table storage structure and implementing dynamic programming algorithms for alignment. Nonetheless, the performance of these sequence alignment-based methods is susceptible to the data quality.

To overcome the reliance on data quality, researchers have shifted their focus toward ML. Earlier works employ ML methods such as support vector machines, decision trees, and random forests as classifiers [30, 31]. Rizzo et al. [19] represent DNA sequences as images using the FCGR for the first time. DeepMicrobes [32] process DNA sequences with one-hot encoding and *k*-mer embedding, and feed the embeddings into the deep neural network. The recently published method [33] enhances its predictive capabilities by balancing the feature data. Though these methods improve efficiency, they still fall short in predicting unseen sequences.

Pre-training models

The paradigm of pre-training and fine-tuning marks a milestone in deep learning, boosting the development of NLP and CV. Typically, they learn general features from large unlabeled datasets via unsupervised learning and then enhance performance by fine-tuning on downstream tasks.

Text pre-training captures the intrinsic patterns of language by training models on a large scale of textual data. Early word embedding techniques such as Word2Vec [34] and GloVe [35] learn vector representations of words by predicting context. Elmo [36] model captures contextual information through a bidirectional LSTM [37] network. BERT [26] and GPT [38] introduce the transformer architecture and different pre-training tasks, significantly enhancing the model's ability of language understanding and generation, respectively. In fields involving DNA sequences, BERTax [39] regards DNA sequences as natural language, utilizing BERT and adopting pre-training and fine-tuning to extract feature representations of DNA sequences for classification.

Image pre-training captures more subtle and complex visual patterns from large image datasets. ImageNet [40] provides a vast number of annotated images, facilitating the rapid development of deep learning in the field of image recognition. Subsequently, ResNet [41] uses deeper networks and introduces residual connections to solve the degradation problem. Moreover, ViT [24] processes images using self-attention mechanisms. MAE [27] self-supervised pre-training learns the representation of image features by reconstructing partially masked images. These image pre-training techniques enhance the model's perception of image features, but the application of image pre-training techniques in DNA sequence classification tasks remains uncharted territory.

Methods

In this section, we elaborate on PCVR from three aspects: the representation of DNA sequence, the MAE pre-training and fine-tuning for taxonomic classification. In PCVR, DNA sequences are first embedded into FCGR and then converted to contextualized representations using a ViT encoder. To obtain a more robust representation, the ViT encoder is pre-trained with the MAE framework. Finally, we append a hierarchical classification head to the pre-trained ViT encoder and fine-tune the whole model to tackle the classification task. The detailed pipeline of DNA sequence classification using PCVR is illustrated in Fig. 1.

Representation of DNA sequence

DNA sequence embedding

When addressing tasks related to DNA sequences using DL models, we face challenges such as varying sequence lengths, excessively long sequences, or sequences that are too short. K-mer counting transforms sequences of varying lengths into vectors of dimension 4^k, where k is a pre-defined value, but it also struggles to capture long-range dependencies. Inspired by previous research [19–21], we consider utilizing FCGR to process DNA sequences. The main distinction between FCGR and k-mer counting encoding lies in the fact that FCGR maps the DNA sequence into a 2-d matrix using the CGR approach [42], capturing richer pattern and structural information compared to 1-d k-mer counting and other commonly used 1-d encoding techniques [43, 44].

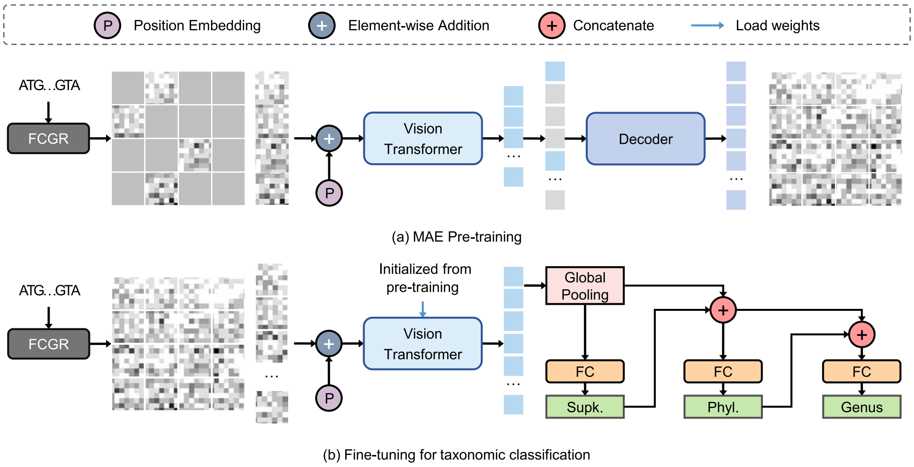

Fig. 1 The pipeline of DNA sequence classification using the proposed PCVR. This pipeline consists of two stages: the MAE pre-training stage for robust features and the fine-tuning stage for taxonomic classification. **a** The upper part of the pipeline illustrates the MAE self-supervised pre-training, which injects robust recognition abilities for images to the ViT encoder through reconstruction learning and obtains PCVR. **b** The lower part shows the fine-tuning stage, depicting the hierarchical feature fine-tuning structure upon the learned PCVR

In CGR, each k-length nucleotide subsequence in the sequence is transformed into a point on a unit square image sequentially as illustrated in Fig. 2. The coordinate P_i for nucleotide s_i in the image depends on the coordinate P_{i-1} for the previous nucleotide s_{i-1} [45]. Formally, the following steps are performed for each nucleotide s_i in the DNA sequence to obtain positions of all nucleotides in the sequence:

$$\begin{aligned}
&\text{If } s_i = \text{“A”, then } P_i = \frac{1}{2}(P_{i-1} + P_A);\\
&\text{If } s_i = \text{“T”, then } P_i = \frac{1}{2}(P_{i-1} + P_T);\\
&\text{If } s_i = \text{“C”, then } P_i = \frac{1}{2}(P_{i-1} + P_C);\\
&\text{If } s_i = \text{“G”, then } P_i = \frac{1}{2}(P_{i-1} + P_G),
\end{aligned}$$

where P_A, P_T, P_C and P_G are coordinates of “A”, “T” “C” and “G”, instantiated as (0, 1), (1, 1), (0, 0) and (1, 0), respectively. P_0 is defined as (0.5, 0.5), meaning that the coordinate computation is started from the center of image. After that, CGR is quantized into an image with a resolution of $2^k \times 2^k$, where each pixel is shaded based on its frequency of related k-length nucleotide subsequence in the fragment. Up to this point, FCGR is obtained and the embedding of the DNA sequence is completed as depicted in Fig. 2.

Contextualized representation

After the DNA sequences are converted into FCGR embeddings, these embeddings are then utilized to extract discriminative patterns and reveal contextual sequence features. ViT [24] is a simple yet powerful model that introduces transformer from the NLP domain to the computer vision domain. It is capable of learning global relationships within images and extracting features at multiple scales. FCGR encapsulates both global DNA sequence statistics and localized short k-mer patterns. Leveraging this property, ViT models contextualized sequence features effectively, forming the backbone of our PCVR framework. ViT comprises token embedding and encoding using the transformer encoder [23], which consists of a multi-head self-attention mechanism (MHSA) and a position-wise fully connected feed-forward network (FFN). This approach effectively extracts generalized DNA sequence features, enabling robust downstream classification.

