OXFORD

*Briefings in Bioinformatics*, 2025, **26(5)**, bbaf459
https://doi.org/10.1093/bib/bbaf459
**Problem Solving Protocol**

# Energy entropy vector: a novel approach for efficient microbial genomic sequence analysis and classification

Hao Wang[1,2], Guoqing Hu[2,3], Stephen S.-T. Yau[2,3,4,*]

[1]Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, P. R. China
[2]Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing 101408, P. R. China
[3]Hetao Institute of Mathematics and Interdisciplinary Sciences (HIMIS), Shenzhen 518000, Guangdong, P. R. China
[4]Department of Mathematical Sciences, Tsinghua University, Beijing 100084, P. R. China

*Corresponding author. Hetao Institute of Mathematics and Interdisciplinary Sciences (HIMIS), Shenzhen 518000, Guangdong, P. R. China; Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing 101408, P. R. China; and Department of Mathematical Sciences, Tsinghua University, Beijing 100084, P. R. China. E-mail: yau@uic.edu

## Abstract

With the rapid development of genomic sequencing technologies, there is an increasing demand for efficient and accurate sequence analysis methods. However, existing methods face challenges in handling long, variable-length sequences and large-scale datasets. To address these issues, we propose a novel encoding method—Energy Entropy Vector (EEV). This method encodes gene sequences of arbitrary length into fixed-dimensional vector representations by modeling nucleotide energy characteristics based on information entropy. Experiments conducted on five microbial datasets demonstrate that, compared to traditional alignment-free methods, EEV achieves higher accuracy in convex hull classification and species classification tasks, with improvements of 15% to 30% in family-level classification. In phylogenetic tree construction, EEV significantly accelerates the process relative to multiple sequence alignment methods while maintaining high tree quality, enabling rapid and accurate phylogenetic reconstruction. Moreover, EEV supports flexible dimensional expansion by superimposing nucleotide energies, enhancing its ability to represent complex genomic sequences while effectively alleviating sparsity issues in high-dimensional representations. This study provides an efficient gene encoding strategy for large-scale genomic analysis and evolutionary research.

**Keywords:** energy entropy; machine learning; species classification; phylogenetic tree

## Introduction

With the rapid development of genomic sequencing technologies, bioinformatics has entered a new stage centered on data-driven approaches [1–3]. Extracting biologically meaningful features from genomic sequences has become a significant challenge in key tasks such as gene classification and evolutionary analysis. Traditional sequence comparison tools, such as BLAST [4], and multiple sequence alignment methods like ClustalW [5] and MAFFT [6] align sequences by analyzing their similarities, providing intuitive results that have played important roles in constructing phylogenetic trees [7–9]. However, as sequence length increases, the computational complexity of these methods rises significantly, making it difficult to meet real-time requirements when processing long sequences [10]. Moreover, relying on local similarities, these methods may fail to fully capture the deep similarities and global dependencies in complex sequences. In recent years, deep learning methods have shown potential in genomic sequence analysis. However, to accommodate the fixed input dimensions of neural networks, gene sequences are often truncated or padded before encoding, which may lead to information loss or noise introduction [11–15].

Alignment-free methods, as an efficient alternative, have gained increasing attention [16–18]. Unlike traditional alignment methods, alignment-free methods directly compute feature vectors from genomic sequences through statistical analysis and mathematical modeling, thereby reducing computational costs [19]. These methods provide new perspectives for sequence similarity analysis, species classification, and phylogenetic tree construction [17, 20, 21]. For example, the k-mer method compares sequences by counting the occurrence frequency of all nucleotide fragments of length $k$ [22]. Feature Frequency Profile (FFP) further enhances the accuracy of k-mer-based genomic comparison by optimizing feature length [23]. More recently, a novel k-mer topology method based on persistent Laplacians has been proposed, which captures the multiscale topological features of sequences and provides improved metrics for genome comparison [24]. Another important alignment-free sequence analysis approach is the natural vector (NV) method [25]. This method constructs a 12-dimensional vector by counting the number of each nucleotide, its average position, and the second-order moment of its positions to characterize genomic sequence features. The k-mer natural vector method represents genomic sequences based on the quantity and distribution of k-mers [26], which further expands the modeling capability and application scope of the NV method. The NV method not only has theoretical advantages but has also been

**Received:** April 7, 2025. **Revised:** July 18, 2025. **Accepted:** August 11, 2025

widely used in genomic classification and phylogenetic studies [27, 28]. The covariance natural vector (CNV) extends the NV by incorporating the covariance of nucleotide positions, constructing an 18-dimensional feature vector [29]. These methods have also been extended to protein sequence analysis, designing a 250-dimensional NV model that includes amino acid distribution, average positions, variances, and covariances, which has been successfully applied to the classification of bacterial and viral families [30].

Although the k-mer method, the NV method, and its extension have achieved some success in certain tasks, they still have significant limitations. The k-mer method's dimensionality grows exponentially with the increase in $k$, resulting in a higher proportion of zero elements in high-dimensional vectors, which causes sparsity. The NV method primarily focuses on basic statistical features of nucleotides, such as position and frequency, which can only reflect local information of the sequence and fail to describe the relationships between these statistical indicators [31, 32]. The CNV method extends the feature dimension by incorporating covariance of positions, but this approach essentially relies on stacking statistical indicators, making it difficult to fully characterize the global distribution features of nucleotides and their complex relationships. In particular, it lacks directional information when describing the interactions between nucleotide positions, for example, it does not include dependency relationships such as nucleotide A to G or G to A. Moreover, when generating vectors, the NV method concatenates the number of nucleotides, their average positions, and the second-order moments of their positions. However, these statistical indicators are independent of each other and cannot reflect the global regularity of sequence features. Therefore, the method shows significant limitations when dealing with long sequences or sequences with complex structures.

In this study, we propose the energy entropy vector (EEV), which effectively represents the features of nucleotide sequences. The method integrates the statistical features of nucleotides, including occurrence probability, conditional probability, and relative position, and then calculates information entropy [33] to quantify the uncertainty of the sequence. It also uses mutual information [34, 35] to enhance the statistical dependencies between nucleotides, thereby providing a more comprehensive measure of the sequence's information content. On this basis, we introduce a weighting factor to construct the EEV, which not only quantifies the uncertainty of the sequence but also enhances the characterization of its global distribution features. The EEV not only captures the dependencies between nucleotides but also reflects the spatial organization of the sequence, resulting in a more comprehensive feature representation. Experimental results demonstrate that the EEV exhibits strong discriminative power in convex hull separation and species classification tasks. Moreover, as a physical quantity with magnitude, energy entropy has clear physical and biological significance and provides an effective method for the quantitative analysis of nucleotide sequences.

