International Journal of
Molecular Sciences

MDPI

Review

Overview and Prospects of DNA Sequence Visualization

Yan Wu, Xiaojun Xie, Jihong Zhu, Lixin Guan and Mengshan Li *

School of Mathematics and Computer Science, Gannan Normal University, Ganzhou 341000, China; wuyan@gnnu.edu.cn (Y.W.); xiexiaojun@gnnu.edu.cn (X.X.); zhujihong@gnnu.edu.cn (J.Z.); lxguan@gnnu.edu.cn (L.G.)
* Correspondence: lms@gnnu.edu.cn

Abstract: Due to advances in big data technology, deep learning, and knowledge engineering, biological sequence visualization has been extensively explored. In the post-genome era, biological sequence visualization enables the visual representation of both structured and unstructured biological sequence data. However, a universal visualization method for all types of sequences has not been reported. Biological sequence data are rapidly expanding exponentially and the acquisition, extraction, fusion, and inference of knowledge from biological sequences are critical supporting technologies for visualization research. These areas are important and require in-depth exploration. This paper elaborates on a comprehensive overview of visualization methods for DNA sequences from four different perspectives—two-dimensional, three-dimensional, four-dimensional, and dynamic visualization approaches—and discusses the strengths and limitations of each method in detail. Furthermore, this paper proposes two potential future research directions for biological sequence visualization in response to the challenges of inefficient graphical feature extraction and knowledge association network generation in existing methods. The first direction is the construction of knowledge graphs for biological sequence big data, and the second direction is the cross-modal visualization of biological sequences using machine learning methods. This review is anticipated to provide valuable insights and contributions to computational biology, bioinformatics, genomic computing, genetic breeding, evolutionary analysis, and other related disciplines in the fields of biology, medicine, chemistry, statistics, and computing. It has an important reference value in biological sequence recommendation systems and knowledge question answering systems.

Keywords: DNA sequences; biological sequence; visualization; knowledge graph; machine learning

check for updates

Academic Editor: Timofey S. Rozhdestvensky

Received: 19 November 2024
Revised: 30 December 2024
Accepted: 4 January 2025
Published: 8 January 2025

Citation: Wu, Y.; Xie, X.; Zhu, J.; Guan, L.; Li, M. Overview and Prospects of DNA Sequence Visualization. *Int. J. Mol. Sci.* **2025**, *26*, 477. https://doi.org/10.3390/ijms26020477

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Background

With the advent of the big data era, the field of life sciences has entered the post-genomic era characterized by "omics". The Human Genome Project has transitioned and transformed from structural genomics to functional genomics [1–4]. Research on gene expression and its regulation, genomic diversity, and model organism genomes has become the core issue [5,6]. The development of sequencing technologies has provided support for obtaining massive amounts of sequencing data. However, the continuous increase in data volume has brought forth a new challenge of how to store, manage, and analyze data [7–9]. Data are not equivalent to knowledge. The interpretation and application of data have become crucial tasks for the future. High-throughput sequencing data are stored in textual

format, which presents characteristics such as being unstructured, multi-source, heterogeneous, and difficult to integrate and utilize due to its high specialization [10,11]. The human brain struggles to directly extract knowledge from textual data, which is why scholars have long been committed to extracting knowledge from textual data through visualization. The traditional representation of DNA sequences is through the use of the Letter Sequence Representation (LSR) method [12]. LSR has made significant contributions to the storage and statistical analysis of sequence information. However, it is often challenging to extract knowledge and analyze sequence features in LSR. Especially for long sequences, it can be difficult to intuitively identify, remember, and compare sequences using LSR. LSR lacks readability and exhibits significant limitations in these aspects [13,14]. How to transform massive biological data into knowledge and how to update existing knowledge structures for efficient information retrieval have become new challenges [15,16]. Therefore, the development of sequence visualization and analysis systems that align with the characteristics of the current era has emerged as a new necessity. DNA sequence visualization (DNA SV) has opened up new avenues for biologists to extract information from massive datasets. At the same time, it has opened up a new research area in DNA sequence research [17,18]. DNA SV performs the local or global representation of sequences, extracting characteristic information from the sequences to enable multiple types of sequence analysis, such as sequence homology and similarity analysis [19,20]. As DNA SV brings convenience to various fields of biology, scientists have also extended visualization techniques to RNA and protein sequences. Therefore, research on DNA SV remains an important topic with profound implications for life sciences [21,22].

1.2. Focus and Vision of This Paper

This review primarily consists of two parts. The first part provides a comprehensive overview of DNA sequence visualization methods, focusing on two-dimensional, three-dimensional, high-dimensional, and dynamic visualization methods. Furthermore, the advantages and limitations of using visualization methods for sequence similarity measurement, gene mutation analysis, and other related analyses are discussed. The second part introduces two potential research directions: the construction of knowledge graphs for DNA sequence big data and the generation of DNA sequence graphics based on machine learning. This paper provides readers of DNA SV with a clear research path. The framework of this review is shown in Figure 1.

Traditional DNA sequence visualization methods are no longer efficient in extracting knowledge networks. There is an urgent need to find a method that can effectively explore the interconnected networks within massive data. A knowledge graph (KG) is an important branch of artificial intelligence that is used to represent concepts and their relationships in the physical world in a symbolic form [23–25]. Therefore, constructing a better KG system that organizes, manages, and understands the relationships between DNA sequences is of great importance for deep mining of massive data. Meanwhile, another challenge in DNA SV is the extraction of features from unstructured data [26]. Existing visualization methods for letter sequences suffer from subjective factors in feature extraction, as different methods may choose different features for visualization representation [27]. As a result, the generated graphics have different emphases and may not provide sufficient semantic information. Therefore, utilizing machine learning to construct visual representations of DNA sequences is another important research direction for the future.

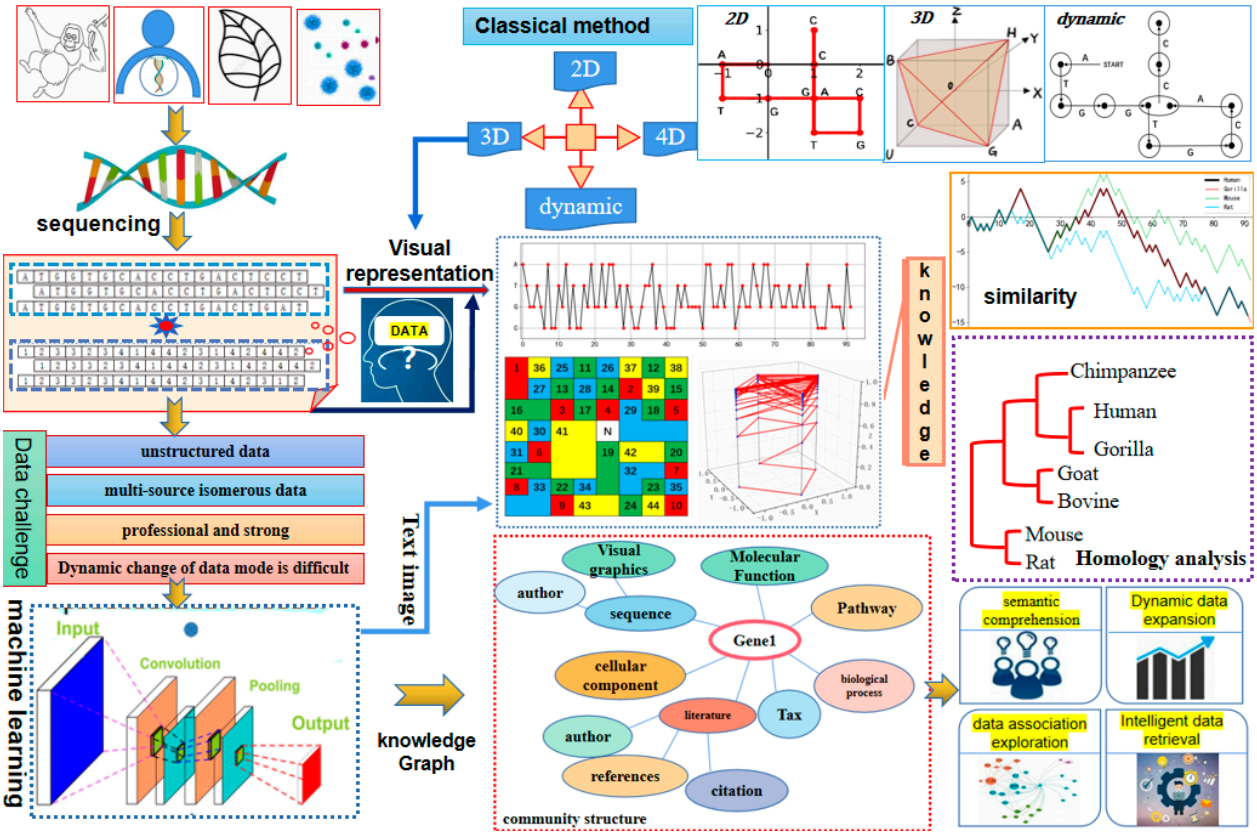

Figure 1. The overall framework of this review. This framework consists of two main parts: (1) Data visualization and knowledge extraction: This part involves 2D, 3D, 4D, and dynamic visualization methods to represent sequence characteristics. (2) Machine learning and knowledge integration: This part focuses on processing multi-source heterogeneous and unstructured data. Machine learning methods are used to extract features from these datasets. The extracted features are visualized to generate knowledge, which can further be organized into a knowledge graph.

2. Two-Dimensional (2D) Visualization

2.1. Random Walk Visualization

A 2D visualization method for DNA sequences was proposed by Gates [28]. The basic principle is to assign the four basic nucleotides (adenine (A), thymine (T), guanine (G), and cytosine (C)) to a Cartesian coordinate system, constructing a symmetric purine–pyrimidine plot, as shown in Figure 2a. The six points in the figure correspond to the ten bases. The curves exhibit significant overlap, leading to pronounced information loss. For instance, points corresponding to repetitive bases tend to aggregate, making it challenging to distinguish sequence variations visually. To address these limitations, Nandy proposed a new 2D visualization method for DNA sequences, called the Gates–Nandy model [29,30]. In this model, Nandy introduced a different allocation scheme for the four nucleotides, as shown in Figure 2b. The curve structure in the figure is clear, and there is a noticeable improvement in reducing information loss. However, the appearance of closed loops introduces the issue of the number of cycles and the direction of traversal, which remains unknown [31]. To address the closed loop issue, Leong and Morgenthaler refined the nucleotide coordinate assignments (Figure 2c), reducing closed loops and improving curve interpretability. However, its complexity posed challenges for path traversal, and subsequent researchers have primarily focused on studying and improving the Gates and Gates–Nandy models. Figure 2d,e illustrate the practical performance of the Gates and Gates–Nandy models, respectively. Figure 2d demonstrates the clustering and repetitive

paths present in the Gates model, particularly for long sequences, while Figure 2e highlights the improved clarity and reduced overlap achieved by the Gates–Nandy model. Figure 2e shows fewer closed loops and repetitive paths compared to Figure 2d. The Gates–Nandy model has an overall good visualization effect, but it still does not solve the problems of graph degeneracy and information loss in the visualization.

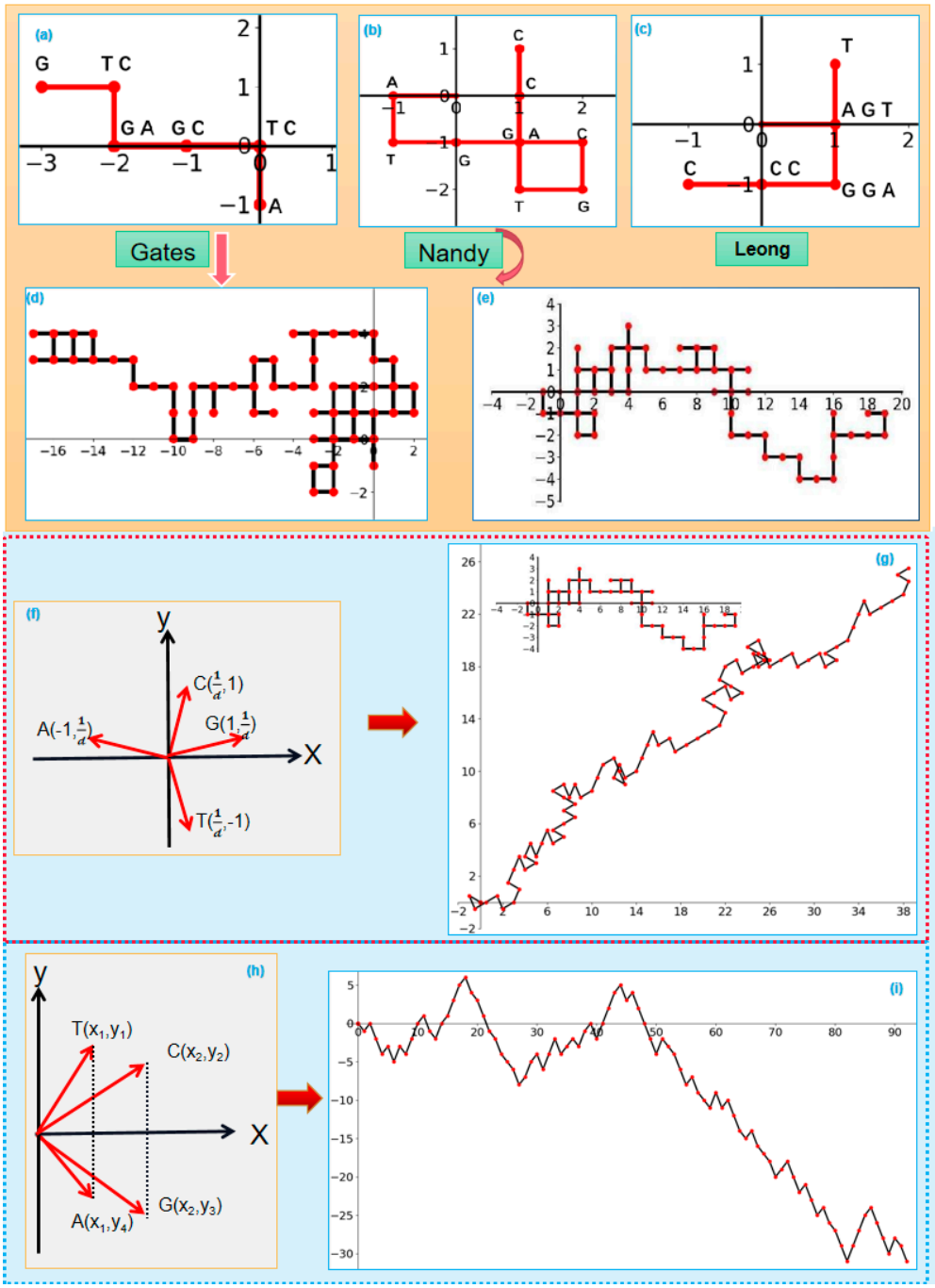

Figure 2. Walking-based visualization method. (**a**) The Gates graph of sequence fragment $\mathcal{S}$. (**b**) The Gates–Nandy graph of sequence segment $\mathcal{S}$. (**c**) The Leong and Morgenthaler graph of sequence segment $\mathcal{S}$. (**d**) The Gates graph of sequence δ_1. (**e**) The Gates–Nandy graph of sequence δ_1. (**f**) The Guo theory model. (**g**) The Guo curve of sequence δ_1. (**h**) The H-L curve theory model. (**i**) The H-L curve of sequence δ_1.

To address the information loss, many scholars have proposed improvement methods based on the Gates–Nandy model. Guo et al. used four special vectors in the Cartesian

coordinate system to represent the four basic nucleotides, with each vector forming a fixed angle with the coordinate axes, as shown in Figure 2f. Figure 2g illustrates the visualization results of the improved algorithm for sequence visualization. The sequence curves constructed using the improved model exhibit an increasing pattern, with a reduced probability of self-crossing and overlapping, leading to a decrease in graph degeneracy. Especially, when compared to the Gates–Nandy graphical representation, the performance is significantly enhanced. However, Figure 2g is not strictly monotonically increasing, and the issue of degeneracy is not completely eliminated. Many improved methods based on the Gates–Nandy model have been proposed, such as the Leong and Morgenthaler models [32], as shown in Figure 2c. While they have improved the issue of degeneracy to some extent, they may also introduce new problems. The classical H-L curve representation allocates the four nucleotides into the first and fourth quadrants based on purines and pyrimidines [33], as shown in Figure 2h. The sequence curves constructed by this model completely solve the issues of overlap and crossing. The curves do not exhibit self-crossing or closed loops, eliminating the problems of degeneracy and information loss. At the same time, the number of projected points on the x-axis can reflect the length of the DNA sequence and also capture some detailed information within the sequence. However, in the case of long sequences, the sequence curves constructed by this model require a significant amount of space, which can still lead to visual clutter and loss of visual effectiveness, thus not fully achieving the desired goals.

2.2. Chaos Game Representation (CGR)

The CGR visualization method abandons the traditional Cartesian coordinate system and instead adopts the principles of chaos theory. It uses scattered points within a square to represent the bases of a sequence [34]. This method has the advantage of being highly compact and occupying less space. Figure 3 illustrates the construction process of the visualization of CGR [35]. First, the four basic nucleotides are assigned to the four vertices of the square. Starting at the origin (0,0), the midpoint between the current point and the vertex corresponding to the current nucleotide is calculated and marked. This process is repeated iteratively for each nucleotide in the sequence, generating a fractal-like pattern. For example, in the sequence ”ATGG”, the first point p1 is the midpoint between (0,0) and the vertex for “A”. The second point p2 is the midpoint between p1 and the vertex for “T”. Similarly, p3 and p4 are calculated as the midpoints between the previous points (p2 and p3, respectively) and the vertex for “G”, as shown in Figure 3a. This method effectively visualizes nucleotide composition and sequence patterns [36]. The reverse construction process starts from the point corresponding to the last nucleotide in the sequence (denoted as p10). It involves moving in the reverse direction along the line connecting vertex A and point p10. The aim is to find the previous point, p9, that is equidistant from vertex A and p10. The nucleotide corresponding to p9 represents the second-to-last nucleotide in the sequence. By continuing this process, we can find the previous point one by one until all the points are identified, completing the reverse construction of the sequence. If the last point is unknown, you can choose any point within one of the quadrants as the starting point. The reverse construction process can be terminated when the point corresponding to the first base of the sequence is reached by reconstructing the local contour based on the starting point.

Figure 3b,c illustrate the visualization of CGR for different sequences. From Figure 3b, it can be observed that the points within the square appear to be random and without a specific pattern. If we divide the square into four quadrants following the principles of the Cartesian coordinate system, we can observe that the point corresponding to base A falls in the first quadrant, the point corresponding to base C falls in the second quadrant,

the point corresponding to base G falls in the third quadrant, and the point corresponding to base T falls in the fourth quadrant. According to the density analysis of points in each quadrant, the content of the four basic nucleotides in the sequence can be determined. In Figure 3b, the density of points in the third quadrant is higher than in the other three quadrants, indicating a rich content of G nucleotides in the sequence. From Figure 3c, it can be inferred that the sequence has a higher content of C and T nucleotides, while the content of G and A nucleotides is significantly lower. Comparing Figure 3b,c, it can be observed that the two sequences have significant differences and a low similarity.

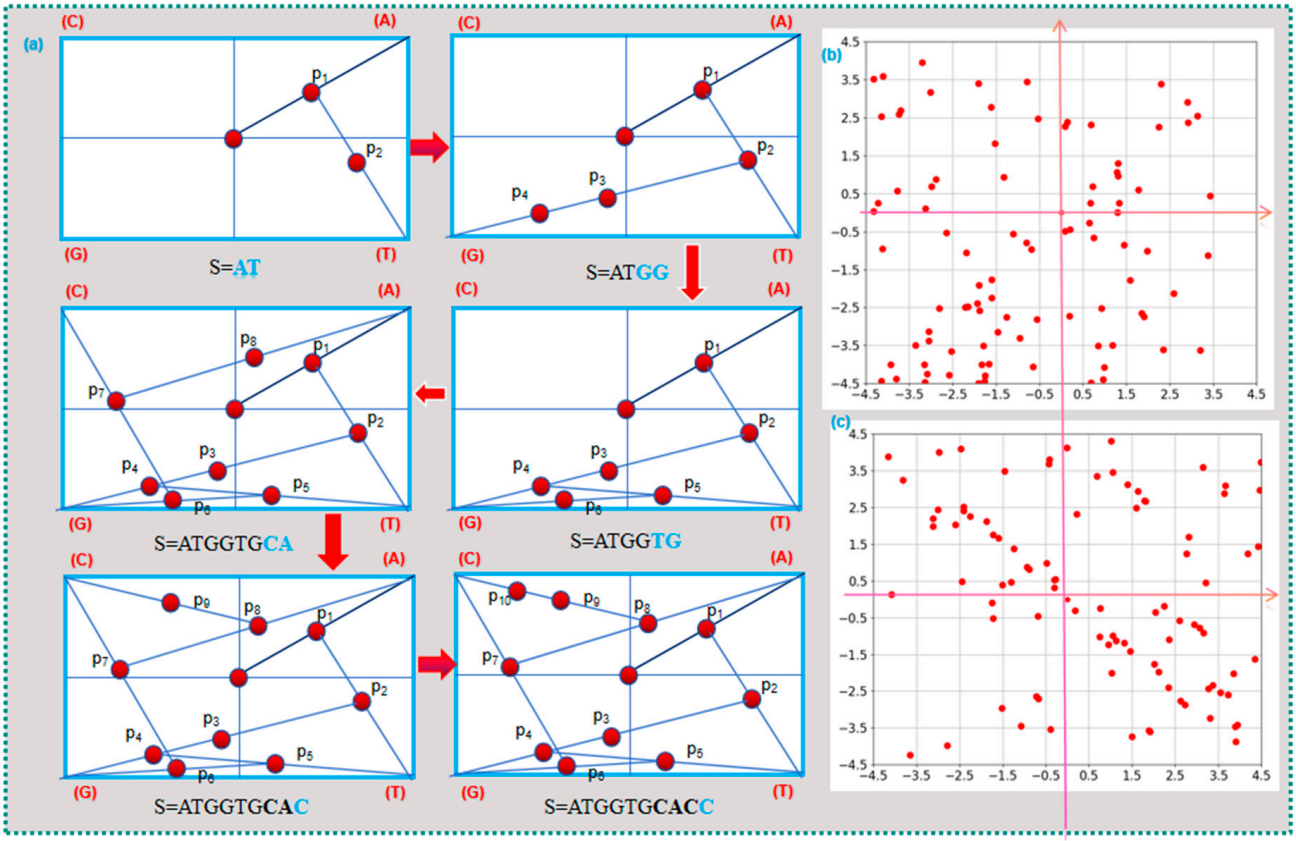

Figure 3. Chaos game representation visualization method. (**a**) The steps to construct a chaos game representation graph. (**b**) The chaos game representation graph of the sequence δ_1. (**c**) The chaos game representation graph of the custom sequence φ.

2.3. Double Vector and Double Nucleotide

Double Vector (DV-Curve) is a visualization method that uses two vectors to represent the basic nucleotides [37]. The method of constructing sequence graphs using the DV-Curve model is shown in Figure 4a. Figure 4a shows that the bases T and C do not move along the Y-axis in the coordinate system, while the bases A and G have displacements of two units up and two units down along the Y-axis, respectively. Therefore, the relationship between the quantities of G and A in a DNA sequence can be determined based on the positive, negative, or zero value of the curve at the Y-axis endpoint. To illustrate the advantages of using the DV-Curve for visualizing long sequences, we will compare the DV-Curve (as shown in Figure 4b) and the spectrum curve (as shown in Figure 4d) for a given sequence. Spectral visualization is one of the most popular two-dimensional methods, mainly consisting of the four-horizontal-line method and the two-horizontal-line method. By comparing the two figures, it can be observed that for sequences longer than 300 bp, the DV-Curve method maintains a good visual representation without any loss of information [38]. In contrast, the spectrum representation tends to lose its visual advantage as the sequence

length increases. The DV-Curve and the spectrum curve for the sequence δ_7 are shown in Figure 4c,e, respectively. It is observed that directly determining similarity through the comparison of the spectrum curve (Figure 4d,e) is quite challenging. However, from the DV-Curve (Figure 4b,c), it is apparent that there are significant differences between the DNA sequences. This indicates that, for long sequences, the DV-Curve has better visualization capabilities compared to the spectrum representation method.