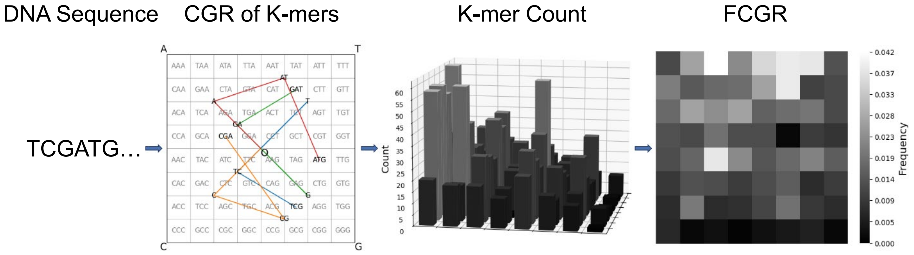
DNA Sequence
CGR of K-mers
K-mer Count
FCGR
TCGATG...
A
T
C
G
AAA TAA ATA TTA AAT TAT ATT TTT
CAA GAA CTA GTA CAT GAT CTT GTT
ACA TCA AGA TGA ACT TCT AGT TGT
CCA GCA CGA GGA CCT GCT CGT GGT
AAC TAC ATC TTC AAG TAG ATG TTG
CAC GAC CTC GTC CAG GAG CTG GTG
ACC TCC AGC TGC ACG TCG AGG TGG
CCC GCC CGC GGC CCG GCG CGG GGG
AT
A
GA
O
TC
C
CG
T
G
Count
60
55
50
45
40
35
30
25
20
15
10
5
0
0.042
0.037
0.033
0.028
0.023
0.019
0.014
0.009
0.005
0.000
Frequency

Fig. 2 DNA sequence FCGR embedding process of PCVR. For clarity, we present an example with $k = 3$. Each k-mer in the DNA sequence is mapped to a grid position in CGR. The locating process of the first four k-mers is displayed using colored lines. The number of each k-mer that appears in a sequence is counted. Finally, these counts are converted into frequencies and each pixel in the FCGR is shaded according to its frequency

Token embedding. ViT uses patch embedding to vectorize each image by splitting it into patches and then treating each patch as a token. Once the tokens are obtained, a position embedding is added to each token to provide the model with positional information, enabling it to understand the global structure and capture contextual information. With these embeddings, the transformer encoder module of ViT processes the input token sequence, converting it into a series of high-level abstract contextualized representations for downstream tasks such as taxonomic classification.

MHSA. MHSA is one of the most critical components of the transformer. Self-attention allows each position in the input sequence to pay attention to information at others, thus enabling the modeling of the global context without introducing sequence order. The multi-head mechanism in the attention mechanism allows the model to attend to information from multiple subspaces, enabling it to extract richer features and enhance the expressive and generalization capabilities of the model. In our context, self-attention and multi-head attention can be understood as a mechanism capturing relationships between individual short sequences and information about the sequential structure at different levels.

The self-attention mechanism first applies a linear transformation to the input tokens X using learnable weights W_Q, W_K, and W_V to obtain the query Q, key K, and value V. Then, the relevance scores between each query and all keys are computed by using dot product with a scaling factor $\sqrt{d_k}$. Subsequently, these relevance scores are normalized via a softmax operation. A weighted summation is applied to values based on the relevance between each corresponding key and the query to yield the final attention output. The self-attention computational formula is illustrated as

$$\mathrm{Att}_i = \mathrm{Softmax}\left(\frac{Q_i {K_i}^T}{\sqrt{d_k}}\right) V_i, \tag{1}$$

where i is symbolic of the index of the i-th attention head. d_k represents the vector dimension of each key which ensures the scaling operation appropriately adjusts the results of the dot product, mitigating issues such as gradient explosion and numerical instability. After getting individual attention heads, we concatenate all attention heads and perform a linear transformation with W_O. Thus far, the output of the multi-head self-attention is acquired.

FFN. The outputs of the MHSA layer are fed into an FFN layer after residual connections and layer normalization. The FFN consists of two simple fully connected layers, and ReLU activation is applied in the first layer. Through this process of dimensionality expansion followed by dimensionality reduction, the model can combine various types of features and eliminate less discriminative feature combinations. This approach enhances the discrimination power of the model while removing redundant information. The FFN is computed as

$$\mathrm{FFN}(x) = \mathrm{ReLU}(W_1 \cdot x + b_1) \cdot W_2 + b_2. \tag{2}$$

MAE pre-training

Masked autoencoders. To obtain more universal and robust feature representations, we employ MAE [27] pre-training. MAE pre-training provides a robust training approach that equips the encoder to recognize intricate patterns and structural information within images. MAE utilizes an asymmetric encoder-decoder structure, with ViT serving as both its encoder and decoder. It learns features through feature recognition and pixel reconstruction, denoted as encoding and decoding, respectively. In essence, MAE pre-training incorporates a masking mechanism akin to BERT [26] for unsupervised pre-training.

The masking strategy in MAE pre-training allows the model to learn intricate information from images. During the encoding phase, MAE randomly masks a substantial number of patches, and in the decoding phase, it strategically leverages these masked patches for image reconstruction. In simpler terms, the encoder processes only a subset of visible image patches, whereas the decoder processes both the image patches output by the encoder and the patches that have been masked during encoding. This asymmetric structure reconstructs the complete image information by leveraging positional encoding obtained during patch embedding.

This randomized masking strategy helps mitigate the risk of selected patches being distributed near the center of the image. Moreover, a high masking ratio effectively prevents the model from easily inferring these masked blocks based solely on neighboring visible patches, prompting the model to learn higher-level and more intricate information between image patches. Ultimately, this strategy results in sparse encoder inputs, as the encoder only processes visible image patches, leading to reduced computational costs and memory footprint.

Pre-training objective. During the image reconstruction process in MAE pre-training, Mean Squared Error is employed as the loss function, computed by summing squared differences between the original and the reconstructed pixels. Since information about unmasked pixel blocks is already known to the encoder and decoder as part of the input, the loss function is computed only for masked patches, akin to BERT. The training objective is designed to enable the model to identify specific patterns within the FCGR. This capability is essential for the encoder to distinguish sequences of different classes by encoding them with clear feature boundaries. Specifically, the training involves optimizing the encoder and decoder simultaneously to minimize the reconstruction error, thus enhancing the autoencoder's effectiveness in reconstructing the input. This optimization is accomplished using gradient descent technique to iteratively refine the parameters, formulated as

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} (I - Dec_\phi(Enc_\theta(I)))^2 \cdot M_I. \tag{3}$$

Here, Enc_θ and Dec_ϕ denote the encoder and decoder, respectively. M_I indicates the mask matrix applied to patches of image I. θ^* is the parameter set of optimized encoder and will be used for subsequent fine-tuning. ϕ^* is the parameter set of optimized decoder.

Fine-tuning for taxonomic classification

Once the pre-trained model captures patterns within the dataset, the ViT encoder can serve as an extractor of image features for the subsequent classification task. We fine-tune the pre-trained ViT with an additional multi-layer perceptron to adapt it to DNA sequence classification tasks.

Hierarchical classification head. For the fine-tuning head, we use lightweight networks. PCVR performs normalization on features output by ViT to obtain the global feature. Following this, PCVR employs the global feature as the classification feature at the superkingdom level and obtains the classification results of the superkingdom. As the difficulty of classification rises at lower taxonomic ranks, i.e., phylum and genus, more information is required compared to the superkingdom level. Hence, we have adopted a hierarchical fine-tuning structure to enhance the classification features of the phylum and genus. In this structure, the classification results of each level are combined with the global features generated by the encoder to obtain composite features, which are then fed into the linear layer at the lower rank as input. In this way, higher-level classification results can provide references for lower-level classifications, allowing the model to effectively leverage label information from each level.

Fine-tuning objective. By incorporating the category information of sequences and fine-tuning the pre-trained parameters with lightweight networks, the model, proficient in recognizing sequence features in FCGR, has transitioned its training focus towards classifying DNA sequences. To facilitate supervision throughout this process, we utilize cross entropy as the loss function for each taxonomic rank. To enhance optimization efficacy across various ranks, weights associated with the loss functions at distinct levels are adjusted accordingly. Ultimately, these weighted losses are aggregated to formulate the final fine-tuning loss function as follows:

$$Loss = \sum_{r=1}^{3} w_r \cdot \sum_{k=1}^{K_r} y_k^r \log \hat{y}_k^r, \tag{4}$$

where w_r is the loss weight and K_r is the class number of the r-th taxonomic rank. y_k^r and $\hat{y}_k^r$ denote the label and predicted probability of the k-th class in the r-th taxonomic rank, respectively.