# Materials and methods

## Data

This study used five microbial datasets: Archaea, Bacteria, Fungi, Virus, and Fungi DNA barcode. The data for Archaea, Bacteria, Fungi, and Virus were sourced from the National Center for Biotechnology Information (NCBI), while the Fungi DNA barcode dataset was obtained from the Barcode of Life Data System (BOLD) [36]. All datasets were published in prior studies and are based on experimentally generated genome or gene sequences. To compare with the natural vector method and the covariance natural vector method, we preprocessed the datasets according to the strategies reported in [36]: removal of sequences lacking family-level taxonomic information; deletion of samples with fewer than three sequences in each family; and retention of only those sequences related to the Internal Transcribed Spacer (ITS) of fungi in the Fungi DNA barcode dataset [37, 38].

After the aforementioned preprocessing, the datasets were summarized as follows: The *Archaea* dataset contains 281 genomic sequences covering 20 families; the *Bacteria* dataset contains 16373 genomic sequences covering 178 families; the *Fungi* dataset contains 387 genomic sequences covering 22 families; the *Virus* dataset contains 7382 genomic sequences covering 83 families; and the *Fungi DNA barcode* dataset contains 95524 sequences covering 467 families.

## Methods

To describe the distribution features of nucleotide sequences, we propose the EEV based on information entropy and a weighting strategy. The EEV is formulated by combining information entropy with the weighting factor of nucleotides, as follows:

$$E = -\left(\sum_{m\in i} w_m\right)\left(\sum_{m\in i} f_m \log f_m\right), \quad i \in \mathrm{Comb}(4,k) \tag{1}$$

where:

- $i$ is a specific subset selected from $\{A, C, G, T\}$;
- $\mathrm{Comb}(4,k)$ denotes all subsets of size $k$, where $k \in \{1,2,3,4\}$;
- $w_m$ is the weighting factor for element $m$, which can be the number of nucleotides, pairs, etc.;
- $f_m$ is the feature proportion of element $m$, which can represent the probability of nucleotide occurrence, positional proportion, etc.

We defined four information metrics that effectively describe nucleotide sequences. These include nucleotide probability distribution, global dependency between nucleotides, relative positions of nucleotides, and mutual information between nucleotides. We utilized these metrics and their corresponding weighting factors in a Hadamard product to obtain the energy entropy of genomic sequences. Figure 1 shows the calculation process of the EEV. The following sections will provide detailed explanations of these metrics.

### *Energy entropy of nucleotide probability distribution ($E_1$)*

In a genomic sequence, the probability of occurrence of each nucleotide is a key factor in describing its distribution characteristics. $E_1$ calculates the information entropy based on the nucleotide probability distribution and combines nucleotide counts as weights to compute the energy entropy. This metric reflects the global probability distribution characteristics of nucleotides and quantifies their contribution to the sequence information.

Taking the energy entropy of individual nucleotides as an example, $E_1$ can be obtained by substituting the weighting factor and feature proportion into Equation 1, where:

- **Weighting Factor**: $w_m = n_m$, representing the count of nucleotide $m$;
- **Feature Proportion**: $f_m = p_m$, the occurrence probability of nucleotide $m$, defined as: $p_m = \frac{n_m}{N}$, where $N$ is the total length of the sequence.

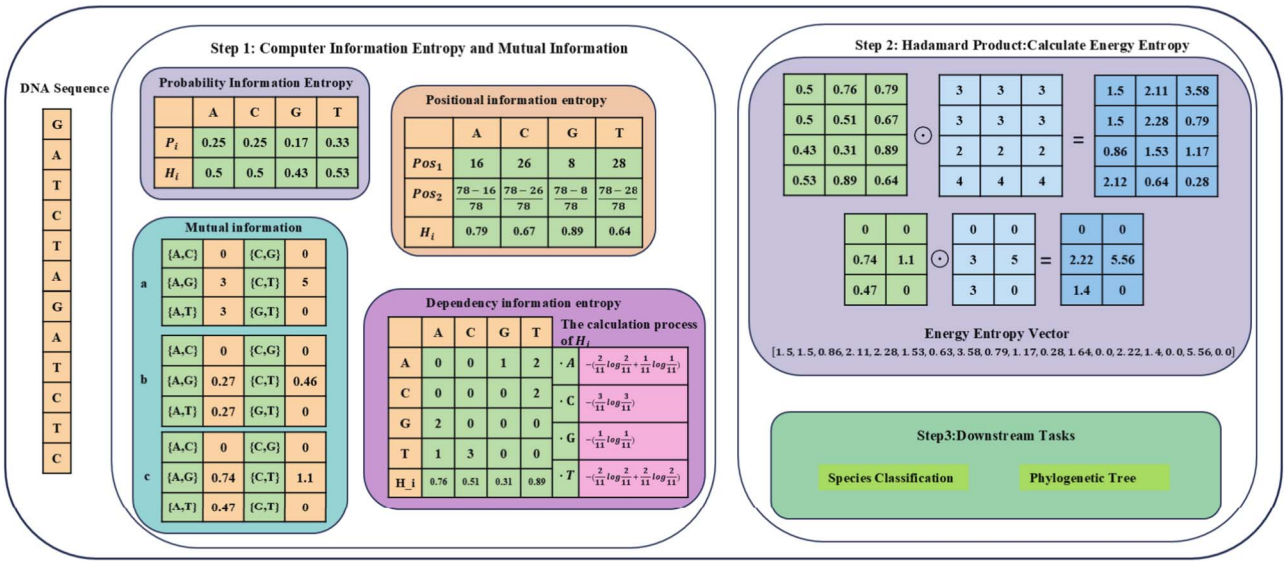


Figure 1. EEV Calculation Process for Nucleotide Sequences. Step 1: (1) **Probability Information Entropy**: Calculate the probability information entropy, where $P_i$ is the probability of the nucleotide; (2) **Positional Information Entropy**: $\mathrm{Pos}_1$ represents the sum of all positions of the nucleotide, and $\mathrm{Pos}_2$ represents the relative position proportion; (3) **Dependency Information Entropy**: Calculate the dependency information entropy through nucleotide pairs, where $(\cdot A)$ represents the mutual information ending with nucleotide $A$; (4) **Mutual Information**: Including (a) the number of nucleotide pairs, (b) the probability of nucleotide pairs, (c) the mutual information calculated by Equation 4. Where $H_i$ is the information entropy of each part. Step 2: Calculate the energy entropy by combining information entropy and weighting factors through the Hadamard product to generate the energy entropy vector. Step 3: Apply the calculated features to downstream tasks, such as species classification and phylogenetic tree construction.

Considering only the energy entropy of individual nucleotides yields with k = 1 yields $C(4,1) = 4$ energy values. By superimposing the energies of nucleotides through combinations, a 15-dimensional vector can be obtained, calculated as $C(4,1)+C(4,2)+C(4,3)+C(4,4) = 15$. This approach helps represent the complexity of sequence features and reveals more comprehensive sequence information patterns.

However, $E_1$ only describes the global distribution characteristics of nucleotides and does not reflect the dependencies between them. To address this, we introduce the dependencies between nucleotides in the following section.

## *Global dependency energy entropy of nucleotide pairs ($E_2$)*

In genomic sequences, the global dependency relationships between nucleotide pairs are important features for revealing sequence regularities. In $E_2$, we use a generalized approach to characterize dependency by quantifying the global dependency of a nucleotide on all other nucleotides through the distribution of nucleotide pairs ending with a specific nucleotide.

