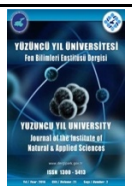
Yuzuncu Yil University
Journal of the Institute of Natural & Applied Sciences

https://dergipark.org.tr/en/pub/yyufbed



Research Article

3D RNA Graph Representation Methods for Classification of RNA Molecules Using Graph Kernel and Graph Neural Network Methods

Enes ALGUL

Bingol University, School of Engineering and Architecture, Department of Computer Science, Bingöl, Türkiye
Enes ALGUL, ORCID No: 0000-0001-6597-4242
Corresponding author e-mail: ealgul@bingol.edu.tr

Article Info

Received: 24.02.2023
Accepted: 25.05.2023
Online December 2023

DOI:10.53433/yyufbed.1256154

Keywords
3D RNA graph representations,
Geometric measurements,
Graph classifications,
Graph kernels,
Graph neural networks

Abstract: Ribonucleic acids (RNAs) are nucleic acid types with 1D/2D/3D structural shapes and are essential for sustaining life. These structural shapes of the RNAs are highly correlated with their functions. While the primary and secondary structures of RNA have been extensively studied, the tertiary structure has received relatively less attention. In this article, we present novel approaches for representing 3D RNA structures as graph data, employing geometric measurements such as Base position, Square root velocity function (SRVF), Arc length, and Curvature. Then, we utilise kernel methods and neural network methods to predict RNA functions. Our findings demonstrate the effectiveness of these methodologies in unraveling the functional attributes of RNA molecules, thus enriching our understanding of their complex biological significance.

Graf Çekirdek ve Graf Sinir Ağı Yöntemlerini Kullanarak RNA Moleküllerini Sınıflandırılmak İçin 3D RNA Graf Temsili Yöntemleri

Makale Bilgileri
Geliş: 24.02.2023
Kabul: 25.05.2023
Online Aralık 2023

DOI:10.53433/yyufbed.1256154

Anahtar Kelimeler
3B RNA graf temsilleri,
Geometrik ölçümler,
Graf çekirdekleri,
Graf sınıflandırmaları,
Graf sinir ağları

Öz: Ribonükleik asitler (RNA'lar), 1B/2B/3B yapısal şekillere sahip nükleik asit türleri olup, yaşamı sürdürmek için hayati öneme sahiptirler. RNA'ların bu yapısal şekilleri, fonksiyonlarıyla yüksek derecede ilişkilidir. RNA'nın birincil ve ikincil yapıları kapsamlı bir şekilde incelenirken, üçüncül yapı nispeten daha az dikkat çekmiştir. Bu makalede, Baz konumu, Karekök hız fonksiyonu (SRVF), Yay uzunluğu ve Eğrilik gibi geometrik ölçümler kullanarak 3B RNA yapılarını grafik verileri olarak temsil etmeye yönelik yeni yaklaşımlar sunuyoruz. Daha sonra, çekirdek (kernel) yöntemleri ve sinir ağı (neural network) yöntemleri kullanarak RNA fonksiyonlarını tahmin ediyoruz. Bulgularımız, bu metodolojilerin RNA moleküllerinin fonksiyonel özelliklerini çözmedeki etkinliğini gösteriyor ve böylece onların karmaşık biyolojik önemine dair anlayışımızı zenginleştiriyor.

1. Introduction

RNA plays significantly important functions in the cell, including protein synthesis, RNA splicing/modification/maturation, cell division, treatment of diseases (cancer and viral/bacterial infections), and other catalytic and regulatory roles (Ding, 2006; Chen et al., 2012; Hajiaghayi et al., 2012; Laing et al., 2013; Laborde et al., 2013; Purzycka et al., 2015; Huang & Lin, 2016; Carrasco-Hernandez et al., 2017; Balcerak et al., 2019). The RNAs' roles depend on their structural shapes.

The information contained in an RNA molecule consists of three different types. The sequence is simply a list of base types in RNA (1D). Paired bases in the RNA structure induce a topological structure (2D). RNA also has complex 3D shapes. 3D RNA shapes have been used in various studies (Laborde et al., 2011; Huang & Lin, 2016; Miao & Westhof, 2017) to find the function of RNA molecules. The shapes of the RNAs are in different sizes and structures. Computationally, extracting structural information from such biochemical components and using them in Machine Learning applications is a challenging problem. Currently, shape alignment methods are applied for RNA comparisons (Needleman & Wunsch, 1970; Lau & Ferré-D'Amaré, 2016).

Organizing and transforming biomolecules and chemical compounds into structured data forms for use in learning algorithms is a challenging problem due to the complexity of these molecules and compounds. Graphs ($G = (V, E, l)$) are flexible data structures that are defined as sets of vertices (V), edges (E), and labels (l); the labels are optional and represent vertex or edge attributes. Biomolecules and chemical compounds can be modelled as graphs, with atoms as vertices, hydrogen bonds as edges, and atom names as labels. Similarly, RNA structures can be represented as a graph with each nucleobase represented as a vertex, the relation between nucleobase pairs as edges, and RNA attributes encoded on the graph's node and edge labels. This graph-structured representation offers potential for graph learning applications.

This research aims to assess the potential of graph representation methods that combine 2D topology and 3D geometric information about RNA structures. We compare the effectiveness of these methods in the RNA classifications using the York RNA dataset (Algul & Wilson, 2019).

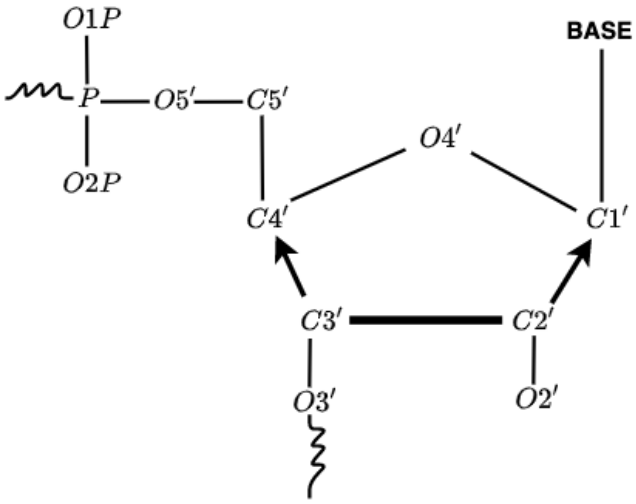


Figure 1. A representation of the nucleotide showing the $C3'$ (Pande & Nilsson, 2008).

Graph kernels compare the similarity between two graphs and have been widely used for structural pattern recognition and RNA classification. Examples include Shortest Path (Borgwardt & Kriegel, 2005), random walk (Kang et al., 2012), Weisfeiler-Lehmann (Shervashidze et al., 2011), and All Paths and Cycles kernels (Giscard & Wilson, 2017). These methods typically have some limitations in the application of graph representations, such as restrictions on negative edge labels and a constrained set of node labels. This restriction in the richness of the representation is investigated in this work, where we introduce various RNA graph representations with limited node labels and 1-dimensional node features.