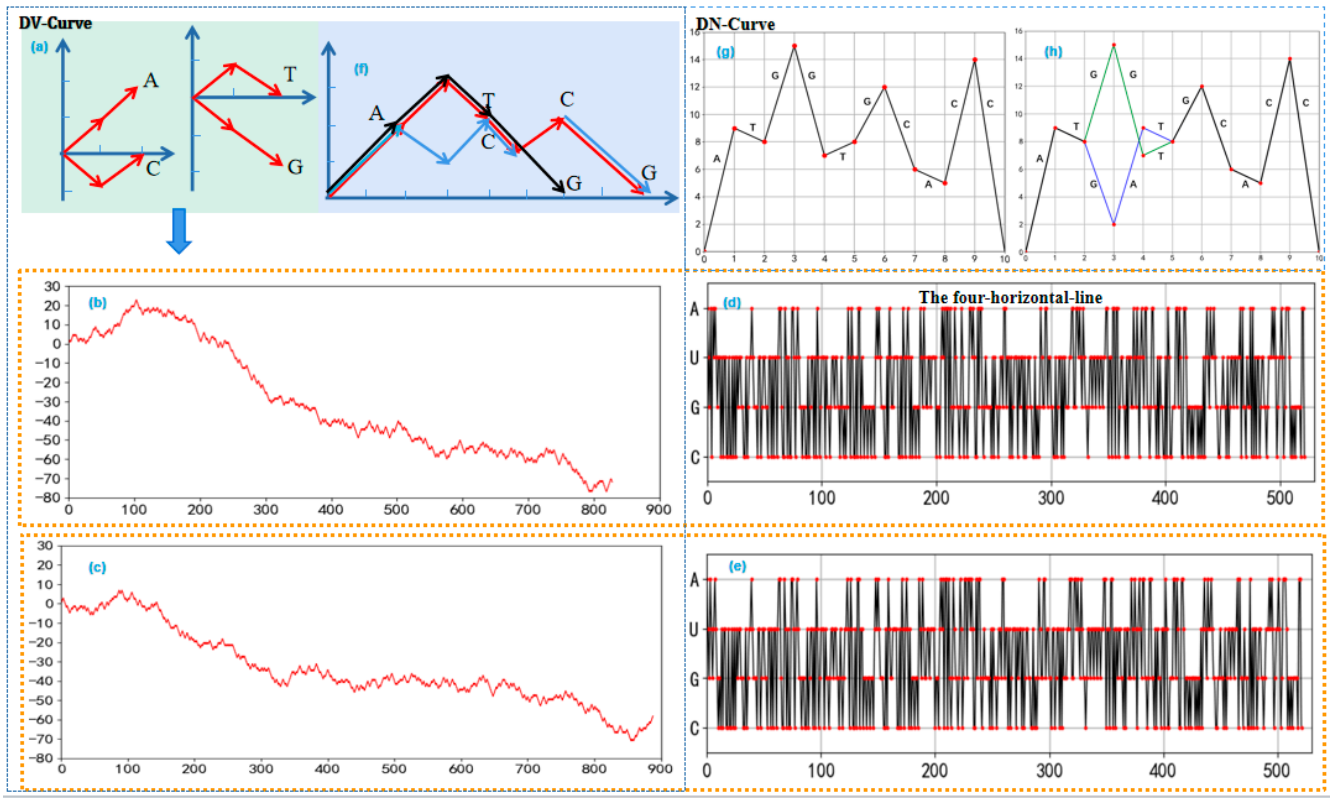

Figure 4. DV-Curve and DN-curve visualization methods. (**a**) Allocation of four nucleotides (A, T, C, and G) of DV-Curve in Cartesian coordinate system. (**b**) DV-Curve of sequence δ_2. (**c**) DV-Curve of sequence δ_7. (**d**) Four horizontal lines of sequence δ_2. (**e**) Four horizontal lines of sequence δ_7. (**f**) Genetic mutation DV-Curve analysis chart. (**g**) DN-Curve of sequence fragment $\mathcal{S}$. (**h**) Comparison of DN-Curves for sequence segment $\mathcal{S}$ and sequence segment $\mathcal{L}'$.

Indeed, the DV-Curve method has significant applications in gene mutation analysis. It assesses the presence of gene mutations by plotting the original sequence and the potential gene mutation sequence on the same Cartesian coordinate system and observing whether the DV-Curves of the two sequences overlap, as shown in Figure 4f. By observing the changes in the length of the DV-Curve, it is possible to infer the type and location of gene mutations. If the length of the DV-Curve increases, it indicates an insertion mutation where additional bases have been inserted into the sequence. If the length decreases, it suggests a deletion mutation where some bases have been removed from the sequence. If the length remains unchanged but the positions do not overlap, it signifies a substitution mutation where one or more bases have been replaced with different bases.

Double Nucleotide (DN-curve) visualization is a method of visualizing sequences using all 16 possible combinations of base pairs [39]. Through this visualization method, we are able to observe the relative positions and spatial relationships between base combinations, which provides unique advantages for understanding the structure and patterns of the sequence as well as identifying important features within the sequence [40,41]. The DN-curve holds clinical significance for the study, diagnosis, and treatment of genetic diseases.

For instance, in mutation detection, the DN-curve identifies mutations by comparing the differences between an individual's genomic sequence and a reference genome sequence. On the curve, each point represents the position of a nucleotide, and the height of the curve denotes the frequency of a specific base observed at that location. The mathematical model of the sequence is given as follows:

$$\varphi(s_i s_{i+1}) = \begin{cases} (i,1) \text{ if } s_i s_{i+1} = AG \\ (i,2) \text{ if } s_i s_{i+1} = GA \\ (i,3) \text{ if } s_i s_{i+1} = CT \\ (i,4) \text{ if } s_i s_{i+1} = TC \\ (i,5) \text{ if } s_i s_{i+1} = AC \\ (i,6) \text{ if } s_i s_{i+1} = CA \\ (i,7) \text{ if } s_i s_{i+1} = GT \\ (i,8) \text{ if } s_i s_{i+1} = TG \\ (i,9) \text{ if } s_i s_{i+1} = AT \\ (i,10) \text{ if } s_i s_{i+1} = TA \\ (i,11) \text{ if } s_i s_{i+1} = CG \\ (i,12) \text{ if } s_i s_{i+1} = GC \\ (i,13) \text{ if } s_i s_{i+1} = AA \\ (i,14) \text{ if } s_i s_{i+1} = CC \\ (i,15) \text{ if } s_i s_{i+1} = GG \\ (i,16) \text{ if } s_i s_{i+1} = TT \end{cases} \quad (1)$$

where i represents the position of the base in the sequence; $i = 1, 2, 3 \cdots \mathcal{N} - 1$. In a Cartesian coordinate system, the coordinates of the base pair combinations are marked on the Cartesian coordinate system, and adjacent points are connected to form a DN-Curve, as shown in Figure 4g. The DN-Curve can highlight the features of base pair combinations, which is highly advantageous for studying the distribution of mutations in base pair combinations. Figure 4h depicts the DN-Curve of two sequences in the same coordinate system. The coordinate of the base pair combination s3s4 in the sequence $\mathcal{S}$ is (3,15), while the coordinate of the base pair combination s3s4 in the sequence $\mathcal{S}'$ is (3,2). The vertical coordinate difference is s4 = 15 − 2 = 13. Therefore, it can be inferred that there is a mutation in the fourth base of the sequence.

2.4. Spectral Visualization

Spectral visualization is one of the most popular two-dimensional methods, mainly consisting of the four-horizontal-line method and the two-horizontal-line method. The four-horizontal-line method does not require pre-assigning four basic nucleotide vectors in the Cartesian coordinate system [42]. First, four horizontal lines are drawn at equal distances, and then these four lines are labeled according to the order in which the four nucleotides appear in the sequence, as shown in Figure 5a. The advantage of this method is that it can visually demonstrate the distribution of bases in the sequence, avoiding the degeneracy and degeneration issues of traditional visualization methods in the Cartesian coordinate system. Additionally, this method can also reflect the local features and similarities of the sequence. Figure 5b illustrates a variant of the four-horizontal-line method, known as the chaotic four-horizontal-line method [43,44]. This method integrates chaos theory to calculate the position of each nucleotide and map it onto a coordinate system. The position of each nucleotide is determined by the coordinate of the preceding nucleotide and its corresponding chaotic factor. Compared to the traditional four-horizontal-line method, the chaotic four-horizontal-line method effectively avoids overlapping points on the vertical axis, providing a clearer representation of local sequence features. This

visualization approach intuitively captures local patterns in sequences, such as nucleotide clustering or repetitive structures, supporting biological function analysis and sequence similarity studies.

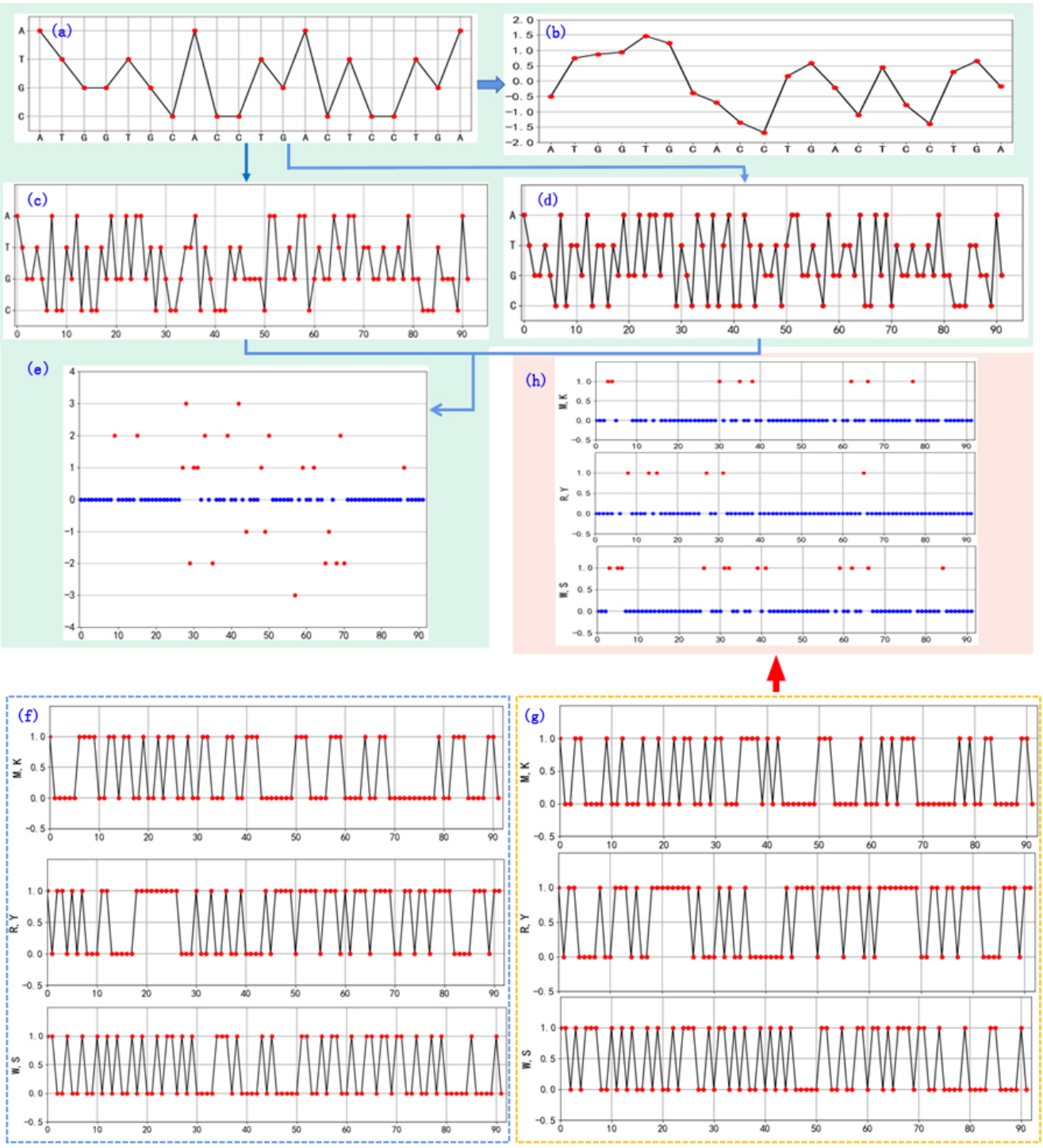

Figure 5. Spectral visualization method. (**a**) Four horizontal lines of sequence fragment γ. (**b**) Chaotic four horizontal lines of sequence fragment γ. (**c**) Four horizontal lines of sequence δ_1. (**d**) Four horizontal lines of sequence δ_7. (**e**) Differences between δ_1 and δ_7. (**f**) Two horizontal lines of sequence δ_1. (**g**) Two horizontal lines of sequence δ_3. (**h**) Differences between δ_1 and δ_3.

The four-horizontal-line visualization method can be employed to analyze the similarity of sequences. Figure 5c,d illustrate the construction of four-horizontal-line repre-

sentations for two DNA sequences. By comparing the two curves, it can be observed that there are significant nucleotide differences between the intervals 28–71. Figure 5e depicts the difference plot between two sequences, where the points on the X-axis ($y = 0$) in the figure represent matching bases between the two sequences, and the skipped points ($y \neq 0$) represent non-matching bases between the two sequences.

The two-horizontal-line method is a visual representation method based on the chemical properties of nucleotides, where three sets of two horizontal lines are used to construct graphical curves as a visual representation [45,46]. Nucleotides are classified into three categories (purine R = [A,G]/pyrimidine Y = [C,T], amino M = [A,C]/keto K = [G,T], and weak-H bond W = [A,T] /strong-H band S = [C,G]). The two-horizontal-line method allows for the observation and analysis of the chemical and structural features of a sequence through a simple and intuitive graphical representation. Figure 5f,g depict the graphical representations of different sequences using the two-horizontal-line method. Figure 5h illustrates the difference plot of two sequences. In Figure 5h, the W/S curve shows the difference in the number of bases with strong and weak hydrogen bonds between the sequences. The M/K curve displays the difference in the positions of amino and keto groups. Additionally, the R/Y curve shows the difference in the positions of purine and pyrimidine bases between the sequences. Therefore, the three sets of feature curves can serve as a coarse-grained description of the DNA primary sequence and can be used for sequence analysis and reconstruction. The visualization effect of this method is intuitive, as it can clearly display the distribution of different types of bases in the sequence and provide a deeper understanding of the structure and chemical properties of DNA sequences. Additionally, this method converts sequences into numerical features, which requires relatively less computational resources. This allows computers to process and compare the sequences more quickly [47].

2.5. Worm Curves and Five-Color Map

The worm curve visualization method (WormBin) represents a sequence of length N in a $\sqrt{n} * \sqrt{n}$ space [48]. The resulting graphical representation is relatively compact, allowing for the display of longer DNA sequences within a smaller space. It provides an intuitive visualization of the positions and relative relationships of different bases [49]. First, the bases A, G, C, and T are encoded as binary labels 00, 01, 10, and 11, respectively, and mapped onto consecutive vertices of a predefined worm curve. This 2D path follows a consistent rule: turning 90° right if unvisited, or 270° left otherwise, ensuring no self-intersections. The path is identical for all sequences and arranged compactly on a 2D plane without biological significance for vertex positions. Only '1' bits are marked as red dots, with multiple '1' bits represented independently. This results of WormBin representation are shown in Figure 6a. Each base in the WormBin has two adjacent vertices, and the curve does not intersect itself, avoiding degeneracy and ambiguity. Figure 6b depicts the worm curves of two sequences in the same Cartesian coordinate system. Black dots represent identical bases in both sequences, while blue and red dots represent different bases in the two sequences. By counting the number of different points, the similarity between two sequences can be directly calculated. The WormBin method can visually demonstrate the similarity and differences between sequences without requiring any special preprocessing of the sequences.

The WormBin method has a notable limitation: it is not able to quickly identify which base corresponds to a specific point on the curve, making it difficult to reconstruct the DNA sequence or perform local analysis. To address this issue, the WormStep visualization method for DNA sequences has been proposed [50]. The construction method of WormStep is overall similar to the rules of the WormBin curve, as shown in Figure 6c. In this process,

the four basic nucleotides are labeled on the curve based on the number of steps taken. If it is necessary to determine the corresponding nucleotide represented by a point in the graph, it can be identified by the number of steps between it and the previous point. WormStep overcomes the drawback of WormBin in not being able to quickly identify the bases, making sequence reconstruction and local sequence analysis easier.

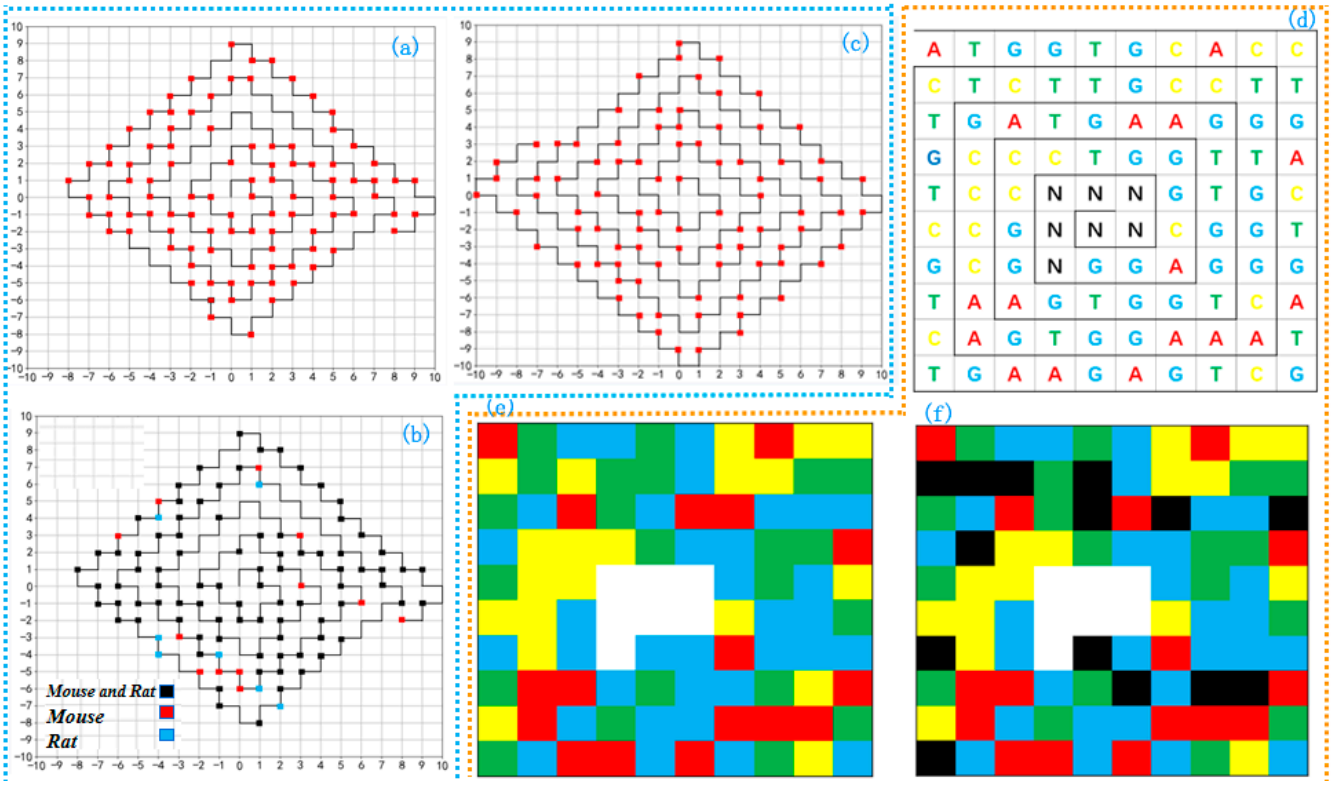

Figure 6. Worm curves and five-color map visualization method. (**a**) WormBin curve of sequence δ_4. (**b**) Differences between δ_4 and δ_5. (**c**) WormStep curve of sequence δ_6. (**d**) Square curve of sequence δ_5. (**e**) Five-color map of sequence δ_5. (**f**) Differences between δ_4 and δ_5.

The WormBin and WormStep visualization methods are effective for short DNA sequences, but they are not suitable for extremely long sequences. To overcome this limitation, the four-color map and five-color map visualization methods have been proposed (five-color map is an improvement of the four-color map). In the case of the five-color map, as shown in Figure 6d, a regular square curve is generated, and the starting position of the bases begins from the top left corner of the square curve, and the bases are allocated to small squares in a clockwise direction. The four nucleotides and the central blank space are marked with different colors. This method provides a better visualization effect. To better observe local features, adjacent small squares with the same base are merged, and then the merged regions are colored according to the color rules (blue for G, yellow for C, green for T, red for A, and white for N), as shown in Figure 6e. The colored graph clearly reveals the distribution and quantity of each base, with a higher abundance of G bases. At the same time, each small square corresponds to a base in the sequence, and there is no degeneracy issue. Additionally, the color-coded representation allows people to easily identify the differences between two images, making it a valuable tool for genetic mutation analysis and similarity analysis. Using this method to analyze the similarity of two sequences, as shown in Figure 6f, it is easy to identify the differing bases. The differing bases are marked in black. The similarity analysis results of the two sequences are consistent with other methods.

3. Three-Dimensional (3D) Visualization

3.1. Z-Curve

The Z-curve is a classical method that converts DNA sequences into a visually intuitive geometric shape, forming a unique folded three-dimensional curve representation [51]. The Z-curve method overcomes some limitations of traditional algebraic methods and exhibits strong uniqueness and reversibility. It is advantageous for the visualization and analysis of sequences, allowing for a visual observation of the structure and features of DNA sequences, such as base pairing, sequence folding, and circular structures, among others [52,53]. The construction process of the Z-curve is depicted in Figure 7.

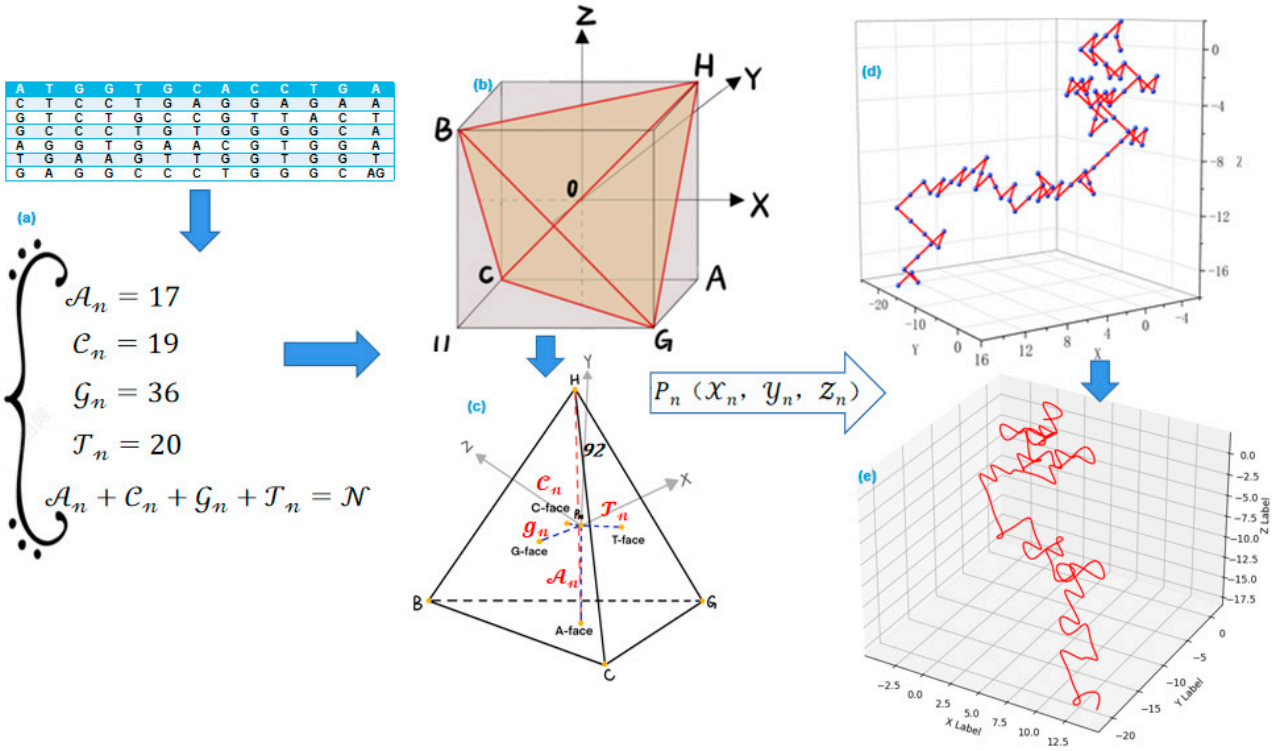

Figure 7. Z-curve construction framework for sequence δ_1. (**a**) The nucleotide counts (A, C, G, T) are computed cumulatively along the sequence. (**b**) The tetrahedral framework maps nucleotide composition into a three-dimensional coordinate system. (**c**) The 2D projection of the tetrahedral framework demonstrates the contributions of each nucleotide to the overall sequence structure. (**d**) The initial Z-curve is constructed as a three-dimensional trajectory based on cumulative nucleotide counts. (**e**) The smoothed Z-curve is obtained by applying B-spline functions to the trajectory of the initial Z-curve.