Taking the energy entropy of single nucleotides as an example, $E_2$ is obtained by substituting the feature proportion $f_m$ and the weighting factor $w_m$ into Equation 1, where:

- **Weighting Factor**: $w_m = n_m$, representing the count of nucleotide $m$;
- **Feature Proportion**: $f_m = P(m|\cdot)$, representing the relative frequency of nucleotide pairs ending with nucleotide $m$, defined as:

$$P(m|\cdot) = \frac{\sum_{X\in\{A,C,G,T\}} n_{Xm}}{\sum_{i,j\in\{A,C,G,T\}} n_{ij}}, \tag{2}$$

where the numerator represents the total count of nucleotide pairs ending with nucleotide $m$, and the denominator represents the total count of all nucleotide pairs in the sequence.

This formula captures the global dependency of nucleotide $m$, rather than the local relationship described by the traditional conditional probability $P(j|i)$. When the distribution of nucleotides $A, C, G, T$ is relatively uniform, $f_m$ is similar to the global probability in $E_1$. However, when there are significant differences in the distribution, $E_2$ can more prominently highlight the statistical dependencies between nucleotides. Additionally, $E_2$ can be extended by stacking nucleotide energies to increase dimensionality, thereby enhancing its ability to express local information patterns in the sequence.

## *Energy entropy of nucleotide relative positions ($E_3$)*

The relative position distribution of nucleotides is an important feature for uncovering spatial regularities in sequences. $E_3$ describes the spatial distribution based on the relative positions of nucleotides. Taking the energy entropy of single nucleotides as an example, $E_3$ can be calculated by substituting the weighting factor $w_m$ and feature proportion $f_m$ into Equation 1, where:

- **Weighting Factor**: $w_m = n_m$, representing the count of nucleotide $m$;
- **Feature Proportion**: $f_m = q_m$, representing the relative position proportion of nucleotide $m$, defined as:

$$q_m = \frac{L - \sum_{i\in\{A,C,G,T\}\setminus\{m\}} \mathrm{Pos}_i}{L} \tag{3}$$

where $\mathrm{Pos}_i$ denotes the position of nucleotide $i$ in the sequence, and $\{A, C, G, T\} \setminus \{m\}$ represents the set of nucleotides excluding $m$ from the sequence, and $L = \frac{N(N+1)}{2}$, the sum of all positions in the sequence, where $N$ represents the sequence length.

Similarly, $E_3$ can also be extended by stacking nucleotide energies to capture more complex spatial distribution characteristics and reveal multi-level spatial patterns in the sequence.

### *Mutual information energy entropy between nucleotides ($E_4$)*

The shared information between nucleotides can reveal underlying biological patterns. Mutual Information is an information-theoretic measure that can be used to quantify the shared information between nucleotide pairs and reflect their interactions.

Considering the mutual information energy between two nucleotides, it is used as the feature proportion $f_m$ and substituted into Equation 1 to obtain $E_4$, where:

- **Weighting Factor**: $w_m = n_{ij} + n_{ji}$, representing the sum of counts of nucleotide pairs $\{i, j\}$ and $\{j, i\}$;
- **Feature Proportion**: $f_m = I(i, j)$, representing the mutual information between nucleotide pairs $\{i, j\}$, defined as:

$$I(i,j) = p_{ij} \cdot \log \frac{p_{ij}}{p_i \cdot p_j}. \tag{4}$$

where:

$$p_{ij} = \frac{n_{ij}}{n}, \quad p_i = \frac{n_i}{n}, \quad p_j = \frac{n_j}{n}$$

The parameters are explained as follows: $n_{ij}$: The count of nucleotide pair $\{i, j\}$; $n$: The total count of all nucleotide pairs in the sequence; $p_i$ and $p_j$: The occurrence probabilities of nucleotides $i$ and $j$.

In this study, we transform nucleotide sequences of arbitrary length into a feature vector concatenated from $E_1$, $E_2$, $E_3$, and $E_4$ by calculating the energy entropy of single nucleotides. The dimensionality of $E_1, E_2$, and $E_3$ is $C(4, 1)$, while $E_4$, representing the mutual information between nucleotides and involving at least two nucleotides, has a dimensionality of $C(4, 2) = 6$. Thus, the dimensionality of the final feature vector is $C(4, 1) + C(4, 1) + C(4, 1) + C(4, 2) = 18$.

Furthermore, the EEV can be further expanded in dimensionality by superimposing nucleotide energies, thereby more comprehensively characterizing sequence features. The specific details of dimensionality expansion and its impact will be detailed in the last experiment of the results section.

## Convex hull analysis

Convex Hull Analysis is used to evaluate the classification capability of feature vectors in high-dimensional spaces and applied in genomic classification and phylogenetic analysis [39, 40]. For each genomic family, feature vectors and their convex hulls are constructed. Let $F = \{x_1, x_2, \ldots, x_n\}$ denote the set of feature vectors of $n$ genomic sequences in a given family, then its convex hull is defined as:

$$S(F) = \left\{ \sum_{i=1}^{n} \lambda_i x_i \,\middle|\, \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1 \right\}. \tag{5}$$

To evaluate classification performance, determine whether the convex hulls $S(F_1)$ and $S(F_2)$ of two genomic families intersect, which can be converted to the following linear programming problem:

$$\sum_{i=1}^{n} \lambda_i x_i = \sum_{j=1}^{m} \mu_j y_j, \quad \sum_{i=1}^{n} \lambda_i = \sum_{j=1}^{m} \mu_j = 1. \tag{6}$$

If this equation has a solution, the convex hulls intersect; otherwise, they are disjoint. Classification capability is measured by the disjoint rate $P_{\text{disjoint}}$, which is calculated as:

$$P_{\text{disjoint}} = \frac{\text{Number of disjoint family pairs}}{\text{Total number of family pairs}}$$

where the total number of family pairs is determined by all possible convex hull combinations $C_n^2$. A higher $P_{\text{disjoint}}$ value indicates stronger separation capability of the feature vectors in high-dimensional space.

# Results

## Comparison of family classification performance

To evaluate the performance of the EEV in species family classification, the experiment compared it with the NV and CNV. We used five datasets, *Archaea*, *Bacteria*, *Fungi*, *Virus*, and *Fungi DNA barcode*, to test their convex hull classification rate and classification performance.

### *Convex hull analysis*

In this study, based on the principle of convex hulls, we analyzed EEV, NV, and CNV. In the experiments, each dataset comprised several families, and the gene sequences of different families were used to construct convex hulls. By calculating the disjoint rate of convex hulls between different families, we evaluated the classification performance of each method on family-level classification tasks.

Table 1 shows that the EEV achieved the highest disjoint rates of convex hulls across all datasets, reaching 100% in the Archaea and Fungi datasets. Compared to the CNV, which is also 18-dimensional, the EEV demonstrated superior discriminative ability for species family classification. This indicates that, at the same dimensionality, the energy entropy vector can more effectively separate species families.