Experiments

Implementation details

We implement the whole PCVR with PyTorch [46]. We choose the 5-mer FCGR as the input for our model with a patch size of 4 for the images. The encoder of our large model has an embedding dimension of 1024, comprising 24 transformer layers with 16 attention heads. The base model has an encoder embedding dimension of 768, with 12 transformer layers and 12 attention heads. The embedding dimension of the decoder is 512, featuring 8 transformer layers and 16 attention heads. During fine-tuning, we utilize layer normalization for global pooling of the features obtained from the encoder to derive the global feature. In the pre-training phase, we employ a batch size of 256 and utilize the AdamW optimizer [47] for both the pre-training and fine-tuning of PCVR. The base learning rate is set to 0.0001, and the learning rate undergoes a warm-up phase of 40 epochs. Following the warm-up, the learning rate decays with a half-cycle cosine schedule until the completion of the remaining epochs. The reported results of the large model correspond to the pre-trained model obtained from the checkpoint at the 520th epoch. The results for the base model are derived from the 540th epoch, with loss weights for the three levels set at 2:3:5. Except for the final result presentation, all experiments in this work are conducted using a balanced loss ratio of 1:1:1. For the distillation, we use KL divergence to align the logits distribution, with temperature coefficients of $\{2, 3, 4\}$ and weights of $\{0.4, 0.4, 0.2\}$ across three taxonomic ranks, and cross-entropy loss to supervise the labels. All models are trained on 8 NVIDIA GeForce RTX 3090 GPUs.

Evaluation metrics

To evaluate the performance of our model, we follow BERTax [39] and use macro average precision (macro AveP) as our primary evaluation metric. Macro AveP reflects the performance of the model on the dataset as a whole without focusing on the performance of a specific category. The macro AveP is computed using

$$\text{macro AveP} = \frac{1}{n}\sum_{i=0}^{n} P_i, \tag{5}$$

where P_i denotes the precision of the i-th class.

In cases of imbalanced class distribution, the macro AveP may be substantially influenced. Therefore, we additionally employ micro average precision (micro AveP) as a supplementary evaluation metric in some comparisons which is not sensitive to imbalanced data. We compute micro AveP as

$$\text{micro AveP} = \frac{\sum_{i=0}^{C} w_i \cdot TP_i}{\sum_{i=0}^{C} w_i \cdot (TP_i + FP_i)}, \tag{6}$$

where w_i signifies the class weight for class i, determined by the proportion of the quantity of this class in the dataset.

To provide a more comprehensive evaluation of our model, we also present classification accuracy (Acc), the micro-averaged Area Under the ROC Curve (AUC), which are metrics commonly used in classification tasks. Acc, the most intuitive evaluation metric in classification tasks, represents the proportion of correctly predicted samples out of the total samples. AUC provides insights into the model's classification ability across various thresholds, facilitating a more balanced assessment of the model's performance. Given the limitations of traditional alignment-based methods that classify only a subset of the data, we use the proportion of the predicted samples for the method (Prop) to assess DNA sequence classification models' ability to handle all available data.

Benchmark dataset

We use the data provided by the developers of BERTax [39] for our study whose number of categories and samples in each dataset are shown in Table 1. For pre-training, the dataset comprises 2,492,474 DNA sequences with a fixed length of 1,500 nucleotides from the four superkingdoms, Eukaryotes, Bacteria, Archaea, and Viruses. Note that only these sequences are used and no class label is provided for pre-training. For fine-tuning, to show the generalization ability of the model when the test set includes unknown sequences, datasets are selected and formed: closely related dataset and distantly related dataset [39]. For the closely related dataset, a fixed number of samples are

Table 1 Statistics of datasets

Dataset	Supk.	Phyl.	Genus	Training data	Test data
Closely related dataset	4	30	146	2,268,584	60,000
Distantly related dataset	4	30	146	2,245,416	53,400
Final dataset	4	44	156	5,311,920	88,000

Table 2 Comparison with existing baseline models of the macro AveP on all three datasets

Method	Closely related		Distantly related		Final		
	Supk.	Phyl.	Supk.	Phyl.	Supk.	Phyl.	Genus
MMseqs2 [7]	92.19	85.66	62.76	41.36	96.94	92.90	74.76
MMseqs2 tax. [29]	94.33	86.56	67.47	43.44	98.11	93.47	75.09
minimap2 [8]	86.12	76.06	44.12	20.03	93.46	86.71	66.68
DeepMicrobes [32]	97.18	86.62	67.25	36.61	98.13	92.11	66.43
BERTax [39]	95.65	83.88	90.06	54.10	98.62	95.10	66.92
Subspace KNN [33]	-	-	88.03	65.77	99.07	95.53	**86.43**
Bagged decision trees [33]	-	-	81.64	69.71	92.51	85.17	76.10
PCVR-Base	98.40	95.40	94.59	75.08	98.97	96.29	74.65
PCVR-Large	**98.87**	**96.32**	**96.00**	**78.67**	**99.22**	**96.93**	74.51

randomly selected from each phylum as the test set, with the remaining data used as the training set. For the distantly related dataset, one or more entire genera samples are selected as the test set from each phylum. It is ensured that there is no overlap between the genera in the test set and the training set on the genus level. Essentially, the distantly related dataset reflects the zero-shot ability of the model for genus-level predictions, akin to its discrimination ability for species in new genera. The discrepancy between the training and test data makes the classification task more challenging on this dataset, so we regard it as a measure of the generalization ability of the model when dealing with data less related to the training set. For the final dataset, the redundancy after clustering is very low in eukaryotes and bacteria. To obtain a dataset that covers as much genomic diversity as possible, 2 million additional sequences are incorporated into these two superkingdoms following the settings in [39]. After that, clustering is performed again based on sequence similarity and the final dataset is constructed.

Comparison with existing baseline models

We implement two models, a base PCVR model and a large PCVR model. In Table 2, we compare the two models' performance of macro AveP on all three datasets with previous methods MMseqs2, MMseqs2 taxonomy, minimap2, DeepMicrobes, BERTax, and feature space balancing approach. In Table 2, where the best performance is highlighted in bold, the performance values for state-of-the-art methods are taken from [33, 39] and the results on the closely related dataset of Subspace KNN and Bagged decision trees are not presented for not released in their original paper.

On the closely related dataset, our large model exhibits notable enhancements, with the macro AveP improving from 95.65% to 98.87% on the superkingdom level and from 83.88% to 96.32% on the phylum level when compared with BERTax. For the distantly related dataset, PCVR achieves 6.0% and 9.95% superior macro AveP on the superkingdom and phylum level to the subspace KNN model. In the case of the final dataset, higher macro AveP values are obtained for a larger dataset compared to the closely related dataset. There is an increase from 99.07% to 99.22% on the superkingdom level and from 95.53% to 96.93% on the phylum level. On the genus level, although optimal performance is not reached, PCVR-Base still outperforms BERTax by 7.73%.

On the superkingdom and phylum levels, our model showcases excellent performance, indicative of its adeptness in capturing complex patterns within sequences. It effectively leverages the feature of the ViT encoder to achieve nuanced differentiation at these hierarchical levels. As a result, even when faced with substantial disparities between testing and training data categories on the distantly related dataset, our model exhibits commendable classification capabilities. We ascribe this outstanding performance to the robust representation capabilities of FCGR and the strong pre-training ability of MAE. For genus level, the approach incorporating feature space balancing [33] achieves the best performance at 86.43%, while PCVR-Base achieves a comparable performance of 74.65% to MMseqs2. Our model, alongside MMseqs2 and DeepMicrobes, relies on *k*-mer counts, in whose feature space certain regions are densely populated while others remain sparsely occupied. The implementation of balanced features effectively addresses the challenge posed by the presence of numerous imbalanced genus classes in the final dataset. Additionally, algorithms such as KNN and decision trees are more sensitive to imbalanced data and tend to perform better when handling such data. The evaluation results on micro AveP are presented in Table 3. The results of the Subspace KNN and Bagged Decision Trees methods are not included because they are not released. Despite the macro AveP of PCVR-Large being only 74.51% on the genus level, the micro AveP reaches 95.01%. Given this phenomenon, we believe that the imbalanced data causes the bad performance on the genus level.

