THE JOURNAL OF PHYSICAL CHEMISTRY B
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCB

Article

A Fiedler Vector Scoring Approach for Novel RNA Motif Selection

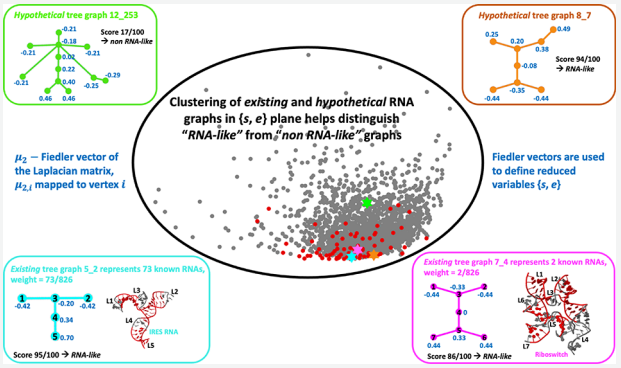
Published as part of The Journal of Physical Chemistry virtual special issue "Ruth Nussinov Festschrift".

Qiyao Zhu and Tamar Schlick*

Cite This: *J. Phys. Chem. B* 2021, 125, 1144–1155

Read Online

ABSTRACT: Novel RNA motif design is of great practical importance for technology and medicine. Increasingly, computational design plays an important role in such efforts. Our coarse-grained RAG (RNA-As-Graphs) framework offers strategies for enumerating the universe of RNA 2D folds, selecting "*RNA-like*" candidates for design, and determining sequences that fold onto these candidates. In RAG, RNA secondary structures are represented as tree or dual graphs. Graphs with known RNA structures are called "existing", and the others are labeled "hypothetical". By using simplified features for RNA graphs, we have clustered the hypothetical graphs into "RNA-like" and "non-RNA-like" groups and proposed RNA-like graphs as candidates for design. Here, we propose a new way of designing graph features by using Fiedler vectors. The new features reflect graph shapes better, and they lead to a more clustered organization of existing graphs. We show significant increases in K-means clustering accuracy by using the new features (e.g., up to 95% and 98% accuracy for tree and dual graphs, respectively). In addition, we propose a scoring model for top graph candidate selection. This scoring model allows users to set a threshold for candidates, and it incorporates weighing of existing graphs based on their corresponding number of known RNAs. We include a list of top scored RNA-like candidates, which we hope will stimulate future novel RNA design.



INTRODUCTION

Aside from RNAs that act as templates for translation into proteins, microRNAs, silencing RNAs, ribozymes, and riboswitches have central roles in catalysis, gene regulation, and gene editing activities.[1,2] The 3D structures of these noncoding RNAs are essential for completing their tasks. Since the first RNA structure published in 1965,[3] thousands of RNA structures have been determined by X-ray crystallography, NMR spectroscopy, cryo-EM, and other experimental techniques. Figure 1 displays the number of RNA structures available in Protein Data Bank (PDB) from 1978 to 2019 (https://www.rcsb.org/stats/growth/growth-rna).

The fast growing RNA databases suggest that our known structural repertoire is just the tip of the iceberg of the RNA universe. Discovering and designing new RNA folds have important implications to technology and medicine. Indeed, RNA nanotechnology is an emerging field for RNA-targeting therapeutics. RNAs like aptamers, silencing RNAs, ribozymes, and riboswitches can be applied for medical diagnosis, targeted drug delivery, and gene silencing and regulation, with possibly reduced side effects and immune responses compared with antibody- and small-molecule-based therapeutics.[4,5] RNA-based vaccines have now become a reality to fight the COVID-19 pandemic, with two mRNA-based vaccines by

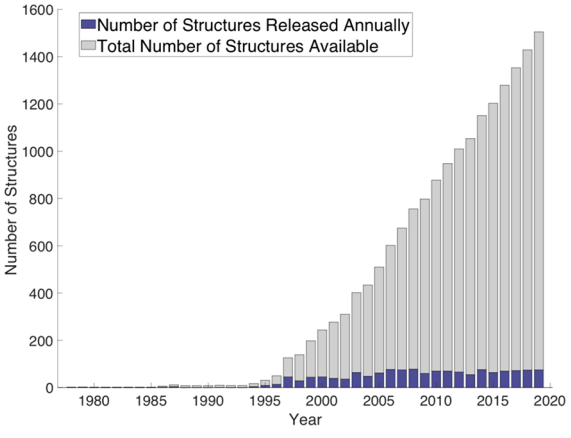


Figure 1. Number of RNA structures available in PDB.

Received: November 28, 2020
Revised: January 6, 2021
Published: January 20, 2021

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144–1155

Pfizer/BioNTech and Moderna with >90% efficacy entering the clinic at warp speed.

The building blocks of these RNA therapeutics often require prediction and design of RNA secondary and tertiary structures.[4] Secondary structures of RNAs refer to the hydrogen bonding networks that form as the single-stranded RNA molecule folds upon itself to form double-stranded regions (stems), imperfect with loops. Tertiary structures involve the folding in space of these networks. While many programs like ViennaRNA,[6] MFOLD,[7] PKNOTS,[8] NUPACK,[9] and INFO-RNA[10] can effectively predict and design RNA secondary structures with or without pseudoknots (or intertwined hydrogen-bonded segments), accurate and consistent RNA tertiary structure prediction remains a challenge.[11−15] The large number of degrees of freedom in building RNA 3D atomic models is a key difficulty, and thus coarse-grained approaches like our RNA-As-Graphs (RAG) framework developed since 2003[16] provide viable alternatives. (See refs 17 and 18 for recent reviews of simplified approaches to RNA modeling.)

Graphs have been used to describe RNA secondary structures since the 1980s.[19−22] Ruth Nussinov, to whom this article is dedicated, made many pioneering contributions to RNA representations and structure analysis, including proposing the usage of circular plots to represent RNA pseudoknots (see Figure 2b).[23]