### *Family classification*

To further assess the performance of the EEV in classification tasks, we conducted classification experiments using three machine learning methods: MLP, Random Forest [41], and XGBoost [42]. The experiments used the EEV, NV, and CNV as feature inputs to train models and evaluate their classification accuracy.

As can be seen from Table 2, the EEV demonstrated good classification performance on the five datasets, with significant improvements in test set accuracy compared to the NV and CNV, particularly on the Archaea, Bacteria, and Fungi datasets. On the Archaea dataset, the test set accuracy of the EEV was 28% higher than that of the CNV when using the MLP model (0.8421 versus 0.5614), and the improvement reached 30% with the Random Forest model (0.8247 versus 0.5263). On the Bacteria dataset, the test set accuracy of the EEV was 17% higher than that of the NV when using the MLP model (0.9282 versus 0.7566). On the Fungi dataset, the EEV achieved higher test set accuracy than other methods across all models. For instance, under XGBoost, the accuracy was approximately 22% higher (0.6154 versus 0.3974). However, on the Virus and Fungi DNA barcode datasets, although the EEV still showed some improvement, the increase was relatively smaller. Overall, the EEV not only improved classification accuracy but also effectively enhanced the generalization ability of models, particularly on complex biological datasets.

Convex hull analysis and classification experiments indicate that the EEV exhibits strong family-level discriminative ability in representing gene sequence features. Its disjoint convex hull rate and classification accuracy outperform those of the NV and the

Table 1. Comparison of convex hull disjoint rates across different datasets using NV, CNV, and EEV methods

| DataSets | Convex hull pairs | NV | CNV | EEV |
|---|---|---|---|---|
| Archaea | $C_{20}^2 = 190$ | 96.8% | 100% | **100%** |
| Bacteria | $C_{178}^2 = 15753$ | 92.5% | 96.7% | **99.5%** |
| Virus | $C_{83}^2 = 3403$ | 94.8% | 97.7% | **98.6%** |
| Fungi | $C_{22}^2 = 231$ | 89.6% | 97.0% | **100%** |
| Fungi DNA barcode | $C_{467}^2 = 108811$ | 69.1% | 82.1% | **85.1%** |

Table 2. Evaluation of classification models with NV, CNV, and EEV across different datasets

| Model | Method | Archaea | | Bacteria | | Virus | | Fungi | | Fungi DNA barcode | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test | Train | Test | Train | Test |
| MLP | NV | 0.5938 | 0.5439 | 0.7927 | 0.7566 | 0.5598 | 0.5586 | 0.5502 | 0.5256 | 0.7499 | 0.7246 |
| | CNV | 0.5759 | 0.5614 | 0.7815 | 0.7542 | 0.5578 | 0.5396 | 0.5857 | 0.5385 | 0.7566 | 0.7259 |
| | EEV | **0.9375** | **0.8421** | **0.9743** | **0.9282** | **0.7617** | **0.7177** | **0.9126** | **0.6667** | **0.8184** | **0.7813** |
| Random Forest | NV | 1.0000 | 0.5263 | 1.0000 | 0.7927 | 1.0000 | 0.7813 | 1.0000 | 0.4487 | 0.9914 | 0.7747 |
| | CNV | 1.0000 | 0.5263 | 1.0000 | 0.7890 | 1.0000 | 0.7793 | 1.0000 | 0.4487 | 0.9913 | 0.7761 |
| | EEV | **1.0000** | **0.8247** | **1.0000** | **0.9127** | **1.0000** | **0.8517** | **1.0000** | **0.7436** | **0.9914** | **0.8112** |
| XGB | NV | 0.5439 | 0.5090 | 0.5596 | 0.5396 | 0.7596 | 0.7637 | 0.4359 | 0.3974 | 0.6946 | 0.7104 |
| | CNV | 0.4561 | 0.5737 | 0.7637 | 0.7759 | 0.7596 | 0.7637 | 0.3974 | 0.3974 | 0.7104 | 0.7104 |
| | EEV | **0.8070** | **0.8070** | **0.8940** | **0.8294** | **0.8294** | **0.8463** | **0.6154** | **0.6154** | **0.7371** | **0.7371** |

CNV. This validates its effectiveness in family-level classification tasks for gene sequences.

## Kingdom analysis

To evaluate the performance of EEV in classifying species kingdoms, we conducted experiments on the Archaea, Fungi, and Virus datasets, which contained 281, 387, and 7,382 sequences, respectively. Each kingdom dataset was randomly split into training (80%), validation (10%), and test (10%) subsets to ensure a fair and robust evaluation. To maintain balanced class sizes, the Virus dataset was downsampled by randomly selecting 387 sequences. To assess variability, downsampling and training were repeated ten times, and the mean and standard deviation on the test set were reported. Additionally, to evaluate the method's performance under real-world class imbalance, the model was also tested once on the original, unbalanced Virus dataset without downsampling.

All methods were trained under the same MLP architecture, which consisted of three hidden layers with 128, 64, and 32 neurons, respectively, using ReLU activation functions. Models were trained for 200 epochs with a learning rate of 0.001, and the checkpoint with the highest validation accuracy was saved for final evaluation on the independent test set.

### *Performance evaluation*

We used the preserved optimal model to predict the test datasets, and the results are summarized in Table 3. The EEV method performed the best, with an overall mean accuracy of 0.9594 and a mean micro-AUC of 0.9800 based on the ten random undersampling runs. In comparison, the mean accuracy rates of NV and CNV were 0.9274 and 0.9349, respectively, with their micro-AUC values lower than those of the EEV method. In addition, the bottom part of Table 3 shows the results obtained on the full, imbalanced Virus dataset, further demonstrating that the EEV method consistently outperforms the other methods under real-world class imbalance.

### *Phylogenetic tree construction*

To validate the effectiveness of the EEV method in biological classification, we constructed phylogenetic trees using the Unweighted Pair Group Method with Arithmetic Mean(UPGMA) [43] and compared it with ClustalW and MAFFT, focusing on evaluating both computational efficiency and the performance of phylogenetic tree construction.

#### Computational Efficiency Comparison

All experiments were conducted on a server equipped with an Intel Xeon Gold 6346 CPU (64 cores, 3.10 GHz) and 512 GB RAM, running Ubuntu 22.04.5 LTS.

Due to the excessive computation time required to process complete genome data (such as Archaea, with some sequence lengths reaching 1-4 million bp) using traditional alignment methods, which is difficult to handle, we extracted partial gene fragments and conducted experiments with sequence lengths up to 5k, 10k, and 80k. We measured the computational time for constructing the phylogenetic tree using EEV, ClustalW, and MAFFT, with the results shown in Table 4.

The experimental results show that, with 80k sequences, ClustalW requires 46,311.31 seconds, MAFFT requires 20,515.12 seconds, while EEV only takes 4.96 seconds, achieving acceleration factors of 9,337 and 4,137, respectively. With 10k sequences, EEV is still 356 times faster than MAFFT and 838 times faster than ClustalW, demonstrating a significant computational advantage. For complete genome data, EEV can still construct the tree in 85.74 seconds, further validating its efficiency.