Let A_n, T_n, G_n, and C_n represent the cumulative counts of bases A, C, G, and T, respectively, up to position n in a DNA sequence of length $\mathcal{N}$. In a three-dimensional coordinate system, we can find a point P_n that uniquely represents a base in a DNA sequence. To facilitate the calculation of the coordinates of points P_n in space, we establish a regular tetrahedron BCGH inside the cube with a height equal to L, as shown in Figure 7b. Each face of the tetrahedron corresponds to a nucleotide (A, T, G, C), and the sum of distances from any point to the four faces equals the tetrahedron's height. The 2D projection of the tetrahedral framework demonstrates the contributions of each nucleotide to the overall sequence structure, as shown in Figure 7c. The coordinates of point P_n within the

regular tetrahedron can be represented as $(\mathcal{X}_n, \mathcal{Y}_n, \mathcal{Z}_n)$. The coordinate calculation formula for point P_n within the regular tetrahedron is as follows [54]:

$$\begin{cases} \mathcal{X}\mathrm{n} = (A_n + G_n) - (C_n + T_n) \\ \mathcal{Y}\mathrm{n} = (A_n + C_n) - (G_n + T_n), \ \mathcal{N} = 0, 1, 2, 3, \cdots, \mathcal{N} \\ \mathcal{Z}\mathrm{n} = (A_n + T_n) - (C_n + G_n) \end{cases} \quad (2)$$

In Formula (2), the coordinates of point P_n are calculated based on the cumulative counts of nucleotides up to position n in the sequence. These cumulative counts reflect the nucleotide composition from the sequence start to position n. Next, mark these coordinates in 3D space and connect them with lines between adjacent points. This will result in the Z-curve representation of the DNA sequence. The directly constructed Z-curve may not be smooth. To achieve a smooth representation, the B-spline interpolation method can be applied. For example, Figure 7d shows the initial Z-curve of sequence δ_1, constructed based on cumulative nucleotide counts. Figure 7e presents the smoothed version of this curve after applying B-spline interpolation, improving the clarity of local and global structural features.

The Z-curve reveals rich spatial folding structures that reflect nucleotide distribution characteristics, including symmetry, periodicity, and local/global features. These features are particularly useful for analyzing repetitive DNA sequences and comparing sequence patterns. By examining the folding and periodicity of Z-curves from different sequences, researchers can identify similarities, differences, and structural variations [55].

3.2. L-Curve

The L-curve method has a low computational cost and does not require three-dimensional projection. It can effectively display the structure and features of DNA sequences in a clear manner. The construction process of the L-curve is simple [56,57]. Firstly, it involves mapping the four basic nucleotides onto a two-dimensional coordinate system to determine their plane coordinates, as shown in Figure 8a. The arrow labeled "multiple distribution modes" indicates that the four nucleotides (A, C, G, and T) can be mapped to different positions in the coordinate system using various allocation schemes. Among these, three commonly used allocation schemes are demonstrated in Figure 8b, each representing a distinct characteristic representation of the nucleotide distribution. Then, iterate through the DNA sequence, and, for each nucleotide, calculate its cumulative occurrence count, which serves as the coordinate value along the Z-axis. Finally, by connecting the adjacent points, the three-dimensional curve graph representing the entire sequence is obtained based on different nucleotide mapping schemes, as shown in Figure 8c,d. The graph clearly shows the positions and quantities of each nucleotide, with a simple structure, and the reconstruction of the DNA sequence is relatively straightforward.

A potential issue with the L-curve method is that there are multiple ways to allocate the planar coordinates for the four basic nucleotides, while different allocation schemes may have an impact on the resulting L-curve [58]. Researchers have proposed an improved L-curve algorithm based on the classification of nucleotides according to their chemical properties. This enhancement aims to provide a more accurate representation of the structure and features of DNA sequences in the L-curve. The improved L-curve algorithm utilizes three different vector representation modes to represent the four basic nucleotides, as shown in Figure 8b. Each DNA sequence corresponds to three L-curve modes, thereby resolving the issue of non-uniqueness in the allocation of plane coordinates in the L-curve method. Figure 8d depicts the L-curve curve with the three modes included. By calculating the geometric center of the L-curve curves for the three modes, a 9D vector can be constructed for sequence similarity measurement [30,59]. If two vectors in the 9D

space point in similar directions, then the two DNA sequences are considered similar. This sequence comparison method is simple to calculate and significantly reduces computational complexity.

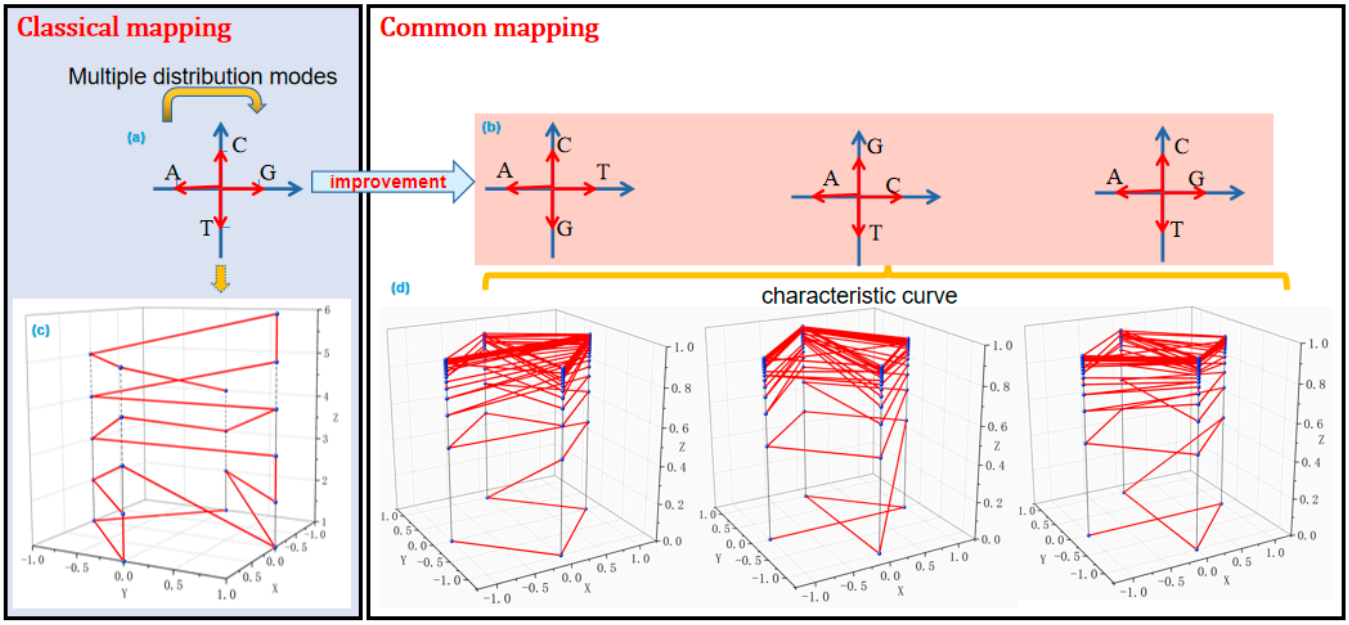

Figure 8. L-curve visualization method. (**a**) Allocation of four basic nucleotides in two-dimensional coordinate system. (**b**) Four nucleotide vector assignment patterns for improved L-curve method. (**c**) Normal L-curve of sequence fragment γ based on initial nucleotide allocation modes. (**d**) Improved L-curve of sequence δ_6 based on optimized nucleotide allocation modes.

3.3. DN-Curve 3D

The Double Nucleotide Curve (DN-Curve 3D) visualization method combines consecutive pairs of nucleotides in the DNA sequence as a single unit and maps it to a point in three-dimensional space [60,61]. By connecting all the points, a curve is formed. The DN-Curve 3D method has 16 possible combinations for DNA sequences consisting of four nucleotides (A, C, G, T) and 10 possible combinations for DNA sequences. The 16 possible combinations of base pairs in the DN-Curve 3D method can be categorized using two different classification methods. (1) Chemical property-based classification (16 combinations): This method categorizes the 16 possible combinations of base pairs into four classes based on the chemical properties of the bases, as shown in Figure 9a. A 4×4 matrix is then constructed to represent these categories. Next, the three-dimensional curve is constructed by determining the planar coordinates for each base pair combination within the two-dimensional coordinate system [62]. (2) Molecular weight-based classification (16 combinations): This method categorizes the 16 base pair combinations into three groups based on the molecular weights of the four nucleotides, as shown in Figure 9b. Similarly, a 4×4 matrix is constructed to represent these categorizations. In this method, the 16 base pair combinations are divided into three groups along the X-axis in the coordinate system. The coordinates of each base pair are determined based on this division, and a three-dimensional curve is constructed using these coordinates [63]. The z-coordinate in the DN-Curve 3D graph is determined by the positional index of the base pair within the DNA sequence.

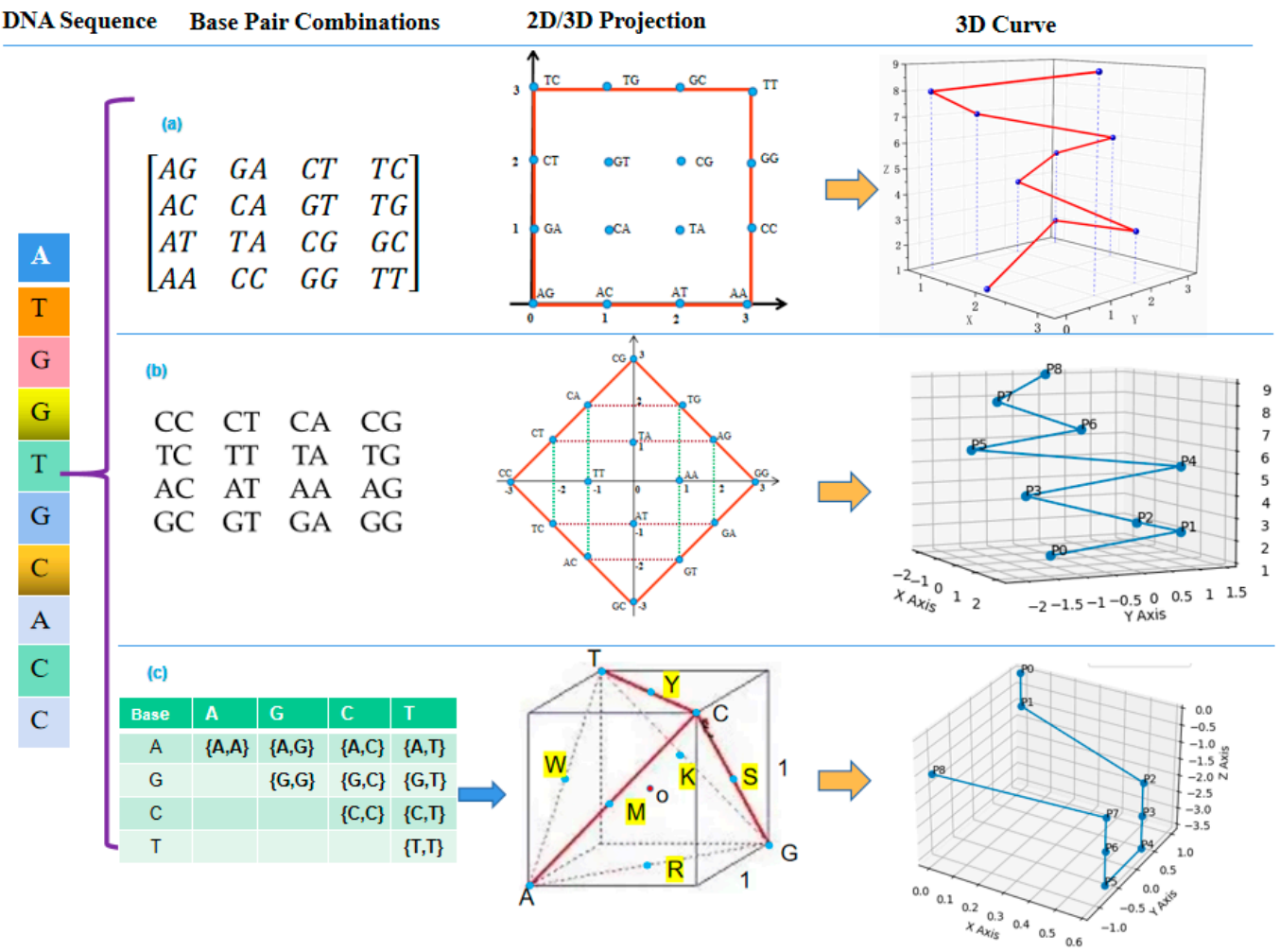

Figure 9. DN-Curve 3D visualization method. (**a**) Chemical property-based visualization principle of 16 base combinations divided into four classes. (**b**) Molecular weight-based visualization principle of 16 base combinations divided into three classes. (**c**) Ten-base combination visualization principle.The process of using the 10-base pair combination DN-Curve 3D method can be described as follows: In this method, the 10 base pair combinations are represented as points mapped onto a regular tetrahedron in three-dimensional space. These points correspond to the vertices and midpoints of the edges of the tetrahedron, forming 10 fixed positions. Each point defines a unique vector originating from the centroid of the tetrahedron $(O(0,0,0))$ and terminating at the corresponding point on the unit sphere. These vectors are normalized to ensure they are unit vectors, meaning all vectors have a length of 1. The z-coordinate for each point in the DN-Curve 3D method is determined by the positional index of the corresponding base pair within the DNA sequence. By connecting the points in three-dimensional space, a curve is constructed to represent the DNA sequence, as depicted in Figure 9c. The spatial arrangement of the points and their associated vectors provides a clear mapping for each base pair combination. This facilitates the visual observation and analysis of the DNA sequence features, including the spatial distribution and structural characteristics of base pair combinations within the DNA sequence.

3.4. C-Curve

The C-Curve technique is a visualization method that maps codons to three-dimensional space. C-Curve is used to display the distribution of codons in DNA sequences and extract certain descriptors that represent both global and local information of the DNA sequence [64]. It assists researchers in studying the encoding information of genomes. Firstly, C-Curve converts the DNA sequence into codon form. Then, each codon is transformed into a set of points, which are mapped to a three-dimensional coordinate

system [65]. In the three-dimensional coordinate system, the points are labeled based on their coordinate values, and the adjacent points are connected to form the C-Curve, as shown in Figure 10a. The coordinate relationships for the nucleotides in a given codon are as follows: For the first nucleotide, if $x > 0$, it is either A or T; otherwise, it is C or G. If $y > 0$, it is either A or G; otherwise, it is C or T. If $x > 0$ and $y > 0$, it is A. For the second nucleotide, its identity is determined by the projection onto the xz-plane (the sign of x and z), while the third nucleotide is determined by the projection onto the yz-plane (the sign of y and z). The z-coordinate corresponds to the codon's position in the sequence and increases linearly, ensuring that the sequential order of codons is preserved in the 3D representation.

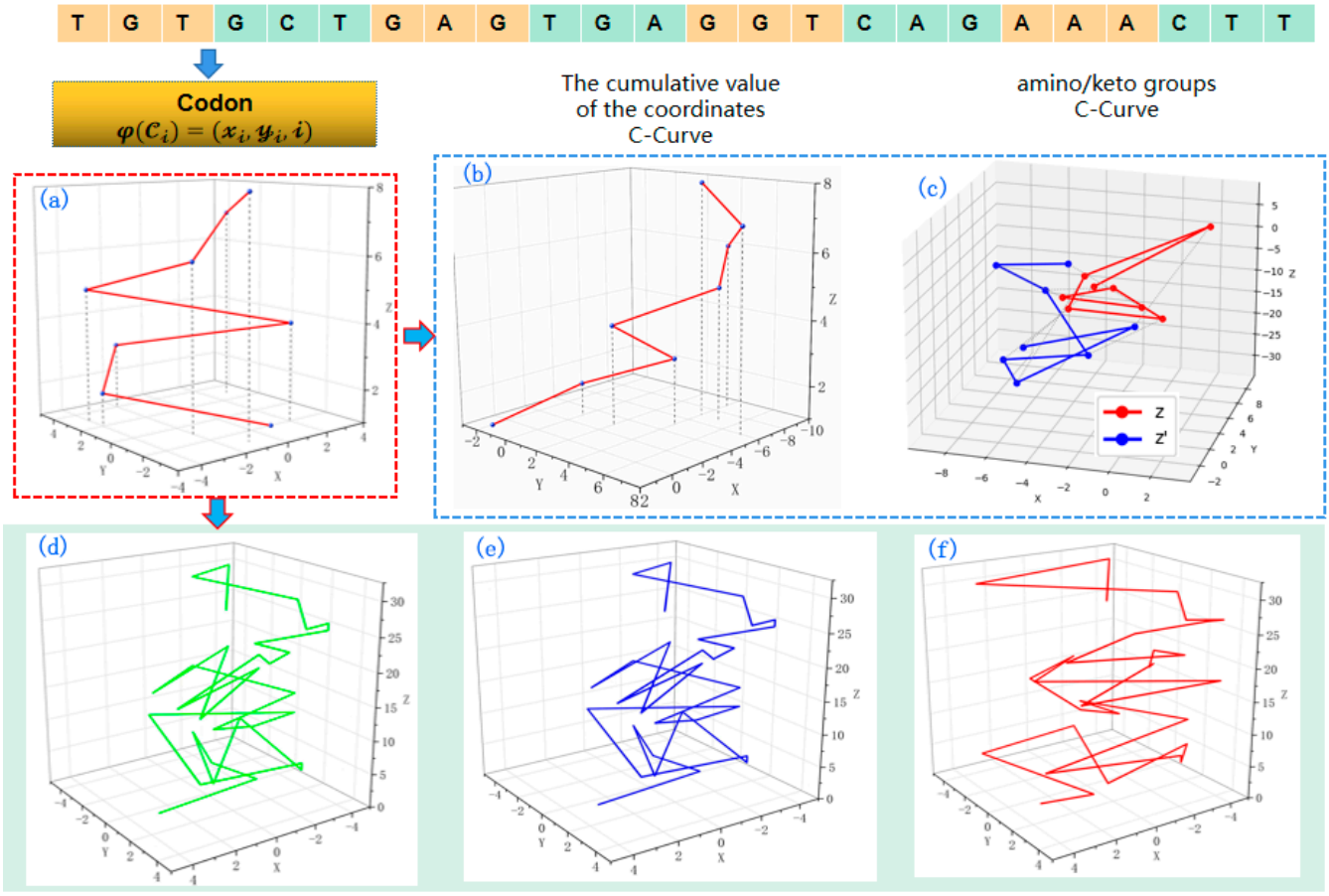

Figure 10. C-Curve visualization method. (**a**) C-Curve of sequence fragment $\mathscr{H}$. (**b**) Cumulative C-Curve of $\mathscr{H}$. (**c**) Variant C-Curve of $\mathscr{H}$. (**d**) C-Curve of sequence δ_1. (**e**) C-Curve of sequence δ_2. (**f**) C-Curve of sequence δ_3.

The alternative invariants inferred by C-Curve can generate various other forms of curves, providing multiple effective methods for sequence information mining. For example, by calculating the cumulative coordinate values of codons, a cumulative effect plot of codons can be generated, as shown in Figure 10b. By referring to Figure 10b, we can gain insights into the global information of the sequence and infer that there is a higher abundance of codons starting with the nucleotides G or C. Additionally, we can also plot $z = x^*y$ and the cumulative value of z to distinguish amino/keto groups, as shown in Figure 10c.

The C-Curve method has significant advantages in the application of sequence similarity measurement [66]. Figure 10d–f depict the C-Curves of different sequences. The curves show that sequence δ_1 (the first exon of human β-globin gene) and sequence δ_2 (the

first exon of Gorilla β-globin gene) are similar. The problem with C-Curve is that when describing very long DNA sequences, it can lose some visual advantages, and comparing them can be complex and time-consuming.

4. Dynamic Visualization

"Dynamic" refers to the ability of visualization methods to dynamically adjust based on the structure, sequential complexity, and characteristics of DNA sequences. Such dynamic methods are particularly useful for sequence similarity analysis, gene function annotation, classification and clustering, as well as applications in drug design and biomedical research. For instance, key features such as periodic patterns, frequency distributions, information entropy, and dynamic changes can be extracted from the spectrum. These features not only comprehensively reflect the global structure and local details of the sequence but also significantly reduce redundant information. Moreover, these features demonstrate a distinct advantage in addressing scalability issues. By extracting high-information-density dynamic features, the original base sequences can be transformed into low-dimensional matrices, thereby reducing the scale of input data and decreasing the computational burden on models. At the same time, these dynamic features are highly compatible with distributed computing frameworks.

4.1. Spectral Dynamic Visualization

Spectral dynamic visualization incorporates spectroscopy and kinetic theory into DNA sequence analysis. It represents the four basic nucleotides using a set of discrete line segments, constructing four sets of B-spectra [67]. Each nucleotide corresponds to a discrete line segment in the B-spectrum. Figure 11a–c depict the spectral dynamic representations of different sequences. The positions of the line segments in the figures represent the positions of the bases in the DNA sequence [68]. The density and positions of the line segments in the B-spectrum provide information about the distribution of specific bases in the DNA sequence [69]. From the B-spectrum, it can be observed that the three sequences have a significantly higher content of the base G. The content of G + C is noticeably higher than A + T in all three sequences. Therefore, it can be inferred that the first exon sequences of the β-globin gene in these three species have a high GC content.

Spectral dynamic visualization has the advantage of being simple and easy to use in sequence quantification analysis. By calculating the rotational inertia of the four sets of B-spectra, four descriptors of the sequence can be obtained [70,71]. These four descriptors can be combined to form a 4D vector representing the sequence. By calculating the similarity index of the 4D vector, we can quantitatively compare the similarity between sequences and quantify the similarity between sequences. Indeed, this method can be applied in various fields, including bioinformatics and chemistry. For example, it is used to compare the similarity of DNA sequences, protein sequences, or chemical structures. The computational complexity of the spectral dynamic graph is not affected by the length of the sequence, making it suitable for addressing similarity measurement problems in long sequences. So, it has significant advantages in addressing large-scale DNA sequence alignment problems.

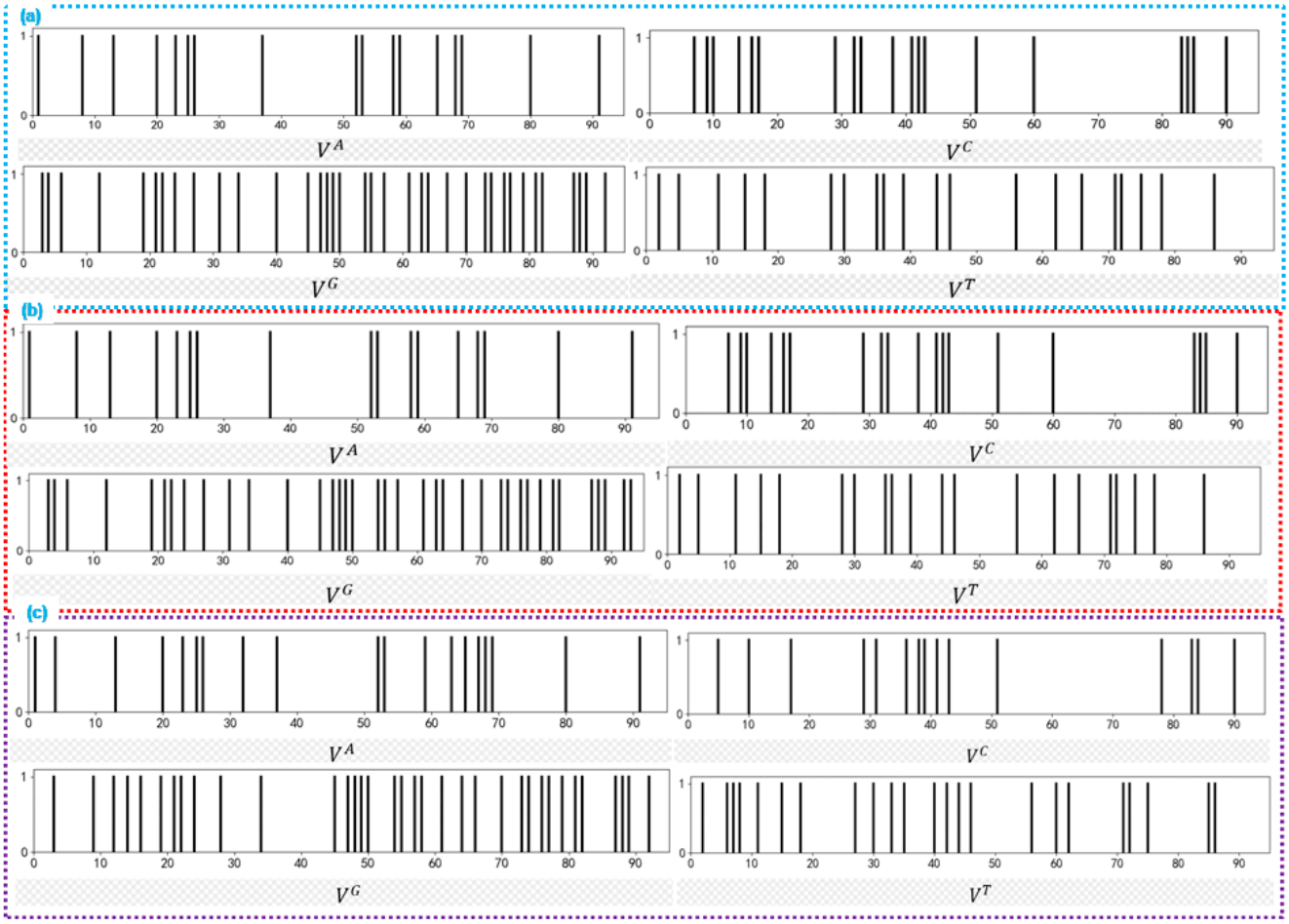

Figure 11. The spectral dynamic method. (**a**) The B-spectrum of sequence δ_1. (**b**) The B-spectrum of sequence δ_2. (**c**) The B-spectrum of sequence δ_3.