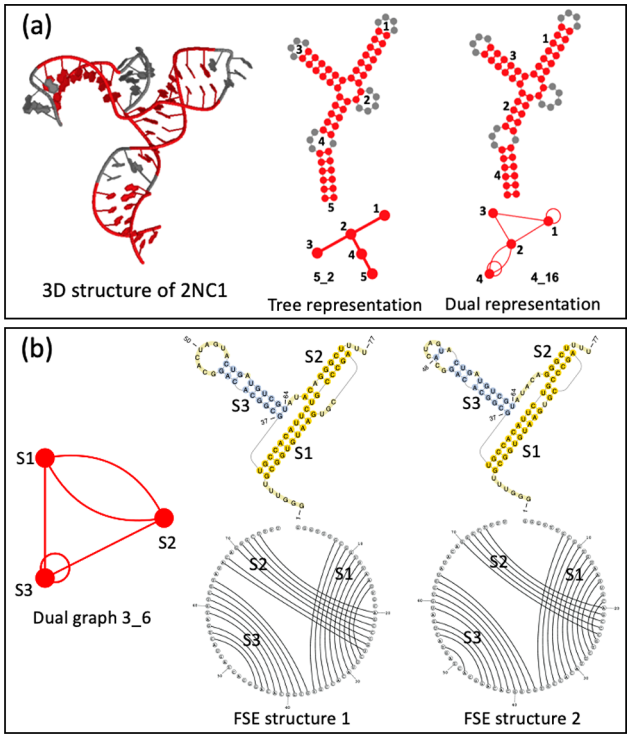


Figure 2. (a) Tree and dual graph representation of the 2D structure of an IRES RNA (PDB code 2NC1). In its 3D structure, stems are colored red and loops are gray. With loops (gray) labeled as vertices and with stems (red) labeled as edges, its 2D structure can be represented as RAG tree graph 5_2. With stems labeled as vertices and loops as edges, the 2D structure can be represented as RAG dual graph 4_16. (b) Two possible 2D structures of a pseudoknot of the SARS-CoV-2 frameshifting element, with associated circular diagrams, and their common dual graph representation 3_6.

In our RAG approach, we represent RNA 2D structures as tree or dual graphs: for tree graphs, loops (hairpins, bulges, internal loops, junctions, dangling ends) are represented as vertices, and stems are edges. For dual graphs, we reverse this definition so that vertices represent stems, and edges denote loops.[24] Dual graphs represent pseudoknots explicitly, while tree graphs are more intuitive.

Figure 2a illustrates the tree and dual graphs of an IRES RNA (PDB code 2NC1). This coarse-grained representation of a 2D strcture significantly reduces the dimension of the conformational space compared to the sequence space and allows us to enumerate all possible nonisomorphic graph topologies for a given number of vertices using graph theory enumeration.[25,26] Another advantage of graph representation is its insensitivity to small variations in base pairing. Figure 2b shows two possible 2D structures of the 3_6 pseudoknot of the SARS-CoV-2 frameshifting element (FSE).[27−29] Although the two structures have different stem and loop sizes (see associated circular diagrams), their overall topologies are the same: both have the dual graph 3_6 representation. When studying RNAs whose functions rely on their 2D structures, such as this FSE pseudoknot, focusing on the overall topology helps us better distinguish among and classify RNA conformations. See our work on this RNA using graph theory to define drug−target residues, interpret COVID-19 related frameshifting mechanisms, and the relevance of several graphs to the conformational space.[29,30]

We label those graphs that have corresponding known RNA structures as "existing" and the others as "hypothetical". Each graph has a unique identification number. Using solved RNA structures in PDB, we have found 80 existing tree graphs out of the total 2287 tree graphs that have 2−13 vertices, and 121 existing dual graphs out of the total 110 667 dual graphs that have 2−9 vertices.[31]

We have further applied graph theory to select features for the graph topologies to classify hypothetical graphs into "RNA-like" and "non-RNA-like" motifs.[31] Thus, an RNA-like motif resembles existing topologies, so it would be more likely to exist in nature. Such candidates can then be designed by "inverse folding" (produce sequences that fold onto the target motif) by our computational pipeline.[32] In our pipeline, we first partition the target tree graph into subgraphs using our partitioning algorithm[33] and extract corresponding atomic fragments from our RAG-3D database.[34] Second, we assemble these atomic fragments using our F-RAG tool.[35] Third, the assembled sequences are screened *in silico* by 2D structure prediction programs like RNAfold and NUPACK to determine whether this inverse folding (IF) is successful. Fourth, we mutate sequences that do not fold onto the target tree graph by our genetic algorithm RAG-IF[36] until we obtain successful designs. Experimental testing of two designed sequences using SHAPE-MaP has shown promise.[30,32,37]

In this paper, we improve our graph clustering approach for identifying novel design candidates by using Fiedler vectors, along with a new scoring model. Prior features were derived from the Laplacian spectra of the graphs using linear or quadratic variables, and both unsupervised clustering algorithm K-means and supervised classification k-nearest-neighbors (k-NN) were used to classify the graph topologies.[31] By use of our new features, the accuracy of K-means clustering significantly increases from 77.22% to 95% for tree graphs (linear variables) and from 75.42% to 98% for dual graphs; for quadratic variables, notable improvements also result.[31] The 10-fold

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

cross validation accuracy of k-NN classification using the new features increases to 66−73% for tree graphs (from 58−63% using full linear variables) but decreases slightly from 76−81% using full quadratic variables; for dual graphs, we improve the accuracy to 73−78% compared to full linear variables (63−69%) and it decreased slightly from 76−81% for full quadratic variables.[31] In addition to notable increased classification accuracy compared to linear variables, the new features allow us to incorporate graphs with two vertices and their large associated pool of known RNA structures as fragments in our clustering work.

An added advantage of the Fiedler vector scoring model is the introduction of a threshold value for novel motif candidates. In contrast, K-means clustering often classifies more than 50% of the total graphs as RNA-like, and thus it is difficult to identify top candidates for novel RNA design. Our scoring model also incorporates weighing of existing graph topologies based on their corresponding number of known RNAs, so it effectively uses existing RNA data. With these new features and scoring model, we can thus propose stronger and more targeted candidates for design of novel RNA motifs.

Another interesting application of the Fiedler vector scoring model is to find motifs similar to a given graph. This can be useful for discovering or creating an alternative fold of an RNA. In our recent paper, we applied RAG-IF to define minimal mutations that transform the SARS-CoV-2 FSE pseudoknot into other graph motifs to identify target residues for antiviral drug and gene editing strategies.[29] In this process, determining related graph motifs can be challenging, especially when facing a large pool of candidates. However, our scoring model can analyze the graph motifs to define a ranked list of candidates. Our mutation results[29] align well with our scoring model ranking: highly ranked candidates require fewer mutations.

In the next section (Methods), we present the new Fiedler vector scoring model followed by its motivation and a simple illustration. The Results section compares the clustering for new versus prior features, assesses the scoring model performance, and tests the predictive power of the Fiedler vector scoring model. In the last section, we summarize our findings, discuss applications of our model, and suggest future improvement areas.

METHODS

Basic Definitions. Both tree and dual graphs can be described by their adjacency matrices (see Figure 3 and Figure 4 for examples). A tree graph with n vertices has an $n \times n$ adjacency matrix $\mathbf{A}$, with entries $a_{ij} = 1$ if there is an edge between vertex i and j and $a_{ij} = 0$ otherwise. For dual graphs, self-loops are allowed, and there can be multiple edges connecting two vertices. Hence, the adjacency matrix $\mathbf{A}$ for a dual graph has entries a_{ij} equal to the number of edges between vertex i and j, and $a_{ii} = 2$ if there is a self-loop on vertex i. The degree matrix $\mathbf{D}$ of a graph is an $n \times n$ diagonal matrix, with diagonal entries d_{ii} equal to the number of edges incident on vertex i. The Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{A}$. By construction, the Laplacian matrix is positive semidefinite, with $\lambda_1 = 0$ as its smallest eigenvalue and associated eigenvector $\boldsymbol{\mu}_1 = (1,1, \ldots, 1)^T$. Because our graphs are connected, the second smallest eigenvalue of $\mathbf{L}$, the *Fiedler value* λ_2, is positive.