#### Classification Performance Comparison

We constructed four phylogenetic trees (Fig. 2). The results showed that when the maximum sequence length was 5000 base pairs, Figures (a), (b), and (c) could effectively distinguish Fungi from Archaea and Virus, but there was still some overlap between Archaea and Virus.

In the phylogenetic tree constructed by EEV, the families Haloarculaceae and Sulfolobaceae within Archaea are clearly

Table 3. Comparison of NV, CNV, and EEV classification performance across kingdoms. The first three rows correspond to 10-time random under-sampling; the last three rows (marked with *) show results on the full imbalanced dataset without downsampling

| Method | Acc | AUC | | | Micro-AUC |
|---|---|---|---|---|---|
| | | Archaea | Virus | Fungi | |
| EEV | 0.9594 ± 0.0194 | 1.0000 ± 0.0001 | 0.9865 ± 0.0087 | 0.9366 ± 0.1049 | 0.9800 ± 0.0220 |
| NV | 0.9274 ± 0.0167 | 1.0000 ± 0.0001 | 0.9679 ± 0.0133 | 0.9402 ± 0.0443 | 0.9743 ± 0.0118 |
| CNV | 0.9349 ± 0.0163 | 0.9997 ± 0.0010 | 0.9667 ± 0.0127 | 0.9405 ± 0.0566 | 0.9767 ± 0.0119 |
| EEV* | 0.9901 | 1.0000 | 0.9953 | 0.9678 | 0.9985 |
| NV* | 0.9863 | 1.0000 | 0.9862 | 0.9639 | 0.9979 |
| CNV* | 0.9826 | 1.0000 | 0.9851 | 0.9454 | 0.9976 |

Table 4. Computation Time of Different Methods for Phylogenetic Tree Construction (in seconds)

| Method | 5k | 10k | 80k | Full sequence |
|---|---|---|---|---|
| **EEV** | **1.43** | **1.66** | **4.96** | **85.74** |
| **MAFFT** | 152.24 | 592.70 | 20515.12 | - |
| **ClustalW** | 431.72 | 1393.87 | 46311.31 | - |
| **Speed-up (EEV versus MAFFT)** | **106×** | **356×** | **4,137×** | - |
| **Speed-up (EEV versus ClustalW)** | **302×** | **838×** | **9,337×** | - |

clustered together (Figure (c)). In contrast, ClustalW show a more ambiguous classification of these two families, with more dispersed sequence distribution. In the Virus dataset, compared to the EEV method, the phylogenetic trees generated by ClustalW and MAFFT exhibit a more dispersed distribution of members belonging to the same kingdom, with them scattered across five different positions. This indicates that EEV is more effective than ClustalW and MAFFT in revealing phylogenetic differences among different species.

Meanwhile, classical MSA methods face a significant increase in computational complexity when dealing with large-scale data, leading to substantial increases in computation time and resource consumption. In contrast, EEV efficiently integrates all the information and completes the computation in a shorter time. Figure (d) shows that, on whole-genome data, EEV fully utilizes the sequence information to completely separate Fungi, Archaea, and Virus. Furthermore, compared to Figures (a) and (b), EEV more clearly distinguishes the Anelloviridae and Closteroviridae families within the Virus dataset. The total computation time was only 85.74 seconds.

To assess the topological consistency of the constructed phylogenetic trees, We built a reference tree at the family level based on known taxonomic information according to established classifications. All trees generated by EEV, ClustalW, and MAFFT used the same sequence set and consistent leaf node labels to ensure comparability. We then compared the clustering structure implied by each method's tree with the reference tree by calculating the Adjusted Rand Index (ARI) [44] and Normalized Mutual Information (NMI) scores, where higher values indicate better agreement with the known family labels. The results show that the phylogenetic tree constructed by the EEV method achieves the highest clustering consistency with the known family classifications, significantly outperforming ClustalW and MAFFT, which further demonstrates the effectiveness of EEV in preserving species relationships, as summarized in Table 5.

Table 5. Comparison of structural similarity between phylogenetic trees generated by traditional MSA methods and the proposed EEV method

| Method | ARI | NMI |
|---|---|---|
| ClustalW | 0.2246 | 0.4200 |
| MAFFT | 0.1990 | 0.4069 |
| EEV 5k | 0.4461 | 0.6502 |
| EEV Full | 0.6825 | 0.8238 |

## Energy superposition and dimensionality expansion

The EEV can achieve dimensionality expansion through the superposition of nucleotide energies. Specifically, the features of nucleotides $A, C, G, T$ can be expanded from a single energy state to four energy states. For Equation 1, Comb$(4, k)$ represents the $k$-element subsets generated from the set $\{A, C, G, T\}$. When $k = 2$, it represents the superposition of energies of two nucleotides, and the corresponding nucleotide pairs generated by Comb$(4, 2)$ are {{A, C}, {A, G}, {A, T}, {C, G}, {C, T}, {G, T}}. Taking the energy superposition of nucleotides $A$ and $G$ as an example, the energy entropy calculation formula is:

$$E_{\{A,G\}} = -(w_A + w_G) \cdot (f_A \log f_A + f_G \log f_G). \tag{7}$$

This expansion can more comprehensively capture the complex information features in gene sequences, thereby enhancing the feature expression ability.

### *Dimensional growth: EEV versus k-mer*

Compared to the traditional $k$-mer method, the dimensionality expansion growth rate of EEV is $O(4 \times 2^k)$, while that of the $k$-mer method is $O(4^k)$. The dimensionality growth rate of the $k$-mer method is significantly faster. For example, when $k = 4$, the $k$-mer method generates 256-dimensional features ($4^4 = 256$), while EEV requires only 56 dimensions. The dimensionality of EEV can be calculated using Equation 8 as follows:

$$d_k = 3 \sum_{i=1}^{k} C(4, i) + \sum_{i=2}^{\max(2,k)} C(4, i) \tag{8}$$

Here, $3 \sum_{i=1}^{k} C(4, i)$ represents the dimensionalities of $E_1$, $E_2$, and $E_3$, which are multiplied by 3; $\sum_{i=2}^{\max(2,k)} C(4, i)$ represents the dimensionality of the mutual information energy $E_4$, as $E_4$ involves mutual information between nucleotides and requires at least two nucleotides, starting from $i = 2$ to calculate $C(4, i)$.