To comprehensively evaluate our method, we incorporate classification accuracy (Acc), the micro-averaged Area Under the ROC Curve (AUC), and the proportion of the predicted sample of the method (Prop) as additional metrics. As shown in Table 4, PCVR surpasses other DL-based methods and achieves comparable Acc to alignment-based approaches. PCVR also demonstrates superior robustness as evidenced by its high AUC, indicating that excellent performance can be attained without complex parameter tuning. Alignment-based methods suffer from strict alignment criteria that prevent

Table 3 Performance comparison in terms of weighted micro AveP

Method	Closely related		Distantly related		Final		
	Supk.	Phyl.	Supk.	Phyl.	Supk.	Phyl.	Genus
MMseqs2	91.71	85.72	62.24	41.46	96.57	92.94	75.44
MMseqs2 tax.	93.99	86.69	67.69	43.62	97.89	93.55	75.86
minimap2	85.00	76.01	42.67	19.88	92.61	86.68	66.26
DeepMicrobes	97.14	87.18	67.47	35.80	98.15	92.38	73.45
BERTax	95.68	84.46	89.91	54.21	98.66	95.20	73.82
PCVR-Base	98.67	95.08	95.88	69.37	98.94	96.29	94.93
PCVR-Large	**99.05**	**96.09**	**96.83**	**74.69**	**99.22**	**96.64**	**97.37**

Table 4 Performance comparison of Acc, AUC, and Prop on closely and distantly related datasets

Method	Closely related						Distantly related					
	Supk.			Phyl.			Supk.			Phyl.		
	Acc	AUC	Prop	Acc	AUC	Prop	Acc	AUC	Prop	Acc	AUC	Prop
MMseqs2	99.63	0.94	87.45	97.30	0.93	87.45	90.73	0.75	52.78	75.04	0.71	52.78
MMseqs2 tax.	98.06	0.96	92.62	92.62	0.93	92.69	80.50	0.79	71.37	59.19	0.73	71.37
minimap2	99.95	0.90	75.59	99.55	0.88	75.59	94.42	0.62	19.88	77.12	0.59	19.88
DeepMicrobes	96.68	0.98	100	87.72	0.94	100	68.39	0.81	100	41.95	0.70	100
BERTax	94.78	0.97	100	85.55	0.93	100	88.95	0.94	100	60.10	0.80	100
PCVR-Base	98.57	0.99	100	95.08	0.99	100	95.31	0.99	100	69.38	0.95	100
PCVR-Large	98.99	0.99	100	96.09	0.99	100	96.48	0.98	100	74.69	0.92	100

Table 5 Results of the one-sided Wilcoxon signed-rank test

Method pair	Median diff	Cohen's d	P-value	Corrected p-value	Significant
PCVR-Base vs. MMseqs2	6.21	0.77	0.016	0.016	✓
PCVR-Base vs. MMseqs2 tax.	4.07	0.68	0.016	0.016	✓
PCVR-Base vs. minimap2	12.28	1.13	0.008	0.013	✓
PCVR-Base vs. DeepMicrobes	8.22	0.72	0.008	0.013	✓
PCVR-Base vs. BERTax	4.53	0.50	0.008	0.013	✓
PCVR-Large vs. MMseqs2	6.68	0.84	0.016	0.016	✓
PCVR-Large vs. MMseqs2 tax.	4.54	0.75	0.016	0.016	✓
PCVR-Large vs. minimap2	12.75	1.18	0.008	0.016	✓
PCVR-Large vs. DeepMicrobes	8.08	0.79	0.008	0.016	✓
PCVR-Large vs. BERTax	5.94	0.58	0.008	0.016	✓
PCVR-Large vs. PCVR-Base	0.64	0.10	0.016	0.016	✓

classification of samples lacking reference sequences, which results in only a small fraction of input samples being successfully categorized. In contrast, PCVR overcomes this fundamental constraint by comparing sequence similarity patterns rather than relying on direct sequence alignment, thereby addressing the low classification coverage issue.

To verify the significance of our experimental results, we conduct statistical analysis on the results of these datasets. We use the one-sided Wilcoxon signed-rank test to assess the performance improvement. We set the significance threshold to $\alpha = 0.05$ and apply Benjamini-Hochberg correction to mitigate errors from multiple comparisons. The performance difference magnitude is measured using Cohen's d. As shown in Table 5, the improvement of PCVR-Base in macro AveP is statistically significant compared to five baseline models, with median diff greater than 4.07% and Cohen's d effect sizes greater than 0.5. Similarly, PCVR-Large also demonstrates statistically significant performance improvements. These tests indicate that the performance enhancements of our model are not only statistically but also practically significant.

Assessment and comparison of encoder

We select BERTax as our primary baseline considering its similar training methodology with our approach. As shown in Table 2, PCVR outperforms BERTax across all datasets.

Table 6 Comparison of the macro AveP with feature retrieval on distantly related and final dataset

Method	Dist. related		Final		
	Supk.	Phyl.	Supk.	Phyl.	Genus
BERTax w/o fine-tuning	87.88	35.90	87.88	46.96	13.76
BERTax w/ fine-tuning	-	-	98.63	95.48	63.62
Ours w/o fine-tuning	81.14	54.10	90.73	80.53	37.25
Ours w/ fine-tuning	90.57	72.21	98.97	96.16	63.89

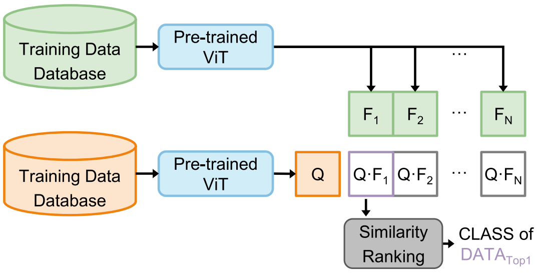

Fig. 3 Framework of feature retrieval

Notably, PCVR demonstrates significant improvement at lower taxonomic ranks. We attribute this performance to the robustness conferred by MAE self-supervised pre-training and the potent capacity of ViT to capture contextual information of features. The combination of ViT and MAE pre-training allows the encoded features to possess a more distinctive signature of FCGR. Moreover, compared to the use of 3-mer tokens as representations of DNA sequences in BERTax, FCGR makes features from different spices exhibit greater distinctiveness.

To further assess the DNA sequence representation quality under the combination of FCGR and MAE pre-training, we empirically investigate the recognition capabilities acquired by the ViT encoder during the pre-training phase. We propose a feature retrieval method for classification based on cosine similarity ranking. Specifically, the category assignments for DNA sequences are determined by identifying the data categories in the database with the highest cosine similarity to the sequence. The framework is depicted in Fig. 3.