4.2. Two-Dimensional Dynamic Visualization

Two-dimensional dynamic visualization represents a sequence using points of different qualities. This approach effectively addresses the issue of degeneracy in sequences. By using points of different qualities, the traversal path of each base in the coordinate system can be clearly visualized, making the structure and features of the sequence more distinct and visible [72]. Unlike traditional static methods, the dynamic approach represents sequences as sets of material points with different qualities, such as mass, to capture more nuanced details of the sequence's structural and spatial properties. The construction process involves three main steps: First, construct the random walk graph for the bases, as shown in Figure 12a. Secondly, determine the quality of the points, where each ellipse represents the endpoint of a vector, and the number of points inside the ellipse represents the quality of that point. Finally, construct a dynamic graph based on the quality of each point, as shown in Figure 12b. This graph clearly visualizes traversal paths and base positions, achieving a two-dimensional dynamic representation of the DNA sequence.

This representation method has found applications in the field of DNA sequence analysis. Two-dimensional dynamic visualization can visually display the similarity and diversity between species, revealing the evolutionary relationships and history among species and constructing phylogenetic trees to study the temporal evolution of species [73,74]. Figure 12c,d present the two-dimensional dynamic representations of different sequences, providing a framework to compare their structural characteristics. The similarity between these sequences can be observed by analyzing specific patterns in the graphs. For instance,

the clustering patterns of points (distinguished by color and shape based on their masses) reveal the distribution characteristics of the sequences. Additionally, the overall shapes, including the trajectories and spreads of points in the $x - y$ plane, highlight structural similarities. While minor variations in clustering or distribution may exist, the consistent trends in point clustering and spatial distributions suggest a high degree of visual similarity between the two sequences. However, for longer DNA sequences, the visualizations may become complex and difficult to directly compare. In such cases, it is necessary to extract sequence descriptors from the two-dimensional dynamic visualizations. Centroid and moment of inertia serve as sequence descriptors that quantitatively characterize the spatial distribution of nucleotides in the 2D dynamic graph. The centroid represents the weighted average position of all nucleotides in the sequence, reflecting the overall nucleotide composition and bias. The moment of inertia measures the spread of nucleotides relative to the centroid, capturing structural features such as clustering, symmetry, and distribution patterns. These descriptors enable more precise sequence comparisons by providing numerical values that reflect both global and local structural characteristics of the sequences.

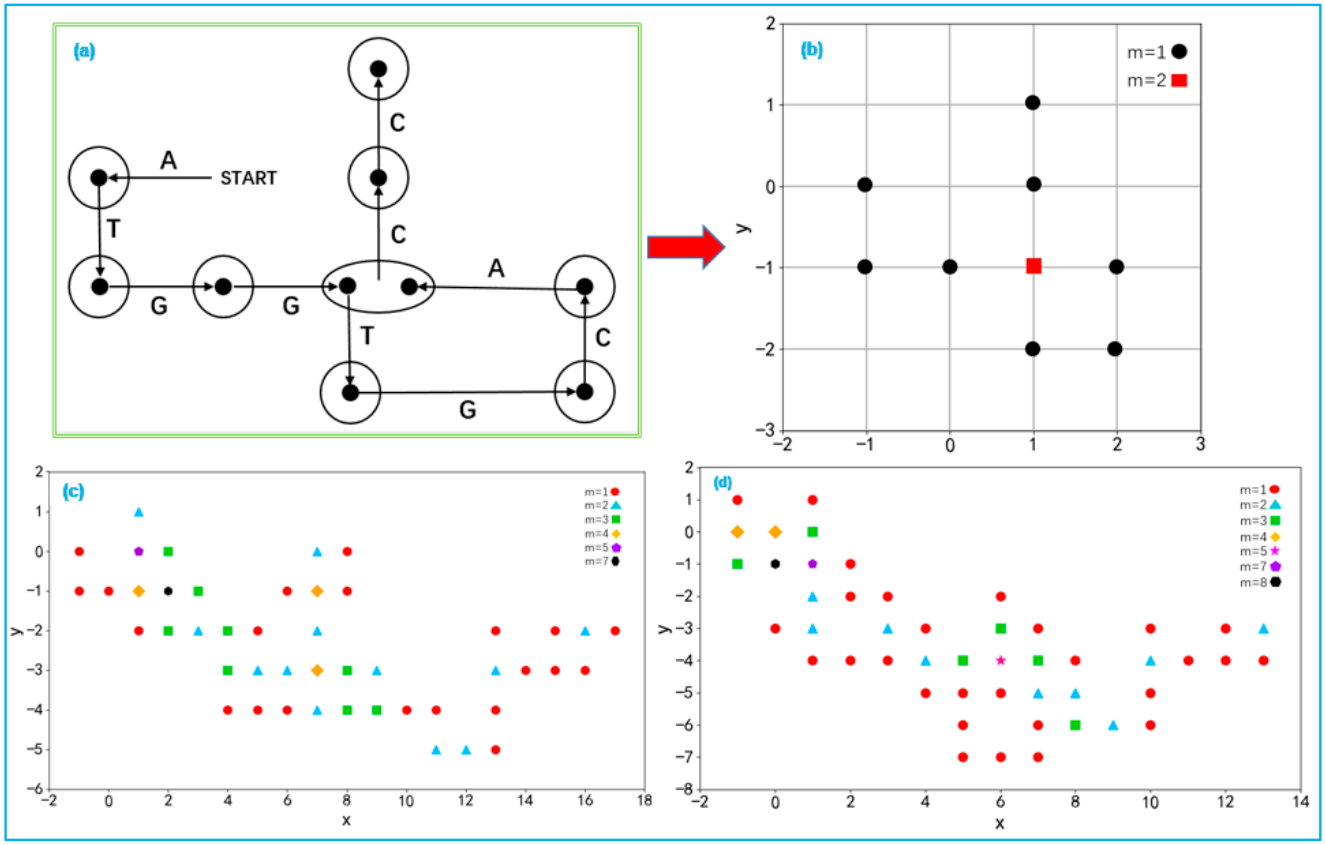

Figure 12. Two-dimensional dynamic visualization method. (**a**) Process of constructing a 2D dynamic visualization of sequence fragment $\mathcal{S}$. (**b**) Two-dimensional dynamic graph of sequence $\mathcal{S}$. (**c**) Two-dimensional dynamic graph of sequence δ_4. (**d**) Two-dimensional dynamic graph of sequence δ_5.

4.3. Coded Mark Inversion (CMI) Encoding Dynamic Visualization

The CMI encoding dynamic visualization method categorizes DNA sequences based on the chemical properties of the bases [31]. It translates DNA sequences into source sequences and converts the source sequences into a 0–1 encoded sequence. On the contrary, the source sequences are obtained by reverse-translating the 0–1 encoded sequence. Based on chemical properties, nucleotides can be divided into three groups, with each group containing two types [75,76]. Therefore, we can obtain six different sets of CMI-encoded

graphics, with each set corresponding to a specific combination of bases, as shown in Figure 13.

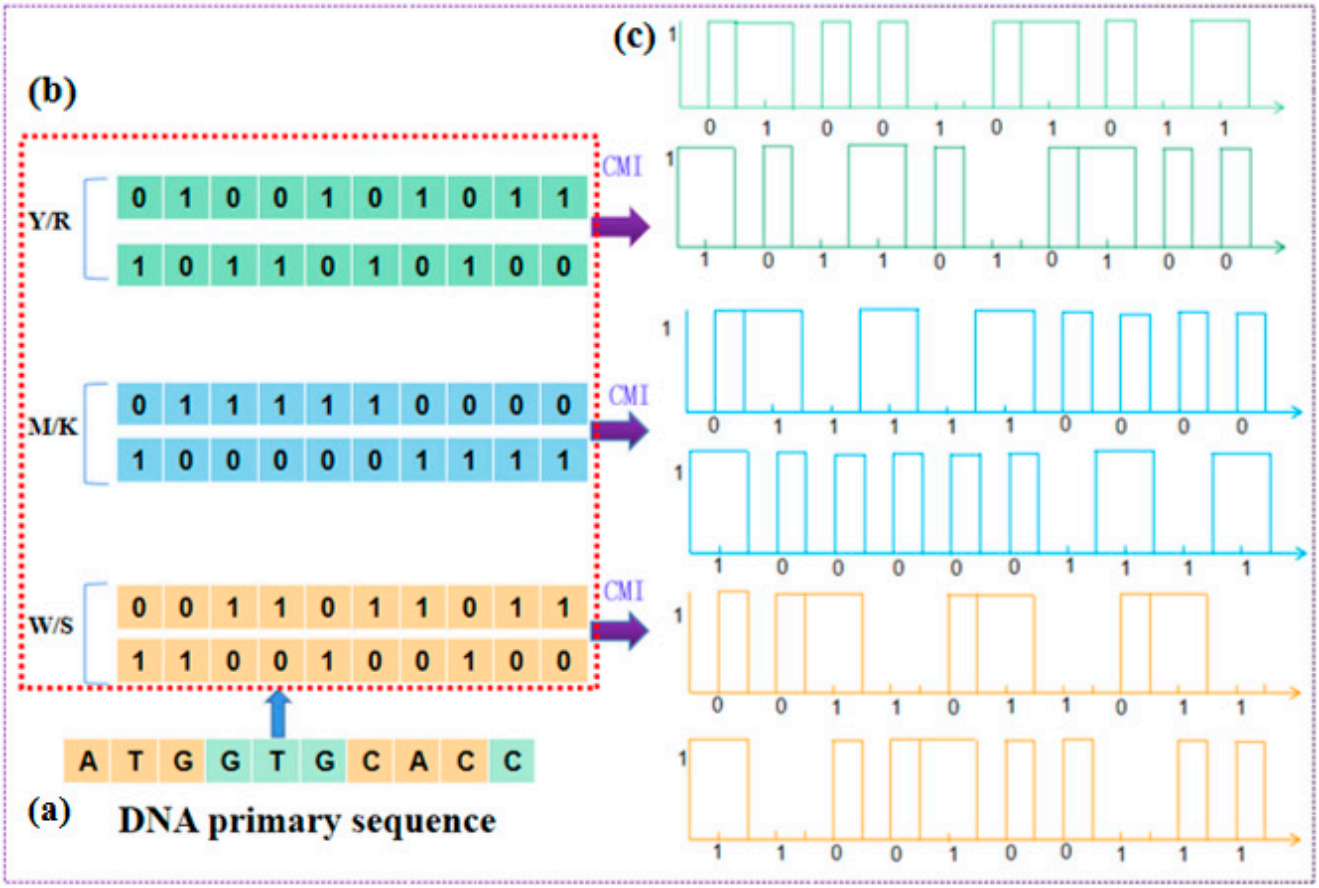

Figure 13. A principle diagram of the CMI encoding visualization method. (**a**) The original DNA sequence is displayed at the bottom. (**b**) Three binary encodings are generated based on the nucleotide classifications: Y/R (pyrimidine/purine), M/K (amino/keto), and W/S (weak/strong hydrogen bonding). (**c**) The CMI representation.

CMI encoding dynamic visualization integrates both the chemical properties and structural features of DNA sequences. Not only does it provide an intuitive way to observe and analyze the features of DNA sequences, but it also allows for the extraction of structural and chemical information from the sequences [77]. Additionally, this representation method allows for quantitative measurement of sequence similarity with relatively low computational cost, providing valuable clues for subsequent sequence analysis and functional research.

5. Four-Dimensional Numerical Representation

5.1. The Bandwidth Numerical Representation

The numerical representation of bandwidth is a method of representing sequences using the bandwidth invariant. By calculating the average of the off-diagonal elements in the distance matrix, a set of numerical features called "band invariants" is obtained [78]. These band invariants can be used for comparing and analyzing sequences. The numerical representation method of bandwidth is simple and efficient, allowing for the fast analysis

and processing of DNA sequences [79]. To construct a distance matrix, the Euclidean distance between two bases in a sequence fragment S is calculated as follows:

$$\sqrt{\left\{ (A_i - A_j)^2 + (T_i - T_j)^2 + (G_i - G_j)^2 + (C_i - C_j)^2 \right\}} \quad (3)$$

Here, A_i, T_i, G_i and C_i represent the coordinate values of the i-th base in the sequence fragment $\mathcal{S}$, where $i = 1, 2, 3, \cdots n$. Similarly, A_j, T_j, G_j and C_j represent the coordinate values of the j-th base in the sequence fragment $\mathcal{S}$, where $j = 1, 2, 3, \cdots n$.

To quantitatively calculate the similarity of DNA sequences, invariant features are extracted from the distance matrix. Figure 14 shows the bar plot of the average bandwidth values from the distance matrices of three sequence fragments [80,81]. Figure 14 clearly demonstrates that the average bandwidth values of sequence fragments $\mathcal{S}$ and $\mathcal{S}'$ in the distance matrices are very similar, while they exhibit significant differences compared to sequence fragment $\mathcal{S}''$. It can be noted that the bandwidth invariants of different DNA sequences exhibit considerable variations in both their form and numerical values. In any local feature analysis, it is possible to set different bandwidths for calculation [31,42]. However, using different values may result in variations in local similarity.

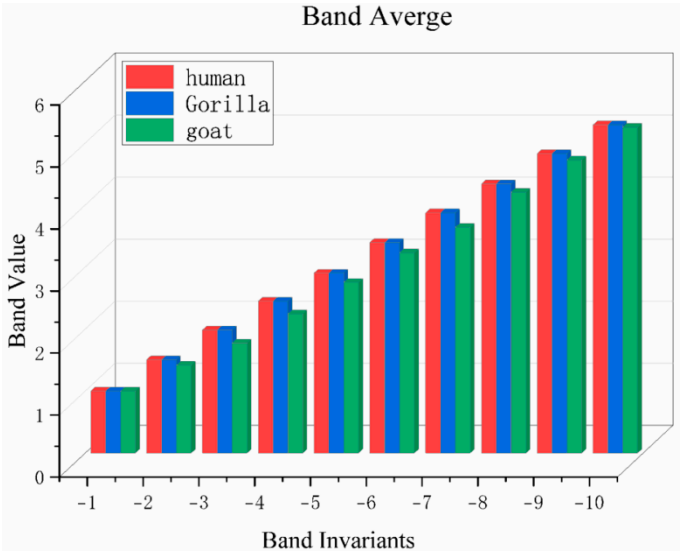

Figure 14. Bar chart of average bandwidth for a 10 × 10 fragment distance matrix of sequences $\mathcal{S}$, $\mathcal{S}'$, and $\mathcal{S}''$.

In Figure 14, the horizontal axis (band invariants) represents the band invariants, which correspond to the indices of different off-diagonals in the distance matrix. For example, -1 represents the first off-diagonal of the distance matrix, -2 represents the second off-diagonal, and so on. Each diagonal represents the distances between different base pairs, with smaller indices indicating larger distance spans between the bases. The vertical axis (band value) represents the band value corresponding to each off-diagonal, which is the average of all elements on that diagonal. These band values reflect the statistical distribution of relative distances between base pairs in 4D space. The numerical representation method of bandwidth offers high flexibility, simple computation, and time efficiency. It allows for comparing sequences of arbitrary lengths. It is widely used in the field of bioinformatics and computational biology, such as DNA sequence alignment, sequence clustering, and phylogenetic tree construction. This method has advantages such as mathematical clarity, determinism, and independence from the direction of the bases [82]. It provides a quantitative analysis tool for studying the similarity and differences between DNA sequences. However, this method loses the ability of graphical visualization

and direct comparison of sequences, making it relatively weaker in terms of sequence visualization and intuitive comparison.

5.2. Chemical and Physical Properties Numerical Representation

The numerical representation method of the chemical and physical properties determines the coordinate values of bases using the classification of their chemical properties (R/Y, M/K, W/S) [83]. This method simultaneously considers the physical and chemical structures of the sequence. The specific construction method is as follows: The initial DNA sequence, $\mathcal{S} = s_1, s_2, s_3, \cdots s_{\mathcal{N}}$, is simplified into a set of points, $\varphi(\mathcal{S}) = \varphi(g_1)\varphi(g_2)\varphi(g_3)\cdots\varphi(g_N)$. The mathematical model corresponding to each point is as follows:

$$\varphi(g_i) = \begin{cases} \left(1, 1, 1, 1 - \frac{1}{i}\right) \text{ if } g_i = \text{A} \\ \left(1, 0, 0, 1 - \frac{1}{i}\right) \text{ if } g_i = \text{G} \\ \left(0, 1, 0, 1 - \frac{1}{i}\right) \text{ if } g_i = \text{C} \\ \left(0, 0, 1, 1 - \frac{1}{i}\right) \text{ if } g_i = \text{T} \end{cases} \tag{4}$$

where $i = 1, 2, 3 \cdots \mathcal{N}$,$\mathcal{N}$ represents the length of the sequence, and the resulting 4D representation of the DNA primary sequence is unique. By performing simple calculations, the geometric center of the base distribution frequencies in the sequence can be obtained [84]. At the same time, we can obtain the relationships between the different bases in the DNA sequence. The geometric center of a sequence is fixed, so it can be used as an invariant in the matrix to compare the similarity between two DNA sequences. This method also avoids information loss during the transition from DNA sequence to mathematical representation [85]. However, a drawback is that it loses the visual representation of the sequence.

5.3. Bases Combinations Numerical Representation

The numeric representation method of base combinations utilizes a combination of four nucleotides (A, T, G, C), 16 base pairs, and 64 codons to establish a new representation method [86]. The 4D spatial coordinates of $\varphi(g_i)$ are defined as ($\mathcal{X}_{\text{m}}$, $\mathcal{Y}_{\mathcal{N}}$, $\mathcal{Z}_p$, $\mathcal{W}_q$). The value of $\mathcal{X}_{\text{m}}$ (m<=$\mathcal{N}$) is determined based on the priority order of individual nucleotides occurring in the sequence, i.e., [3, 2, 1, 0]. $\mathcal{Y}_{\mathcal{N}}(\mathcal{N} <= \mathcal{N})$ represents the values associated with the 16 possible combinations of nucleotide pairs. When $\mathcal{N} < \mathcal{N}$, the specific calculation of each nucleotide pair is given by equation (5). When the nucleotide pair consists of the $\mathcal{N}$ th nucleotide and a space, the value is equal to the average of the values of the four nucleotide pairs formed by the $\mathcal{N}$ th nucleotide with the four nucleotides. $\mathcal{Z}_p$(p<=$\mathcal{N}$)

represents the numerical value of the 61 codons encoding amino acids. $\mathscr{W}_q$ represents the number of occurrences of the four nucleotides (A, T, G, C) in the sequence [87,88].

$$\varphi(g_i g_{i+1}) = \begin{cases} 1 \text{ if } g_i g_{i+1} = \text{AA} \\ 2 \text{ if } g_i g_{i+1} = \text{AT} \\ 3 \text{ if } g_i g_{i+1} = \text{AG} \\ 4 \text{ if } g_i g_{i+1} = \text{AC} \\ 5 \text{ if } g_i g_{i+1} = \text{TA} \\ 6 \text{ if } g_i g_{i+1} = \text{TT} \\ 7 \text{ if } g_i g_{i+1} = \text{TG} \\ 8 \text{ if } g_i g_{i+1} = \text{TC} \\ 9 \text{ if } g_i g_{i+1} = \text{GA} \\ 10 \text{ if } g_i g_{i+1} = \text{GT} \\ 11 \text{ if } g_i g_{i+1} = \text{GG} \\ 12 \text{ if } g_i g_{i+1} = \text{GC} \\ 13 \text{ if } g_i g_{i+1} = \text{CA} \\ 14 \text{ if } g_i g_{i+1} = \text{CT} \\ 15 \text{ if } g_i g_{i+1} = \text{CG} \\ 16 \text{ if } g_i g_{i+1} = \text{CC} \end{cases} \quad (5)$$

Taking a base segment $\mathcal{S}$ as an example, the calculation of coordinates in 4D space is shown in Table 1. Based on the coordinate values of the points in 4D space, it can be inferred that there is a one-to-one correspondence between any DNA sequence and its 4D graphical representation [89,90]. This representation fully captures the biological information of the DNA sequence and prevents information loss.

Table 1. Four-dimensional coordinate values of each base in sequence fragment $\mathcal{S}$ = ATGGTGCACC.

Single Nucleotide	$\mathscr{X}_m$	Base Combinations	$\mathscr{Y}_N$	Triplet Bases	$\mathscr{Z}_p$	$\mathscr{W}_q$	4D Coordinates
A	3	AT	2	ATG	−2	1	(3,2,−2,1)
T	2	TG	7	TGG	1	1	(2,7,1,1)
G	1	GG	11	GGT	0	1	(1,11,0,1)
G	1	GT	10	GTG	−2	2	(1,10,−2,2)
T	2	TG	7	TGC	0	2	(2,7,0,2)
G	1	GC	12	GCA	−2	3	(1,12,2,3)
C	0	CA	13	CAC (ACA + ACG + ACU + ACC)	2	1	(0,13,2,1)
A	3	AC (CA + CU + CC + CG)	4	ACC	1	2	(3,4,1,2)
C	0	CC	16	CC_	0	2	(0,16,0,2)
C	0	C_	14.5	C__	0	3	(0,14.5,0,3)

.

By using the 4D coordinates of each nucleotide in a DNA sequence fragment, it is possible to calculate the geometric center of the sequence fragment. The computed geometric center can be used as an invariant feature of the sequence. By calculating the Euclidean distance and the angle between two feature vectors, we can measure the similarity between the two sequences. This numerical representation method based on single, double, and triple base combinations combines the physical and chemical properties of DNA sequences, allowing for an effective representation of the structural features of the sequences [91]. This method has extensive applications in various fields such as DNA sequence alignment, classification, and analysis.

The four-dimensional numerical representation method not only possesses significant computational efficiency advantages but also offers unique insights into evolutionary

biology that other visualization methods cannot provide. Firstly, by integrating multi-dimensional attributes, it provides a more comprehensive perspective for revealing similarities and variabilities among sequences. Secondly, it can precisely quantify the similarities and differences of sequences of any length, offering an efficient tool for studying sequence similarities and variabilities. Additionally, it can reveal the distribution characteristics and patterns of base properties within sequences, which aids in analyzing key functional regions or specific pattern changes within the sequences.

6. Discussion and Prospect

6.1. Analysis of Different Sequence Visualizations

6.1.1. Two-Dimensional Representation

The two-dimensional representation of DNA sequences is used to intuitively visualize and analyze key features of DNA sequences in computational biology. Its primary advantages are as follows [92–94]:

(1) Intuitiveness: two-dimensional representation enables quick identification of base distribution patterns and probabilistic features.
(2) Information richness: two-dimensional representation captures and conveys more information than one-dimensional methods, providing a richer display of data.
(3) Handling large data volumes: Two-dimensional representation methods can effectively process large datasets. For example, DNA sequence visualization models based on grayscale images can handle extensive DNA sequences with minimal loss of information.
(4) Low computational complexity: graphical representation methods provide a low computational complexity approach to describing gene sequences, significantly accelerating the research process compared to traditional approaches.

However, the two-dimensional representation of DNA sequences also has several notable drawbacks [95,96]:

(1) Information loss: some two-dimensional models may be unsuitable for representing long DNA sequences, potentially resulting in the loss of detailed information and increasing the risk of observers overlooking critical data.
(2) Limited distinguishability: visualization based on grayscale images may have limited distinguishability, potentially hindering the effective extraction of useful information.
(3) Localization ambiguity: when distribution differences arise in the graphical representation of gene sequences, pinpointing the exact locations of the differing sequences becomes challenging.
(4) Technical limitations: Two-dimensional visualization models, such as spectral-based models, are effective in revealing the properties of shorter DNA sequences. However, they face challenges when applied to long DNA sequences due to limitations in resolution and clarity. In such cases, key structural and compositional features may become obscured or difficult to interpret within a two-dimensional framework.

Despite these limitations, two-dimensional representation methods excel in handling large datasets composed of multiple shorter DNA sequences. Their efficiency and intuitive visualization capabilities make them particularly suitable for analyzing high-volume datasets. Nonetheless, challenges such as potential information loss and limited distinguishability remain [97]. Therefore, selecting an appropriate representation method should be guided by specific research objectives and data characteristics.

6.1.2. Three-Dimensional and Higher-Dimensional Representation Approaches

Three-dimensional DNA sequence representation methods are widely used to capture and analyze the spatial organization and interactions of DNA sequences in three-dimensional space, providing distinct advantages in computational biology [98,99].

(1) Understanding spatial conformation mechanisms: three-dimensional genomics provides a clearer understanding of the mechanisms underlying changes in chromatin spatial conformation and gene transcriptional regulation.
(2) Comprehensive mapping of intracellular chromatin interactions: three-dimensional genomics facilitates the comprehensive mapping of chromatin interactions within cells.
(3) Revealing genome functional mechanisms: three-dimensional genomics, combined with 3C-based techniques and other derivative sequencing technologies, has provided scientists with an increasingly clear understanding of the spatial conformation of genomes in plants, animals, and microorganisms.
(4) Predicting Three-Dimensional Interactions: recent machine learning approaches can predict three-dimensional spatial interactions from one-dimensional epigenomic data or DNA sequences, as shown by models like DeepC and Akita [100,101].