Deep Learning (DL) methods can process multidimensional features as inputs and have been effectively used on structured data like grids or sequences represented in Cartesian Space. The Graph data, however, are dynamic and lack fixed node order and reference points. Graph neural networks (GNN) have been developed to address this (Dai et al., 2016; Kipf & Welling, 2017; Zhang et al., 2018; Xu et al., 2019). In GNNs, a novel graph convolutional layer has been added to extract local features, extending the capabilities of convolutional neural networks (Zhang & Chen, 2018). DL methods have found application in a range of tasks related to graphs, including the classification of nodes and graphs, prediction of edges or links, and detection of communities within structured data (Dai et al., 2016; Zhang et al., 2018; Zhang & Chen, 2018; Xu et al., 2019; Gao & Ji, 2019; Ren et al., 2021; Chen et al., 2017; Zhao et al., 2021). Node classification, crucial for social network analysis, involves using neighbours to identify node features in graphs (Zhao et al., 2021). The edge prediction aims to predict connections in the graph networks. Community detection discovers clusters or segments in a graph (Chen et al., 2017). GNNs are utilized to classify graph data based on their labels in graph classification tasks. Many GNN techniques have been developed in bioinformatics for classification problems (Dai et al., 2016; Zhang et al., 2018; Xu et al., 2019; Gao & Ji, 2019; Du et al., 2019; Zhang et al., 2019b; Zhao et al., 2021). In this article, the cutting-edge GNN methods are also analysed for classifying 3D RNA graphs.

In this paper, we explore graph representation techniques for 3D RNA structures and introduce multiple 3D RNA graph representations with continuous multi-dimensional node labels based on the geometric coordinates (x, y, z) of the RNA backbone sugar ($C3$ atom). These representations apply to all RNA strands available in our dataset and allow comparison between RNA molecules for classification. Our literature analysis and experiments show that RNA molecules with similar 3D shapes are typically classified under the same functional RNA group (Kerpedjiev et al., 2015; Purzycka et al., 2015; Miao & Westhof, 2017; Wilson & Algul, 2018; Algul & Wilson, 2019; Magnus et al., 2019). The geometric shape of the RNA's backbone sugar is an important factor in the prediction of RNA functions.

Therefore, we extract the x, y, z coordinates of each RNA nucleotide's $C3$ atom (See Figure 1) to provide the representations. We then utilize geometric measures (base position, SRVF, arc length, curvature) for representing RNA shapes. Each of these representations is in the form of feature matrices, where each row is being RNA nucleotide's $C3$ feature, and the matrix consists of continuous feature columns. We employ k-medoids and Learning Vector Quantization (LVQ) for clustering and labelling rows as nodes, as graph kernel methods require limited distinct node labels and only accept 1D node features.

2. Problem Statement

The 3D RNA shape, which is more complex than the 2D RNA shape, has received less attention than its 1D/2D structure in determining the biological tasks of RNA molecules (Laborde et al., 2013). The main issues are,

- Is it possible to transform 3D RNA structures into graph data with geometric measurement techniques?
- What are the most suitable methods to classify RNA graphs by RNA function?

The objective is to discover effective representations for converting the 3D RNA structures into graphs. Then, to classify RNA molecules according to their functional categories by applying machine learning techniques to these representations, along with the sequences and topological data.

3. Related Work

According to our review of available literature, there are currently two primary methods for representing the 3D structure of RNA in a graph data format. The first set of methods uses 2D RNA topologies and the coordinates (x, y, z) of RNA backbone sugar (Laing et al., 2013; Zahran et al., 2015; Kim et al., 2015). These techniques produce 2D topologies of RNA and obtain geometric structure information inferred from PDB files for providing corresponding 3D RNA shapes. This process involves encoding structural elements as nodes and connecting these nodes with edges. Additionally, geometric data is employed to set the helices, and an extra node is placed at the centre of the junctions to construct the 3D RNA graphs. The second set of approaches operates by grouping structural elements of the 2D RNAs (Petrov et al., 2013; Reinharz et al., 2018; Chen et al., 2017; Oliver et al., 2022). This approach

represents each cluster of elements as nodes and edges connecting clusters with shared/matched elements.

RNAJAG (RNA-Junction-As-Graph) (Laing et al., 2013) encodes RNA molecules by transforming helical regions and junctions into a tree graph, using 2D RNAs as input and predicting the junction topologies. RNAJAG estimates 3D structures by considering a collection of helical settings within the scope of the junctions. By leveraging geometric data ($C1, C6$, and $C8$) from RNA strands, RNAJAG constructs a tree graph to position the edges at the junctional region.

RAGTOP (RNA-As-Graph-Topologies) (Kim et al., 2015) encodes RNA molecules in a hierarchical structured data form for analysing the riboswitch's 3D topology. RAGTOP uses RNAJAG to determine junction topologies and puts geometrical data of the helical settings in 3D. It uses knowledge-based statistical potentials to represent the loops as nodes and helices as edges in pseudoknot-free structures. Furthermore, the vertices in RAGTOP are assigned 3D coordinates at the centres of loops and helices, and as extra edges, it adds pseudoknot interconnections.

RAG-3D (RNA as Graph 3D) (Zahran et al., 2015) is an online tool and a dataset providing 3D RNA structures into tree graph data forms with a maximum of 10 nodes (approximately 240 nucleobases). The tool compares the tree graphs and their substructures to discover almost identical topologies using a Laplacian Matrix. RAG-3D identifies similar graphs by considering the number and label of nodes, as well as the eigenvalue of the Laplacian matrix. It uses RNAView (Yang et al., 2003) to generate pseudoknot-free 2D RNA structures and predicts the 3D RNA building components. It is important to note that RAG-3D can only represent graphs with a minimum of 2 nodes and does not include linear/straight strands.

The RNA 3D Motif Atlas (Petrov et al., 2013) introduces secondary RNA structures and their structural elements using VARNA (Darty et al., 2009) and generates 3D motif groups. These motifs consist of loops with similar structures and are represented as vertices with a weighted edge connecting groups of motifs with similar motifs. RNA Bricks (Chojnowski et al., 2013), VeRNAl (Oliver et al., 2022), and CaRNAval (Reinharz et al., 2018) also encode 3D RNA motifs as graphs.

RNAComposer (Purzycka et al., 2015) provides 3D RNA structures using 2D RNA tree graphs, but it faces significant challenges in predicting the 3D structures of large RNA sequences. It has a limitation of generating 3D RNA structures with a maximum length of 500 nucleotides (Purzycka et al., 2015).

However, these methods have limitations in their size and fail to consider straight/linear RNA strands. Therefore, to tackle this problem, it is necessary to develop graph representations of 3D RNA of any size. A new approach, Elastic Shape Analysis, as described in (Ding, 2006; Laborde et al., 2011; Chen et al., 2012; Laborde et al., 2013; Purzycka et al., 2015; Carrasco-Hernandez et al., 2017), considers RNA strands as 3D curves and characterizes it using a square root velocity function (SRVF), which is particularly useful for the geometric analysis of RNA shape.