The Fiedler value describes the algebraic connectivity of a graph, and its corresponding eigenvector is called the *Fiedler vector*. When two simple graphs (no self-loops or multiple edges between two vertices) are compared, the more compact

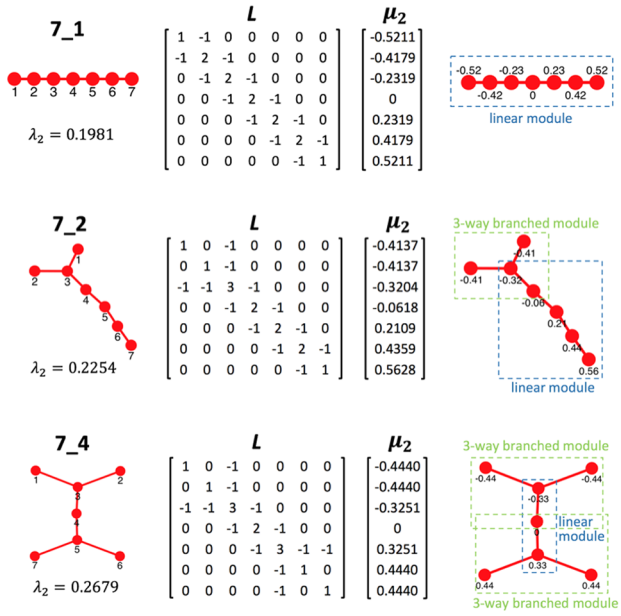


Figure 3. Analysis of three tree graphs 7_1, 7_2, and 7_4. For each tree graph, the vertices are numbered from 1 to 7, and the corresponding Fiedler value λ_2 is shown. The corresponding Laplacian matrix $\mathbf{L}$ and the Fiedler vector $\boldsymbol{\mu}_2$ are shown at center. At right, the Fiedler vector components are mapped onto their corresponding vertices, i.e., $\mu_{2,i}$ for vertex i, and the different modules that make up the graph are colored.

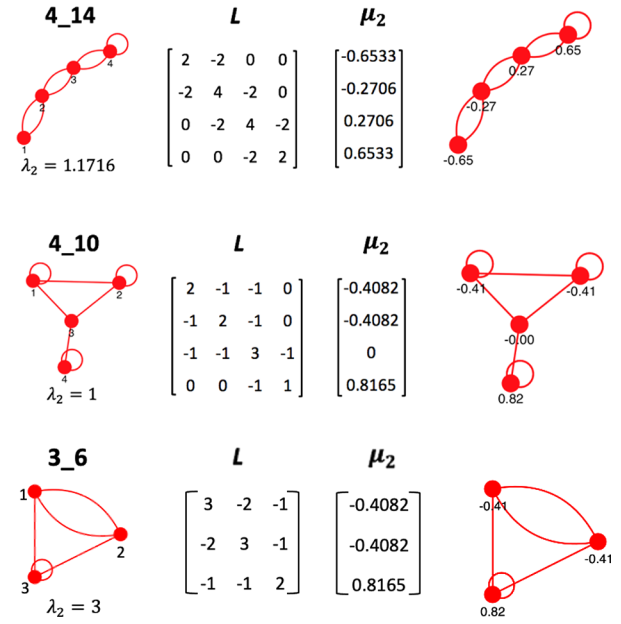


Figure 4. Analysis of three dual graphs 3_6, 4_14, and 4_10. For each dual graph, the vertices are numbered, and the corresponding Fiedler value λ_2 is shown. The corresponding Laplacian matrix $\mathbf{L}$ and the Fiedler vector $\boldsymbol{\mu}_2$ are shown at center. At right, the Fiedler vector components are mapped onto their corresponding vertices, i.e., $\mu_{2,i}$ for vertex i.

graph has a larger Fiedler value (a simple explanation is provided in Appendix C in Supporting Information). Tree graphs are simple graphs, and more compact tree graphs correspond to more branched RNAs or RNAs with more junctions. Figure 3 illustrates three tree graphs with increasing compactness.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

Because the Laplacian **L** is symmetric, its eigenvectors are orthogonal. Hence, the Fiedler vector $\boldsymbol{\mu_2}$ is orthogonal to $\boldsymbol{\mu_1} = (1,1, \ldots, 1)^T$; i.e., its components sum up to 0. In spectral partitioning, Fiedler vectors are utilized to identify graph cuts that optimize different conditions.[38] For tree graphs, we have used Fiedler vectors to partition RNAs.[33]

Fiedler Vector Scoring Model. Though the Fiedler value λ_2 measures a graph's compactness, using it alone is insufficient to distinguish among graphs. Our previous approach derived features from the Laplacian spectra $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_n$ for all graphs with $n \geq 3$ vertices, as follows (implementation details in Appendix A).[31]

Prior Features

(1) Perform linear regression for eigenvalue points $(1, \lambda_2)$, $(2, \lambda_3)$, ..., $(n-1, \lambda_n)$ to obtain slope α_1 and y-intercept β_1. Scale α_1 as $n\alpha_1$ to be independent of n.

(2) Perform linear OR quadratic regression for squared eigenvalue points $(1, \lambda_2^2)$, $(2, \lambda_3^2)$, ..., $(n-1, \lambda_n^2)$ to obtain scaled slope $n\alpha_2$ and y-intercept β_2. Alternatively, we derive coefficients a, b, c for the polynomial $ax^2 + bx + c$ by quadratic regression.

(3) Together, we call $[n\alpha_1, \beta_1, n\alpha_2, \beta_2]$ full linear variables and we call $[n\alpha_1, \beta_1, a, b, c]$ full quadratic variables. To ensure the variables contribute equally, we normalize them to obtain same mean values as $n\alpha_1$.

(4) Use principal component analysis (PCA) to select two features from the full linear/quadratic variables, and call the features reduced linear/quadratic variables

This feature selection is heuristic. To develop features that better reflect the graph topologies, we are motivated by the observation of correspondence between Fiedler vector components and graph structure (see next section). This leads us to the following definition of features s and e.

New Features s and e

(1) Calculate the normalized Fiedler vector $\boldsymbol{\mu_2} = (\mu_{2,1}, \mu_{2,2}, \ldots, \mu_{2,n})^T$ of the Laplacian matrix **L**.