The main drawbacks of three-dimensional and higher-dimensional DNA sequence representation methods include the high complexity of data integration analysis and high computational resource requirements. Higher-dimensional representation methods generate vast and complex datasets, making the integration and analysis of three-dimensional structural differences across different sequence types highly challenging. Predicting three-dimensional structures from one-dimensional sequences demands substantial computational resources to extract meaningful information [102,103].

Higher-dimensional DNA sequence representation methods offer significant advantages for understanding the three-dimensional structure of genomes. However, they also face challenges such as complex data processing, high costs, and limited analysis tools. With continued technological advancements, these limitations are expected to be mitigated, enabling broader and more in-depth applications of multi-dimensional DNA sequence representation in computational biology [104,105].

6.1.3. Influence of DNA Sequence Representation on Machine Learning

(1) Differences in feature extraction capability: DNA sequence representation methods convert sequence data into numerical features that can be processed by machine learning models. Different representation methods capture various local features (such as domains or motifs) and global features (such as information from entire coding regions), thereby influencing the performance of machine learning models.
(2) Differences in computational complexity: two-dimensional and higher-dimensional sequence representation methods vary in data volume, storage requirements, and modeling mechanisms within machine learning models, resulting in differences in computational complexity.
(3) Differences in machine model selection: model selection is often guided by the specific DNA sequence representation method used, as different models have varying requirements for DNA sequence representations.
(4) Differences in model generalization ability: Different DNA sequence representation methods exhibit varying levels of generalization ability. Due to differences in the underlying mechanisms of these representations, machine learning models demonstrate different adaptability to sparse or imbalanced training data. Low-dimensional representations generally offer better generalization than higher-dimensional representations.

DNA sequence representation methods impact various aspects of machine learning models [106,107], including feature extraction, model performance, computational complexity, prediction accuracy, and generalization ability [108–110]. Therefore, selecting an appropriate DNA sequence representation method is essential for building effective machine learning models.

6.2. Application Fields of Sequence Visualization

6.2.1. Application Field Overview

DNA sequence visualization is not only a tool for displaying genetic information but also a crucial method for analyzing biological data and decoding complex biological phenomena [111,112]. Ensemble learning models that integrate DNA visualization with machine learning hold especially significant value in the field of biology. DNA visualization not only provides the model with rich input information but also allows for an intuitive display of output results, thereby enhancing the model's accuracy, interpretability, and practical effectiveness. Such ensemble learning models demonstrate broad applicability and value in the fields of biology and medicine [92,113].

(1) Genome Analysis and Disease Decoding

In genome analysis and disease decoding, DNA sequence visualization can integrate information from multiple data sources, such as genomic sequences, transcriptomic data, protein interaction networks, and epigenetic information. By using these multi-source data as inputs for machine learning models, researchers can effectively identify structural variations in chromosomal regions and gene regulatory networks. Presenting these complex data relationships in a visual format allows machine learning models to better reveal associations between genetic variations and diseases, offering new perspectives for studying disease mechanisms.

(2) Personalized Medicine

In the field of personalized medicine, DNA sequence visualization combined with machine learning offers new possibilities for developing precise medical treatments. This approach shows significant value, particularly in gene mutation prediction and the design of gene editing and targeted therapies. By visually presenting machine learning predictions, researchers can intuitively interpret a patient's specific mutations, providing strong support for the development of personalized diagnostic and treatment plans.

(3) Gene Expression and Regulatory Prediction

In gene expression and regulatory prediction, machine learning models (such as BigRNA) can predict regulatory features from DNA sequences, including RNA expression levels, splice sites, and microRNA binding sites. By visualizing these predictions, researchers can identify key non-coding variants and regulatory regions, further exploring their roles in gene expression regulation and their potential impact on disease.

(4) Promoter Engineering Research

In promoter engineering, visualization combined with machine learning models can be used to build quantitative models of promoter sequences. This approach allows for the effective analysis of their functional characteristics. The visualization of promoter sequences can reveal the regulatory effects of different sequences on gene expression. This provides scientists with data support for designing promoters to meet specific expression needs, enabling the precise control of gene expression.

(5) Genomics Research

In genomics research, DNA sequence visualization combined with machine learning models is used to predict gene regulatory networks, identify essential genes, and analyze complex genome structures. By using DNA visualization for multi-dimensional data integration and analysis, researchers can gain deeper insights into the relationships between genes and phenotypes, thereby advancing progress in genomics research.

6.2.2. Multidimensional Biological Data Integration and Analysis

Multidimensional omics data, including genomics, transcriptomics, proteomics, and metabolomics, have been increasing exponentially [114,115]. The integration and analysis of these diverse biological datasets have become essential for understanding complex biological systems. DNA sequence visualization techniques play a crucial role in this field, facilitating the systematic integration, data mining, and feature analysis of various omics datasets [116,117]. These techniques enable researchers to understand the functions and regulatory mechanisms of complex biological networks at multiple levels. Integrative analysis enables a more comprehensive elucidation of the interactions and dynamic changes in various biological information at the cellular, tissue, or organismal levels [118,119].

(1) Heterogeneous Integration and Standardization of Multi-Omics Data

The integration of multi-omics data is a major research focus in bioinformatics. With the rapid accumulation of multidimensional omics datasets, integrating these heterogeneous data to uncover the complex regulatory networks within biological systems has become a critical challenge. However, traditional sequence numerical encoding faces notable limitations in multi-omics data integration. These limitations primarily include inconsistent expression standards across different data sources, complex quantification methods, and considerable variability in sequence lengths. Consequently, numerical encoding often results in inconsistencies in data dimensionality, which further complicates feature extraction and increases model processing complexity.

DNA sequence visualization techniques provide an effective solution for multi-omics data integration. Through visual encoding, heterogeneous data such as genomics and transcriptomics can be intuitively displayed and compared on a unified platform, ensuring consistency in data format and scale. Sequence visualization not only highlights the key features of each omics data type but also effectively reveals inter-layer relationships, facilitating cross-platform integration of multi-omics data. Additionally, sequence visualization assists in identifying and correcting outliers or biases, reducing error accumulation stemming from format discrepancies. This integration and standardization process enhances data comparability and consistency, establishing a robust foundation for multi-omics integrative analysis.

(2) Complex Feature Extraction from High-Dimensional Sequence Data

Feature extraction is essential in high-dimensional sequence data analysis. However, the context-dependent features commonly found in sequence data are often difficult to capture accurately through numerical encoding methods. Numerical encoding not only fails to represent the complex relationships and regulatory information between sequences but also, due to its high dimensionality, results in incomplete feature extraction and introduces redundant features. This encoding approach further increases the time complexity and computational cost of data processing, ultimately diminishing the model's accuracy and efficiency.

Transforming high-dimensional sequence data into graph structures is an effective strategy for feature extraction. Graph structures can flexibly accommodate sequence data of varying lengths and extract multidimensional features from inter-sequence relationships.

For instance, dynamic visualization techniques can illustrate temporal changes in gene expression, DNA mutations, and metabolomic features, making them particularly suited for studying processes like cell differentiation and drug response, and revealing the temporal dependencies and dynamic mechanisms underlying these biological phenomena. Graph structures are highly effective at capturing spatial relationships, functional modules, and other biologically relevant information within high-dimensional data, thereby enhancing the biological significance of extracted features. Additionally, through network analysis and clustering methods, graph structures can effectively reduce redundant features, improving both the representativeness and discriminative power of the features. This multidimensional feature extraction approach not only simplifies the processing of high-dimensional data but also significantly enhances data interpretability, providing a solid foundation for subsequent data analysis and biological research.

6.2.3. Disease-Associated Mutation Analysis and Personalized Therapy

(1) Mutation Prediction in Personalized Medicine

Predicting gene mutations is essential for personalized medicine. Accurate mutation prediction assists in assessing genetic disease risk, optimizing drug response, and developing targeted treatment plans, thus providing scientific support for early diagnosis and personalized therapeutic strategies. However, gene mutation data originate from diverse individuals, with variations in mutation locations and types causing inconsistencies in data dimensionality and scale [120,121]. Conventional numerical encoding methods struggle to flexibly adapt to these variations, often resulting in information loss or error accumulation, which subsequently impacts the accuracy and stability of mutation predictions. Moreover, the complex characteristics of gene mutations depend not only on sequence context but also on spatial and regulatory interactions with other genes and regulatory elements—interactions that numerical encoding fails to capture effectively.

To address the limitations of numerical encoding, DNA sequence visualization techniques offer an efficient solution. By visualizing mutations and polymorphisms, the complex information within gene mutations can be intuitively displayed, making their distribution, frequency, and associations with diseases easily discernible. Tools such as heatmaps, distribution plots, and 3D structural visualizations enable researchers to rapidly identify mutation hotspots, rare variants, and regions with high-frequency polymorphisms, thereby uncovering potential biological functions. The intuitive nature of visualization techniques enhances data interpretability, allowing clinicians to analyze and interpret mutation data more effectively and providing strong support for mutation analysis in personalized medicine [122–124].

(2) Gene Editing and Targeted Therapy Design

In gene editing and targeted therapy design, DNA sequence visualization techniques provide essential support for target identification and mutation repair design. By leveraging 3D visualization to reveal the spatial structure of mutation sites within the genome, researchers can precisely locate editing targets and surrounding regulatory regions, thereby enhancing the accuracy and efficiency of gene editing [125,126]. This visualization approach not only aids in optimizing gene editing design and minimizing off-target effects but also ensures the specific impact of editing operations on the target region, laying a robust technical foundation for the advancement of precision medicine and targeted therapies [127,128].

6.3. Potential Research Directions

Traditional DNA sequence visualization methods have limitations when dealing with long sequences, mainly manifested as the loss of global structure and local details due to

information density, high complexity in data integration, and the significant consumption of computational resources. To address these issues, this paper proposes two potential approaches: Firstly, the construction of a knowledge graph for DNA sequence big data is proposed, which integrates genome data; annotates functional regions, mutation hotspots, and sequence patterns; and combines high-dimensional visualization techniques to achieve the precise parsing and efficient display of long sequences. Secondly, the integration of machine learning with DNA sequence visualization is suggested, which through feature extraction and pattern recognition, automatically identifies important patterns and functional areas, reduces redundant information, and generates concise and meaningful visualization images, providing an efficient solution for the analysis of long sequences.

6.3.1. Construction of Knowledge Graph for DNA Sequence Big Data

Biological sequence data exhibit characteristics such as being sourced from multiple sources, having heterogeneity, and being loosely organized [129]. There are many challenges in extracting knowledge from massive biological data. Biologists often use specialized terms, such as SNOMED, CT, ICD-10, PubChem, Gene Ontology (GO), and so on [130,131]. These terms convey important semantics and consensus among biologists. However, for computer experts, these terms are often dry and obscure, making them difficult to understand [132,133]. Especially when dealing with large-scale data, it becomes challenging to precisely locate the required knowledge. When biological researchers use computer programs or software for bioinformatics analysis, the time complexity is often high. Moreover, they need to filter out potentially useful data from numerous search results, and the process of handling long sequences of genes and genomes is very time-consuming [134,135]. Therefore, traditional computer programs have limitations in retrieving and analyzing sequencing data, and they cannot meet the scientific community's needs for information connectivity and search. To address this issue, establishing a language that can be understood by biologists, computer scientists, and machines alike through the use of machines and machine learning is a fundamental solution [136].

The advancement of sequencing technologies has a major advantage in providing low-cost digital data and high-resolution sequencing results. This is a prerequisite for knowledge extraction. When it comes to storing large-scale sequencing data, there is a growing focus on building more efficient data structures and management systems [137,138]. Indeed, data are not equivalent to knowledge, so further exploration is needed on how to perform optimal searches on data to extract meaningful knowledge from them. To be exact, the development of suitable analysis software and integrated platforms is crucial to better establish a semantic network of data.

Knowledge graphs (KGs) are a technique that describes concepts, entities, and their relationships in the objective world in a structured manner [139,140]. It can represent information on the Internet in a form that is closer to the human cognitive world, providing a better ability to organize, manage, and understand the vast amount of information on the Internet [141–144]. On one hand, they can be continuously updated with new knowledge and information. On the other hand, knowledge may be distributed across different data sources. KGs play a significant role in addressing the challenges of biological sequence big data [145]. By utilizing KGs, it becomes possible to establish richer semantic networks within sequencing data by linking them with other biological knowledge and entities. By leveraging inferences and querying on the concepts and relationships within the KG, it is possible to achieve semantic search and knowledge extraction from sequencing data [146,147]. Therefore, establishing a graphical association KG system for biological sequences is an important direction of development.

(1) General framework

As an opening for further discussion and exploration, this article proposes a concept and research directions for constructing a KG of DNA sequence big data by utilizing KG-related technologies and focusing on the characteristics of DNA sequence big data [137,148]. The aim is to provide researchers with a new solution for the in-depth exploration and decision analysis of massive DNA sequence data [149,150]. The framework and key technologies for constructing KGs are shown in Figure 15.

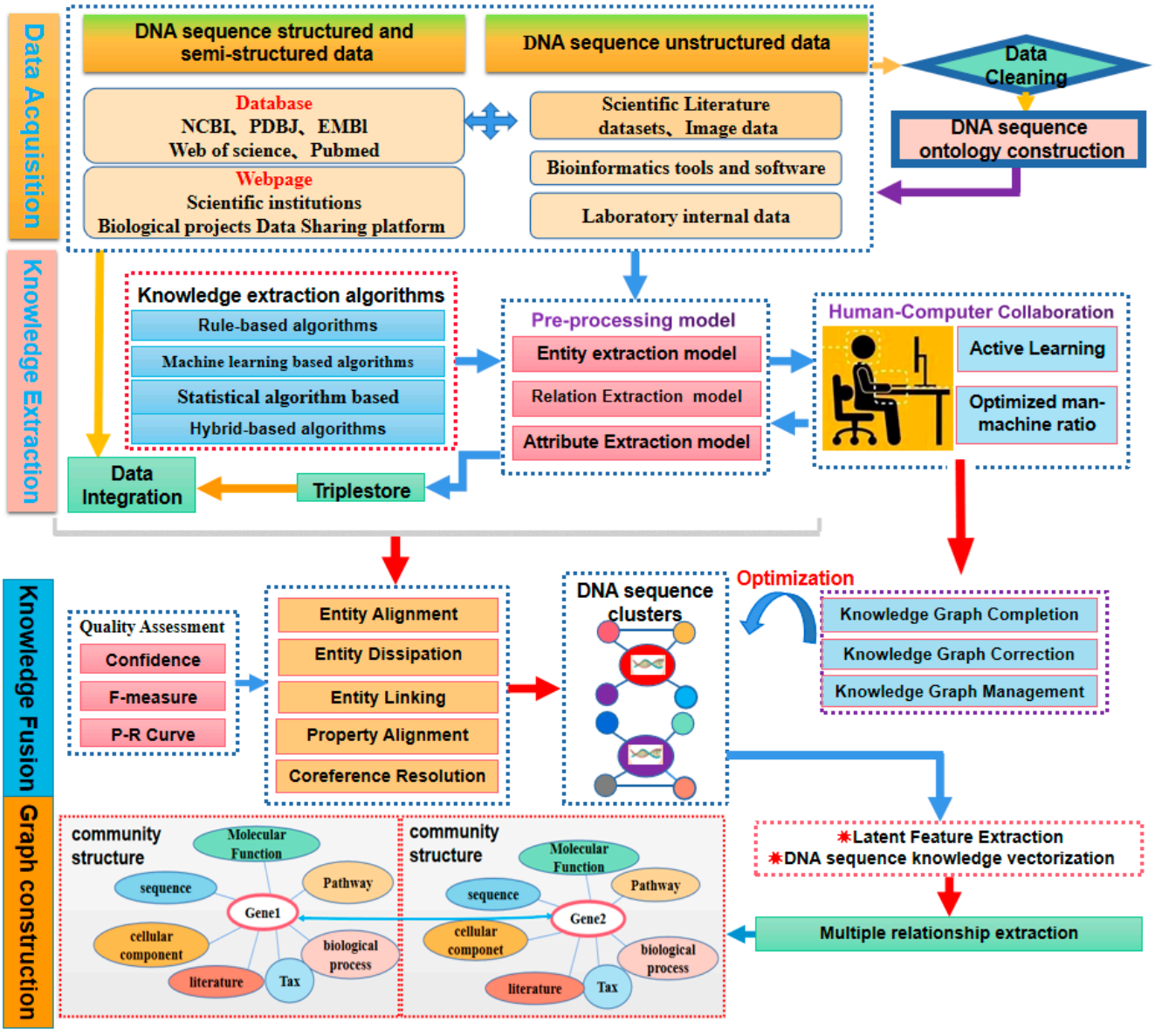

Figure 15. Framework diagram for building a knowledge graph of DNA sequences.

(2) Ontology Construction

To ensure the reliability of entity recognition results in biological sequence, it is necessary to construct a DNA sequence ontology library. It is recommended to adopt a combined approach of top-down and bottom-up methods to construct the DNA sequence ontology library [151,152]. The framework of ontology construction is illustrated in Figure 16.

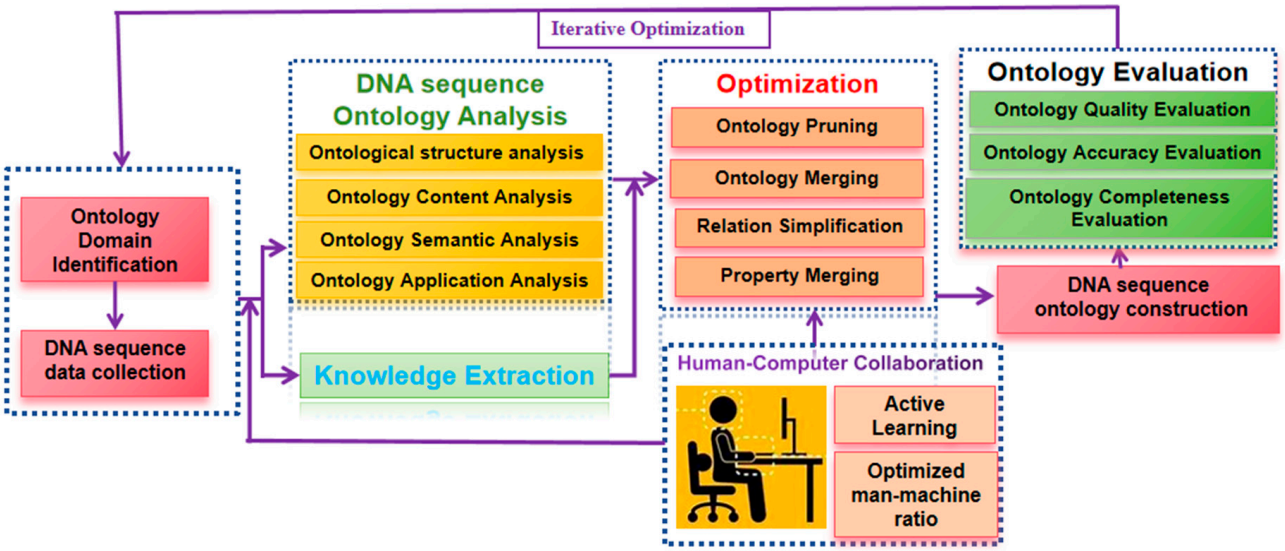

Figure 16. Framework for DNA sequence ontology construction.

Firstly, it is to determine the professional domain and application scope of the DNA sequence ontology. For the DNA sequence KG, the ontology is oriented towards the fields of DNA sequence function, structure, origin, organization, biological processes, and sequence variations. The scope of application involves the semantic description and inference capabilities of a DNA sequence. It is used to support applications such as data management, data mining, knowledge discovery, and knowledge inference in the fields of bioinformatics, genomics, biomedical research, and other related areas [153,154]. Secondly, DNA sequence data collection should be carried out based on the identified professional domains and application scope. Thirdly, a collaborative approach involving both human experts and machine analysis should be employed to extract the key terms for ontology from the data sources [155]. At the same time, the entities obtained in the subsequent knowledge extraction stage should also be included as key terms in the ontology [156–158]. Fourth, ontology optimization and simplification should be conducted, also using a collaborative approach involving both human expertise and computational methods [159–161]. Fifth, the DNA sequence ontology should be constructed and evaluated using three metrics. Sixth, the professional domain and application scope of the DNA sequence ontology should be iteratively optimized based on evaluation metrics, and then the above steps should be repeated until the requirements are met [162–164].

(3) Community structure fusion

Community structure can be seen as a network structure composed of different "clusters". Within each "cluster", the nodes are more densely connected, while the connections between nodes from different "clusters" are sparser. In the construction of a DNA sequence big data KG, genes with similar functions in different species may form distinct "clusters" [149]. For example, the first exon sequences of the human β-globin gene and the first exon sequences of the gorilla β-globin gene form two different "clusters". How to establish connections between these different "clusters" and form a community structure is an important question. Furthermore, the identification of existing sequence community structures, the acquisition and querying of community data, as well as the analysis of the overall network structure and functionality have important implications for gene function prediction, essential gene identification, and other related aspects [165,166]. However, traditional community detection algorithms typically only consider the structural char-

acteristics of the network, lacking the necessary considerations for semantic information within the KG.

(4) Knowledge fusion

Different biological knowledge bases use different entity identifier systems, which can lead to ambiguity when merging entities in a consolidated biological database [167]. When integrating heterogeneous data, typical problems include entity alignment, attribute alignment, entity disambiguation, entity linking, and co-reference resolution [168,169]. The knowledge fusion framework is illustrated in Figure 17.

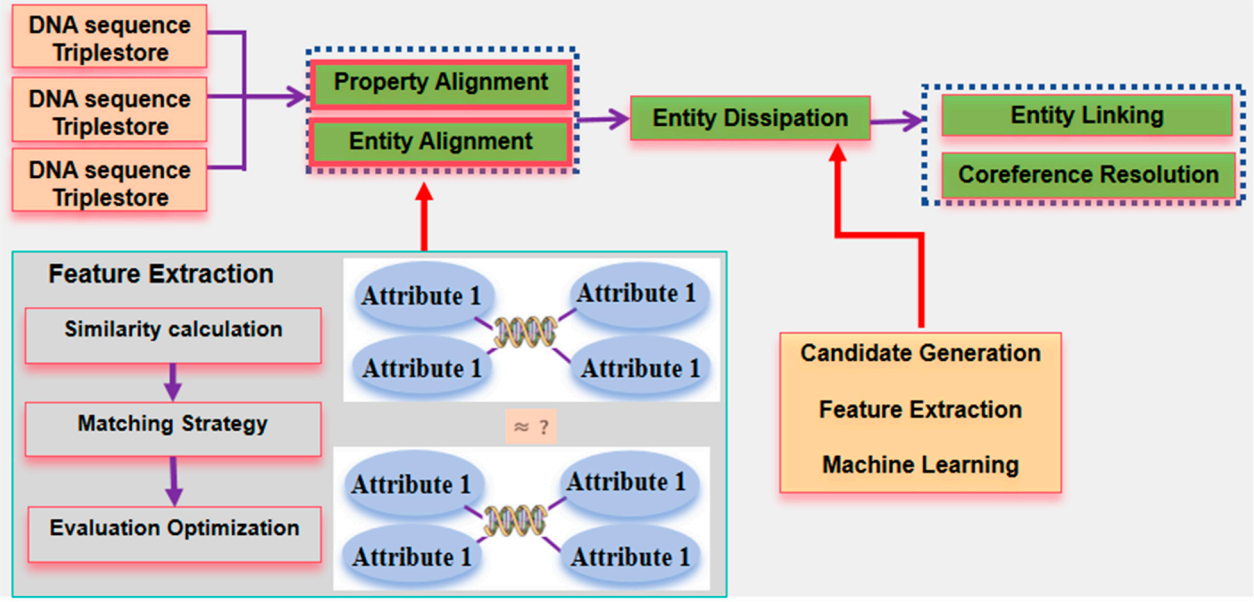

Figure 17. Knowledge fusion framework for DNA sequence.

DNA sequence entity alignment is a core step in knowledge fusion. First, extract entity data features from DNA sequence triplets, such as attribute values, labels, descriptions, relationships, etc., as input for entity alignment [170]. Based on the obtained entity features, calculate the similarity or distance between entities (entity attribute similarity measure). Common similarity calculation methods include cosine similarity, edit distance, Jaccard similarity, etc. Secondly, entity alignment is performed based on the calculated entity similarity [24]. Different matching strategies can be used, such as a threshold-based strategy, similarity ranking strategy, machine learning methods, etc. Finally, the entity alignment results are evaluated using metrics such as precision, recall, F1 score, etc. [171,172]. Based on the evaluation results, the entity alignment can be optimized and adjusted.

Entity disambiguation is performed after entity alignment and attribute alignment. First, generate a set of possible entity candidates. Secondly, extract features from the entity candidate set and contextual information [173]. Finally, use machine learning or rule-based methods to calculate the weights or probabilities of the entity candidate set based on the extracted features, and choose the most likely entity as the disambiguation result [174].

After entity disambiguation, entity linking and coreference resolution are performed. The process of entity linking mainly involves two steps: (1) candidate entity generation; (2) candidate entity ranking. After entity linking, co-reference resolution is performed to improve the accuracy of entity linking.