Graph classification is currently an area of interest in machine learning, with many different approaches proposed. We focus on two recent approaches. The first is based on kernels. Graph kernels are organized for structural data and have shown promising performance. Indeed, a graph kernel is a similarity technique that assesses pairwise graph similarities. Various newer techniques have emerged, such as the All Paths and Cycles (APC) Embedding (Giscard & Wilson, 2017), which explores the similarity in paths and cycles between graph pairs. The Weisfeiler Lehman Optimal Assignment Kernel (WL-OA) (Kriege et al., 2016) is another advanced method for comparing labelled pairwise graphs. This method uses the Weisfeiler-Lehman label enrichment procedure, and additionally only measures the best match between the label sets. Furthermore, the Shortest-path kernel is also employed in our research to assess the similarity between the shortest paths in two graphs.

Graph Neural Networks (GNNs) are new techniques for graph classification that works similarly to convolutional neural networks (CNNs). In GNNs, convolution and aggregation operations are typically based on the local structures of the network. The process includes iterative computation of the features of neighbouring nodes of each node, aggregation of this information through message passing, and updating the features of the current node (Gilmer et al., 2017; Xu et al., 2018; Xu et al., 2019). This approach has similarities to the WL subtree kernel method. Early work in GNNs includes GCN (Kipf & Welling, 2017), which introduced the graph Laplacian approach for graph convolution. A relative example, gUNets (Gao & Ji, 2019), has a node computation process that is similar to graph kernels. Another deep learning method, such as DGCNN, has analogies with the graph kernels (Propagation

Kernel (Neumann et al., 2015), WL subtree kernel (Shervashidze et al., 2011). These kernels iteratively update node labels based on information from their node neighbours.

Using non-Euclidean data as input is challenging in traditional CNN architectures since they can only handle input data of the same size (Zhang et al., 2019a), and fixed-size arrays requires to use in the fully connected layer in CNN. To solve this problem, DGCNN resizes the graphs in the Sort Pooling layer, allowing the use of traditional CNNs. On the other hand, GIN does not include fully connected layers and employs the SoftMax function directly for minimizing the loss after convolution and pooling. The most crucial aspect of GNNs in a convolution layer is to train local features for the graph network embeddings.

4. 3D RNA Representation Methods

We begin with an RNA graph representation where nodes represent RNA bases, and the nodes are linked in sequence i.e., nodes representing two consecutive bases are linked by an edge. We used X3DNA (x3dna.org, n.d.) to identify base pairs, and these pairs are also linked by an edge in our representation. We can also include node labels that represent the base type, although, in the later experiments, these are not always used. This standard representation, therefore, encodes the sequence and 2D topology of the RNA. Treating 3D RNA as 2D structures and relying solely on their topology neglects a significant amount of information about their geometry. Furthermore, comparing small RNA strands which lack base pairs is not possible, where most information resides in the 3D shape. In this section, our objective is to explore 3D representations of RNA that take into account information inferred from the 3D structure of the RNAs. Since we have chosen to use a graph representation, this will take the form of additional geometry labels on the nodes or edges.

Initially, our focus is on Elastic Shape Analysis (ESA), where the RNA sequences are treated as 3D curves and reformed with the square root velocity function (SRVF) (Liu et al., 2010). Then, we test local geometric indicators such as arc length and curvature to characterise the compactness and bending of the strand. Finally, we investigate a direct 3D representation by utilizing the backbone position (the position of the $C3$ atom) and the RNA nucleobase's centroid position.

4.1. Elastic shape analysis (ESA)

Our approach involves utilizing Elastic Shape Analysis (ESA) to represent the geometry of RNA shapes. ESA method considers 3D RNA shapes as parameterized continuous 3D curves, represented as $\beta : [0,1] \rightarrow R^3$. The primary purpose of the ESA method is to maintain the curve's essential characteristics during transformations such as rotation, scaling, and translation. To explore the stretching and bending of RNA strands, we further represent the 3D curves using the SRVF (Liu et al., 2010; Laborde et al., 2011). The 3D RNA curve is represented in Euclidean space (R^3) using the geometric information of RNA strands. We obtain the sequence of coordinates (x_i, y_i, z_i) of the $C3$ atom in each base and represent the sequence as a continuous curve $\beta : [0,1] \rightarrow R^3$ with $\beta(t_i) = (x_i, y_i, z_i)$ where

$$t_1 = 0, t_{i+1} = t_i + \frac{1}{L}\left|\left|\beta_{i+1} - \beta_i\right|\right| \ \forall 1 \leq i < n \tag{1}$$

Here, L indicates the total length of the RNA curve, n refers to the number of bases of RNA strands, and t_i corresponds to the time at the point β_i such that $\beta(t_i) = \beta_i$. Now we represent the geometry of β using SRVF ($q(t)$).

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{\left|\left|\dot{\beta}(t)\right|\right|}} \tag{2}$$

Where, $||\,.\,||$ is the euclidean norm, and $\dot{\beta}(t) = \frac{d\beta(t)}{d(t)}$ re-adjust each curve β to a length of 1, and $\dot{\beta}(t)$ is calculated using the given sample points (t_i, β_i) as

$$\dot{\beta}(t) = \frac{\beta_{i+1} - \beta_i}{t_{i+1} - t_i} \tag{3}$$

$q(t)$, which comprises both directions $(q(t) \,/\, ||q(t)|| = \dot{\beta}(t) \,/\, ||\dot{\beta}(t)||)$ and speeds $(q(t)^2 = ||\dot{\beta}(t)||)$, is invariant to translation of β due to the time derivative (Liu et al., 2010). Thus, we represent RNA strands as curves with SRVF, derived from their geometric shapes (Laborde et al., 2011).

4.2. Arc length

Again, we consider the RNA strand to be a curve passing through the $C3$ atom on the base and want to encode the arc length between bases. Since the strand is bending differently in different locations. This is not simply the distance between bases. We divide the curve into sub-intervals, which are arcs between two nucleobases. We calculated the arc lengths (i.e., the length of the sub-intervals) by using (x_i, y_i, z_i) coordinates of the $C3$ atoms as below:

$$d_i = \sqrt{(x_i - x_{i-1},)^2 + (y_i - y_{i-1},)^2 + (z_i - z_{i-1},)^2} \tag{4}$$

$$S_i = \frac{d_i - d_{i-1}}{2} \tag{5}$$

$$S_1 = d_1 \tag{6}$$

4.3. Curvature

Curvature is a geometry measurement technique that computes the speed and sharpness of changes in a curve direction (Verbeek & Vliet, 1993). However, the process of computing curvature is more complex than arc length. We assume that the curvature is approximately constant between three consecutive residues and fit an osculating circle through these points using the method of (Mjaavatten, 2020). The curvature is then the inverse radius of this circle, $\kappa i \;=\; 1 \,/\, R_i$.

4.4. Base position

In (Wilson & Algul, 2018; Algul & Wilson, 2019), we presented 3D graph representations of RNA geometry using three labels (1, 2, and 3) to distinguish paired (label: 1) and unpaired bases (label: 2, within 6.5Å; label 3: otherwise), as well as bases located both inside and outside of the loops (label: 2, 3). Edges in the RNA graph connect paired and adjacent vertices.