(2) Sort the Fiedler vector components $\{\mu_{2,i}\}_{i=1}^n$ in ascending order and denote the ordered components $\{v_i\}_{i=1}^n$.

(3) Scale each v_i to be

$$\tilde{v}_i = \frac{v_i(n-1)}{v_n - v_1}$$

(4) Perform linear regression on the points $(1, \tilde{v}_1)$, $(2, \tilde{v}_2)$, ..., $(n, \tilde{v}_n)$ to obtain slope s and mean squared error e.

Using new features $\{s, e\}$, we score tree and dual graphs separately with additional weighing information for existing graphs. For M existing graphs and N total graphs, we order *existing* graphs as $1 \leq i \leq M$ and all graphs as $1 \leq j \leq N$. Each existing graph i has a weight w_i, which is the number of known RNAs corresponding to this graph topology. Then we score the graphs as follows; see below for motivation.

Scoring Model

(1) For existing graph i with weight w_i, its initial score is

$$ES_i = \sigma \log(w_i) + \varepsilon \quad (1)$$

where σ and ε are adjustable parameters.

(2) Suppose existing graph i has features (s_i, e_i) and graph j has (s_j, e_j), then the score that graph j receives from existing graph i is

$$S_{j,i} = ES_i \exp[-(rd_{ij}/x_i)] \quad (2)$$

where r is a parameter, $x_i = \sqrt{(s_i^2 + e_i^2)}$, and $d_{ij} = \sqrt{(s_i - s_j)^2 + (e_i - e_j)^2}$.

(3) Sum up the scores that graph j receives from all the existing graphs, i.e.,

$$S_j = \sum_{i=1}^{M} S_{j,i} \quad (3)$$

(4) Normalize the scores to be in the range from 0 to 100 by $\frac{100S_j}{\max_j S_j}$.

Note that the scoring model works for any pair of features $\{f_1, f_2\}$. Here this pair is $\{s, e\}$.

Feature Selection Motivation. If we examine the Fiedler vectors for different tree graphs, we observe a one-to-one correspondence between the Fiedler vector components and the tree graph vertices. For a tree graph with n vertices, each eigenvector has n components. If we assume the Fiedler value λ_2 is simple, then the normalized Fiedler vector $\boldsymbol{\mu_2}$ is unique up to a sign change. Once the vertices of the graph are numbered, $\boldsymbol{\mu_2}$ is fixed so that we can associate each vector component $\mu_{2,i}$ with vertex i.

To describe the correspondence between Fiedler vector components and graph vertices, we first define two basic modules that make up tree graphs (see Figure 3). A linear module is composed of m vertices that are connected in a line by $m-1$ edges ($m \geq 3$). In this line, the two end vertices have degree 1 and the others have degree 2. A k-way branched module is a k-way junction represented by RAG tree graph. Its k branch vertices are adjacent to the center vertex, making its degree k. Any two modules can be combined by overlapping a common edge. Because we seek to divide a tree graph into distinct topologies, there is no combination of a linear module with another linear one.

In the last column of Figure 3, we label the component modules of sample tree graphs and indicate corresponding Fielder vector components. We see that within linear modules, the Fiedler vector components increase in value. Within 3-way branched modules, the two free end branch vertices have same Fielder vector component values. Moreover, if the free end branch vertices precede the center vertex, their Fiedler vector components are smaller than that of the center vertex; otherwise, they are larger. In this way, the Fiedler vector components increase monotonically from one end of the tree graph to the other. Mathematical explanations for these observations are in Appendices B and C.

Although dual graphs may contain pseudoknots that complicate the topology, similar observations apply. Figure 4 shows three dual graphs with their Fiedler vectors. Dual graph 4_14 is an analogue of a linear module, and again its Fiedler vector components monotonically increase. Dual 4_10 contains a 3-way junction, with vertices 1 and 2 representing the two free end helical arms. These two vertices are analogous to the free end branches in tree graph 3-way branched module, so again, they have the same Fiedler vector component value, and it is smaller than that of the following center vertex 3. Dual

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

graph 3_6 (associated with the SARS-CoV-2 frameshifting element[29,30]) is a 3-stem pseudoknot, where stems 1 and 2 are intertwined and stem 3 is a hairpin. Vertices 1 and 2 are connected to other vertices in the same way, just like the free end branch vertices. As expected, they have the same Fiedler vector component values.

Thus, the way vertices are arranged in graphs is reflected in the Fiedler vector components $\mu_{2,i}$. Because adjacent vertices have similar $\mu_{2,i}$ values and the values increase monotonically from one end of the graph to the other, the distribution of the ordered components v_i may capture graph topology. To make the distribution independent of n, we scale $v_n - v_1$ to be $n - 1$, i.e.,

$$\tilde{v}_i = \frac{v_i(n-1)}{v_n - v_1}$$

Figure 5 plots points $\{(i, \tilde{v}_i)\}_{i=1}^{n}$ for four sample tree graphs. For simple linear-moduled tree graphs like 7_1, the points $(i, \tilde{v}_i)$

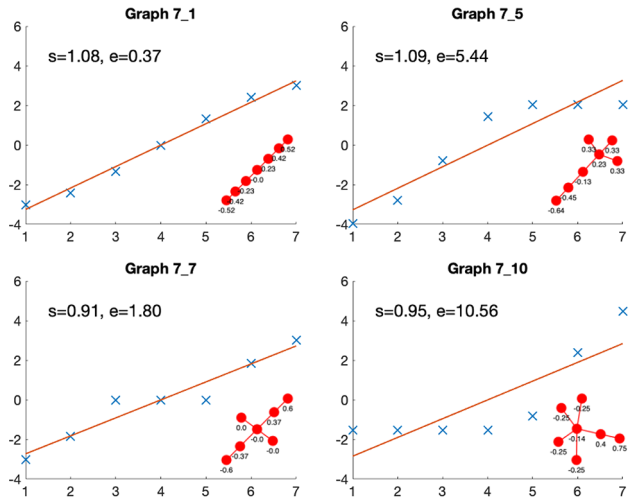


Figure 5. Plots of scaled ordered Fiedler vector points $(i, \tilde{v}_i)$ of four tree graphs 7_1, 7_5, 7_7, and 7_10. Linear least-squares regressions are drawn as red lines, and the slopes s and mean suqared errors e are calculated.

increase in a straight line. For k-way branched modules, the free end branch vertices give repeated $\tilde{v}_i$ values, and the distributions reflect this. For example, the 4-way branched module of 7_5 has three repeated values for i = 5, 6, 7, so branching is at the end. For graph 7_7, the three repeated values for i = 3, 4, 5 indicate branching in the middle.