(5) Named Entity Recognition (NER)

Entities are an essential component in building KGs. The accuracy of entity recognition is crucial for building KGs [175,176]. The data sources for constructing a KG of

DNA sequence big data include structured, unstructured, and semi-structured data such as research papers, biological software and tools, and image data [177,178]. Therefore, performing entity recognition from these vast amounts of textual and image data is a crucial task for constructing a KG. The NER for a DNA sequence is illustrated in Figure 18.

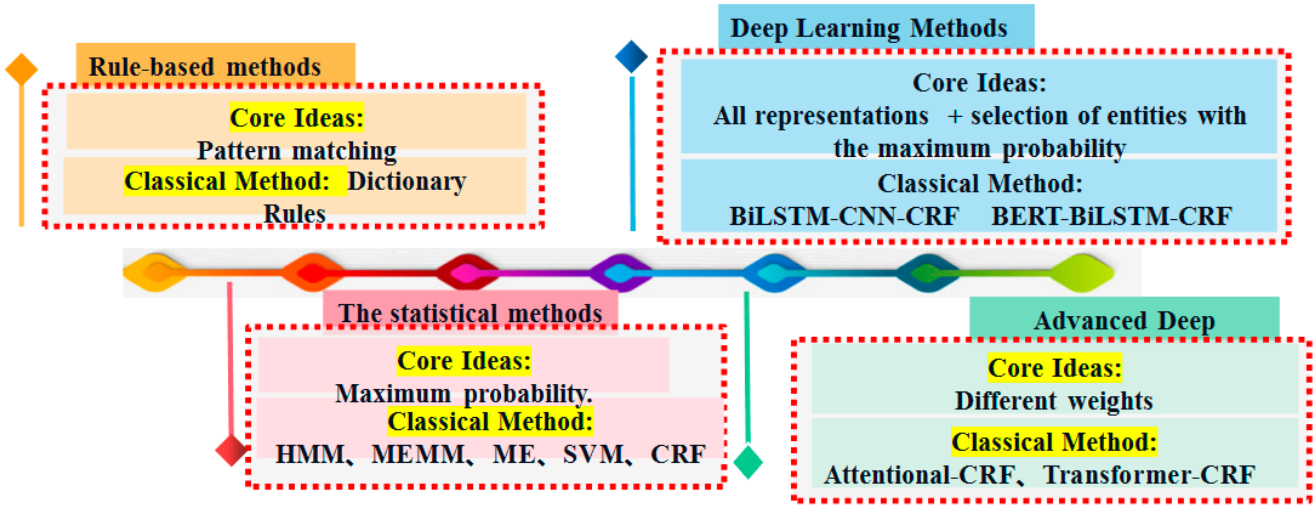

Figure 18. Named entity recognition methods for DNA sequence.

Currently, commonly used entity recognition methods include rule-based entity recognition, statistical machine learning-based entity recognition, and deep learning-based entity recognition, among others [179,180]. These methods utilize specific patterns, statistical models, or deep neural networks to extract entities from text or image data, thereby extracting the necessary entity information for constructing a KG [181].

6.3.2. Machine Learning-Based DNA Sequence Image Generation

There is a lack of unified standards for visualizing DNA sequences. Different methods are used to visualize different DNA sequences, and different features are selected for visualization representation using different methods [182–185]. For example, some methods focus on global features, while others focus on local features, resulting in generated graphics with different emphases, and may not provide sufficient semantic information to meet the requirements of customized views for biological sequences [186–188]. This limitation hinders the extension and application of knowledge. Therefore, building image generation models that transform DNA sequence textual descriptions into DNA sequence images that reflect the textual information is an important task.

The current mainstream generation models for addressing the text-to-image generation task include auto-regressive models, variational autoencoders, and Generative Adversarial Networks (GANs) [189–191]. An AM generates images by progressively generating pixels or local features of the image step by step. A VA generates new samples by learning the latent space of the data. The GAN has become the most popular method due to its powerful sample generation capability. It continuously improves the quality of generated images through adversarial training between the generator and discriminator. Therefore, there is a wide range of promotion and application value in generating DNA sequence images using advanced machine learning algorithms.

(1) DNA sequence image generation framework

As an opening for further discussion and exploration, this review combines the features of DNA sequences and leverages the advantages of machine learning algorithms to propose a research idea for constructing visual representations of DNA sequences using

machine learning methods [192]. The purpose of this idea is to provide researchers in interdisciplinary fields such as biology and computer science with a new perspective on handling and analyzing large-scale DNA sequence data [193,194]. This research perspective holds the potential to provide a new method for constructing high-quality image generation models in DNA sequence visualization. By leveraging machine learning techniques and considering the specific requirements of different application scenarios, it aims to address the need for customizable views of biological sequences [195,196]. The image generation framework and key techniques are illustrated in Figure 19.

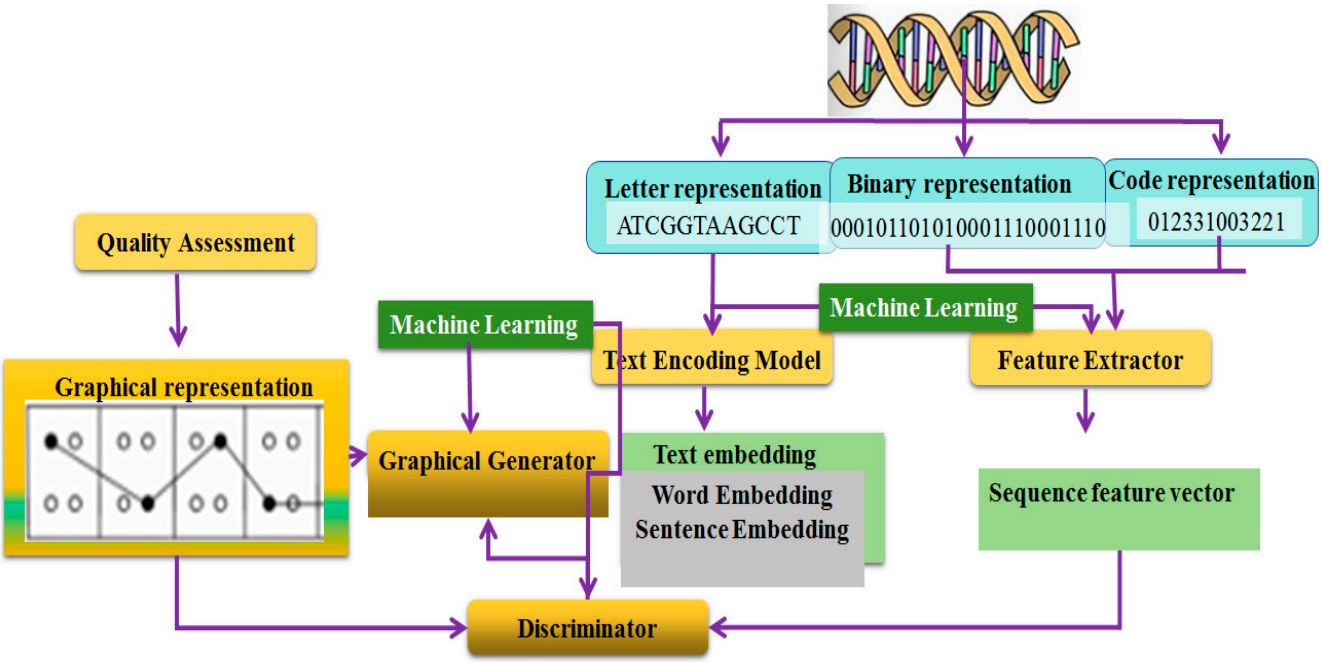

Figure 19. Machine learning-based framework for DNA sequence image generation.

First, the DNA sequence structure is represented through textual and numerical descriptions. Secondly, a machine learning algorithm is employed to construct a text encoding model, which generates text embedding vectors (such as word embeddings and sentence embeddings) through training [197]. Simultaneously, a machine learning model is used to construct a feature extractor for numerical descriptions, which extracts numerical features from the sequence. Third, a machine learning pre-trained image generator is employed to generate draft images using the text embedding vectors as input. Fourth, a machine learning algorithm is employed to pre-train the discriminator. The sequence's numerical feature is then associated with the generated draft image to provide fine-grained supervisory feedback, thereby facilitating the training of an effective image generator. Finally, the generated images are subjected to quality evaluation [198].

(2) Machine Learning Algorithms in DNA Sequence Image Generation

The textual description of DNA sequences is stored as a one-dimensional sequence, while DNA sequence images are stored in the form of two-dimensional structural information. Therefore, there is a significant difference between textual description and image representation in terms of feature representation and information expression [199,200]. This difference can potentially lead to issues such as the generated image having an unreasonable layout and being inconsistent with the textual description. Therefore, when selecting algorithms, it is necessary to consider the differences in feature representation and information expression between text and images and to choose suitable algorithms to generate DNA sequence images with a reasonable layout and accurate representation.

There are currently various machine learning algorithms available, but there is no single algorithm that is optimal for all tasks [201,202]. The main machine learning algorithms to generate DNA sequence images are depicted in Figure 20.

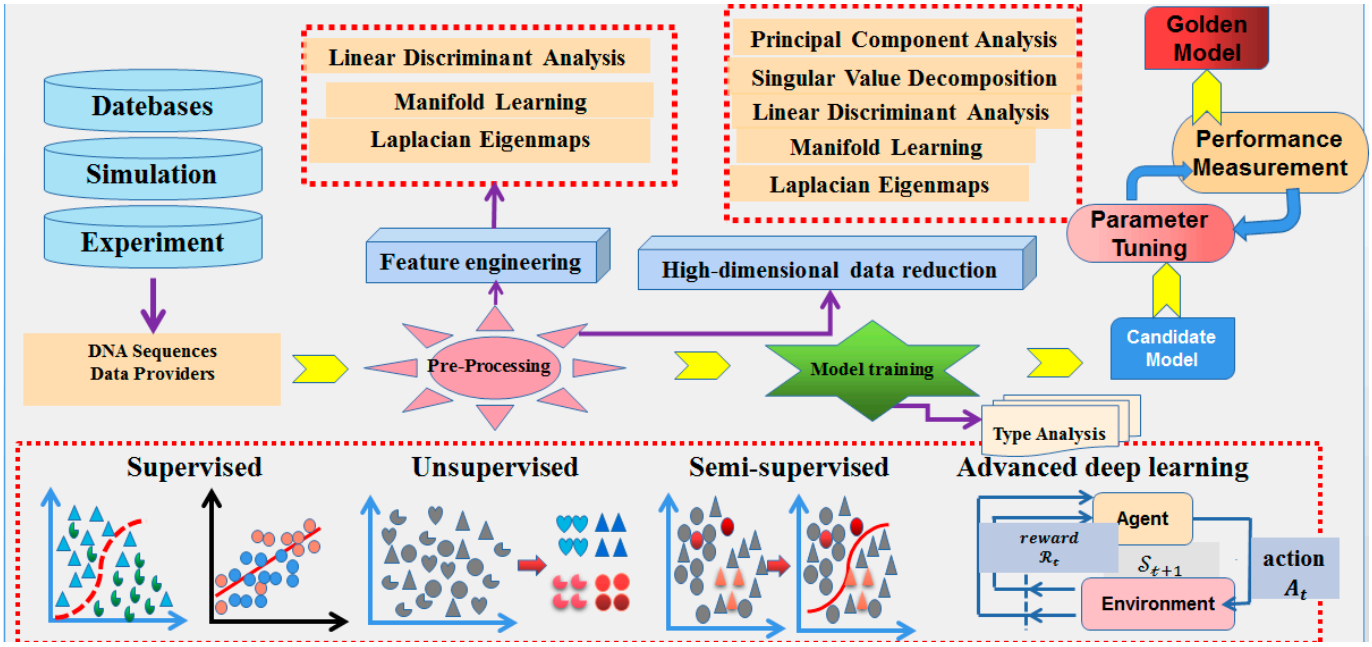

Figure 20. Machine learning algorithms for DNA sequence image generation.

The entire process of machine learning includes steps such as analyzing objectives, collecting data, organizing data, preprocessing data, training models, evaluating models, and optimizing models [203–205]. Depending on the type of data collected, different types of machine learning algorithms can be chosen for model training. Based on learning methods, the main types of machine learning algorithms are supervised learning, unsupervised learning, semi-supervised learning, and advanced deep learning [206–209].

Supervised learning: Training is performed using input data along with their corresponding labels or target values. The goal is to establish a mapping relationship between the input and the output [210,211]. By learning from known input–output data, the model can predict the output for new, unseen data. It mainly includes decision trees, support vector machines, Bayesian networks, neural networks, and ensemble learning methods such as random forests. Unsupervised learning: By learning the internal structure and relationships of the input data, it is possible to discover patterns, categories, or other structural information within the data [212]. Unlike supervised learning, unsupervised learning does not have explicit labels or target values to guide the learning process. It mainly includes clustering algorithms, principal component analysis (PCA), and association rule mining, among others. Semi-supervised learning: Semi-supervised learning algorithms are a learning approach that falls between supervised learning and unsupervised learning. It utilizes both labeled training data and unlabeled training data to learn, aiming to improve learning performance and generalization by leveraging information from both labeled and unlabeled data [213]. Semi-supervised learning algorithms are often applied in situations where labeled data are difficult to obtain. Advanced deep learning: Advanced deep learning refers to more complex and advanced techniques and methods within the field of deep learning. It builds upon the foundation of basic deep learning models and algorithms. It involves the use of deeper and more complex neural network architectures, as well as more advanced training techniques, to tackle more challenging and intricate problems. Some commonly used advanced deep learning methods include Convolutional Neural Networks

(CNNs), Recurrent Neural Networks (RNNs), Advanced Recurrent Neural Networks, Deep Belief Networks (DBNs), Generative Adversarial Networks, Reinforcement Learning (RL) [214,215], and so on. These different types of machine learning algorithms are suitable for various problems and scenarios. Choosing the appropriate learning algorithm based on specific task requirements and data characteristics can help achieve better learning and prediction performance.

7. Summary

This article presents a comprehensive review of DNA sequence visualization methods based on two-dimensional, three-dimensional, four-dimensional, and dynamic visualization. Additionally, this article discusses and analyzes the strengths and limitations of each method. Furthermore, this article introduces the applications of visualization methods in sequence similarity measurement, gene mutation analysis, and homology analysis. It analyzes the advantages and limitations of these applications. In addition, this article proposes two potential research directions: the construction of a KG for DNA sequence big data and the application of machine learning methods for DNA sequence image generation. Visual methods play a crucial role in DNA sequence analysis, but they also have some limitations. For example, there are limitations in terms of visualization effectiveness, data capacity for visualization, and high algorithmic complexity. Future research directions include improving visualization effectiveness, enhancing the processing capabilities of visualization techniques, and increasing the interactivity of data visualization. These advancements aim to better meet the requirements of DNA sequence analysis. This article contributes to the comparison and evaluation of research from different disciplines, promoting the improvement and development of visualization methods for biological sequences.

Author Contributions: Y.W. and L.G. designed the study; Y.W. and M.L. performed the research; X.X. contributed to the conception of the study; M.L. and L.G. contributed significantly to analysis and manuscript preparation; J.Z. and M.L. provided and analyzed the data. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Numbers: 51663001, 52063002, 42061067) and the Degree and postgraduate education and teaching reform research project of Jiangxi province (Multi thinking and "439" scientific research ability training mode for interdisciplinary academic degree postgraduate training).

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviation

Abbreviation	Detail
Attentional-CRF	Attentional CRF
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
BN	Bayesian Network
CGR	Chaos Game visualization
CMI	Coded Mark Inversion
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DBN	Deep Belief Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNA SV	DNA sequence visualization

DN-curve	Double Nucleotide Curve
DT	Decision Tree
DV-Curve	Double Vector
EA	Evolutionary Algorithm
GANs	Generative Adversarial Networks
HMM	Hidden Markov Model
KG	Knowledge Graph
LSR	Letter Sequence Representation
LSTM	Long Short-Term Memory
ME	Maximum Entropy
MEMM	Maximum Entropy Markov Model
NER	Named Entity Recognition
NN	Neural Network
PCA	Principal Component Analysis
RL	Reinforcement Learning
RNNs	Recurrent Neuron Networks
SVM	Support Vector Machine
Transformer-CRF	Transformer with Conditional Random Field
δ_1	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGT
	GGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAG
δ_2	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGT
	GGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
δ_3	ATGACTTTGCTGAGTGCTGAGGAGAATGCTCATGTCACCTCTCT
	GTGGGGCAAGGTGGATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG
δ_4	ATGGTGCACCTGACTGATGCTGAGAAGTCTGCTGTCTCTTGCCTGTGGG
	CAAAGGTGAACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
δ_5	ATGGTGCACCTAACTGATGCTGAGAAGGCTACTGTTAGTGGCCTG
	TGGGGAAAGGTGAACCCTGATAATGTTGGCGCTGAGGCCCTGGGCAG
δ_6	ATGCTGACTGCTGAGGAGAAGGCTGCCGTCACCGGCTTCTGGGGC
	AAGGTGAAAGTGGATGAAGTTGGTGCTGAGGCCCTGGGCAG
δ_7	ATGGTGCACTTGACTTCTGAGGAGAAGAACTGCATCACTACCATC
	TGGTCTAAGGTGCAGGTTGACCAGACTGGTGGTGAGGCCCTTGGCAG
ρ_1	ACATTTGCTTCTGACACAACTGTGTTCACTAGCACCTCAAACAGACA
	CCATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCG
	TGGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGC
	AGGCTGCTGGTGGTCTACCCTTGGACCCAGAGGTTCTTTGAGTCC
	TTTGGGGATCTGTCCACTCCTGATGCTGTTATGGGCAACCCTAA
	GGTGAAGGCTCATGGCAAGAAAGTGCTCGGTGCCTTTAGTGAT
	GGCCTGGCTCACTGGACAACCTCAAGGGCACCTTTGCCACACTG
	AGTGAGCTGCACTGTGCAAGCTGCACGTGGATCCTGAGAACTTCAAGC
	TCCTGGGCAATGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGCAAAG
ρ_2	ATGGTGCACTTGACTTCTGAGGAGAAGAACTGCATCACTACCATC
	TGGTCTAAGGTGCAGGTTGACCAGACTGGTGGTGAGGCCCTTGGC
	AGGATGCTCGTTGTCTACCCCTGGACCACCAGGTTTTTTGGGAGCT
	TTGGTGATCTGTCCTCTCCTGGCGCTGTCATGTCAAATTCTAAGGTT
	CAAGCCCATGGTGCTAAGGTGTTGACCTCCTTCGGTGAAGCAGTCAA
	GCATTTGGACAACCTGAAGGGTACTTATGCCAAGTTGAGTGAGCTCC
	ACTGTGACAAGCTGCATGTGGACCCTGAGAACTTCAAGATGCTGGG
	GAATATCATTGTGATCTGCCTGGCTGAGCACTTTGGCAAGGATTTT
	ACTCCTGAATGTCAGGTTGCTTGGCAGAAGCTCGTGGCTGGAG
	TTGCCCATGCCCTGGCCCACAAGTACCACTAA
φ	CTAACTTTCCTCATTGCTAACCAGCATTCTCATGTCACCTCTCTG
	TCCTTCAAGCTTCATGTATATAAACTTTCTTTCCAGCCCTTGGGCAT

$\mathcal{S}$	ATGGTGCACC
$\mathcal{S}'$	ATGGTGCACC
$\mathcal{S}''$	ATGCTGACTG
γ	ATGGTGCACCTGACTCCTGA
$\mathcal{L}'$	ATGATGCACC
$\mathcal{H}$	TGTGCTGAGTGAGGTCAGAAACTT
$\mathcal{J}$	ATGACTTTGCTGAGT

References

1. Pique-Regi, R.; Degner, J.F.; Pai, A.A.; Gaffney, D.J.; Gilad, Y.; Pritchard, J.K. Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. *Genome Res.* **2011**, *21*, 447–455. [CrossRef] [PubMed]
2. Zhou, J.X.; Zhang, B.; Li, H.Y.; Zhou, L.X.; Li, Z.X.; Long, Y.K.; Han, W.K.; Wang, M.R.; Cui, H.H.; Li, J.J.; et al. Annotating TSSs in Multiple Cell Types Based on DNA Sequence and RNA-seq Data via DeeReCT-TSS. *Genom. Proteom. Bioinform.* **2022**, *20*, 959–973. [CrossRef]
3. Zullo, J.M.; Demarco, I.A.; Pique-Regi, R.; Gaffney, D.J.; Epstein, C.B.; Spooner, C.J.; Luperchio, T.R.; Bernstein, B.E.; Pritchard, J.K.; Reddy, K.L.; et al. DNA Sequence-Dependent Compartmentalization and Silencing of Chromatin at the Nuclear Lamina. *Cell* **2012**, *149*, 1474–1487. [CrossRef]
4. Ao, C.Y.; Jiao, S.H.; Wang, Y.S.; Yu, L.; Zou, Q. Biological Sequence Classification: A Review on Data and General Methods. *Research* **2022**, *2022*, 11. [CrossRef] [PubMed]
5. Butz, S.; Schmolka, N.; Karemaker, I.D.; Villasenor, R.; Schwarz, I.; Domcke, S.; Uijttewaal, E.C.H.; Jude, J.; Lienert, F.; Krebs, A.R.; et al. DNA sequence and chromatin modifiers cooperate to confer epigenetic bistability at imprinting control regions. *Nat. Genet.* **2022**, *54*, 1702. [CrossRef]
6. Osama, S.; Shaban, H.; Ali, A.A. Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review. *Expert Syst. Appl.* **2023**, *213*, 118946. [CrossRef]
7. Petti, S.; Eddy, S.R. Constructing Benchmark Test Sets for Biological Sequence Analysis Using Independent Set Algorithms. *PLoS Comput. Biol.* **2023**, *19*, e1010971. [CrossRef] [PubMed]
8. Xie, Y.B.; Li, H.Q.; Luo, X.T.; Li, H.Y.; Gao, Q.Y.; Zhang, L.W.Y.; Teng, Y.Y.; Zhao, Q.; Zuo, Z.X.; Ren, J. IBS 2.0: An upgraded illustrator for the visualization of biological sequences. *Nucleic Acids Res.* **2022**, *50*, W420–W426. [CrossRef] [PubMed]
9. Cangialosi, A.; Yoon, C.; Liu, J.; Huang, Q.; Guo, J.K.; Nguyen, T.D.; Gracias, D.H.; Schulman, R. DNA sequence-directed shape change of photopatterned hydrogels via high-degree swelling. *Science* **2017**, *357*, 1126–1129. [CrossRef] [PubMed]
10. Blatnik, M.C.; Gallagher, T.L.; Amacher, S.L. Keeping development on time: Insights into post-transcriptional mechanisms driving oscillatory gene expression during vertebrate segmentation. *Wiley Interdiscip. Rev.-RNA* **2023**, *14*, e1751. [CrossRef] [PubMed]
11. Allan, M.F.; Brivanlou, A.; Rouskin, S. RNA levers and switches controlling viral gene expression. *Trends Biochem. Sci.* **2023**, *48*, 391–406. [CrossRef] [PubMed]
12. Nandy, A.; Basak, S.C.; Gute, B.D. Graphical representation and numerical characterization of H5N1 avian flu neuraminidase gene sequence. *J. Chem. Inf. Model.* **2007**, *47*, 945–951. [CrossRef] [PubMed]
13. Miska, E.A.; Ferguson-Smith, A.C. Transgenerational inheritance: Models and mechanisms of non-DNA sequence-based inheritance. *Science* **2016**, *354*, 59–63. [CrossRef] [PubMed]
14. Li, M.Y.; Wang, I.X.; Li, Y.; Bruzel, A.; Richards, A.L.; Toung, J.M.; Cheung, V.G. Widespread RNA and DNA Sequence Differences in the Human Transcriptome. *Science* **2011**, *333*, 53–58. [CrossRef] [PubMed]
15. Rozas, J.; Ferrer-Mata, A.; Sanchez-DelBarrio, J.C.; Guirao-Rico, S.; Librado, P.; Ramos-Onsins, S.E.; Sanchez-Gracia, A. DnaSP 6: DNA Sequence Polymorphism Analysis of Large Data Sets. *Mol. Biol. Evol.* **2017**, *34*, 3299–3302. [CrossRef]
16. Jing, R.Y.; Xue, L.; Li, M.L.; Yu, L.Z.; Luo, J.S. layerUMAP: A tool for visualizing and understanding deep learning models in biological sequence classification using UMAP. *Iscience* **2022**, *25*, 105530. [CrossRef] [PubMed]
17. Lee, D.; Karchin, R.; Beer, M.A. Discriminative prediction of mammalian enhancers from DNA sequence. *Genome Res.* **2011**, *21*, 2167–2180. [CrossRef] [PubMed]
18. Qi, Z.; Redding, S.; Lee, J.Y.; Gibb, B.; Kwon, Y.; Niu, H.Y.; Gaines, W.A.; Sung, P.; Greene, E.C. DNA Sequence Alignment by Microhomology Sampling during Homologous Recombination. *Cell* **2015**, *160*, 856–869. [CrossRef] [PubMed]
19. Liu, L.Z.; Wu, F.X.; Zhang, W.J. Reverse engineering of gene regulatory networks from biological data. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2012**, *2*, 365–385. [CrossRef]
20. Kaur, A.; Chauhan, A.P.S.; Aggarwal, A.K. Prediction of Enhancers in DNA Sequence Data using a Hybrid CNN-DLSTM Model. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **2023**, *20*, 1327–1336. [CrossRef]