Additionally, we developed a novel graph representation using the backbone and centroid positions to encode RNA geometry. First, we inferred the distances between any base pairs. Second, we keep the min distances between per RNA base and its nearest neighbour. Then, we cluster minimum distances using k-medoids and Learning Vector Quantisation (LVQ). The resulting vertex labels correspond to base positions.

Our explanation in Section 5 will cover the clustering of these geometric representations (base position, curvature, arc length, SRVF) values, as well as the encoding of each of these values in a graph-structured form.

5. Encoding RNA Representations as Graph

While graph kernels have produced excellent results in graph classification problems, one drawback is that they are typically restricted to limited/few numbers of discrete labels on the nodes and edges (Giscard & Wilson, 2017; Kriege et al., 2016). To overcome this limitation, we use LVQ for

clustering node labels and obtain 1- dimensional discrete labels for use in kernel methods. On the other hand, GNNs typically operate on a vector of continuous features. In this section, we describe both our discrete and continuous node features.

5.1. Label representation

A supervised data clustering method known as Learning Vector Quantization (LVQ) utilizes a group of codebook vectors (Schneider et al., 2009; Nova & Estévez, 2013). These vectors serve as reference points to determine the data point closest to them, thereby acting as suitable measures of distance (Schneider et al., 2009). In order to use LVQ effectively, it is essential to identify efficient measures of distances, and we achieved this by using k-medoids. Our approach involved applying base position, curvature, arc length, and SRVF to RNA strands, providing features for each nucleotide. We then combined these features of all nucleotides in the dataset to construct an $(n \times m)$ matrix where m indicates the number of feature channels and n indicates the total number of nucleobases. Subsequently, using k-medoids, we identified k suitable distance measures and used them as inputs in the LVQ as codebook vectors. To explore the most efficient discrete labels of nodes, we utilized LVQ. Consequently, we grouped each row of the RNA matrix $(n \times m)$ using k-medoids and LVQ for encoding each of these rows as node labels.

Thus, RNA graph representations are constructed with a distinct set of node labels, determined by the features used to represent the RNA strands. The labels correspond to 'A: arc-length', 'B: base-position', and 'C: curvature'. For each RNA representation, the optimal k values are selected by extensive trials.

- RNA_ABC: This representation uses all three features (A: arc-length, B: base-position, C: curvature) and constructs a joint representation ($(n\ x\ 3)$ matrix). The matrix is then processed using k-medoids with $k = 6$ and LVQ for encoding each row as a distinct node label. The resulting graph consists of nodes with one of six distinct labels.
- RNA_AB: This representation uses both arc-length and base-position to construct a graph, resulting in 5 distinct node labels. The encoding process is similar to RNA_ABC, but with $k = 5$.
- RNA_AC: This representation uses arc-length and curvature to construct a graph with $k = 5$ discrete node labels.
- RNA_A, RNA_B, RNA_C: These representations only utilise single features to construct the graph with 5, 6, and 5 discrete node labels, respectively.
- RNA_SRVF: These representations use SRVF to convert each RNA molecule with a vector feature $q(\mathrm{t_i})$ for each node. The features are encoded into $k = 4$ distinct node labels using k-medoids and LVQ.
- RNA_SRVF_P: The principal component analysis (PCA) is employed to three coordinates of the $C3$ atom of the RNA strands to pre-align the strands. The RNA strands are then encoded using the same method as RNA_SRVF to construct a graph with four distinct node labels.

5.2. Continuous feature representations

This section presents various representations for encoding 3D RNA structures to determine the optimal approach for classifying RNA molecules using geometric deep learning (GDL) applications. Our graph representations incorporate node labels with multi-dimensional continuous features, enabling learning algorithms to utilize these features while operating on graphs.

- RNA_A-II: This approach applies arc length to the RNA strands. Each node has a continuous feature corresponding to the calculated arc length.
- RNA_C-II: Similarly, this approach employs curvature as the single node feature.
- RNA_AC-II: This representation combines arc length with curvature to construct 2D node features.

- RNA_XYZ: This approach directly represents the spatial positions of the $C3$ atoms in each base via their (x, y, z) coordinates. This is similar to a point-cloud representation, although we retain the topological information in the graph.

6. Classification Methods

The task in the York RNA Dataset is to classify each RNA molecule into one of eight classes. In this article, one of our goals is to provide a comprehensive comparison of various graph-based classification methods for this problem. Here we briefly review the graph kernels and GNN-based methods we have applied.

6.1. Graph kernel methods

The *Weisfeiler-Lehman Optimal Assignment Kernel (WL-OA)* is an advanced technique for comparing labelled graphs. This technique utilises the Weisfeiler-Lehman label refinement approach. The WL kernel (de Vries, 2013) counts the number of label matches at each level of the refinement hierarchy. The optimal assignment method (Kriege et al., 2016) augments this by counting only the labels of the best match between the two graphs. For *h* refinement levels, the WL-OA kernel is described as

$$K(G_1, G_2) = \max_{B} \sum_{u,v \in B} \sum_{i=0}^{h} k_\delta(\tau_i(v), \tau_i(v')) \tag{7}$$

where k_δ denotes a base (Dirac) kernel, $\tau_i(.)$ denotes the level-h label of a vertex, and B is a bijection between the two graphs. The max runs over all possible bijections.

The walk-based kernels calculate the number of similar walks between graph pairs (G_1, G_2) and measure their pairwise similarities. A walk is considered a path only when it doesn't involve replicated edges. The *Shortest Path (SP) kernel* (Borgwardt & Kriegel, 2005; Hermansson et al., 2015) is an example of the walk-based kernel that exclusively counts the shortest path between each node pair. It counts all the shortest distances between every possible pair of nodes in the graphs (G_1, G_2). The paths can be characterized by their sequence of edge and node labels and their respective lengths. Here we use the length and the start and end-point labels (Kriege et al., 2020). The shortest paths can be calculated in $O(n^3)$using the Floyd-Warshall algorithm, making this method quite computationally expensive for large graphs.

$$K(G_1, G_2) = \sum_{p_i \in SP(G_1)} \sum_{p_j \in SP(G_2)} k_\delta(\tau(p_i), \tau(p_j)) \tag{8}$$

Here, $SP(.)$ refers to a collection of shortest paths, p_i is a shortest path between node pairs in (G_1) and likewise p_j for G_2. $\tau(.)$ is the path labelling function.

The *All Paths and Cycles kernel (APC)* (Giscard & Wilson, 2017) is designed to count all simple cycles and possible paths within a graph up to a predetermined maximum length instead of solely computing the shortest paths.

$$K_{APC}(G_1, G_2) = \sum_{p_i \in PC_h(G_1)} \sum_{p_j \in PC_h(G_2)} k_\delta(\tau(p_i), \tau(p_j)) \tag{9}$$

In this equation, $PC_h(G)$ refers to the collection of all simple cycles and possible paths on the graph G that have a length of at most h. Due to the complexity of counting all paths, the number of distinct node labels that can be used is restricted to a maximum of 3.