Similarly, for protein sequences, when $k = 4$, the $k$-mer method generates 160000-dimensional features ($20^4 = 160000$), whereas

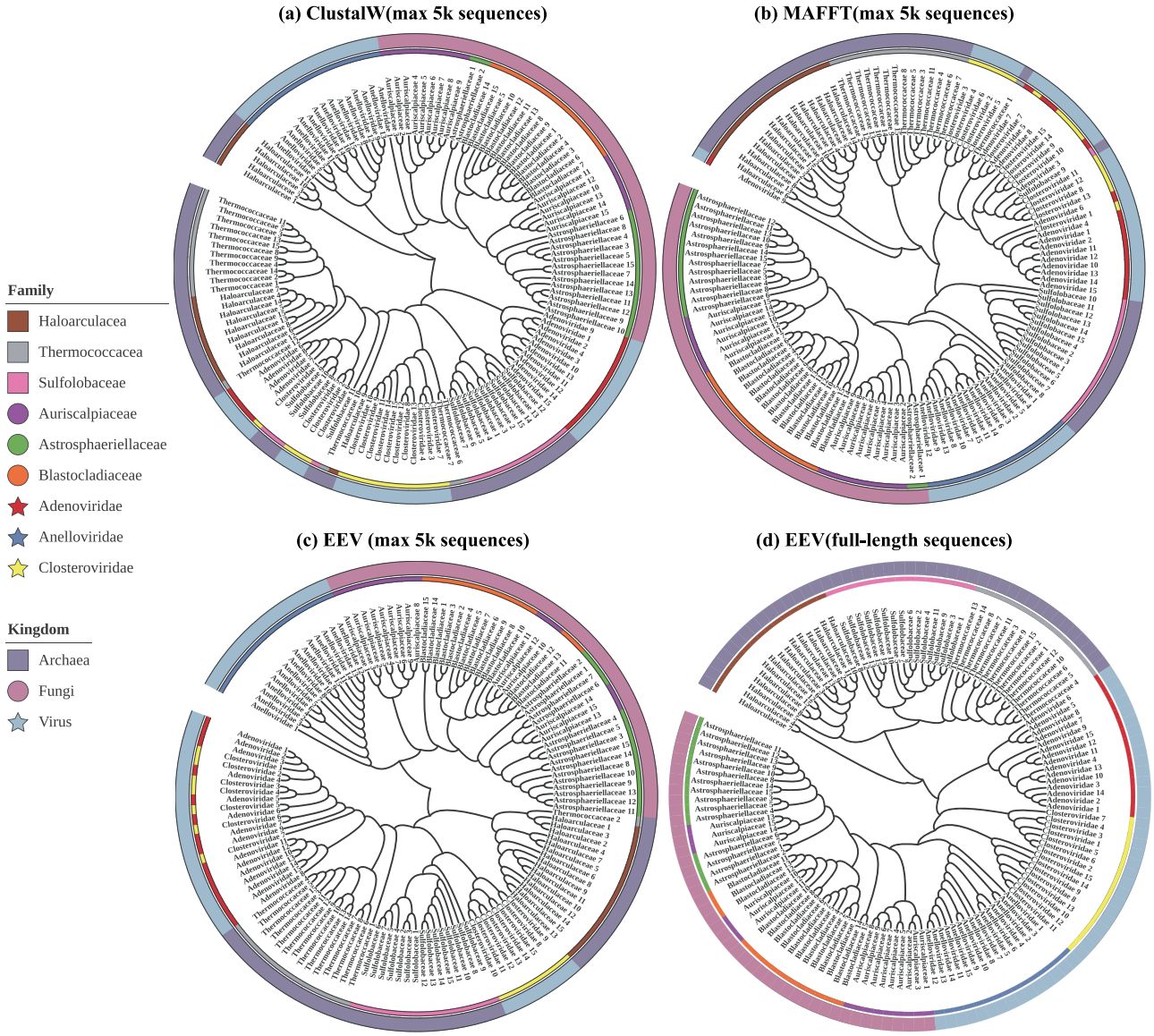


Figure 2. Phylogenetic Tree of Archaea, Fungi, and Virus. The tree is based on 135 sequences from three kingdoms: Archaea, Fungi, and Virus. Outer ring colors represent kingdoms: purple for Archaea, pink for Fungi, and blue for Virus. Inner ring colors distinguish families within each kingdom: Archaea includes Haloarculaceae, Thermococcaceae, and Sulfolobaceae; Fungi includes Astrosphaeriellaceae, Auriscalpiaceae, and Blastocladiaceae; and Virus includes Adenoviridae, Anelloviridae, and Closteroviridae.

the EEV only requires 24760 dimensions, computed as:

$$d_4 = 3\sum_{i=1}^{4} C(20, i) + \sum_{i=2}^{4} C(20, i) = 24760.$$

As $k$ increases, the $k$-mer method faces not only the problem of dimensionality explosion but also vector sparsity. In contrast, the EEV significantly reduces dimensional growth through cumulative energy superposition, effectively avoiding the issue of vector sparsity.

Figure 3 shows the disjoint rate of convex hulls for the 18-dimensional EEV, the 16-dimensional standard $k$-mer method, and the $k$-mer with FFP method [23] across three datasets. The results indicate that the EEV more effectively separates the data.

## *Energy superposition and classification performance*

This section will analyze the impact of energy superposition on feature separation performance. The experiments selected Bacteria, Virus, and Fungi DNA barcode datasets that had not yet reached the convex hull separation limit, to better demonstrate the differences under more challenging classification conditions. By progressively superposing energy states of nucleotides, We observed their effect on the disjoint rate of convex hulls in high-dimensional space.

Figure 4 shows the trends of convex hull separation performance on different datasets during the energy superposition process. The X-axis represents the change in feature dimensionality with the superposition of nucleotide energies. According to Equation 2, the feature dimensionality for a single nucleotide is 18, for two nucleotides is 36, for three nucleotides is 52, and for four nucleotides is 56. For the 24-dimensional case, the first three

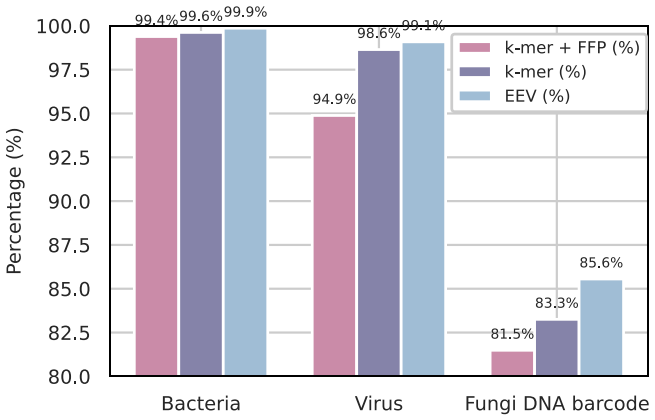


Figure 3. Convex Hull Disjoint Rates. The convex hull disjoint rates of the standard $k$-mer and $k$-mer with FFP ($k = 2$, vector length 16) and the EEV method (single nucleotide energy, vector length 18) across Bacteria, Virus, and Fungi DNA barcode datasets.