21. Cui, X.; Lu, F.L.; Qiu, Q.; Zhou, B.; Gu, L.F.; Zhang, S.B.; Kang, Y.Y.; Cui, X.K.; Ma, X.; Yao, Q.Q.; et al. REF6 recognizes a specific DNA sequence to demethylate H3K27me3 and regulate organ boundary formation in Arabidopsis. *Nat. Genet.* **2016**, *48*, 694–699. [CrossRef]
22. Bainbridge, M.N.; Wang, M.; Wu, Y.Q.; Newsham, I.; Muzny, D.M.; Jefferies, J.L.; Albert, T.J.; Burgess, D.L.; Gibbs, R.A. Targeted enrichment beyond the consensus coding DNA sequence exome reveals exons with higher variant densities. *Genome Biol.* **2011**, *12*, R68. [CrossRef] [PubMed]
23. Naresh, V.S.; Thamarai, M. Privacy-preserving data mining and machine learning in healthcare: Applications, challenges, and solutions. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2023**, *13*, e1490. [CrossRef]
24. Guo, Q.Y.; Zhuang, F.Z.; Qin, C.; Zhu, H.S.; Xie, X.; Xiong, H.; He, Q. A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3549–3568. [CrossRef]
25. Wani, M.A.; Sultan, B. Deep learning based image steganography: A review. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2022**, *13*, e1481. [CrossRef]
26. Chakraborty, N.; Lukovnikov, D.; Maheshwari, G.; Trivedi, P.; Lehmann, J.; Fischer, A. Introduction to neural network-based question answering over knowledge graphs. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2021**, *11*, e1389. [CrossRef]
27. Marcinkevics, R.; Vogt, J.E. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2023**, *13*, e1493. [CrossRef]
28. Gates, M.A. A simple way to look at DNA. *J. Theor. Biol.* **1986**, *119*, 319–328. [CrossRef]
29. Cui, F.; Zhang, Z.; Zou, Q. Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. *Brief. Funct. Genom.* **2021**, *20*, 61–73. [CrossRef] [PubMed]
30. Li, C.C.; Dai, Q.; He, P.A. A time series representation of protein sequences for similarity comparison. *J. Theor. Biol.* **2022**, *538*, 111039. [CrossRef] [PubMed]
31. Das, S.; Das, A.; Bhattacharya, D.K.; Tibarewala, D.N. A new graph-theoretic approach to determine the similarity of genome sequences based on nucleotide triplets. *Genomics* **2020**, *112*, 4701–4714. [CrossRef] [PubMed]
32. Mu, Z.C.; Yu, T.; Liu, X.P.; Zheng, H.Y.; Wei, L.Y.; Liu, J.T. FEGS: A novel feature extraction model for protein sequences and its applications. *BMC Bioinform.* **2021**, *22*, 297. [CrossRef] [PubMed]
33. Liu, Y.; Peng, Y. A Novel 2-D Graphical Representation for DNA Sequences Based on the Chemical Properties. *J. Comput. Theor. Nanosci.* **2013**, *10*, 2102–2105. [CrossRef]
34. Mu, Z.C.; Yu, T.; Qi, E.F.; Liu, J.T.; Li, G.J. DCGR: Feature extractions from protein sequences based on CGR via remodeling multiple information. *BMC Bioinform.* **2019**, *20*, 351. [CrossRef]
35. Zheng, K.; You, Z.H.; Li, J.Q.; Wang, L.; Guo, Z.H.; Huang, Y.A. iCDA-CGR: Identification of circRNA-disease associations based on Chaos Game Representation. *PLoS Comput. Biol.* **2020**, *16*, e1007872. [CrossRef]
36. Lochel, H.F.; Heider, D. Chaos game representation and its applications in bioinformatics. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 6263–6271. [CrossRef]
37. Gupta, M.K.; Niyogi, R.; Misra, M. A new adjacent pair 2d graphical representation of DNA sequences. *J. Biol. Syst.* **2013**, *21*, 1350005. [CrossRef]
38. Itaman, S.; Enikolopov, G.; Podgorny, O.V. Detection of De Novo Dividing Stem Cells In Situ through Double Nucleotide Analogue Labeling. *Cells* **2022**, *11*, 4001. [CrossRef] [PubMed]
39. Xie, G.S.; Jin, X.B.; Yang, C.L.; Pu, J.X.; Mo, Z.X. Graphical Representation and Similarity Analysis of DNA Sequences Based on Trigonometric Functions. *Acta Biotheor.* **2018**, *66*, 113–133. [CrossRef] [PubMed]
40. Chen, Z.; Liu, Z.J.; Sheng, Y.H. On Double Vector Bundles. *Acta Math. Sin.* **2014**, *30*, 1655–1673. [CrossRef]
41. Meinrenken, E. Quotients of double vector bundles and multigraded bundles. *J. Geom. Mech.* **2022**, *14*, 307–329. [CrossRef]
42. Waz, P.; Bielinska-Waz, D. Non-standard similarity/dissimilarity analysis of DNA sequences. *Genomics* **2014**, *104*, 464–471. [CrossRef] [PubMed]
43. Quan, L.J.; Sun, X.Y.; Wu, J.; Mei, J.; Huang, L.Q.; He, R.J.; Nie, L.P.; Chen, Y.; Lyu, Q. Learning Useful Representations of DNA Sequences From ChIP-Seq Datasets for Exploring Transcription Factor Binding Specificities. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 998–1008. [CrossRef]
44. Shu, H.; Zhou, J.; Lian, Q.; Li, H.; Zhao, D.; Zeng, J.; Ma, J. Modeling gene regulatory networks using neural network architectures. *Nat. Comput. Sci.* **2021**, *1*, 491–501. [CrossRef] [PubMed]
45. Kania, A.; Sarapata, K. Multifarious aspects of the chaos game representation and its applications in biological sequence analysis. *Comput. Biol. Med.* **2022**, *151*, 106243. [CrossRef]
46. Chen, X.L.; Xie, H.R.; Li, Z.X.; Cheng, G. Topic analysis and development in knowledge graph research: A bibliometric review on three decades. *Neurocomputing* **2021**, *461*, 497–515. [CrossRef]
47. Rosnik, A.M.; Curutchet, C. Theoretical Characterization of the Spectral Density of the Water-Soluble Chlorophyll-Binding Protein from Combined Quantum Mechanics/Molecular Mechanics Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2015**, *11*, 5826–5837. [CrossRef] [PubMed]

48. Liu, H.L. A Joint Probabilistic Model in DNA Sequences. *Curr. Bioinform.* **2018**, *13*, 234–240. [CrossRef]
49. Wu, R.X.; Liu, W.J.; Mao, Y.Y.; Zheng, J. 2D Graphical Representation of DNA Sequences Based on Variant Map. *IEEE Access* **2020**, *8*, 173755–173765. [CrossRef]
50. Zhang, Z.J.; Zeng, X.X.; Song, T.; Chen, Z.H.; Wang, X.; Ye, Y.M. WormStep: An Improved Compact Graphical Representation of DNA Sequences Based on Worm Curve. *J. Comput. Theor. Nanosci.* **2013**, *10*, 189–193. [CrossRef]
51. Zhang, R.; Zhang, C.T. Z curves, an intutive tool for visualizing and analyzing the DNA sequences. *J. Biomol. Struct. Dyn.* **1994**, *11*, 767–782. [CrossRef] [PubMed]
52. Yan, M.; Lin, Z.S.; Zhang, C.T. A new fourier transform approach for protein coding measure based on the format of the Z curve. *Bioinform. (Oxf. Engl.)* **1998**, *14*, 685–690. [CrossRef]
53. Zhang, C.-T.; Zhang, R.; Ou, H.-Y. The Z curve database: A graphic representation of genome sequences. *Bioinformatics* **2003**, *19*, 593–599. [CrossRef] [PubMed]
54. Zheng, W.X.; Wang, S.X.; Liu, H. Highly Accurate Gene Essentiality Prediction with W-Nucleotide Z Curve Features and Feature Selection Technique in Saccharomyces cerevisiae. *Curr. Bioinform.* **2021**, *16*, 1081–1088. [CrossRef]
55. Zhang, D.; Wang, Y.; Liu, Z.; Dai, S. Improving NoSQL Storage Schema Based on Z-Curve for Spatial Vector Data. *IEEE Access* **2019**, *7*, 78817–78829. [CrossRef]
56. Wang, S.C.; Huang, S.X.; Lu, S.; Yan, B. 3-D Ionospheric Tomography Using Model Function in the Modified L-Curve Method. *IEEE Trans. Geosci. Remote* **2019**, *57*, 3135–3147. [CrossRef]
57. Liu, J.; Xie, J.S.; Li, B.; Hu, B.B. Regularized Cubic B-Spline Collocation Method With Modified L-Curve Criterion for Impact Force Identification. *IEEE Access* **2020**, *8*, 36337–36349. [CrossRef]
58. Paul, T.; Chi, P.W.; Wu, P.M.; Wu, M.K. Computation of distribution of relaxation times by Tikhonov regularization for Li ion batteries: Usage of L-curve method. *Sci. Rep.* **2021**, *11*, 12624. [CrossRef]
59. Vujovic, M.; Marcatili, P.; Chain, B.; Kaplinsky, J.; Andresen, T.L. Signatures of T cell immunity revealed using sequence similarity with TCRDivER algorithm. *Commun. Biol.* **2023**, *6*, 357. [CrossRef] [PubMed]
60. Yu, H.J.; Huang, D.S. Graphical Representation for DNA Sequences via Joint Diagonalization of Matrix Pencil. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 503–511. [CrossRef]
61. Peng, Y.; Liu, Y.W. An Improved Mathematical Object for Graphical Representation of DNA Sequences. *Curr. Bioinform.* **2015**, *10*, 332–336. [CrossRef]
62. Liu, X.Q.; Dai, Q.; Xiu, Z.; Wang, T. Pnn-curve: A new 2d graphical representation of DNA sequences and its application. *J Theor Biol* **2006**, *243*, 555–561. [CrossRef] [PubMed]
63. Peng, Y.; Liu, Y.W. A Novel Numerical Characterization for Graphical Representations of DNA Sequences. *Mini-Rev. Org. Chem.* **2015**, *12*, 534–539. [CrossRef]
64. Jafarzadeh, N.; Iranmanesh, A. C-curve: A novel 3D graphical representation of DNA sequence based on codons. *Math. Biosci.* **2013**, *241*, 217–224. [CrossRef]
65. Li, Y.S.; Liu, Q.; Zheng, X.Q. DUC-Curve, a highly compact 2D graphical representation of DNA sequences and its application in sequence alignment. *Phys. A-Stat. Mech. Its Appl.* **2016**, *456*, 256–270. [CrossRef]
66. Qin, X.Y.; Zhang, L.; Liu, M.; Xu, Z.W.; Liu, G.Z. ASFold-DNN: Protein Fold Recognition Based on Evolutionary Features With Variable Parameters Using Full Connected Neural Network. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 2712–2722. [CrossRef] [PubMed]
67. Bielinska-Waz, D.; Waz, P. Spectral-dynamic representation of DNA sequences. *J. Biomed. Inform.* **2017**, *72*, 1–7. [CrossRef]
68. Li, X.L.; Zhou, R.Q.; Xu, Y.F.; Wei, X.; He, Y. Spectral unmixing combined with Raman imaging, a preferable analytic technique for molecule visualization. *Appl. Spectrosc. Rev.* **2017**, *52*, 417–438. [CrossRef]
69. Vracko, M.; Basak, S.C.; Dey, T.; Nandy, A. Cluster Analysis of Coronavirus Sequences using Computational Sequence Descriptors: With Applications to SARS, MERS and SARS-CoV-2 (COVID-19). *Curr. Comput-Aided Drug Des.* **2021**, *17*, 936–945. [CrossRef]
70. Paul, T.; Vainio, S.; Roning, J. Detection of intra-family coronavirus genome sequences through graphical representation and artificial neural network. *Expert Syst. Appl.* **2022**, *194*, 116559. [CrossRef]
71. Bielinska-Waz, D.; Waz, P. Non-standard bioinformatics characterization of SARS-CoV-2. *Comput. Biol. Med.* **2021**, *131*, 104247. [CrossRef]
72. Bielinska-Waz, D.; Waz, P.; Nandy, A. Graphical Representations of Biological Sequences. *Comb. Chem. High Throughput Screen.* **2022**, *25*, 347–348. [CrossRef] [PubMed]
73. Martin, H.; Sébastien, O.; Christos, B.; Tom, V. Deep homography estimation in dynamic surgical scenes for laparoscopic camera motion extraction. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2022**, *10*, 321–329.
74. Yu, J.F.; Dou, X.H.; Wang, H.B.; Sun, X.; Zhao, H.Y.; Wang, J.H. A Novel Cylindrical Representation for Characterizing Intrinsic, Properties of Protein Sequences. *J. Chem. Inf. Model.* **2015**, *55*, 1261–1270. [CrossRef] [PubMed]
75. Lichtblau, D. Alignment-free genomic sequence comparison using FCGR and signal processing. *BMC Bioinform.* **2019**, *20*, 742. [CrossRef]

76. Ni, H.M.; Qi, D.W.; Mu, H.B. Applying MSSIM combined chaos game representation to genome sequences analysis. *Genomics* **2018**, *110*, 180–190. [CrossRef]
77. Haghighatlari, M.; Vishwakarma, G.; Altarawy, D.; Subramanian, R.; Kota, B.U.; Sonpal, A.; Setlur, S.; Hachmann, J. ChemML: A machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data. *Wiley Interdiscip. Rev.-Comput. Mol. Sci.* **2020**, *10*, e1458. [CrossRef]
78. Kumar, M.R.; Vaegae, N.K. A new numerical approach for DNA representation using modified Gabor wavelet transform for the identification of protein coding regions. *Biocybern. Biomed. Eng.* **2020**, *40*, 836–848. [CrossRef]
79. Dey, S.; Das, S.; Bhattacharya, D.K. Biochemical Property Based Positional Matrix: A New Approach Towards Genome Sequence Comparison. *J. Mol. Evol.* **2023**, *91*, 93–131. [CrossRef]
80. Das, S.; Das, A.; Mondal, B.; Dey, N.; Bhattacharya, D.K.; Tibarewala, D.N. Genome sequence comparison under a new form of tri-nucleotide representation based on bio-chemical properties of nucleotides. *Gene* **2020**, *730*, 144257. [CrossRef] [PubMed]
81. Munteanu, C.R.; Aguiar-Pulido, V.; Freire, A.; Martinez-Romero, M.; Porto-Pazos, A.B.; Pereira, J.; Dorado, J. Graph-Based Processing of Macromolecular Information. *Curr. Bioinform.* **2015**, *10*, 606–631. [CrossRef]
82. Sun, Z.J.; Pei, S.J.; He, R.L.; Yau, S.S.T. A novel numerical representation for proteins: Three-dimensional Chaos Game Representation and its Extended Natural Vector. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1904–1913. [CrossRef]
83. Xiao, N.; Cao, D.S.; Zhu, M.F.; Xu, Q.S. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics* **2015**, *31*, 1857–1859. [CrossRef] [PubMed]
84. Sun, N.; Zhao, X.; Yau, S.S.T. An efficient numerical representation of genome sequence: Natural vector with covariance component. *Peerj* **2022**, *10*, e13544. [CrossRef] [PubMed]
85. Bai, F.; Wang, T. On graphical and numerical representation of protein sequences. *J. Biomol. Struct. Dyn.* **2006**, *23*, 537–546. [CrossRef]
86. Randic, M.; Vracko, M.; Nandy, A.; Basak, S.C. On 3-D graphical representation of DNA primary sequences and their numerical characterization. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1235–1244. [CrossRef] [PubMed]
87. Ma, R.M.; Liu, Z.Y.; Zhang, Q.W.; Luo, T.F. Evaluating Polymer Representations via Quantifying Structure-Property Relationships. *J. Chem. Inf. Model.* **2019**, *59*, 3110–3119. [CrossRef] [PubMed]
88. Shankar, A.; Jagota, A.; Mittal, J. DNA Base Dimers Are Stabilized by Hydrogen-Bonding Interactions Including Non-Watson-Crick Pairing Near Graphite Surfaces. *J. Phys. Chem. B* **2012**, *116*, 12088–12094. [CrossRef] [PubMed]
89. Zhou, J.; Sears, R.L.; Xing, X.Y.; Zhang, B.; Li, D.F.; Rockweiler, N.B.; Jang, H.S.; Choudhary, M.N.K.; Lee, H.J.; Lowdon, R.F.; et al. Tissue-specific DNA methylation is conserved across human, mouse, and rat, and driven by primary sequence conservation. *BMC Genom.* **2017**, *18*, 724. [CrossRef]
90. Xie, G.S.; Mo, Z.X. Three 3D graphical representations of DNA primary sequences based on the classifications of DNA bases and their applications. *J. Theor. Biol.* **2011**, *269*, 123–130. [CrossRef]
91. Qui, Y.H.; Yu, H.; Gong, X.J.; Xu, J.H.; Lee, H.S. On the prediction of DNA-binding proteins only from primary sequences: A deep learning approach. *PLoS ONE* **2017**, *12*, e0188129. [CrossRef]
92. Lee, B.D. Squiggle: A user-friendly two-dimensional DNA sequence visualization tool. *Bioinformatics* **2019**, *35*, 1425–1426. [CrossRef] [PubMed]
93. Zheng, L.; Liu, D.Y.; Yang, W.; Yang, L.; Zuo, Y.C. RaacLogo: A new sequence logo generator by using reduced amino acid clusters. *Brief. Bioinform.* **2021**, *22*, bbaa096. [CrossRef] [PubMed]
94. Kim, J.; Hong, S.P.; Lee, S.; Lee, W.; Lee, D.; Kim, R.; Park, Y.J.; Moon, S.; Park, K.; Cha, B.; et al. Multidimensional fragmentomic profiling of cell-free DNA released from patient-derived organoids. *Hum. Genom.* **2023**, *17*, 96. [CrossRef]
95. Zhang, Y.K.; Lang, M.; Jiang, J.H.; Gao, Z.Q.; Xu, F.; Litfin, T.; Chen, K.; Singh, J.; Huang, X.S.; Song, G.L.; et al. Multiple sequence alignment-based RNA language model and its application to structural inference. *Nucleic Acids Res.* **2024**, *52*, e3. [CrossRef] [PubMed]
96. Pal, M.; Satish, B.; Srinivas, K.; Rao, P.M.; Manimaran, P. Multifractal detrended cross-correlation analysis of coding and non-coding DNA sequences through chaos-game representation. *Phys. A-Stat. Mech. Its Appl.* **2015**, *436*, 596–603. [CrossRef]
97. Lu, Y.; Zhao, L.; Li, Z.; Dong, X.J. Genetic Similarity Analysis Based on Positive and Negative Sequence Patterns of DNA. *Symmetry* **2020**, *12*, 2090. [CrossRef]
98. Chen, M.J.; Liu, X.Y.; Liu, Q.Y.; Shi, D.S.; Li, H. 3D genomics and its applications in precision medicine. *Cell. Mol. Biol. Lett.* **2023**, *28*, 19. [CrossRef]
99. Bernaola-Galván, P.; Carpena, P.; Gómez-Martín, C.; Oliver, J.L. Compositional Structure of the Genome: A Review. *Biology* **2023**, *12*, 849. [CrossRef] [PubMed]
100. Liu, Y.X.; Saito, R.; Matsukawa, N.; Hishinuma, E.; Saigusa, D.; Liu, H.; Yamamoto, M.; Uruno, A. Nrf2 deficiency deteriorates diabetic kidney disease in Akita model mice. *Redox Biol.* **2022**, *58*, 102525. [CrossRef] [PubMed]
101. Schwessinger, R.; Gosden, M.; Downes, D.; Brown, R.C.; Oudelaar, A.M.; Telenius, J.; Teh, Y.W.; Lunter, G.; Hughes, J.R. DeepC: Predicting 3D genome folding using megabase-scale transfer learning. *Nat. Methods* **2020**, *17*, 1118–1124. [CrossRef]

102. Mu, Z.C.; Tan, Y.L.; Liu, J.; Zhang, B.G.; Shi, Y.Z. Computational Modeling of DNA 3D Structures: From Dynamics and Mechanics to Folding. *Molecules* **2023**, *28*, 4833. [CrossRef] [PubMed]
103. Duan, Y.; Zhang, C.Y.; Wang, Y.Y.; Chen, G.F. Research progress of whole-cell-SELEX selection and the application of cell-targeting aptamer. *Mol. Biol. Rep.* **2022**, *49*, 7979–7993. [CrossRef] [PubMed]
104. Tompkins, J.D. Transgenerational Epigenetic DNA Methylation Editing and Human Disease. *Biomolecules* **2023**, *13*, 1684. [CrossRef]
105. Kim, K.D.; Lieberman, P.M. Viral remodeling of the 4D nucleome. *Exp. Mol. Med.* **2024**, *56*, 799–808. [CrossRef] [PubMed]
106. Mokoatle, M.; Marivate, V.; Mapiye, D.; Bornman, R.; Hayes, V.M. A review and comparative study of cancer detection using machine learning: SBERT and SimCSE application. *BMC Bioinform.* **2023**, *24*, 112. [CrossRef]
107. Zhang, X.Y.; Venkatachalapathy, S.; Paysan, D.; Schaerer, P.; Tripodo, C.; Uhler, C.; Shivashankar, G.V. Unsupervised representation learning of chromatin images identifies changes in cell state and tissue organization in DCIS. *Nat. Commun.* **2024**, *15*, 6112. [CrossRef]
108. Bohnsack, K.S.; Kaden, M.; Abel, J.; Villmann, T. Alignment-Free Sequence Comparison: A Systematic Survey From a Machine Learning Perspective. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **2023**, *20*, 119–135. [CrossRef] [PubMed]
109. Asim, M.N.; Ibrahim, M.A.; Fazeel, A.; Dengel, A.; Ahmed, S. DNA-MP: A generalized DNA modifications predictor for multiple species based on powerful sequence encoding method. *Brief. Bioinform.* **2023**, *24*, bbac546. [CrossRef]
110. Freschlin, C.R.; Fahlberg, S.A.; Romero, P.A. Machine learning to navigate fitness landscapes for protein engineering. *Curr. Opin. Biotechnol.* **2022**, *75*, 102713. [CrossRef]
111. Lee, B.D.; Timony, M.A.; Ruiz, P. DNAvisualization.org: A serverless web tool for DNA sequence visualization. *Nucleic Acids Res.* **2019**, *47*, W20–W25. [CrossRef] [PubMed]
112. Fang, M.H.; Fang, J.W.; Luo, S.W.; Liu, K.; Yu, Q.N.; Yang, J.X.; Zhou, Y.Y.; Li, Z.K.; Sun, R.M.; Guo, C.; et al. eccDNA-pipe: An integrated pipeline for identification, analysis and visualization of extrachromosomal circular DNA from high-throughput sequencing data. *Brief. Bioinform.* **2024**, *25*, bbae034. [CrossRef] [PubMed]
113. Li, T.; Zou, Y.R.; Li, X.H.; Wong, T.K.F.; Rodrigo, A.G. Mugen-UMAP: UMAP visualization and clustering of mutated genes in single-cell DNA sequencing data. *BMC Bioinform.* **2024**, *25*, 308. [CrossRef] [PubMed]
114. Silva, J.M.; Qi, W.; Pinho, A.J.; Pratas, D. AlcoR: Alignment-free simulation, mapping, and visualization of low-complexity regions in biological data. *GigaScience* **2023**, *12*, giad101. [CrossRef]
115. Józwik, I.K.; Li, W.; Zhang, D.W.; Wong, D.; Grawenhoff, J.; Ballandras-Colas, A.; Aiyer, S.; Cherepanov, P.; Engelman, A.N.; Lyumkis, D. B-to-A transition in target DNA during retroviral integration. *Nucleic Acids Res.* **2022**, *50*, 8898–8918. [CrossRef] [PubMed]
116. Abdennur, N.; Fudenberg, G.; Flyamer, I.M.; Galitsyna, A.A.; Goloborodko, A.; Imakaev, M.; Venev, S.V.; Ma, J.; Ay, F. Pairtools: From sequencing data to chromosome contacts. *PLoS Comput. Biol.* **2024**, *20*, e1012164. [CrossRef]
117. Filatov, D.A. ProSeq4: A user-friendly multiplatform program for preparation and analysis of large-scale DNA polymorphism datasets. *Mol. Ecol. Resour.* **2024**, *24*, 13962. [CrossRef] [PubMed]
118. Shen, Y.; Gomez-Blanco, J.; Petassi, M.T.; Peters, J.E.; Ortega, J.; Guarné, A. Structural basis for DNA targeting by the Tn7 transposon. *Nat. Struct. Mol. Biol.* **2022**, *29*, 143–151. [CrossRef]
119. Ruprecht, N.A.; Kennedy, J.D.; Bansal, B.; Singhal, S.; Sens, D.; Maggio, A.; Doe, V.; Hawkins, D.; Campbel, R.; O'Connell, K.; et al. Transcriptomics and epigenetic data integration learning module on Google Cloud. *Brief. Bioinform.* **2024**, *25*, bbae352. [CrossRef]
120. Cochetel, N.; Minio, A.; Guarracino, A.; Garcia, J.F.; Figueroa-Balderas, R.; Massonnet, M.; Kasuga, T.; Londo, J.P.; Garrison, E.; Gaut, B.S.; et al. A super-pangenome of the North American wild grape species. *Genome Biol.* **2023**, *24*, 290. [CrossRef] [PubMed]
121. Kanafi, M.M.; Tavallaei, M. Overview of advances in CRISPR/deadCas9 technology and its applications in human diseases. *Gene* **2022**, *830*, 146518. [CrossRef] [PubMed]
122. Lake, N.J.; Zhou, L.; Xu, J.; Lek, M. MitoVisualize: A resource for analysis of variants in human mitochondrial RNAs and DNA. *Bioinformatics* **2022**, *38*, 2967–2969. [CrossRef] [PubMed]
123. Hong, Y.Y.; Liu, L.Q.; Feng, Y.; Zhang, Z.Q.; Hou, R.; Xu, Q.; Shi, J.T. mHapBrowser: A comprehensive database for visualization and analysis of DNA methylation haplotypes. *Nucleic Acids Res.* **2024**, *52*, D929–D937. [CrossRef]
124. Pessoa, J.; Carvalho, C. Human RNA Polymerase II Segregates from Genes and Nascent RNA and Transcribes in the Presence of DNA-Bound dCas9. *Int. J. Mol. Sci.* **2024**, *25*, 8411. [CrossRef] [PubMed]
125. Martinez, J.; Ant, T.H.; Murdochy, S.M.; Tong, L.; Filipe, A.D.; Sinkins, S.P. Genome sequencing and comparative analysis of *Wolbachia* strain *w*AlbA reveals *Wolbachia*-associated plasmids are common. *PLoS Genet.* **2022**, *18*, e1010406. [CrossRef] [PubMed]
126. Barshai, M.; Engel, B.; Haim, I.; Orenstein, Y. G4mismatch: Deep neural networks to predict G-quadruplex propensity based on G4-seq data. *PLoS Comput. Biol.* **2023**, *19*, e1010948. [CrossRef]
127. Trajkovski, M.; Pastore, A.; Plavec, J. Dimeric structures of DNA ATTTC repeats promoted by divalent cations. *Nucleic Acids Res.* **2024**, *52*, 1591–1601. [CrossRef]