6.2. Graph convolutional networks

The *Deep Graph Convolutional Neural Network (DGCNN)* (Zhang et al., 2018) intends to be a graph analogue of the convolution neural network (CNN). It employs a graph convolutional and pooling layer, which respects the structure of the graph. The convolution operation is somewhat similar to the label refinement process in kernels (WL subtree (Kriege et al., 2016), PK (Neumann et al., 2015)). A Sort Pooling layer is used in DGCNN to arrange node feature descriptors in a specific order and ensure a uniform size across all input graphs. Thus, the gap between the traditional pattern vector and the graph-based representation is bridged.

The *Graph Isomorphism Network (GIN)* provides a straightforward convolution operation that is capable of capturing local features and providing a novel vector representation of nodes to address classification tasks for graphs (Xu et al., 2019). GIN iteratively updates nodes' feature vectors via an aggregation operation on information from the neighbours, by applying an aggregation function such as max, mean, and sum.

Structure2Vec is a graph embedding method (Ribeiro et al., 2017). The model is based on inferring latent variables to represent node information and then constructing discriminative information in a feature space for the particular problem at hand.

Graph U-Nets (Gao & Ji, 2019) is a network based on the U-net architecture. It utilises two operations (gPool and gUnpool) to downsample and upsample the network, and a trainable projection vector p to project the nodes from one layer to the next. The downsampling phase aims to compress the graph into a more compact form, and the upsampling checks the information is properly preserved. The gPool and gUnpool layers operate as an encoder-decoder mechanism where the gPool layers encode node features of higher order, and the gUnpool layers reconstruct the previous graph structure. Finally, a GCN layer is applied for final predictions, and then a soft-max function is applied to predict the class.

Label Contrastive Coding based Graph Neural Network (LCGNN) (Ren et al., 2021) uses a base graph encoder with an addition to the loss function to enhance the contrast between same-label and different-label pairs:

$$L_{total} = L_C + \beta L_{LC} \tag{10}$$

where L_C is the classification loss and L_{LC} is the label contrast loss. The method maintains a group of encoded labelled graphs, which aims to ensure that the graph embeddings are both similar to same-label examples in the set, as well as dissimilar to different-label examples.

7. Results and Discussion

In this part, we explore the efficacy of classical kernel methods and GNNs on this problem and the difference between discrete label encoding of properties vs. continuous features.

7.1. Data

We employed the RNA Graph Classification Data Set that includes 3178 RNA strands compiled by the University of York (Algul & Wilson, 2019). Another alternative dataset, the SCOR database of Klosterman et al. (2002), has 419 RNA structures and is too small to draw comparative conclusions about the methods.

7.2. Classification methodology

For the kernel-based methods, we first compute the all-pairs kernel values for the dataset. We then perform the kernel embedding to obtain a feature representation. Finally, we apply the subspace-kNN classification method, which produced the best results from our experiments. The training/test split was 85%/15%. The GNN methods are configured as described in section 6.2, with classification using the method described in the original work.

7.3. Kernel methods

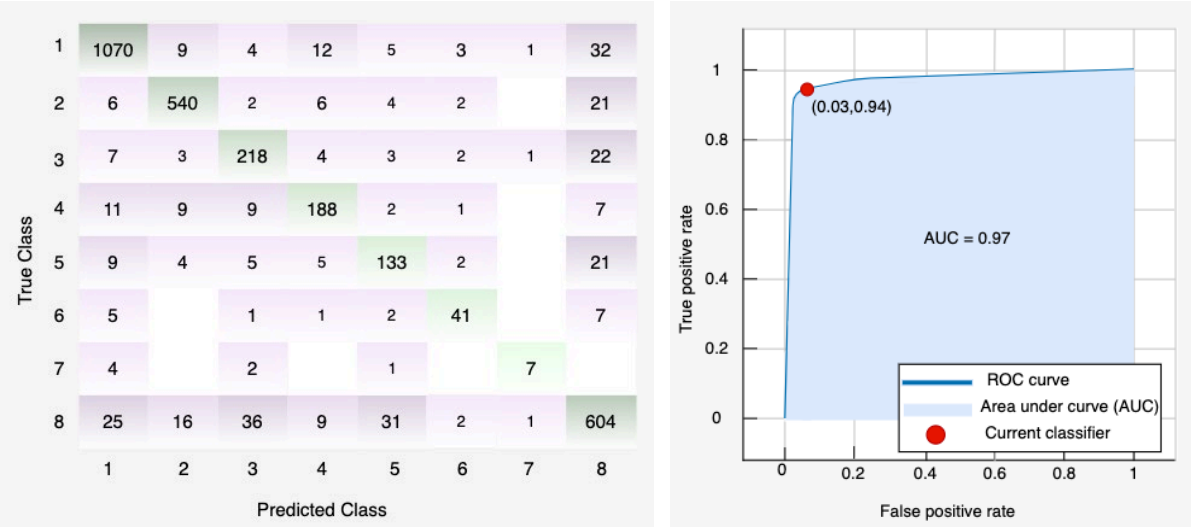
We aim to assess the efficacy of various 3D RNA graph representation techniques in classifying RNA molecules. Based on our literature review, we have found that there are currently no original or standard graph representations available that specifically encode the geometric data of RNA. We evaluate introduced graph representations utilising advanced graph kernels.

In table 1, we show the outcomes of the graph kernel techniques (WL-OA, SP, and APC) on the shape representations using the sequence graph only, i.e., using the sequence but not the topological edges from base pairs. The graphs use discrete labels as described in section 5. We see an improvement in performance from using multiple shape labels (RNA-AB, RNA-BC, RNA-ABC). The elastic shape analysis features are not particularly effective, and R-ABC is the best overall representation. The APC kernel does not perform well, and SP is the most effective kernel overall. The best single result is SP+RNA-AB with 88.1% accuracy.

Table 1. The performance of Graph kernels on the introduced variety of RNA graph representations for Dataset Categorization

	RNA-A	RNA -C	RNA -B	RNA-AB	RNA-AC	RNA-BC	RNA-ABC	RNA-SRVF	RNA-SRVF-P	RNA-type
SP	81.9	84.9	87.8	**88.1**	82.7	87.7	87.5	82.9	83.3	86.7
APC	76.1	82.9	84.6	85.3	76.1	85.9	**86.2**	80.1	81.5	85.5
WL-OA	78.7	85.7	86.9	**87.4**	82.4	**87.4**	**87.4**	81.6	81.6	87.1

Table 1 presents the results of applying different graph kernels to a range of RNA graph representations in order to categorize the RNA dataset. The introduced representations consist of 1D distinct node characteristics. In the table, bold numbers highlight the best result achieved for each representation, while red indicates the best kernel method employed for that particular representation.



(a) Confusion Matrix (b) ROC Curve

Figure 2. Confusion matrix and ROC curve generated by applying method SP+RNA-ABC.