We use linear least-squares regression to describe the point distributions and look at slopes s and mean squared errors e. For linear modules like graph 7_1, the points fit the linear regression well, so slopes s are close to 1 and errors e are very small. For k-way branched modules, repeated $\tilde{v}_i$ values make the points deviate from the linear fit, and errors e are larger. The locations of these repeated values also influence s and e. Having the values in the middle like graph 7_7 decreases s and e, compared to branching at the end like graph 7_5. The situation is similar for dual graphs. Thus, we let the slope s and the mean squared error e be features for our graphs, and this feature selection works for all graphs with $n \geq 2$.

The slope s can be calculated explicitly (derivation provided in Appendix D):

$$s = \frac{12}{(n-1)n(n+1)} \sum_{i=1}^{n} i\tilde{v}_i = \frac{12}{n(n+1)} \sum_{i=1}^{n} \frac{iv_i}{v_n - v_1}$$

Because $\sum_i v_i = 0$ and

$$\frac{v_n}{v_n - v_1} - \frac{v_1}{v_n - v_1} = 1$$

the $\frac{v_i}{v_n - v_1}$ terms are of scale ~1 and the summation is

$$\sum_{i=1}^{n} \frac{iv_i}{v_n - v_1} \sim n(n+1)$$

Hence, our scaling makes the slope s independent of the vertex number n.

Scoring Model Motivation. With any defined pair of features $\{f_1, f_2\}$, we can represent a graph as a point (f_1, f_2) in the plane. Good features should capture key information about the graph's arrangement, so we expect RNA-like topologies to be clustered together, and the closer a graph is to an existing one, the more likely it is to find corresponding RNA structures.

We incorporate existing graphs and their weights (number of known RNAs) to build the scoring model, where the score assigned to a graph represents the likelihood of finding RNAs of this topology. The basic idea of our scoring model is to treat every existing graph like a hotspot radiating heat. The radiation decreases exponentially with distance. An existing graph with larger weight exhibits more radiation. To model more distant graphs with respect to an origin with larger radiation ranges and to reflect absorption of energy from more neighbors, we score graphs by the total amount of heat they receive.

This visual helps explain the first two steps of our scoring model:

(1) The initial score assigned to existing graph i is

$$ES_i = \sigma \log(w_i) + \varepsilon$$

where the weight w_i is the number of known RNAs corresponding to this graph topology. This is an increasing function of weight. Using a large σ considers graphs with higher weights to be more important, while a small σ treats all existing graphs equally. Note that ε is added to have nonzero scores to graphs of weight 1. Using a large ε diminishes the impact of weights on initial scores.

(2) Each existing graph contributes scores to the graphs nearby, and the score that graph j receives from existing graph i is

$$S_{j,i} = ES_i \exp[-(rd_{ij}/x_i)]$$

The d_{ij} value is the distance between the two graphs, so this score exponentially decays as distance increases. The parameter r controls the score's decay rate, with larger r meaning more rapid decay. The x_i term is the distance of existing graph i from the origin (0,0); including this term allows us to use existing graphs to influence a range of graphs.

Simple Illustration. To illustrate, we show how two existing tree graphs 2_1 and 7_2 contribute scores to points in the plane in Figure 6. We use our newly derived features s and e. Since we have many existing graphs with weights 1, we set σ = 1 not large. We choose ε = 5 to let the weights have a moderate impact on the initial scores. On the basis of trials, we set r = 1.5. For points in the plane, we sum the scores they receive from 2_1 and 7_2 using eqs 1 and 2. Then we draw a filled contour plot using the scores, i.e., yellow to white for

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

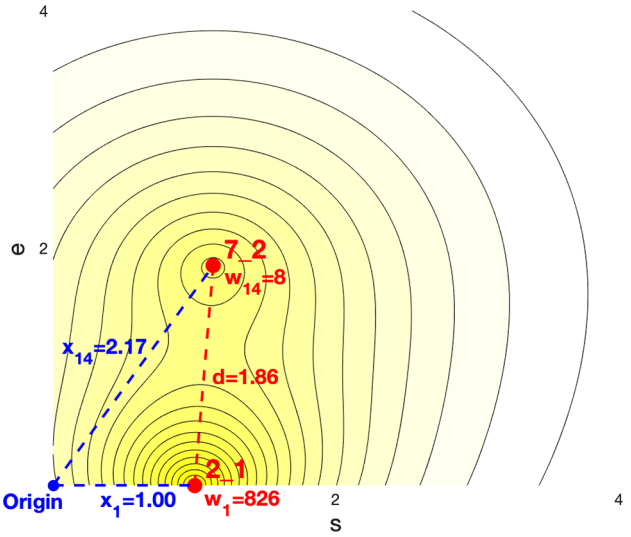


Figure 6. Illustration of how tree graphs 2_1 and 7_2 contribute scores to points in the plane. The new feature s is used for x-axis, and e is used for y-axis.

scores high to low, with 20 contour lines at evenly spaced score levels. As expected, the plot looks like a heat map, with two graphs 2_1 and 7_2 at centers of two "hotspots" that "radiate heat" outward.

There are M = 80 existing tree graphs with 2−13 vertices, and 2_1 is enumerated as the first one with weight w_1 = 826, and 7_2 is enumerated as the 14th with weight w_{14} = 8. By eq 1, the initial score of 2_1 is ES_1 = 11.72, and that of 7_2 is ES_{14} = 7.08. As a result, the neighborhood around 2_1 looks slightly hotter than that of 7_2. Graph 2_1 has s = 1 and e = 0, so its distance from the origin is x_1 = 1; graph 7_2 has s = 1.13 and e = 1.85, so x_{14} = 2.17. Because 7_2 is further away from the origin, it has a larger "radiation range", which can be seen from the contour lines being further apart from each other there.

We can also calculate scores for some graphs. For 7_2, its distance from 2_1 is d = 1.86, so the score 7_2 receives from 2_1 (eq 2) is

$$S_{j,1} = 11.72 \exp[-(1.5 \times 1.86)/1] = 0.72$$

The summed score for 7_2 (eq 3) is S_j = 205.97. The maximum score is 206.71, so following normalization, the score for 7_2 is 99.65. For tree graph 7_1, we obtain score 89.00, and for tree graph 7_4, the score is 86.00. The scores of these three tree graphs should reflect their likelihood to exist in nature.

RESULTS

Clustering Comparisons with Prior Features. *K-Means Comparison.* To see how well our new features s and e work, we compare the clustering results using these new features with those of prior features. We first apply K-means clustering (see details in Appendix A.1), with reduced linear or quadratic variables for comparison. By mapping feature 1/feature 2 (x-axis/y-axis) into the plane, we represent graphs as points in the plane. K-means is then applied to cluster the points into two groups. We label the group with more existing graphs as "RNA-like", and the other as "non-RNA-like". Note that using s and e allows us to add tree graph 2_1 and three dual graphs 2_1, 2_2, 2_3 as existing graphs. They were not considered before because previous feature derivation required graphs to have at least 3 vertices.