For the retrieval data, we employ the training set data as the database, and the test set data serves as the query. Firstly, all the data is encoded into latent features by the encoder. We then compute the similarity between each query and the data in the database, and sort the results based on cosine similarity. Such classification without category information supervision directly reflects the quality of features encoded by the encoder and the representation ability of DNA sequences in different embedding approaches. As BERTax also adopts a combination of self-supervised pre-training and fine-tuning, we employ the same strategy to evaluate the representation of DNA sequences of BERTax. To ensure a fair comparison, we utilize the encoder in PCVR-Base. We do not perform feature retrieval on fine-tuned BERTax on the distantly related dataset as its model weights are not publicly available, and other results of feature retrieval are shown in Table 6. From the results, we found that after pre-training, our encoder exhibits

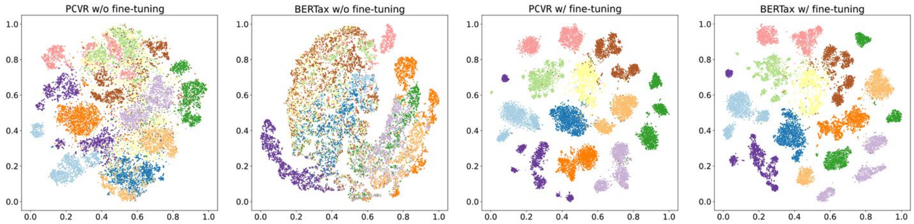

Fig. 4 Visualization of the clustered latent features of PCVR-Base and BERTax via T-SNE. We display PCVR's and BERTax's features before and after fine-tuning. Points of different colors represent different phylum categories, where greater spatial distances between points indicate lower similarity between features

Table 7 Macro AveP of feature retrieval in three layers

FC layer	**Closely related**			**Dist. related**			**Final**		
	Supk.	**Phyl.**	**Genus**	**Supk.**	**Phyl.**	**Genus**	**Supk.**	**Phyl.**	**Genus**
Supk. Layer	**98.78**	96.07	63.50	**90.57**	72.21	18.80	98.98	96.16	63.89
Phyl. Layer	98.71	96.06	62.79	98.71	96.06	62.79	98.90	96.25	63.42
Genus Layer	98.68	**96.37**	**65.72**	98.68	**96.37**	**65.72**	**99.17**	**96.90**	**67.77**

significantly stronger abstraction capabilities on lower taxonomic ranks compared to BERTax. Interestingly, without fine-tuning, PCVR-Base demonstrates superior performance across all datasets except on the superkingdom level of the distantly related dataset. It is evident that after fine-tuning with category information, the performance has seen a substantial improvement from 80.53% to 96.16% on the phylum level and from 37.25% to 63.89% on the genus level.

In order to visually depict the encoding quality of features, we visualize the high-dimensional latent abstract feature data. We select one-fourth of the phylum classes from the test set of the final dataset and map the latent features onto a two-dimensional space using the T-SNE [48] tool. As depicted in Fig. 4, even without category information before fine-tuning, PCVR-Base successfully encodes features with clearer category boundaries in the two-dimensional space compared to BERTax. After fine-tuning, the features of each category can be largely separated in feature space, revealing the ability to learn distinct features for each category.

Assessment of hierarchical classification

To make sure our fine-tuned layers are integrated in a proper way, we assess the input of the classifier on each taxonomic level as Table 7, where the best performance in each column is shown in bold. We employ the strategy for the evaluation of the encoder. We extract the features before each fully connected layer (FC), named Supk. layer, Phyl. layer, and Genus layer, respectively, and conduct feature retrieval depicted in Sect. 4.5 on each taxonomic level.

We expect that incorporating information from higher taxonomic levels would positively impact predictions at lower levels. However, these results suggest that predictions at the phylum level are minimally influenced by the inclusion of superkingdom outputs.

This is primarily due to the small dimension (5-d) of the superkingdom output compared to the encoder features. Regarding predictions at the genus level, integrating features from phylum outputs led to improvements on both closely related and final datasets, resulting in a nearly 3% increase in macro AveP. Notably, the distantly related dataset does not follow this pattern. This deviation can be attributed to the absence of genus categories in the training set of the distantly related dataset, which are present in the test data. Therefore, these anomalous results align with our expectations. Overall, these observations suggest that the design of our fine-tuning network is reasonable.

Ablation study

To investigate how different components of PCVR contribute to its performance, we conduct ablation studies on the pre-trained model settings, FCGR size, fine-tuning structure, and loss weights. We also investigate lighter models by using distillation with smaller models as student models. Furthermore, we explore the impact of training data volume. Due to limitations in computational resources, we opt for ablation studies using the ViT-Base as the encoder. By default, our base model configuration includes a decoder width of 512, a depth of 8, a mask ratio of 75%, and follows the fine-tuning structure depicted in the framework diagram. The loss weights are evenly distributed at a ratio of 1:1:1 by default.

Impact of pre-training. We train ViT using various ViT initialization strategies as shown in Table 8. We employ randomly initialized ViT and ImageNet [40] initialized ViT for fine-tuning for taxonomic classification. These two model initialization strategies both exacerbate the complexity of the training process. Additionally, we simulate random classification using randomly generated numbers, represented as the "Random" case. These suboptimal outcomes show that MAE provides a better initial state for the fine-tuning of ViT, making the model converge to superior performance more quickly. By contrast, it is hard for the randomly initialized ViT and ImageNet pre-trained ViT to attain optimization within an equivalent number of training epochs. These results suggest that initializing ViT with MAE pre-training is the most appropriate apt. It also corroborates the conclusion that MAE pre-training endows ViT with more robust feature representations.

Impact of MAE pre-train settings. In order to verify the effectiveness of model settings of MAE pre-training, we conduct ablation experiments focusing on the masking ratio, decoder layers, and embedding dimensions of the decoder. Our base model uses a masking ratio of 75%, a decoder with 8 blocks, and a width of 512-d.

Table 8 The impact of encoder initialization on distantly related and final datasets

Initialization	Dist. related		Final		
	Supk.	Phyl.	Supk.	Phyl.	Genus
MAE pre-train initialized	94.59	75.08	98.97	96.29	74.65
Randomly initialized	6.67	0.11	2.27	0.05	0.00
ImageNet initialized	12.38	0.30	23.80	5.21	2.33
Random	24.98	3.31	24.98	2.27	0.64

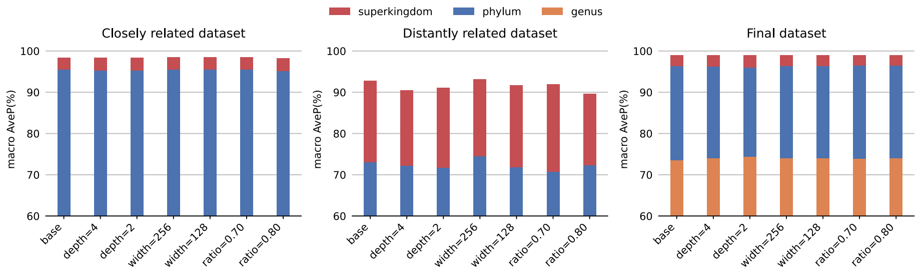

Fig. 5 Comparison of various decoder settings in terms of macro AveP on all three datasets. The results are presented from top to bottom, corresponding to closely related, distantly related, and the final dataset

Table 9 Ablation study of *k*-mer in FCGR on distantly related and final datasets

Settings		Dist. related		Final		
k	patch size	Supk.	Phyl.	Supk.	Phyl.	Genus
6	8	89.76	73.02	99.02	96.45	75.45
5	4	92.81	73.09	98.96	96.36	73.51
4	2	89.30	68.76	98.44	94.92	70.01

From the visualized results shown in Fig. 5, we observe improvement on the distantly related dataset when the encoder width is set to 256. In our task, a significant difference in encoding dimensions between the encoder and decoder may have adverse effects. Apart from the impact of decoder width, the effects of other decoder settings are marginal. This is because the encoder focuses on pattern recognition, and the decoder is primarily involved in reconstruction. During fine-tuning for classification, we mainly utilize the recognition capabilities of the encoder. The results also align with the previous findings in MAE.