We plotted a ROC curve and utilized the Subspace KNN classifier to assess the classification performance for SP+R-ABC. The resulting analysis showed a true positive rate (TPR) of 94% and 97% of an area under the curve (AUC).

Based on our observations, we conclude that using joint graph representations gives superior results compared to using single graph representations alone. As shown in Table 1, the most optimal result is achieved by employing the SP kernel to RNA-ABC, resulting in an 88.1% accuracy rate in three graph kernel methods. Furthermore, the SP method demonstrates superiority over other graph kernels in 8 out of 10 3D graph representations.

Table 2. The RNA dataset's classification accuracy utilising 3 kernel methods and 10 RNA graph representations. R-type is introduced in (Algul & Wilson, 2019) as a type

Full graph	RNA-A	RNA-C	RNA-B	RNA-AB	RNA-AC	RNA-BC	RNA-ABC	RNA-SRVF	RNA-SRVF-P	RNA-type
SP	83.4	85.1	87.2	87.8	81.0	87.6	88.1	80.9	82.6	86.7
APC	76.1	82.6	84.7	85.6	76.1	85.7	86.1	80.1	81.4	84.3
WLOA	82.3	85.0	87.0	87.4	79.0	87.5	87.6	81.2	81.7	86.8

In Table 2, we now add the base-pair edges to the graph to represent the 2D topology of the molecule. The addition of topology improves the majority (6 out of 10) of the representations, but in many cases only marginally. The previous best-performing method decreases slightly in performance, and now the best result is received from the RNA-ABC representation (88.1 percent) with the SP method. However, this is the same headline performance with no topology. We conclude that the kernel methods are highly dependent on label quality and not the connectivity of the representation.

7.4. Deep learning methods

We now investigate whether the deep learning architectures, which use graph learning and continuous features, are advantageous for classifying graph representations of RNA. In section 6.2, we described 5 graph neural network methods including DGCNN (Zhang et al., 2018), gUNets (Gao & Ji, 2019), structure2vec (Dai et al., 2016), GIN (Xu et al., 2019), and LCGNN$_{GIN}$ (Ren et al., 2021) utilised in this work. Collectively we refer to these as graph deep learning (GDL) methods. Our implementation of the GDL methods involved the utilization of PyTorch in Python. For gradient descent, we employed the Adam optimizer in all models. To receive the best test results, we employed 10-fold cross-validation and trained each model for around 600 epochs, and recorded the optimal outcomes. Following several iterations for each technique, we have determined the optimal hyperparameters and the values.

Table 3. Selection of k and graph size for DGCNN. The numbers in parentheses indicate the graph sizes after fixed. These classification accuracy scores were achieved using the DGCNN on RNA-R graph representations

	k = 0.9	k = 0.8 (158)	k= 0.75 (122)	k=0.6 (77)
DGCNN	93.063	94.006	91.798	92.744

Our implementation of the DGCNN model includes five convolutional layers with output channels of $[16, 32, 32, 16, 1]$. We configured the SortPooling with k by 0.8, which indicates that in 80% of the graphs, the number of vertices is less than k. We set the learning rate for our DGCNN model to

0.0001 and kept all other parameters of the original implementation. Based on the results shown in Table 3, the most outstanding results were achieved when the graph sizes were fixed at 158 nodes. Our GIN model implementation leveraged the sum aggregator function for node embedding and had four convolutional layers, a dropout proportion of 0.4, and a learning rate of 0.005. The remaining elements of the GIN model were identical to the original model. The Structure2Vec model performed the highest results when we utilized the identical hyperparameters found in the original Structure2Vec model implementation. We built our gUNets model using four gPool layers with gPool ratios [0.8 0.8 0.8 0.8], a layer dimension of 32, a hidden size of 128, a batch size of 32, and a learning rate of 0.004. All other parameters are identical to those in the original gUNets model.

Table 4. A range of methods (DGCNN, S2V, GIN, LCGNN, and G-U-Net) were applied to the RNA graph representations. These representations contain continuous node labels with dimensions of (1 – 2 - 3)

	RNA-XYZ	RNA-A-II	RNA-A-II	RNA-ABC-II	RNA-AC-II	RNA-C-II
DGCNN	84.277	83.596	83.596	85.174	80.757	82.019
GIN	66.038	62.264	62.264	64.151	64.151	64.151
S2V	...	81.073	81.073	...	...	82.650
gUNets	68.553	72.642	72.642	67.610	67.610	73.899

Table 5. The results are from a range of graph kernels (WL-OA, SP, APC) and deep learning methods (DGCNN, S2V, GIN, $LCGNN_{GIN}$, G-U-Net) on introduced a variety of graph representations. The nodes labels are discrete labels. RNA_R is described in (Algul and Wilson, 2019) as seq+top. Table 1 and Table 2 only represent the result obtained from 3D RNA representation

	RNA-A	RNA-B	RNA-C	RNA-AB	RNA-AC	RNA-BC	RNA-ABC	RNA-SRVF	RNA-SRVF-P	RNA-R	RNA-type
DGCNN	86.1	87.4	84.9	88.0	84.2	85.5	87.1	81.7	82.6	94.0	85.8
GIN	67.29	84.9	80.8	86.8	67.6	86.2	85.2	72.6	76.7	91.2	77.4
S2V	82.3	86.7	84.5	88.0	80.1	85.2	90.2	75.3	79.5	95.3	86.8
gUNets	74.21	86.5	79.9	85.8	74.2	85.2	86.5	73.0	74.8	89.9	75.5
LCGNN$_{GIN}$	72.64	86.2	83.3	85.2	74.5	86.2	85.8	73.0	76.1	93.1	78.0
SP	81.9	87.8	84.9	88.1	82.7	87.7	87.5	82.9	83.4	91.1	86.7
APC	76.1	84.6	82.9	85.3	76.1	85.9	86.2	80.1	81.5	89.9	85.5
WL-OA	78.7	86.9	85.7	87.4	82.4	87.4	87.4	81.6	81.6	92.4	87.1

Our results are presented in Table 4, for the continuous node features discussed in section 5.2. S2V model can only be able to consume graphs with one-dimensional node features. Based on our experiments, we found that the DGCNN method yielded the highest test accuracy results. This method outperformed other methods for all RNA graph representations shown in Table 4. We achieved the

highest accuracy of 84.277 using the RNA-XYZ graph representation. However, this is inferior to the 88.1% obtained by SP+R-ABC.

Then, we applied GNN methods to RNA graph representations that use distinct node labels. The results obtained by the GDL methods are similar to those for the kernel methods and better than the results in Table 4. The DGCNN and SP methods are superior and comparable to each other. The results for all representations are presented in Table 5. In the article (Xu et al., 2019), GIN claims that its method is comparable in effectiveness to WL in classification tasks. Nevertheless, our applications reveal that the GIN method, as shown in Table 5, did not outperform the WL methods. Moreover, in our extensive experiments, the graph kernels achieved the highest accuracy in 6 of the 11 RNA graph representations.