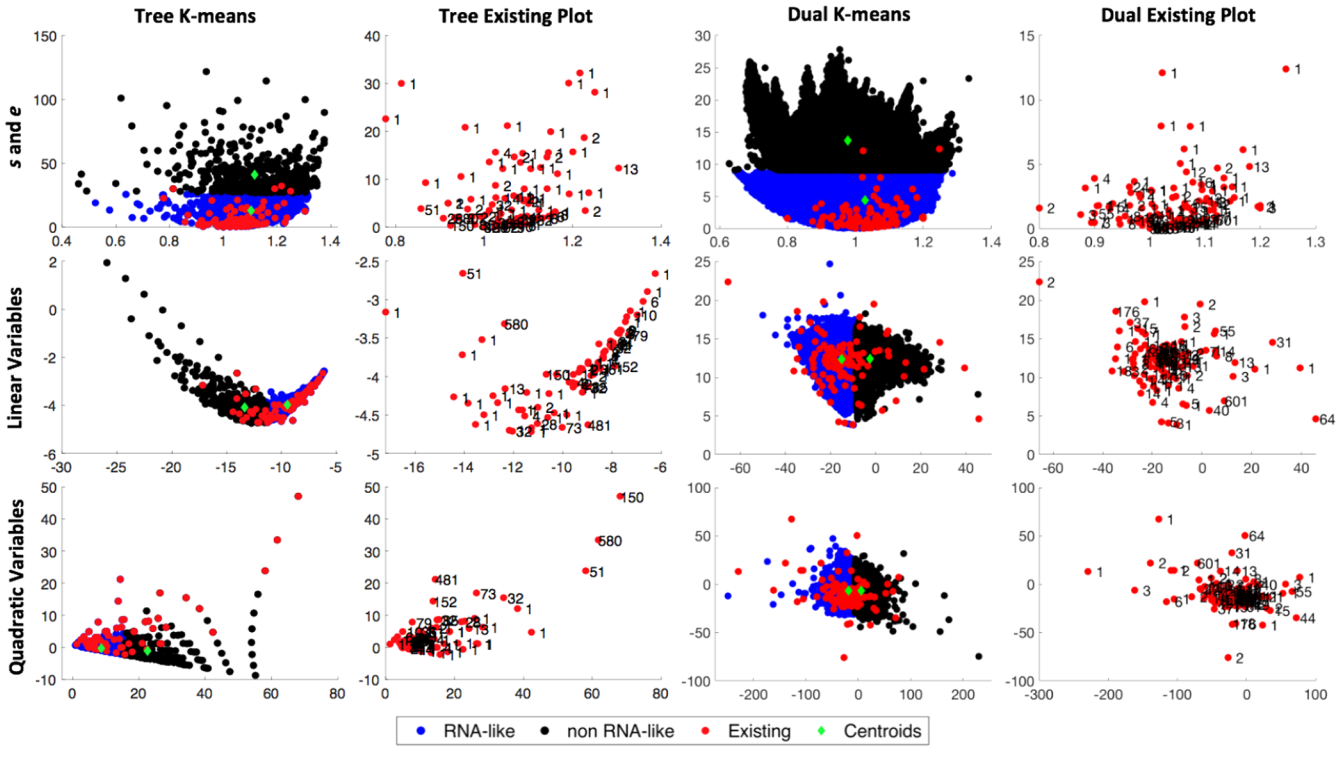


Figure 7. K-means clustering results for the three different feature selections. For s and e features, x-axis is for s and y-axis is for e. For linear/quadratic variables, x-axis is for feature 1 obtained using PCA and y-axis is for feature 2. Distributions for existing graphs are enlarged, with graph weights (number of known RNA structures) shown to the right of the overall distributions.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

Figure 7 shows the K-means plots with a zoom into existing graphs (red). The RNA-like hypothetical points are colored blue, and the non-RNA-like points are black. Compared to prior features, our new s and e approach spreads out the graphs while still clustering existing motifs. Those existing motifs tend to have low e values with s values around 1 (bottom center of plots). On the basis of our observations in Figure 5, graphs with higher-order junctions with branching at the end tend to have larger e values. In our database, there are indeed only a few graphs with five or more-way junctions.[39]

For the existing graphs, weights (from number of known RNA structures) are listed. With the proposed features s and e, existing graphs with heavy weights are highly concentrated at the bottom center, and the graphs far from this center mostly have weights 1. With our prior reduced linear variables, existing tree graphs concentrate at the right end of the plot, but some graphs with heavy weights are far from the center, even the graph with the heaviest weight 580. The observation for dual graphs is similar: existing dual graphs concentrate at the center, with some heavy-weighted graphs further away. Using reduced quadratic variables, some heavy-weighted graphs are positioned away from the centroid of existing graphs.

We can evaluate the clustering performance by calculating the accuracy, which is defined as the percentage of existing graphs correctly classified as RNA-like. The more clustered existing graphs using s and e allow K-means to achieve a higher accuracy. In Figure 7, graphs at the bottom half plane are classified as RNA-like using our new features s and e, and almost all existing graphs fall in this cluster. The accuracies using new and prior features are listed in Table 1. By use of s and e, the accuracy is as high as 95% for tree graphs and 98.35% for dual graphs, compared to 77.22% (linear) and 73.42% (quadratic) for tree graphs and 75.42% (linear) and 72.88% (quadratic) for dual graphs. In Table 1, we also show the percentages of graphs classified as RNA-like and non-RNA-like. By use of s and e, the RNA-like percentage increases from ~50% to 71.15% for dual graphs due to the high densities of graphs with low e values.

Table 1. K-Means Accuracy and Predictions[a]

	tree K-means		
	s and e	linear	quad
accuracy (%)	95.00	77.22	73.42
RNA-like (%)	78.62	71.87	82.68
non-RNA-like (%)	21.38	28.13	17.32
	dual K-means		
	s and e	linear	quad
accuracy (%)	98.35	75.42	72.88
RNA-like (%)	71.15	49.93	51.50
non-RNA-like (%)	28.85	50.07	48.50

[a]For both tree and dual graphs, the K-means clustering accuracy and the percentages of graphs classified as RNA-like and non-RNA-like are calculated, using new s and e features and prior reduced linear/quadratic variables.

k-NN Comparison. We also perform comparative (untrained) k-NN classification (see Appendix A.2 for details) to see how cross validation accuracy changes. The hyperparameter k, number of neighbors, is set to be the odd numbers between 1 and 19. Because we only have M existing graphs as positive data, we synthesize negative data sets by randomly selecting M graphs among the hypothetical and create 10 such negative data sets. Using s and e, we now have M = 80 for tree graphs and M = 121 for dual graphs, while using prior features results in M = 79 for tree graphs and M = 118 for dual graphs. To avoid bias from negative data set selection, the same negative data sets used before[31] are taken for the prior features, and we add additional random hypothetical graphs for the new features. The average accuracies of 10-fold cross validations over the 10 negative data sets are calculated for new features.