128. Zhang, T.; Li, L.; Sun, H.; Xu, D.; Wang, G. DeepICSH: A complex deep learning framework for identifying cell-specific silencers and their strength from the human genome. *Brief. Bioinform.* **2023**, *24*, bbad316. [CrossRef] [PubMed]
129. Li, X.; Wu, C.K.; Yang, Z.L.; Guo, Y.J.; Jiang, R. Knowledge graph-enabled adaptive work packaging approach in modular construction. *Knowl.-Based Syst.* **2023**, *260*, 110115. [CrossRef]
130. Mohamed, S.K.; Nounu, A.; Novacek, V. Biological applications of knowledge graph embedding models. *Brief. Bioinform.* **2021**, *22*, 1679–1693. [CrossRef] [PubMed]
131. Khan, N.; Ma, Z.M.; Ullah, A.; Polat, K. Categorization of knowledge graph based recommendation methods and benchmark datasets from the perspectives of application scenarios: A comprehensive survey. *Expert Syst. Appl.* **2022**, *206*, 117737. [CrossRef]
132. Liu, Y.S.Y.; Gu, F.; Wu, Y.J.; Gu, X.J.; Guo, J.F. A metrics-based meta-learning model with meta-pretraining for industrial knowledge graph construction. *Comput. Ind.* **2022**, *143*, 103753. [CrossRef]
133. Deng, J.F.; Chen, C.; Huang, X.Y.; Chen, W.Y.; Cheng, L.L. Research on the construction of event logic knowledge graph of supply chain management. *Adv. Eng. Inform.* **2023**, *56*, 101921. [CrossRef]
134. Nicholson, D.N.; Greene, C.S. Constructing knowledge graphs and their biomedical applications. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1414–1428. [CrossRef]
135. Fei, H.; Ren, Y.F.; Zhang, Y.; Ji, D.H.; Liang, X.H. Enriching contextualized language model from knowledge graph for biomedical information extraction. *Brief. Bioinform.* **2021**, *22*, bbaa110. [CrossRef]
136. Lan, Y.Y.; He, S.Z.; Liu, K.; Zeng, X.R.; Liu, S.P.; Zhao, J. Path-based knowledge reasoning with textual semantic information for medical knowledge graph completion. *BMC Med. Inf. Decis. Mak.* **2021**, *21*, 335. [CrossRef]
137. Bonner, S.; Barrett, I.P.; Ye, C.; Swiers, R.; Engkvist, O.; Bender, A.; Hoyt, C.T.; Hamilton, W.L. A review of biomedical datasets relating to drug discovery: A knowledge graph perspective. *Brief. Bioinform.* **2022**, *23*, bbac404. [CrossRef] [PubMed]
138. Opdahl, A.L.; Al-Moslmi, T.; Dang-Nguyen, D.T.; Ocana, M.G.; Tessem, B.; Veres, C. Semantic Knowledge Graphs for the News: A Review. *Acm. Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]
139. Ali, M.; Hoyt, C.T.; Domingo-Fernandez, D.; Lehmann, J.; Jabeen, H. BioKEEN: A library for learning and evaluating biological knowledge graph embeddings. *Bioinformatics* **2019**, *35*, 3538–3540. [CrossRef]
140. Cai, X.J.; Xie, L.J.; Tian, R.; Cui, Z.H. Explicable recommendation based on knowledge graph. *Expert Syst. Appl.* **2022**, *200*, 117035. [CrossRef]
141. Su, X.R.; Hu, L.; You, Z.H.; Hu, P.W.; Zhao, B.W. Attention-based Knowledge Graph Representation Learning for Predicting Drug-drug Interactions. *Brief. Bioinform.* **2022**, *23*, bbac140. [CrossRef] [PubMed]
142. Chen, C.M.; Ross, K.E.; Gavali, S.; Cowart, J.E.; Wu, C.H. COVID-19 Knowledge Graph from semantic integration of biomedical literature and databases. *Bioinformatics* **2021**, *37*, 4597–4598. [CrossRef] [PubMed]
143. Domingo-Fernandez, D.; Baksi, S.; Schultz, B.; Gadiya, Y.; Karki, R.; Raschka, T.; Ebeling, C.; Hofmann-Apitius, M.; Kodamullil, A.T. COVID-19 Knowledge Graph: A computable, multi-modal, cause-and-effect knowledge model of COVID-19 pathophysiology. *Bioinformatics* **2021**, *37*, 1332–1334. [CrossRef]
144. Dogan, T.; Atas, H.; Joshi, V.; Atakan, A.; Rifaioglu, A.S.; Nalbat, E.; Nightingale, A.; Saidi, R.; Volynkin, V.; Zellner, H.; et al. CROssBAR: Comprehensive resource of biomedical relations with knowledge graph representations. *Nucleic Acids Res.* **2021**, *49*, gkab543. [CrossRef] [PubMed]
145. Shao, B.L.; Li, X.J.; Bian, G.Q. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Syst. Appl.* **2021**, *165*, 113764. [CrossRef]
146. Li, Z.R.; Zhong, Q.; Yang, J.; Duan, Y.J.; Wang, W.J.; Wu, C.K.; He, K.L. DeepKG: An end-to-end deep learning-based workflow for biomedical knowledge graph extraction, optimization and applications. *Bioinformatics* **2022**, *38*, 1477–1479. [CrossRef]
147. Hu, J.J.; Lepore, R.; Dobson, R.J.B.; Al-Chalabi, A.; Bean, D.M.; Iacoangeli, A. DGLinker: Flexible knowledge-graph prediction of disease-gene associations. *Nucleic Acids Res.* **2021**, *49*, W153–W161. [CrossRef] [PubMed]
148. Chen, X.J.; Jia, S.B.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948. [CrossRef]
149. Morris, J.H.; Soman, K.; Akbas, R.E.; Zhou, X.Y.; Smith, B.; Meng, E.C.; Huang, C.C.; Cerono, G.; Schenk, G.; Rizk-Jackson, A.; et al. The scalable precision medicine open knowledge engine (SPOKE): A massive knowledge graph of biomedical information. *Bioinformatics* **2023**, *39*, btad080. [CrossRef]
150. Waagmeester, A.; Stupp, G.; Burgstaller-Muehlbacher, S.; Good, B.M.; Griffith, M.; Griffith, O.L.; Hanspers, K.; Hermjakob, H.N.; Hudson, T.S.; Hybiske, K.; et al. SCIENCE FORUM Wikidata as a knowledge graph for the life sciences. *eLife* **2020**, *9*, e52614. [CrossRef] [PubMed]
151. Al-Aswadi, F.N.; Chan, H.Y.; Gan, K.H. Automatic ontology construction from text: A review from shallow to deep learning trend. *Artif. Intell. Rev.* **2020**, *53*, 3901–3928. [CrossRef]
152. Geng, Q.; Deng, S.Y.; Jia, D.P.; Jin, J. Cross-domain ontology construction and alignment from online customer product reviews. *Inf. Sci.* **2020**, *531*, 47–67. [CrossRef]
153. Fionda, V.; Gutierrez, C.; Pirro, G. Building knowledge maps of Web graphs. *Artif. Intell.* **2016**, *239*, 143–167. [CrossRef]

154. Chen, Y.; Mensah, S.; Ma, F.; Wang, H.; Jiang, Z.A. Collaborative filtering grounded on knowledge graphs. *Pattern Recognit. Lett.* **2021**, *151*, 55–61. [CrossRef]
155. Liang, S.; Shao, J.; Zhang, D.Y.; Zhang, J.S.; Cui, B. DRGI: Deep Relational Graph Infomax for Knowledge Graph Completion. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 2486–2499. [CrossRef]
156. Song, X.Y.; Li, J.X.; Tang, Y.F.; Zhao, T.G.; Chen, Y.L.; Guan, Z.Y. JKT: A joint graph convolutional network based Deep Knowledge Tracing. *Inf. Sci.* **2021**, *580*, 510–523. [CrossRef]
157. Wang, S.K.; Xu, F.; Li, Y.Y.; Wang, J.; Zhang, K.; Liu, Y.; Wu, M.; Zheng, J. KG4SL: Knowledge graph neural network for synthetic lethality prediction in human cancers. *Bioinformatics* **2021**, *37*, I418–I425. [CrossRef]
158. Guan, N.N.; Song, D.D.; Liao, L.J. Knowledge graph embedding with concepts. *Knowl.-Based Syst.* **2019**, *164*, 38–44. [CrossRef]
159. Wang, Q.; Mao, Z.D.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
160. Li, Z.F.; Liu, H.; Zhang, Z.L.; Liu, T.T.; Xiong, N.N. Learning Knowledge Graph Embedding With Heterogeneous Relation Attention Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3961–3973. [CrossRef] [PubMed]
161. Ott, S.; Barbosa-Silva, A.; Samwald, M. LinkExplorer: Predicting, explaining and exploring links in large biomedical knowledge graphs. *Bioinformatics* **2022**, *38*, 2371–2373. [CrossRef]
162. Zhu, C.Y.; Yang, Z.H.; Xia, X.Q.; Li, N.; Zhong, F.; Liu, L. Multimodal reasoning based on knowledge graph embedding for specific diseases. *Bioinformatics* **2022**, *38*, 2235–2245. [CrossRef] [PubMed]
163. Li, Z.F.; Zhao, Y.; Zhang, Y.; Zhang, Z.L. Multi-relational graph attention networks for knowledge graph completion. *Knowl.-Based Syst.* **2022**, *251*, 109262. [CrossRef]
164. Alshahrani, M.; Khan, M.A.; Maddouri, O.; Kinjo, A.R.; Queralt-Rosinach, N.; Hoehndorf, R. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics* **2017**, *33*, 2723–2730. [CrossRef] [PubMed]
165. Song, Q.Q.; Su, J.; Zhang, W. scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nat. Commun.* **2021**, *12*, 3826. [CrossRef]
166. Balabin, H.; Hoyt, C.T.; Birkenbihl, C.; Gyori, B.M.; Bachman, J.; Kodamullil, A.T.; Ploger, P.G.; Hofmann-Apitius, M.; Domingo-Fernandez, D. STonKGs: A sophisticated transformer trained on biomedical text and knowledge graphs. *Bioinformatics* **2022**, *38*, 1648–1656. [CrossRef] [PubMed]
167. Smirnov, A.; Levashova, T. Knowledge fusion patterns: A survey. *Inf. Fusion* **2019**, *52*, 31–40. [CrossRef]
168. Mao, W.T.; Liu, J.; Chen, J.X.; Liang, X.H. An Interpretable Deep Transfer Learning-Based Remaining Useful Life Prediction Approach for Bearings With Selective Degradation Knowledge Fusion. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3159010. [CrossRef]
169. Yang, J.; Yang, L.T.; Wang, H.; Gao, Y.; Zhao, Y.L.; Xie, X.; Lu, Y. Representation learning for knowledge fusion and reasoning in Cyber-Physical-Social Systems: Survey and perspectives. *Inf. Fusion* **2023**, *90*, 59–73. [CrossRef]
170. Yu, Y.; Huang, K.X.; Zhang, C.; Glass, L.M.; Sun, J.M.; Xiao, C. SumGNN: Multi-typed drug interaction prediction via efficient knowledge graph summarization. *Bioinformatics* **2021**, *37*, 2988–2995. [CrossRef]
171. Sun, Y.; Li, Y.S.; Yang, Y.Y.; Yue, H.D. Differential evolution algorithm with population knowledge fusion strategy for image registration. *Complex Intell. Syst.* **2022**, *8*, 835–850. [CrossRef]
172. Yang, L.Y.; Wang, X.; Dai, Y.X.; Xin, K.J.; Zheng, X.L.; Ding, W.P.; Zhang, J.; Wang, F.Y. HackRL: Reinforcement learning with hierarchical attention for cross-graph knowledge fusion and collaborative reasoning. *Knowl.-Based Syst.* **2021**, *233*, 107498. [CrossRef]
173. Ji, S.X.; Pan, S.R.; Cambria, E.; Marttinen, P.; Yu, P.S. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 494–514. [CrossRef] [PubMed]
174. Wang, H.; Wang, W.J.; Zhou, X.Y.; Zhao, J.; Wang, Y.; Xiao, S.Y.; Xu, M.Y. Artificial bee colony algorithm based on knowledge fusion. *Complex Intell. Syst.* **2021**, *7*, 1139–1152. [CrossRef]
175. Huang, M.S.; Lai, P.T.; Lin, P.Y.; You, Y.T.; Tsai, R.T.H.; Hsu, W.L. Biomedical named entity recognition and linking datasets: Survey and our recent development. *Brief. Bioinform.* **2020**, *21*, 2219–2238. [CrossRef]
176. Mi, B.G.; Fan, Y. A review: Development of named entity recognition (NER) technology for aeronautical information intelligence. *Artif. Intell. Rev.* **2023**, *56*, 1515–1542. [CrossRef]
177. Mao, Y.R.; Hao, Y.; Liu, W.W.; Lin, X.M.; Cao, X. Class-Imbalanced-Aware Distantly Supervised Named Entity Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 12117–12129. [CrossRef] [PubMed]
178. Song, B.S.; Li, F.; Liu, Y.S.; Zeng, X.X. Deep learning methods for biomedical named entity recognition: A survey and qualitative comparison. *Brief. Bioinform.* **2021**, *22*, bbab282. [CrossRef]
179. Perera, N.; Dehmer, M.; Emmert-Streib, F. Named Entity Recognition and Relation Detection for Biomedical Information Extraction. *Front. Cell Dev. Biol.* **2020**, *8*, 673. [CrossRef]
180. Geng, R.S.; Chen, Y.P.; Huang, R.Z.; Qin, Y.B.; Zheng, Q.H. Planarized sentence representation for nested named entity recognition. *Inf. Process. Manag.* **2023**, *60*, 103352. [CrossRef]

181. Ye, Q.; Hsieh, C.Y.; Yang, Z.Y.; Kang, Y.; Chen, J.M.; Cao, D.S.; He, S.B.; Hou, T.J. A unified drug-target interaction prediction framework based on knowledge graph and recommendation system. *Nat. Commun.* **2021**, *12*, 6775. [CrossRef] [PubMed]
182. Charlier, J.; Nadon, R.; Makarenkov, V. Accurate deep learning off-target prediction with novel sgRNA-DNA sequence encoding in CRISPR-Cas9 gene editing. *Bioinformatics* **2021**, *37*, 2299–2307. [CrossRef] [PubMed]
183. Muntoni, A.P.; Pagnani, A.; Weigt, M.; Zamponi, F. adabmDCA: Adaptive Boltzmann machine learning for biological sequences. *BMC Bioinform.* **2021**, *22*, 528. [CrossRef]
184. Danilevsky, A.; Polsky, A.L.; Shomron, N. Adaptive sequencing using nanopores and deep learning of mitochondrial DNA. *Brief. Bioinform.* **2022**, *23*, bbac251. [CrossRef] [PubMed]
185. Kurian, N.C.; Sethi, A.; Konduru, A.R.; Mahajan, A.; Rane, S.U. A 2021 update on cancer image analytics with deep learning. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2021**, *11*, e1410. [CrossRef]
186. Jing, R.Y.; Li, Y.Z.; Xue, L.; Liu, F.J.; Li, M.L.; Luo, J.S. autoBioSeqpy: A Deep Learning Tool for the Classification of Biological Sequences. *J. Chem. Inf. Model.* **2020**, *60*, 3755–3764. [CrossRef]
187. Liu, B.; Gao, X.; Zhang, H.Y. BioSeq-Analysis2.0: An updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res.* **2019**, *47*, e127. [CrossRef] [PubMed]
188. Liu, B. BioSeq-Analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Brief. Bioinform.* **2019**, *20*, 1280–1294. [CrossRef]
189. Cao, F.; Zhang, Y.; Cai, Y.C.; Animesh, S.; Akincilar, S.C.; Loh, Y.P.; Li, X.Y.; Chng, W.J.; Tergaonkar, V.; Kwoh, C.K.; et al. Chromatin interaction neural network (ChINN): A machine learning-based method for predicting chromatin interactions from DNA sequences. *Genome Biol.* **2021**, *22*, 226. [CrossRef] [PubMed]
190. Zhang, J.X.; Yordanov, B.; Gaunt, A.; Wang, M.X.; Dai, P.; Chen, Y.J.; Zhang, K.; Fang, J.Z.; Dalchau, N.; Li, J.M.; et al. A deep learning model for predicting next-generation sequencing depth from DNA sequence. *Nat. Commun.* **2021**, *12*, 4387. [CrossRef]
191. Manco, G.; Ritacco, E.; Rullo, A.; Sacca, D.; Serra, E. Machine learning methods for generating high dimensional discrete datasets. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2022**, *12*, e1450. [CrossRef]
192. Liu, Y.; Wang, T.; Lei, Y.; Roper, J.; Bradley, J.; Liu, T.; Yang, X. Synthetic Contrast MR Image Generation Using Deep Learning. *Med. Phys.* **2021**, *48*, TI9WO.
193. Yu, C.; Li, K.; Zhang, Y.X.; Xiao, J.; Cui, C.Z.; Tao, Y.H.; Tang, S.J.; Sun, C.; Bi, C.K. A survey on machine learning based light curve analysis for variable astronomical sources. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2021**, *11*, e1425. [CrossRef]
194. Gao, M.Q.; Zheng, F.; Yu, J.J.Q.; Shan, C.F.; Ding, G.G.; Han, J.G. Deep learning for video object segmentation: A review. *Artif. Intell. Rev.* **2023**, *56*, 457–531. [CrossRef]
195. Schaefferkoetter, J.; Yan, J.H.; Moon, S.; Chan, R.; Ortega, C.; Metser, U.; Berlin, A.; Veit-Haibach, P. Deep learning for whole-body medical image generation. *Eur. J. Nucl. Med. Mol. Imaging* **2021**, *48*, 3817–3826. [CrossRef] [PubMed]
196. Kauffman, C.; Karypis, G. Computational tools for protein-DNA interactions. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2012**, *2*, 14–28. [CrossRef]
197. Huang, H.M.; Lin, C. A kernel-based image denoising method for improving parametric image generation. *Med. Image Anal.* **2019**, *55*, 41–48. [CrossRef] [PubMed]
198. Cardoso, F.M.; Moraes, M.C.; Furuie, S.S. Realistic ivus image generation in different intraluminal pressures. *Ultrasound Med. Biol.* **2012**, *38*, 2104–2119. [CrossRef] [PubMed]
199. Li, W.Y.; Li, J.Y.; Polson, J.; Wang, Z.C.; Speier, W.; Arnold, C. High resolution histopathology image generation and segmentation through adversarial training. *Med. Image Anal.* **2022**, *75*, 102251. [CrossRef] [PubMed]
200. Deserno, T.M.; Handels, H.; Meinzer, H.P.; Tolxdorff, T. Recent Advances in 3D Medical Image Generation and Analysis. *Curr. Med. Imaging Rev.* **2013**, *9*, 77–78.
201. Zeleznikow, J. The benefits and dangers of using machine learning to support making legal predictions. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2023**, *13*, e1505. [CrossRef]
202. Askr, H.; Elgeldawi, E.; Ella, H.A.; Elshaier, Y.; Gomaa, M.M.; Hassanien, A.E. Deep learning in drug discovery: An integrative review and future challenges. *Artif. Intell. Rev.* **2022**, *56*, 5975–6037. [CrossRef]
203. Wang, R.H.; Jiang, Y.; Jin, J.R.; Yin, C.L.; Yu, H.Q.; Wang, F.S.; Feng, J.X.; Su, R.; Nakai, K.; Zou, Q.; et al. DeepBIO: An automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. *Nucleic Acids Res.* **2023**, *51*, 3017–3029. [CrossRef]
204. Bonet, J.; Chen, M.D.; Dabad, M.; Heath, S.; Gonzalez-Perez, A.; Lopez-Bigas, N.; Lagergren, J. DeepMP: A deep learning tool to detect DNA base modifications on Nanopore sequencing data. *Bioinformatics* **2022**, *38*, 1235–1243. [CrossRef] [PubMed]
205. Li, J.Q.; Wei, L.; Zhang, X.L.; Zhang, W.; Wang, H.C.; Zhong, B.X.; Xie, Z.; Lv, H.R.; Wang, X.W. DISMIR: Deep learning-based noninvasive cancer detection by integrating DNA sequence and methylation information of individual cell-free DNA reads. *Brief. Bioinform.* **2021**, *22*, bbab250. [CrossRef]
206. Kim, P.; Tan, H.; Liu, J.J.; Yang, M.Y.; Zhou, X.B. FusionAI: Predicting fusion breakpoint from DNA sequence with deep learning. *Iscience* **2021**, *24*, 103164. [CrossRef] [PubMed]

207. Phan, N.N.; Chattopadhyay, A.; Lee, T.T.; Yin, H.I.; Lu, T.P.; Lai, L.C.; Hwa, H.L.; Tsai, M.H.; Chuang, E.Y. High-performance deep learning pipeline predicts individuals in mixtures of DNA using sequencing data. *Brief. Bioinform.* **2021**, *22*, bbab283. [CrossRef]
208. Li, Y.; Quan, L.J.; Zhou, Y.T.; Jiang, Y.L.; Li, K.L.; Wu, T.F.; Lyu, Q. Identifying modifications on DNA-bound histones with joint deep learning of multiple binding sites in DNA sequence. *Bioinformatics* **2022**, *38*, 4070–4077. [CrossRef]
209. Chen, Z.; Zhao, P.; Li, F.Y.; Marquez-Lago, T.T.; Leier, A.; Revote, J.; Zhu, Y.; Powell, D.R.; Akutsu, T.; Webb, G.I.; et al. iLearn: An integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Brief. Bioinform.* **2020**, *21*, 1047–1057. [CrossRef] [PubMed]
210. Hollerer, S.; Papaxanthos, L.; Gumpinger, A.C.; Fischer, K.; Beisel, C.; Borgwardt, K.; Benenson, Y.; Jeschek, M. Large-scale DNA-based phenotypic recording and deep learning enable highly accurate sequence-function mapping. *Nat. Commun.* **2020**, *11*, 3551. [CrossRef] [PubMed]
211. Wang, Y.R.; Alangari, M.; Hihath, J.; Das, A.K.; Anantram, M.P. A machine learning approach for accurate and real-time DNA sequence identification. *BMC Genom.* **2021**, *22*, 525. [CrossRef]
212. Piecyk, R.S.; Schlegel, L.; Johannes, F. Predicting 3D chromatin interactions from DNA sequence using Deep Learning. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 3439–3448. [CrossRef]
213. Hasan, Z.; Roy, N. Trending machine learning models in cyber-physical building environment: A survey. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2021**, *11*, e1422. [CrossRef]
214. Emmert-Streib, F.; Dehmer, M. Taxonomy of machine learning paradigms: A data-centric perspective. *Wiley Interdiscip. Rev.-Data Min. Knowl. Discov.* **2022**, *12*, e1470. [CrossRef]
215. Rehman, A.; Khan, A.; Fatima, G.; Naz, S.; Razzak, I. Review on chest pathogies detection systems using deep learning techniques. *Artif. Intell. Rev.* **2023**, *56*, 12607–12653. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.