The data presented in Table 5, indicated by the red highlights, demonstrate that $RNA{-}R$ outperforms all other representations. Furthermore, when utilizing the RNA-R representation, the Structure2Vec approach attains the highest performance, reaching 95.3%. RNA-R is the representation of a sequence of the RNA molecules added to the 2D RNA topology. The representation is explained in (Algul & Wilson, 2019) as (Sequence + Topology).

Our comprehensive empirical analysis employed a range of techniques, including classification methods (GNNs and graph kernels) and 3D RNA graph representations for predicting the biological tasks of RNA strands. Through our research, we explored that using discrete node labels was more successful than using continuous node labels for graph-based representations. When comparing graph kernel methods to the GNNs, we discovered that the GNNs generally performed better than the graph kernels in most cases.

8. Conclusion

In this article, we begin by discussing the representation issues and the challenges associated with non-Euclidean data in the context of GDL applications. Next, we review existing 3D RNA representations and introduce novel 3D RNA graph representations that utilize various techniques to encode the geometric RNA shape, including base position, curvature, arc length, and SRVF. We then present the outcomes generated from the 3D RNA graph representations employing graph kernels and compare these results with the ones received from the GNNs.

In the experiments, we successfully transformed the 3D RNA structure as a graph using various representations, where nodes reveal one to three-dimensional features. We applied advanced classification methods to our 3D RNA graph representation and discover that using the introduced graph representations is helpful in accurately predicting RNA functions in comparisons of the 3D RNA representations.

Furthermore, our experiments revealed that kernel methods are successful for RNA graph classification tasks, despite their limitation in using multi-dimensional continuous node labels as input. However, this does not seem to be an issue with RNA graphs, where classification is best using limited node labels. We applied the GNNs using this type of representation and found that graph neural networks provided the best results. GNNs is the most flexible methods as they can be applied to all introduced graph representations. In all our experiments, we achieved significant results on nucleobase sequences with the added 2D RNA topology, obtaining an accuracy of 95.3% in the use of Structure2Vec.

References

Algul, E., & Wilson, R. C. (2019). A Database and Evaluation for Classification of RNA Molecules Using Graph Methods. In D. Conte, J.Y. Ramel & P. Foggia (Eds.), *Graph-Based Representations in Pattern Recognition: 12th IAPR-TC-15 International Workshop, GbRPR 2019. Lecture Notes in Computer Science, vol. 11510* (pp. 78-87). Springer, Cham. doi:10.1007/978-3-030-20081-7_8

Balcerak, A., Trebinska-Stryjewska, A., Konopinski, R., Wakula, M., & Grzybowska, E. A. (2019). RNA–protein interactions: disorder, moonlighting and junk contribute to eukaryotic complexity. *Open Biology*, 9(6), 190096. doi:10.1098/rsob.190096

Borgwardt, K. M., & Kriegel, H. P. (2005). *Shortest-path kernels on graphs*. Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, USA. doi:10.1109/ICDM.2005.132

Carrasco-Hernandez, R., Jácome, R., López Vidal, Y., & Ponce de León, S. (2017). Are RNA viruses candidate agents for the next global pandemic? A review. *ILAR Journal*, 58(3), 343-358. doi:10.1093/ilar/ilx026

Chen, L., Calin, G. A., & Zhang, S. (2012). Novel insights of structure-based modeling for RNA-targeted drug discovery. *Journal of Chemical Information and Modeling*, 52(10), 2741-2753. doi:10.1021/ci300320t

Chen, Z., Li, X., & Bruna, J. (2017). Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*. doi:10.48550/arXiv.1705.08415

Chojnowski, G., Waleń, T., & Bujnicki, J. M. (2013). RNA Bricks-a database of RNA 3D motifs and their interactions. *Nucleic Acids Research*, 42(D1), D123-D131. doi:10.1093/nar/gkt1084

Dai, H., Dai, B., & Song, L. (2016). Discriminative embeddings of latent variable models for structured data. *Proceedings of International Conference on Machine Learning, PMLR*, 48, 2702-2711.

Darty, K., Denise, A., & Ponty, Y. (2009). VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15), 1974-1975. doi:10.1093/bioinformatics/btp250

de Vries, G. K. D. (2013). A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data. In H. Blockeel, K. Kersting, S. Nijssen, F. Železný, (Eds.) *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2013. Lecture Notes in Computer Science(), vol. 8188.* Berlin, Germany: Springer. doi:10.1007/978-3-642-40988-2_39

Ding, Y. (2006). Statistical and Bayesian approaches to RNA secondary structure prediction. *RNA*, 12(3), 323-331. doi:10.1261%2Frna.2274106

Du, S. S., Hou, K., Póczos, B., Salakhutdinov, R., Wang, R., & Xu, K. (2019). Graph neural tangent kernel: Fusing graph neural networks with graph kernels. Advances in Neural Information Processing Systems, 32, *ArXiv, abs/1905.13192.* doi:10.48550/arXiv.1905.13192

Gao, H., & Ji, S. (2019). Graph U-Nets. *Proceedings of the 36th International Conference on Machine Learning, PMLR,* 97, 2083-2092.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning, PMLR,* 70, 1263-1272.

Giscard, P.-L., & Wilson, R. C. (2017). The all-paths and cycles graph kernel. *arXiv preprint arXiv:1708.01410.* doi:10.48550/arXiv.1708.01410

Hajiaghayi, M., Condon, A., & Hoos, H. H. (2012). Analysis of energy-based algorithms for RNA secondary structure prediction. *BMC Bioinformatics*, 13(1), 22. doi:10.1186/1471-2105-13-22

Hermansson, L., Johansson, F. D., & Watanabe, O. (2015). *Generalized shortest path kernel on graphs*. In Discovery Science,18th International Conference, DS 2015, Banf, AB, Canada.

Huang, H.-Y., & Lin, C.-J. (2016). Linear and kernel classification: When to use which? *Proceedings of the 2016 SIAM International Conference on Data Mining*, 216-224. doi:10.1137/1.9781611974348.25

Kang, U., Tong, H., & Sun, J. (2012). Fast random walk graph kernel. *Proceedings of the 2012 SIAM International Conference on Data Mining,* 828-838. doi:10.1137/1.9781611972825.71

Kerpedjiev, P., Höner zu Siederdissen, C., & Hofacker, I. L. (2015). Predicting RNA 3D structure using a coarse-grain helix-centered model. *RNA*, 21, 1110-1121. doi:10.1261%2Frna.047522.114

Kim, N., Zahran, M., & Schlick, T. (2015). Computational prediction of riboswitch tertiary structures including pseudoknots by RAGTOP: a hierarchical graph sampling approach. *Methods in Enzymology*, 553, 115-135. doi:10.1016/bs.mie.2014.10.054

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*. doi:10.48550/arXiv.1609.02907

Klosterman, P. S., Tamura, M., Holbrook, S. R., & Brenner, S. E. (2002). SCOR: A structural classification of RNA database. *Nucleic Acids Research,* 30(1), 392-394. doi:10.1093/nar/30.1.392

Kriege, N. M, Giscard, P-L., & Wilson, R. C. (2016). On valid optimal assignment kernels and applications to graph classification. In D. D. Lee, U. von Luxburg, R. Garnett, M. Sugiyama, & I. Guyon (Eds.), *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (pp. 1623-1631). Curran Associates Inc.