In previous work,[31] we found that using full linear/quadratic variables yields higher average accuracies than the reduced ones, so we use those accuracies to compare with corresponding new values using s and e in Table 2. For tree graphs, our new features increase the average accuracy by about 10%, from ~60% to ~70% for full linear variables, but decrease by about 8% from ~78% for full quadratic variables. For dual graphs, we observe a similar increase of around 10% (from ~67% to ~77%) for full linear variables, but there is no significant decrease with full quadratic variables.

Table 2. k-NN Cross Validation Accuracy[a]

	average accuracy (%)					
	tree graphs			dual graphs		
k	s and e	linear	quad	s and e	linear	quad
1	67.19	60.51	76.08	72.93	63.86	78.81
3	66.38	62.78	76.71	74.09	65.59	81.06
5	71.56	59.43	78.61	77.81	66.53	80.17
7	69.38	60.32	80.32	77.40	67.67	79.24
9	71.56	59.43	80.70	77.40	68.01	78.81
11	72.00	58.73	80.32	77.07	68.35	78.94
13	71.88	59.62	78.23	78.18	68.35	78.22
15	72.31	59.62	77.47	78.02	69.19	77.50
17	72.56	60.95	76.39	77.89	68.31	77.25
19	72.75	59.56	76.27	77.69	68.39	76.57

[a]The average 10-fold cross validation accuracy of k-NN classification is taken over 10 negative datasets, using new s and e features and prior full linear/quadratic variables. The average accuracy is shown for k = 1, 3, ..., 19.

Scoring Model Performance. By use of our (untrained) previous clustering methods, many graphs were classified as RNA-like, and it is difficult to select candidates for RNA design. Our current scoring model solves this problem by incorporating the weight information of existing graphs, thus producing far less "false positives".[25] Setting the parameters σ = 1, ε = 5, r = 1.5 (see Methods for how these parameter values are chosen) and using our new features s and e, we show the scores for all tree and dual graphs in Figure 8, where a color bar displays how different colors represent different scores. As expected, high scores are assigned to graphs at the bottom. Moreover, we separate the score histograms for existing and hypothetical graphs. The majority of existing graphs have scores higher than 70 (65% for tree and 73.6% for dual), while only 15.7% of hypothetical tree graphs and 12.4% of hypothetical dual graphs have scores higher than 70.

As our scoring model works for all feature selections, we also calculate the scores for reduced linear/quadratic variables. We list the average scores for all existing and hypothetical graphs for both tree and dual graphs in Table 3. Although using different parameter values, especially different r values, influences the average scores, the overall average score pattern

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

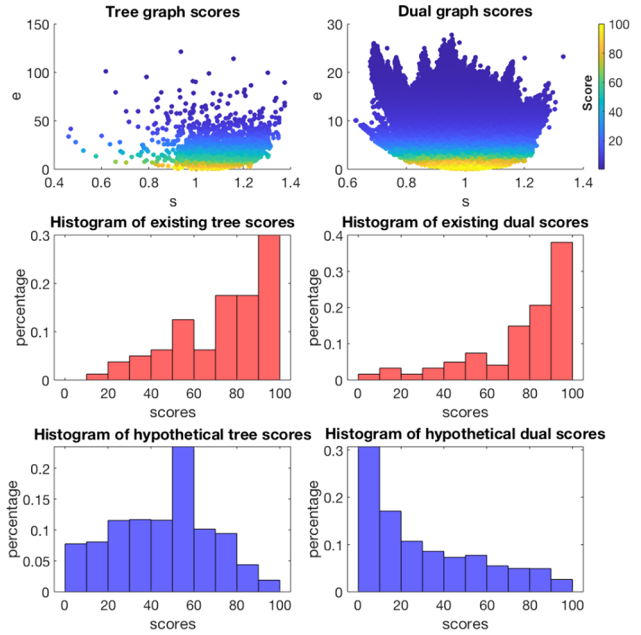


Figure 8. Scoring results for new features s and e. Top two plots show scores of tree and dual graphs, with a color bar indicating how different colors represent different scores. The two middle plots are histograms of existing graph scores on a probability scale. The two bottom plots are histograms of hypothetical graph scores on a probability scale.

Table 3. Average Score Comparisons[a]

	tree graph average scores		
	s and e	linear	quad
all graphs	46.87	88.60	78.12
existing	73.84	90.45	74.51
hypothetical	45.89	88.54	78.25
	dual graph average scores		
	s and e	linear	quad
all graphs	31.11	78.32	71.26
existing	76.94	78.82	67.66
hypothetical	31.06	78.32	71.27

[a]Tree and dual graph average scores using new s and e features or prior reduced linear/quadratic variables. For each feature selection, the average scores of all graphs, of existing graphs, and of hypothetical graphs are listed.

for existing and hypothetical graphs depends on the graph distributions. Compared with prior features, our new s and e approach spreads out the graphs while clustering the existing graphs. Hence, s and e lead to a much larger average score for existing graphs compared to hypothetical graphs. The prior features yielded more similar values for all graphs; for quadratic variables, scores for the hypothetical graphs were even higher than those of existing graphs.

Predictions Using 2015 and 2018 Known RNA Databases. To analyze the predictive power of our model, we perform prediction tests on newly identified existing tree and dual graphs since 2015 and 2018, respectively. The 2015 tree graph database[25] contains 46 of the 80 existing tree graphs, and the 2018 dual graph database[40] contains 87 of the 121 existing dual graphs. We take these older existing graphs as our training set T, and we include the newly added existing graphs and all hypothetical graphs as our test set P. We train our scoring model using existing graphs in set T, and we calculate scores for all graphs. We set a threshold $t = \gamma \times$ average score of set T, where $0 < \gamma \leq 1$. The graphs in P that have scores of $\geq t$ will be considered RNA-like, and the others will be considered as non-RNA-like. Again, we define the accuracy to be the percentage of existing graphs in P that are correctly classifed as RNA-like. We also calculate the percentage of graphs in P that are considered RNA-like. We perform such predictions using new and reduced linear/quadratic variables.

The prediction accuracy and percentage of RNA-like graphs depend on the threshold we set. Since we are clustering graphs, not selecting top candidates, we seek high RNA-like percentages. On the basis of our observations of average scores in Table 3 and some trial and error, we find that for tree graphs, setting γ = 0.5, 1, and 1, respectively for new features, reduced linear and quadratic variables yields 49−63% RNA-like percentage and 76−82% prediction accuracy; for dual graphs, setting γ = 0.4, 1, and 1 yields 33−65% RNA-like percentage and 73−88% prediction accuracy. The corresponding accuracies and RNA-like percentages are recorded in Table 4. We plot the prediction results in Figure 9, where correctly classified and misclassifed newly added existing graphs are represented by different symbols.