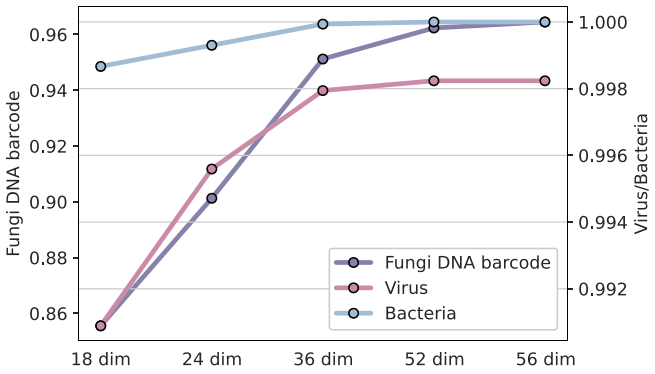


Figure 4. Proportion of Disjoint Convex Hulls with Nucleotide Energy Superposition. Change in the proportion of disjoint convex hulls as nucleotide energies are progressively superposed. The x-axis represents the dimensional changes resulting from nucleotide energy superposition. The y-axis represents the proportion of disjoint convex hulls, where the left side corresponds to the Fungi DNA barcode dataset and the right side corresponds to the Bacteria and Virus datasets.

statistical metrics (E1, E2, E3) only calculate the energy between each pair of nucleotides, $3 \times C(4, 2) = 18$, while E4 calculates the mutual information energy between two nucleotides, which is 6-dimensional, resulting in a total dimensionality of 24. The experimental results indicate that the family-level proportion of disjoint convex hulls gradually increases on different datasets with the superposition of nucleotide energies. This demonstrates that energy superposition has a significant effect on enhancing feature representation.

### *Ablation study on energy terms*

To evaluate whether the four energy terms ($E_1$–$E_4$) provide complementary information, we conducted an ablation experiment by using each term individually and comparing the results with the full EEV. This experiment uses the same settings as described in the Family Classification section, and we selected a random forest classifier for the evaluation. Table 6 shows that the full EEV consistently outperforms any single term alone. Notably, while $E_4$ alone achieves relatively high accuracy, removing the other terms still results in a performance drop, demonstrating that all terms contribute complementary information.

## Discussion

This study proposes a nucleotide sequence feature representation method based on energy entropy. By defining energy entropy from different perspectives, $E_1$ describes the global probability distribution, $E_2$ characterizes global dependencies, $E_3$ reflects spatial distribution properties, and $E_4$ quantifies mutual information between nucleotides. Specifically, $E_4$ is computed based on the logarithm of probabilities, representing a nonlinear operation. Meanwhile, the first three metrics integrate probability distribution, positional information, and dependency through a unified weighting factor, while $E_4$ uses the count of nucleotide pairs as its weighting factor. Ultimately, the information content of nucleotides is transformed into energy representation through a Hadamard product with the weighting factors.

We compared the EEV, NV, and CNV, evaluating their performance in convex hull separation at the family level and species classification tasks. Experimental results show that the EEV outperforms the other two methods in both classification performance and generalization ability. Moreover, EEV can construct high-quality phylogenetic trees in a short time, with efficiency superior to traditional MSA methods (ClustalW and MAFFT). Additionally, the EEV supports flexible dimensional expansion, making it adaptable to complex datasets. For complex tasks, dimensions can be expanded by the superposition of multiple nucleotide energy terms to meet higher task requirements. For simpler datasets, a single nucleotide energy (18-dimensional) is sufficient. The choice of dimensionality should be determined based on the complexity of the dataset and research needs. Compared to the $k$-mer method, the EEV avoids the dimensionality explosion and sparsity issues in high-dimensional spaces caused by increasing $k$, thereby maintaining superior feature representation and classification performance. Meanwhile, energy entropy is a quantity with magnitude, and through its definition, we can perform various types of quantitative analysis in biological analysis. This comprehensive feature representation helps to reveal structural and evolutionary patterns in microbial genomes, providing additional insights for interpreting their genomic diversity and relationships.

Despite the excellent performance of the EEV in multiple experiments, some issues require further investigation. For example, optimizing the design of weighting factors and developing new statistical metrics to further enhance classification performance. Furthermore, the potential applications of the energy entropy vector in protein sequences and other biological data warrant further exploration. Although this study focused on microbial genomes, the EEV framework could also be extended to human genome analysis, which will be investigated in future work.

In summary, the EEV provides a novel and efficient solution for feature representation of gene sequences. Its remarkable performance advantages and flexible scalability make it highly valuable for applications in bioinformatics analysis.

**Key Points**

- EEV encodes gene sequences of arbitrary length into fixed-length vector representations by modeling nucleotide energy characteristics via information entropy.
- EEV effectively captures global sequence features, demonstrating superior accuracy and efficiency in

Table 6. Classification Accuracy: Full EEV versus Individual Components

| Dataset | Full EEV | $E_1$ Only | $E_2$ Only | $E_3$ Only | $E_4$ Only |
|---|---|---|---|---|---|
| Archaea | **0.8247** | 0.6491 | 0.6491 | 0.6491 | 0.7894 |
| Bacteria | **0.9127** | 0.7728 | 0.7728 | 0.7911 | 0.9071 |
| Virus | **0.8517** | 0.7440 | 0.7468 | 0.7379 | 0.7765 |
| Fungi | **0.7436** | 0.4744 | 0.4744 | 0.4872 | 0.6794 |
| Fungi DNA barcode | **0.8112** | 0.6764 | 0.6778 | 0.6699 | 0.7428 |

species classification and phylogenetic tree construction.

- EEV supports flexible dimensional expansion while alleviating high-dimensional sparsity, exhibiting strong scalability and robust representation capability.

## Author contributions

H.W., G.H., and S.S.-T.Y. conceived the experiment. H.W. conducted the experiment and analyzed the results. H.W. wrote the manuscript. G.H. and S.S.-T.Y. reviewed the manuscript.

Conflict of interest: The authors declare no competing interests.

## Funding

This work was supported by the National Natural Science Foundation of China (No. 12171275) and the Tsinghua University Education Foundation.

## Data, materials, and software availability

The code used for implementing experimental methods and evaluating validation algorithms is available in the GitHub repository (https://github.com/karlieswift/EEV). All other relevant data are included in the manuscript.