Impact of *k*-mer size. To ascertain the optimal image size for FCGR, we proportionally adjust several parameters to segment images with sizes of 32 × 32, 64 × 64, and 128 × 128 into 64 patches. The respective patch sizes are calculated to be 2, 4, and 8 for each image size. The model performs worst when *k* is set to 4 as shown in Table 9. Though the 6-mer FCGR achieves superior performance on the final dataset, its efficacy diminishes on distantly related datasets compared to the 5-mer FCGR. As of our current understanding, each sequence tends to have more similar sequences in the final dataset after incorporating additional eukaryotic and bacterial sequences as mentioned in Sect. 4.3. Thus, discriminating sequences in the final dataset requires more distinguishing features. In line with the observation of our results, 6-mer surpasses others on the final dataset, implying that larger *k*-mers incorporate higher-order sequence information and act as a better input for our model. Considering the overall performance across all datasets, we opt for the 5-mer FCGR as the preferred input for our model.

Impact of fine-tuning architecture design. To validate the effectiveness of the hierarchical fine-tuning head we adopt, we design four additional fine-tuning structures for comparison. Firstly, we design a variation using the output of an additional "CLS" token as the input of the fine-tuning head to explore the impact of global pooling, denoted as “CLS pooling”. To verify the effectiveness of hierarchical structure, we substitute the hierarchical head with an FC. Furthermore, we design two variations named “shared encoder + FC” and “individual encoder + FC” to investigate the impact of sharing encoder parameters. Besides, we observe that many samples belong to “unknown” genus in the final dataset. We try an “unknown” branch to determine whether a sequence belongs to the “unknown” before the classification for genus rank to diminish the negative implications for prediction, denoted as “w/ unknown branch”. Note that the loss function only calculates the loss for genera that do not belong to the “unknown” genus in “w/ unknown branch”. “hierarchical” denotes the fine-tuning structure we ultimately adopt.

Results in Table 10 suggest that the model performs better when global pooling is used for fine-tuning, as opposed to solely using the “CLS” token output by ViT. Moreover, it is difficult to conclude whether sharing ViT parameters is better. Considering the memory, we opt to share ViT parameters. Out of our expectation, the “unknown” branch does not work. A possible explanation is that numerous other imbalanced genera affect the prediction of classifiers. Comparison between “shared encoder + FC” and “hierarchical” reveals that our hierarchical head is capable of providing more complex pattern representations.

Impact of loss weights. Further experiments are conducted to explore the impact of weights of classification losses for various taxonomic levels during model fine-tuning. We configure the loss weight ratios for superkingdom, phylum, and genus as 1:3:6, 2:3:5, 3:3:3, and 3:3:4, as depicted in Fig. 6. Notably, when the loss weight ratio is set to 2:3:5, the model demonstrates an enhancement compared to the 3:3:3 configuration, indicating improved optimization of individual loss components at that particular ratio.

Impact of model reduction. We explore the performance of smaller models by applying distillation. We use ViT-Small and ViT-Tiny as the student model backbone and the fine-tuned PCVR-Base as the teacher model in our distillation. As shown in Table 11, distillation endows smaller models with capabilities, but it is challenging to achieve the performance of the teacher model. To further explore the potential of smaller models, we retrain them using the same methodology as PCVR, obtaining performance that surpasses that of distillation. Nonetheless, smaller models still fail to achieve the comparable performance of PCVR-Base. It suggests that larger models are more capable of leveraging the full potential of PCVR, whereas smaller models should only be considered under limited computation resources.

Table 10 Results with different fine-tuning structural designs

Fine-tuning structure	Dist. related		Final		
	Supk.	Phyl.	Supk.	Phyl.	Genus
CLS pooling	91.24	72.03	98.90	96.13	73.92
shared encoder + FC	92.30	72.18	97.79	96.37	71.16
individual encoder + FC	88.80	72.46	98.52	94.96	72.25
w/ unknown branch	-	-	98.85	96.01	52.16
hierarchical	92.81	73.09	98.96	96.36	73.51

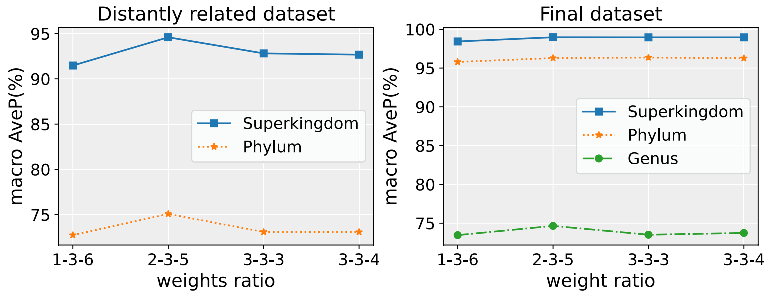

Fig. 6 Impact of weighting proportions on classification losses during fine-tuning. The horizontal axis represents the weights of the loss function for superkingdom-phylum-genus

Table 11 Results with different model reduction strategy

Method	Param.(M)	Dist. related		Final		
		Supk.	Phyl.	Supk.	Phyl.	Genus
PCVR-Large	302.62	96.00	78.67	99.22	96.93	74.51
PCVR-Base	85.29	94.59	75.08	98.97	96.29	74.65
PCVR-Small	21.41	89.00	66.38	98.73	95.71	72.29
PCVR Base-to-Small distiilation	21.41	81.35	64.03	97.79	93.28	64.98
PCVR-Tiny	5.4	88.09	69.15	98.48	95.14	70.63
PCVR Base-to-Tiny distillation	5.4	80.80	60.62	96.74	91.44	60.09

Table 12 Results with different data reduction

Pre-training dataset proportion	Fine-tuning dataset proportion	Dist. related		Final		
		Supk.	Phyl.	Supk.	Phyl.	Genus
100%	100%	94.59	75.08	98.97	96.29	74.65
50%	100%	88.59	68.35	98.92	96.07	73.75
50%	50%	89.31	68.90	98.52	95.21	71.37
50%	25%	89.10	66.27	98.10	94.23	68.35
25%	100%	6.67	0.11	8.08	0.05	0.35
25%	50%	6.67	0.11	8.08	0.05	0.35
25%	25%	6.67	0.11	8.08	0.05	0.35

Impact of data reduction. We investigate the impact of training data volume on PCVR-Base's performance. We set the pre-training data to 50% and 25% of its original volume with fine-tuning data to 100%, 50%, and 25% of its original volume. The results in Table 12 indicate that a 50% reduction in pre-training data achieves similar performance on the final dataset compared to the full 100% pre-training dataset. However, the model's performance on distantly related dataset declines noticeably. Considering the substantial difference between the training and test data in the distantly related dataset, we safely draw the conclusion that pre-training of PCVR is more advantageous in terms of generalization ability than barely training on specific downstream tasks. When pre-training data is reduced to 25% of the original volume, the model fails to train on all fine-tuning settings, further illustrating the vital role of pre-training.

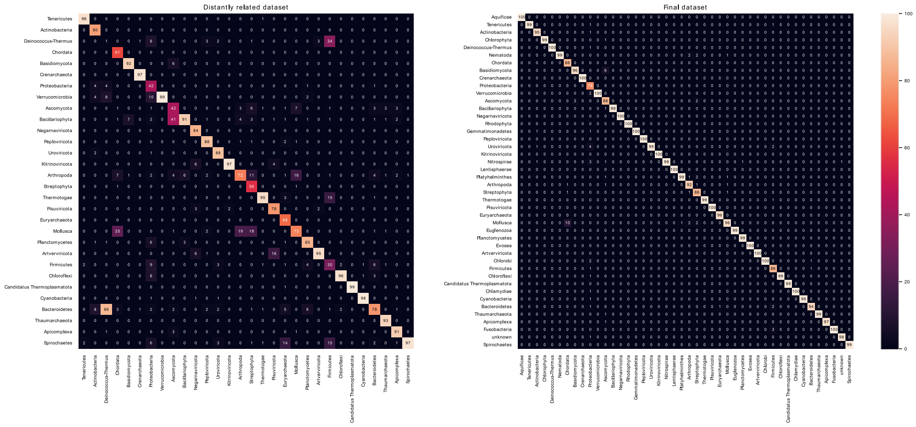

Fig. 7 Confusion matrices of PCVR-Large for the rank phylum of the distantly related and final dataset. The values of each row are percentages of the true positive number of samples from the respective taxonomic class