Kriege, N. M., Johansson, F. D., & Morris, C. (2020). A survey on graph kernels. *Applied Network Science*, 5(1), 1-42. doi:10.1007/s41109-019-0195-3

Laborde, J., Srivastava, A., & Zhang, J. (2011). Structure-based RNA function prediction using elastic shape analysis. *IEEE International Conference on Bioinformatics and Biomedicine,* 16-21. doi:10.1109/BIBM.2011.119

Laborde, J., Robinson, D., Srivastava, A., Klassen, E., & Zhang, J. (2013). RNA global alignment in the joint sequence–structure space using elastic shape analysis. *Nucleic Acids Research*, 41(11), e114. doi:10.1093/nar/gkt187

Laing, C., Jung, S., Kim, N., Elmetwaly, S., Zahran, M., & Schlick, T. (2013). Predicting helical topologies in RNA junctions as tree graphs. *PLoS ONE*, 8(8), e71947. doi:10.1371/journal.pone.0071947

Lau, M., & Ferré-D'Amaré, A. (2016). Many activities, one structure: Functional plasticity of ribozyme folds. *Molecules*, 21(11), 1570. doi:10.3390/molecules21111570

Liu, W., Srivastava, A., & Zhang, J. (2010). Protein structure alignment using elastic shape analysis. *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, 62-70. doi:10.1145/1854776.1854790

Magnus, M., Kappel, K., Das, R., & Bujnicki, J. M. (2019). RNA 3D structure prediction guided by independent folding of homologous sequences. *BMC Bioinformatics*, 20(1), 512. doi:10.1186/s12859-019-3120-y

Miao, Z., & Westhof, E. (2017). RNA structure: Advances and assessment of 3D structure prediction. *Annual Review of Biophysics*, 46(1), 483-503. doi:10.1146/annurev-biophys-070816-034125

Mjaavatten, A. (2020). *Curvature of a 1D curve in a 2D or 3D space*. MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/69452-curvature-of-a-1d-curve-in-a-2d-or-3d-space Access date: 20 March 2023.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3), 443-453. doi:10.1016/0022-2836(70)90057-4

Neumann, M., Garnett, R., Bauckhage, C., & Kersting, K. (2015). Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102, 209-245. doi:10.1007/s10994-015-5517-9

Nova, D., & Estévez, P. A. (2013). A review of learning vector quantization classifiers. *Neural Computing and Applications*, 25(3-4), 511-524. doi:10.1007/s00521-013-1535-3

Oliver, C., Mallet, V., Philippopoulos, P., Hamilton, W. L., & Waldispühl, J. (2022). Vernal: a tool for mining fuzzy network motifs in RNA. *Bioinformatics*, 38(4), 970-976. doi:10.1093/bioinformatics/btab768

Pande, V., & Nilsson, L. (2008). Insights into structure, dynamics and hydration of locked nucleic acid (LNA) strand-based duplexes from molecular dynamics simulations. *Nucleic Acids Research*, 36(5), 1508-1516. doi:10.1093/nar/gkm1182

Petrov, A. I., Zirbel, C. L., & Leontis, N. B. (2013). Automated classification of RNA 3D motifs and the RNA 3D Motif Atlas. *RNA*, 19(10), 1327-1340. doi:10.1261%2Frna.039438.113

Purzycka, K. J., Adamiak, R. W., Blazewicz, J., Popenda, M., Szachniuk, M., Antczak, M., & Lukasiak, P. (2015). Automated 3D RNA structure prediction using the RNAComposer method for Riboswitches1. *Methods in Enzymology,* 553, 3-34. doi:10.1016/bs.mie.2014.10.050

Reinharz, V., Soulé, A., Westhof, E., Waldispühl, J., & Denise, A. (2018). Mining for recurrent long-range interactions in RNA structures reveals embedded hierarchies in network families. *Nucleic Acids Research*, 46(8), 3841-3851. doi:10.1093/nar/gky197

Ren, Y., Bai, J. & Zhang, J. (2021). Label contrastive coding based graph neural network for graph classification. *Database Systems for Advanced Applications*, 123-140. doi:10.1007/978-3-030-73194-6_10

Ribeiro, L., Saverese, P., & Figueiredo, D. (2017). struc2vec: Learning node representations from structural identity. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 385-394. doi:10.1145/3097983.3098061

Schneider, P., Biehl, M., & Hammer, B. (2009). Distance learning in discriminative vector quantization. *Neural Computation*, 21(10), 2942-2969. doi:10.1162/neco.2009.10-08-892

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., & Borgwardt, K. M. (2011). Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12, 2539-2561.

Verbeek, P., & Vliet, L.V. (1993). *Curvature and bending energy in digitized 2D and 3D images*. In 8th Scandinavian Conference on Image Analysis, Tromso, Norway.

Wilson, R. C., & Algul, E. (2018). Categorization of RNA Molecules Using Graph Methods. In: X. Bai, E. Hancock, T. Ho, R. Wilson, B. Biggio, & A. Robles-Kelly (Eds) *Structural, Syntactic, and Statistical Pattern Recognition*. *S+SSPR 2018. Lecture Notes in Computer Science*, vol 11004 (pp. 439-448). Springer, Cham. doi:10.1007/978-3-319-97785-0_42

x3dna.org. (n.d.). *3dna: a suite of software programs for the analysis, rebuilding and visualization of 3-dimensional nucleic acid structures*. http://x3dna.org/ Access date: 20 March 2023.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., & Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. *Proceedings of the 35th International Conference on Machine Learning, PMLR,* 80, 5453-5462.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? *arXiv preprint*, 1810.00826. doi:10.48550/arXiv.1810.00826

Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H., & Westhof, E. (2003). Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Research*, 31(13), 3450-3460. doi:10.1093/nar/gkg529

Zahran, M., Sevim Bayrak, C., Elmetwaly, S., & Schlick, T. (2015). RAG-3D: a search tool for RNA 3D substructures. *Nucleic Acids Research*, 43(19), 9474-9488. doi:10.1093/nar/gkv823

Zhang, M., & Chen, Y. (2018). *Link prediction based on graph neural networks*. Advances in Neural Information Processing Systems, Curran Associates, Inc.

Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018). An end-to-end deep learning architecture for graph classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 4438-4445. doi:10.1609/aaai.v32i1.11782

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019a). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1), 11. doi:10.1186/s40649-019-0069-y

Zhang, Z., Bu, J., Ester, M., Zhang, J., Yao, C., Yu, Z., & Wang, C. (2019b). Hierarchical graph pooling with structure learning. *arXiv:1911.05954*. doi:10.48550/arXiv.1911.05954

Zhao, T., Zhang, X., & Wang, S. (2021). GraphSMOTE: Imbalanced node classification on graphs with graph neural networks. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 833-841. doi:10.1145/3437963.3441720