## References

1. Stephens ZD, Lee SY, Faghri F. *et al.* Big data: astronomical or genomical? *PLoS Biol* 2015;**13**:e1002195. https://doi.org/10.1371/journal.pbio.1002195
2. Goodwin S, McPherson JD, Richard W. *et al.* Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* 2016;**17**:333–51. https://doi.org/10.1038/nrg.2016.49
3. Alharbi WS, Rashid M. A review of deep learning applications in human genomics using next-generation sequencing data. *Hum Genomics* 2022;**16**:26. https://doi.org/10.1186/s40246-022-00396-x
4. Altschul SF, Gish W, Miller W. *et al.* Basic local alignment search tool. *J Mol Biol* 1990;**215**:403–10. https://doi.org/10.1016/S0022-2836(05)80360-2
5. Thompson JD, Higgins DG, Gibson TJ. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* 1994;**22**:4673–80. https://doi.org/10.1093/nar/22.22.4673
6. Katoh K, Misawa K, Kuma K-ı. *et al.* Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Res* 2002;**30**:3059–66. https://doi.org/10.1093/nar/gkf436
7. Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 2004;**32**:1792–7. https://doi.org/10.1093/nar/gkh340
8. Sievers F, Higgins DG. Clustal omega for making accurate alignments of many protein sequences. *Protein Sci* 2018;**27**:135–45. https://doi.org/10.1002/pro.3290
9. Edgar RC. Muscle5: high-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny. *Nat Commun* 2022;**13**:6968.
10. Katoh K, Standley DM. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol* 2013;**30**:772–80. https://doi.org/10.1093/molbev/mst010
11. Alipanahi B, Delong A, Weirauch MT. *et al.* Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nat Biotechnol* 2015;**33**:831–8. https://doi.org/10.1038/nbt.3300
12. Singh R, Lanchantin J, Robins G. *et al.* Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* 2016;**32**:i639–48. https://doi.org/10.1093/bioinformatics/btw427
13. Zou J, Huss M, Abid A. *et al.* A primer on deep learning in genomics. *Nat Genet* 2019;**51**:12–8. https://doi.org/10.1038/s41588-018-0295-5
14. Brandes N, Ofer D, Peleg Y. *et al.* Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics* 2022;**38**:2102–10. https://doi.org/10.1093/bioinformatics/btac020
15. Rio A L-D, Martin M, Perera-Lluna A. *et al.* Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Sci Rep* 2020;**10**:14634. https://doi.org/10.1038/s41598-020-71450-8
16. Haubold B. Alignment-free phylogenetics and population genetics. *Brief Bioinform* 2014;**15**:407–18. https://doi.org/10.1093/bib/bbt083
17. Zielezinski A, Vinga S, Almeida J. *et al.* Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biol* 2017;**18**:1–17.
18. Bohnsack KS, Kaden M, Abel J. *et al.* Alignment-free sequence comparison: a systematic survey from a machine learning perspective. *IEEE/ACM Trans Comput Biol Bioinform* 2022;**20**:119–35.
19. Wan L, Reinert G, Sun F. *et al.* Alignment-free sequence comparison (ii): theoretical power of comparison statistics. *J Comput Biol* 2010;**17**:1467–90. https://doi.org/10.1089/cmb.2010.0056
20. Bussi Y, Kapon R, Reich Z. Large-scale k-mer-based analysis of the informational properties of genomes, comparative

genomics and taxonomy. *PloS One* 2021;**16**:e0258693. https://doi.org/10.1371/journal.pone.0258693
21. Cattaneo G, Petrillo UF, Giancarlo R. *et al.* The power of word-frequency-based alignment-free functions: a comprehensive large-scale experimental analysis. *Bioinformatics* 2022;**38**:925–32. https://doi.org/10.1093/bioinformatics/btab747
22. Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc Natl Acad Sci* 1986;**83**:5155–9. https://doi.org/10.1073/pnas.83.14.5155
23. Sims GE, Jun S-R, Wu GA. *et al.* Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proc Natl Acad Sci* 2009;**106**:2677–82. https://doi.org/10.1073/pnas.0813249106
24. Hozumi Y, Wei G-W. Revealing the shape of genome space via k-mer topology. arXiv preprint arXiv:241220202, 2024.
25. Deng M, Chenglong Y, Liang Q. *et al.* A novel method of characterizing genetic sequences: genome space with biological distance and applications. *PloS One* 2011;**6**:e17293. https://doi.org/10.1371/journal.pone.0017293
26. Wen J, Chan RHF, Yau S-C. *et al.* K-mer natural vector and its application to the phylogenetic analysis of genetic sequences. *Gene* 2014;**546**:25–34. https://doi.org/10.1016/j.gene.2014.05.043
27. Li Y, He L, He RL. *et al.* A novel fast vector method for genetic sequence comparison. *Sci Rep* 2017;**7**:12226. https://doi.org/10.1038/s41598-017-12493-2
28. Zhang YY, Wen J, Yau SS-T. Phylogenetic analysis of protein sequences based on a novel k-mer natural vector method. *Genomics* 2019;**111**:1298–305. https://doi.org/10.1016/j.ygeno.2018.08.010
29. Sun N, Zhao X, Yau SS-T. An efficient numerical representation of genome sequence: natural vector with covariance component. *PeerJ* 2022;**10**:e13544. https://doi.org/10.7717/peerj.13544
30. Guan M, Zhao L, Yau SS-T. Classification of protein sequences by a novel alignment-free method on bacterial and virus families. *Genes* 2022;**13**:1744. https://doi.org/10.3390/genes13101744
31. Vinga S, Almeida J. Alignment-free sequence comparison—a review. *Bioinformatics* 2003;**19**:513–23. https://doi.org/10.1093/bioinformatics/btg005
32. Reinert G, Chew D, Sun F. *et al.* Alignment-free sequence comparison (i): statistics and power. *J Comput Biol* 2009;**16**:1615–34. https://doi.org/10.1089/cmb.2009.0198
33. Shannon CE. A mathematical theory of communication. *Bell Syst Tech J* 1948;**27**:379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
34. Grosse I, Herzel H, Buldyrev SV. *et al.* Species independence of mutual information in coding and noncoding dna. *Phys Rev E* 2000;**61**:5624–9. https://doi.org/10.1103/PhysRevE.61.5624
35. Vinga S. Information theory applications for biological sequence analysis. *Brief Bioinform* 2014;**15**:376–89. https://doi.org/10.1093/bib/bbt068
36. Zhao X, Tian K, Yau SS-T. A new efficient method for analyzing fungi species using correlations between nucleotides. *BMC Evol Biol* 2018;**18**:1–13. https://doi.org/10.1186/s12862-018-1330-y
37. Schoch CL, Seifert KA, Huhndorf S. *et al.* Nuclear ribosomal internal transcribed spacer (its) region as a universal dna barcode marker for fungi. *Proc Natl Acad Sci* 2012;**109**:6241–6. https://doi.org/10.1073/pnas.1117018109
38. Begerow D, Nilsson H, Unterseher M. *et al.* Current state and perspectives of fungal dna barcoding and rapid identification procedures. *Appl Microbiol Biotechnol* 2010;**87**:99–108. https://doi.org/10.1007/s00253-010-2585-4
39. Zhao X, Tian K, He RL. *et al.* Convex hull principle for classification and phylogeny of eukaryotic proteins. *Genomics* 2019;**111**:1777–84. https://doi.org/10.1016/j.ygeno.2018.11.033
40. Zhao R, Pei S, Yau SS-T. New genome sequence detection via natural vector convex hull method. *IEEE/ACM Trans Comput Biol Bioinform* 2020;**19**:1782–93. https://doi.org/10.1109/TCBB.2020.3040706
41. Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32. https://doi.org/10.1023/A:1010933404324
42. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–94, 2016.
43. Sneath PHA. The principles and practice of numerical classification. *Num Taxon* 1973;**573**.
44. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985;**2**:193–218. https://doi.org/10.1007/BF01908075


*Briefings in Bioinformatics*, 2025, **26(5)**, bbaf459
https://doi.org/10.1093/bib/bbaf459
Problem Solving Protocol