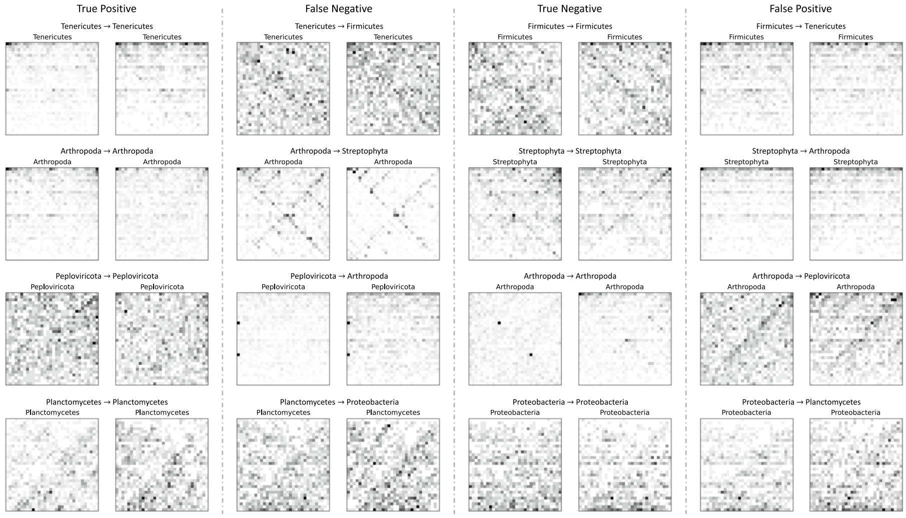

Fig. 8 FCGR images of true positive, false negative, true negative and false positive samples. True classes of DNA sequences are annotated right above their FCGR images. A ⟶ B indicates that the sample with true class A and is predicted as class B by PCVR

Case study

As the confusion matrices in Fig. 7 show, our model achieves accurate classification across most phylum categories, but still exhibits lower performance in some other categories. To provide a more intuitive representation of our results, we select several categories for visualization. Figure 8 illustrates FCGR images of true positive, false negative, true negative and false positive samples. These cases reveal similar patterns between true and false positives, as well as analogous trends between true and false negatives. Both phenomena indicate that PCVR tends to predict the category of DNA sequences according to the discriminative patterns in their FCGR images. For instance, the second line of Fig. 8 shows that sequences with cross-shaped FCGR patterns are consistently classified as "Streptophyta", whereas those lacking this feature are assigned to "Arthropoda". However, similar patterns may exist across different categories, like the "Planctomycetes" and "Proteobacteria" in the fourth line of Fig. 8, which would hinder the classification accuracy of PCVR. Thus, while the model performs well in distinguishing sequences with distinct patterns, sequences exhibiting cross-category pattern similarity are prone to misclassification.

Discussion

In our study, we propose PCVR, a generalized and robust alignment-free model combining ViT and MAE self-supervised pre-training on FCGR. PCVR outperforms the state-of-the-art models on superkingdom and phylum levels, especially in scenarios where a gap exists between the testing and training data. By using FCGR as DNA sequence representation, PCVR diminishes the model's reliance on data length and quality. By enhancing the recognition capabilities of ViT via MAE pre-training, ViT captures high-level abstract features in FCGR, revealing both local and global structural information within DNA sequences. After incorporating downstream sequence category information, it acquires a deeper understanding of DNA sequence structural patterns from hierarchical fine-tuning.

PCVR achieves the best performance on classification on the superkingdom and phylum levels across all three datasets. It is worth noting that, PCVR achieves the most significant improvement on the distantly related dataset, where lower similarity between test and training data. It indicates that it effectively learns the patterns of DNA sequences and holds promising practical applications. This also demonstrates the potential for the generalization and flexibility of pre-trained ViT across a wider range of downstream tasks on DNA sequences. However, on the genus level, PCVR achieves commendable performance compared to many methods, albeit not the best. We attribute such results to the class imbalance in the data, which increases the difficulty of model learning. We also conduct ablation experiments on FCGR, ViT, and fine-tuning structures. These results indicate that PCVR architecture achieves good performance across multiple setting combinations, showing its robustness.

In comparison with BERTax, the combination of FCGR+MAE surpasses the performance of the tokenizer+BERT. It substantiates that methodologies predicated on FCGR representations achieve superior performance when augmented with sophisticated image encoders and commendable training strategies. We posit that treating DNA sequence classification as an FCGR image classification problem may be more appropriate than regarding it as a language task. Thus, we consider PCVR a successful attempt to apply computer vision techniques to address DNA sequence problems. Moreover, we believe FCGR will be preferable for representing DNA sequences in future research. We also reckon MAE pre-training is a powerful tool for other biological image tasks.

In case study, we visualize FCGR images of several samples, finding that PCVR predicts the category of DNA sequences mainly according to the visual patterns in their FCGR. However, we also observe the intra-class diversity and the inter-class similarity of sequence patterns, which may lead to some samples being misclassified. Based on these phenomena, we propose that in practical applications, if the pattern features between classes are clearly distinguished, our model can effectively extract key sequence features and demonstrate strong classification capabilities. For instance, confronting certain genetic diseases or cancers, where specific mutation sites in disease-associated genes are highly conserved across affected individuals, PCVR would exhibit significantly accurate classification due to the variability of conserved mutations. Similarly, in viral classification and origin-tracing tasks, highly species-specific sequence regions provide strong discriminative signals, enabling the model to differentiate between viral species. Furthermore, specific functional genes, e.g., transcription factors binding site, promoter, often have structurally conserved sequence patterns. The conserved patterns provide intuitive feature bases for the classifier to effectively identify and classify DNA sequences. We will apply our method to these real-world scenarios in the future.

Limitations While PCVR demonstrates promising results in DNA taxonomy classification, there are still some limitations. Similar to most of deep learning approaches, the prediction accuracy of PCVR may be influenced by unbalanced distribution of fine-tuning data, e.g., the genus rank in our dataset. Besides, the better DNA sequence representation of PCVR is built upon extra pre-training, whose performance would be discounted if the computation resources are limited. Therefore, we anticipate more applications of advanced computer vision frameworks for data balancing and computation reduction to improve existing models in the future. Furthermore, some sequential information is inevitably lost using FCGR since it mainly extracts frequency information of DNA sequences. We will consider integrating multiple encoding strategies to leverage their respective strengths and overcome the information loss in our future work.

Conclusions

We propose a framework named PCVR for DNA sequence classification. To the best of our knowledge, PCVR is the first model that introduces the ViT to DNA sequence classification and obtains contextualized representations of DNA sequence by MAE pre-training. PCVR optimizes the modeling of long-range dependencies and global information in DNA sequences. Experimental results show that PCVR achieves superior performance across multiple datasets, significantly improving DNA sequence classification accuracy at the superkingdom and phylum levels. Overall, the generalization and robustness of the PCVR establish a promising approach for discovering new species and broad applicability to various genomic tasks.

Abbreviations

FCGR	Frequency chaos game representation
ML	Machine Learning
DL	Deep Learning
CNNs	Convolutional neural networks
ViT	Vision Transformer
MLM	Masked Language Modeling
MAE	Masked Autoencoder
PCVR	Pre-trained Contextualized Visual Representation
MHSA	Multi-head self-attention mechanism
FFN	Feed-forward network
AveP	Average precision
Acc	Classification accuracy
AUC	Micro-averaged Area Under the ROC Curve
Prop	Proportion of the predicted sample of method
FC	Fully connected layer

Acknowledgements

The authors would like to thank Zongmeng Zhang for the helpful discussions and comments.

Author Contributions

J.Z. contributed to the investigation, data acquisition, methodology, coding, experiments, and manuscript drafting and revising. H.W. contributed to the methodology, coding and manuscript revising. K.D. contributed to the investigation, data acquisition, analysis and validation. W.Z. contributed to the conceptualization, methodology, manuscript revising and project administration. C.Z. contributed to the conceptualization and manuscript revising. H.L. contributed to the conceptualization and project administration. All authors read and approved the final manuscript.

Funding

Not applicable.

Data Availability

The datasets of DNA sequence classification can be downloaded from https://osf.io/qg6mv. The source code of our study is available at https://github.com/jiaruizhou/PCVR.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no Conflict of interest.

Received: 13 September 2024 Accepted: 7 April 2025
Published online: 09 May 2025

References

1. Cheng K-O, Wu P, Law N-F, Siu W-C. Compression of multiple DNA sequences using intra-sequence and inter-sequence similarities. IEEE/ACM Trans Comput Biol Bioinf. 2015;12(6):1322–32.
2. Wei Y, Zou Q, Tang F, Yu L. WMSA: A novel method for multiple sequence alignment of DNA sequences. Bioinformatics. 2022;38(22):5019–25.
3. Zhuang Z, Shen X, Pan W. A simple convolutional neural network for prediction of enhancer-promoter interactions with DNA sequence data. Bioinformatics. 2019;35(17):2899–906.
4. Zhang J, Liu B, Wu J, Wang Z, Li J. DeepCAC: A deep learning approach on DNA transcription factors classification based on multi-head self-attention and concatenate convolutional neural network. BMC Bioinformatics. 2023;24(1):345.
5. Sheena K, Nair MS. GenCoder: A novel convolutional neural network based autoencoder for genomic sequence data compression. IEEE/ACM Trans Comput Biol Bioinf. 2024;21(3):405–15.
6. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
7. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol. 2017;35(11):1026–8.
8. Li H. Minimap2: Pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100.
9. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, Wang Z. MetaBAT 2: An adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ. 2019;7: e7359.
10. Ye SH, Siddle KJ, Park DJ, Sabeti PC. Benchmarking metagenomics tools for taxonomic classification. Cell. 2019;178(4):779–94.
11. Imelfort M, Parks D, Woodcroft BJ, Dennis P, Hugenholtz P, Tyson GW. GroopM: An automated tool for the recovery of population genomes from related metagenomes. PeerJ. 2014;2:603.
12. Sieber C, Probst A, Sharrar A, Thomas B, Hess M, Tringe S, Banfield J. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. Nat Microbiol. 2018;3:836–43.
13. Shen C, Chen Y, Xiao F, Yang T, Wang X, Chen S, Tang J, Liao Z. BAT-Net: An enhanced RNA secondary structure prediction via bidirectional GRU-based network with attention mechanism. Comput Biol Chem. 2022;101(C): 107765.
14. Shen C, Mao D, Tang J, Liao Z, Chen S. Prediction of lncRNA-protein interactions based on kernel combinations and graph convolutional networks. IEEE J Biomed Health Inform. 2024;28(4):1937–48.
15. Akkaya UM, Kalkan H. Classification of DNA sequences with k-mers based vector representations. In: Innovations in Intelligent Systems and Applications Conference. 2021;pp. 1–5
16. Gunasekaran H, Ramalakshmi K, Rex Macedo Arokiaraj A, Deepa Kanmani S, Venkatesan C, Suresh Gnana Dhas C. Analysis of DNA sequence classification using CNN and hybrid models. Comput Math Methods Med. 2021;2021(1):1835056.
17. Bae H, Min S, Choi H-S, Yoon S. DNA Privacy: Analyzing malicious DNA sequences using deep neural networks. IEEE/ACM Trans Comput Biol Bioinf. 2020;19(2):888–98.
18. Du Z, Xiao X, Uversky VN. Classification of chromosomal DNA sequences using hybrid deep learning architectures. Curr Bioinform. 2021;15(10):1130–6.
19. 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 International Conference on Computer Systems and Technologies. 2016;pp. 222–228

20. Abd-Alhalem SM, Soliman NF, Eldin S, Abd Elrahman SE, Ismail NA. El-Rabaie E-SM, El-Samie FEA, Bacterial classification with convolutional neural networks based on different data reduction layers. Nucleosides, Nucleotides & Nucleic Acids. 2020;39(4):493–503.
21. Millán Arias P, Alipour F, Hill KA, Kari L. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. PLoS ONE. 2022;17(1):1–25.
22. Jin M, Koh HY, Wen Q, Zambon D, Alippi C, Webb GI, King I, Pan S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. IEEE Trans Pattern Anal Mach Intell. 2024;46(12):10466–85.
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Proceedings of the International Conference on Neural Information Processing Systems. 2017;30:6000–10.
24. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations. 2021;pp. 1–21
25. Zhang C, Zhang M, Zhang S, Jin D, Zhou Q, Cai Z, Zhao H, Liu X, Liu Z. Delving deep into the generalization of vision transformers under distribution shifts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022;pp. 7277–7286
26. Kenton JDM-WC, Toutanova LK. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019;pp. 4171–4186
27. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022;pp. 16000–16009
28. Yue Y, Huang H, Qi Z, Dou H-M, Liu X-Y, Han T-F, Chen Y, Song X-J, Zhang Y-H, Tu J. Evaluating metagenomics tools for genome binning with real metagenomic datasets and CAMI datasets. BMC Bioinformatics. 2020;21(1):1–15.
29. Mirdita M, Steinegger M, Breitwieser F, Söding J, Levy Karin E. Fast and sensitive taxonomic assignment to metagenomic contigs. Bioinformatics. 2021;37(18):3029–31.
30. Mangkunegara IS, Purwono P. Analysis of DNA sequence classification using SVM model with hyperparameter tuning grid search CV. In: Proceedings of the IEEE International Conference on Cybernetics and Computational Intelligence. 2022;pp. 427–432
31. Habib MA, Manik MMH. Classification of DNA sequence using machine learning techniques. 2022;pp. 1–5
32. Liang Q, Bible PW, Liu Y, Zou B, Wei L. DeepMicrobes: Taxonomic classification for metagenomics with deep learning. NAR Genomics and Bioinformatics. 2020;2(1):009.
33. Fuhl W, Zabel S, Nieselt K. Improving taxonomic classification with feature space balancing. Bioinformatics Advances. 2023;3(1):092.
34. Mikolov T, Chen K, Corrado GS, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations. 2013;pp. 1–12
35. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2014;pp. 1532–1543
36. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018;pp. 2227–2237
37. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
38. Radford A, Narasimhan K, Salimans T, Sutskever I, et al. Improving language understanding by generative pre-training. 2018;1–12
39. Mock F, Kretschmer F, Kriese A, Böcker S, Marz M. Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. Proc Natl Acad Sci. 2022;119(35):2122636119.
40. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2009;pp. 248–255
41. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2016;pp. 770–778
42. Almeida JS, Carriço JA, Maretzek A, Noble PA, Fletcher M. Analysis of genomic sequences by chaos game representation. Bioinformatics. 2001;17(5):429–37.
43. Löchel HF, Heider D. Chaos game representation and its applications in bioinformatics. Comput Struct Biotechnol J. 2021;19:6263–71.
44. Adetiba E, Badejo J A, Thakur S, Matthews VO, Adebiyi MO, Adebiyi EF. Experimental investigation of frequency chaos game representation for in silico and accurate classification of viral pathogens from genomic sequences. In: Proceedings of the International Work-Conference on Bioinformatics and Biomedical Engineering, 2017;pp. 155–164
45. Almeida JS, Carriço JA, Maretzek A, Noble PA, Fletcher M. Analysis of genomic sequences by chaos game representation. Bioinformatics. 2001;17(5):429–37.
46. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, et al. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, 2019;pp. 8024–8035
47. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations. 2017;pp. 1–19
48. Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res. 2008;9(11):2579–605.

Publisher's Note