Table 4. Prediction Test Results[a]

		tree predictions		
		s and e	linear	quad
scoring	accuracy (%)	82.35	76.47	76.47
	RNA-like (%)	49.49	62.03	63.41
k-NN	accuracy (%)	43.82	35.00	41.76
	RNA-like (%)	11.95	25.69	9.46
K-means	accuracy (%)	97.06	79.41	82.35
	RNA-like (%)	78.62	71.96	82.68
		dual predictions		
		s and e	linear	quad
scoring	accuracy (%)	88.24	73.53	85.29
	RNA-like (%)	33.31	62.34	65.89
k-NN	accuracy (%)	59.12	57.65	62.94
	RNA-like (%)	16.30	23.60	15.22
K-means	accuracy (%)	97.06	85.29	79.41
	RNA-like (%)	71.15	49.93	51.50

[a]By use of new s and e features or prior reduced linear/quadratic variables, the classification accuracy of newly added existing graphs and the percentage of RNA-like graphs are calculated for scoring model, k-NN, and K-means.

We can also compare our scoring model with K-means and k-NN by conducting similar prediction tests. For k-NN, we use T as positive data set, and we randomly generate 10 equal sized negative data sets from P, and we perform predictions for the test set P. The prediction accuracy is defined as above, and we take the average accuracy and RNA-like percentage over 10 trials with the 10 negative data sets. The prediction results are recorded in Table 4, and we plot the results of trial 1 in Figure 9 for comparison. The K-means clustering is not affected by our specification of existing graphs because this unsupervised approach does not rely on our labeling of graphs, but the relevant accuracy changes. These accuracy results are recorded in Table 4 for comparison.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

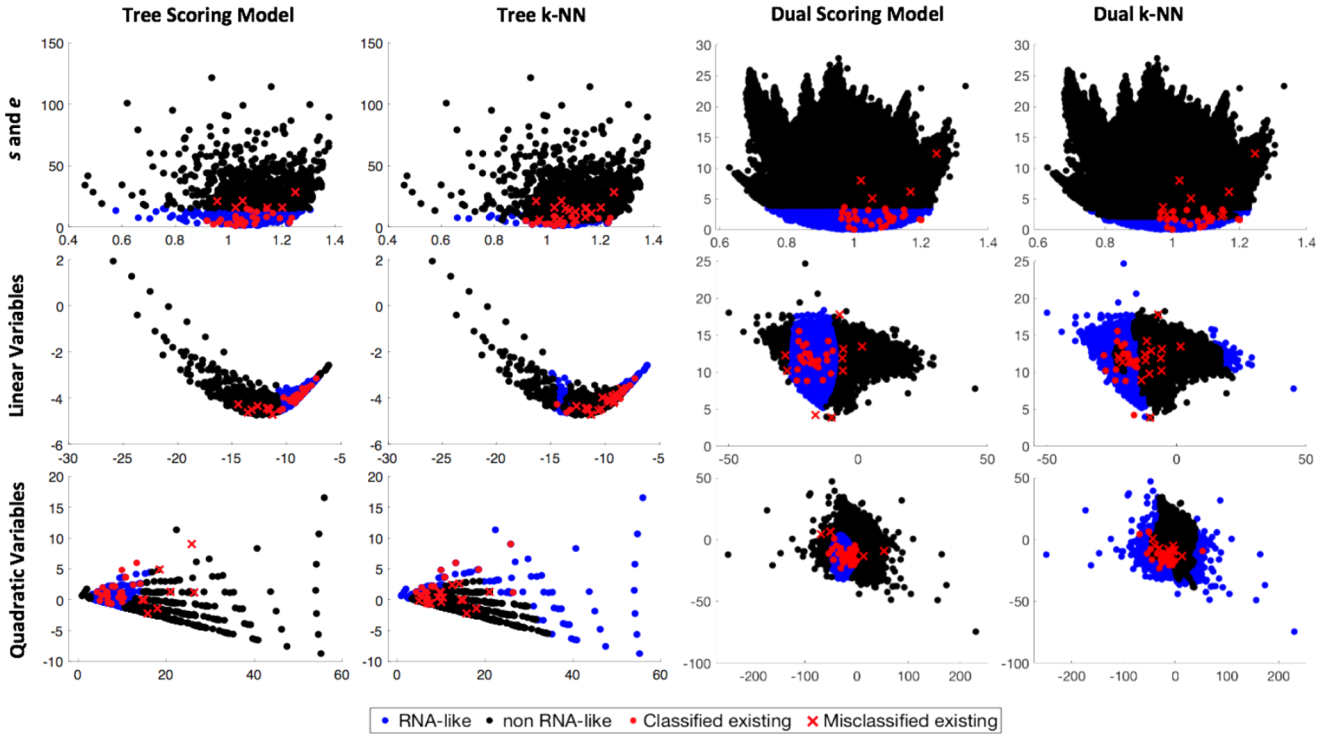


Figure 9. Scoring model and k-NN prediction results using the three feature selections. The newly identified existing graphs are represented by red dots if correctly classified as RNA-like and by red crosses if misclassified as non-RNA-like.

The comparisons in Table 4 show that using new *s* and *e* features achieves highest accuracy except for dual graph predictions using k-NN. Among the three prediction methods, K-means often obtains the highest accuracy of 79−97%, and this is followed by our scoring model with accuracy of 73−88%. The k-NN approach comes last, with accuracy of 35−62%. We observe the same pattern for the RNA-like percentage: K-means obtains 50−82%, scoring model has 33−65%, and k-NN finds 9−25%. Moreover, if we compare the clustering plots of K-means in Figure 7 to the plots of scoring model and k-NN in Figure 9, we see that K-means and our scoring model have similar clustering patterns. These two methods cluster RNA-like graphs together in one region, and the region is similar but smaller for our scoring model, which is consistent with its lower RNA-like percentage. However, except for the *s* and *e* features, k-NN identifies two separate clusters of RNA-like graphs. This may be because using prior features, the existing graphs are more spread out, so the algorithm identifies two RNA-like clusters. Also the negative training data set for k-NN is different each time, so there is some randomness in its clustering.

Overall, we see that our scoring model works best with the new *s* and *e* features. On the basis of the average scores, it distinguishes existing from hypothetical graphs using new features. The average scores are important for selecting suitable threshold in clustering so that the RNA-like percentage is not too low. In the prediction test, our scoring model with new features achieves 82.35% and 88.24% accuracy for tree and dual graphs, respectively. Though the accuracy is lower than the 97.06% using K-means clustering, its RNA-like percentage (49.49% and 33.31%) is lower than that of K-means (78.62% and 71.15%).

DISCUSSION

We have developed a new way of defining feature variables for RAG graph clustering using Fiedler vectors. This feature selection is based on the one-to-one correspondence between graph vertices and Fiedler vector components. By using the slope *s* and the mean squared error *e* of the linear regression for sorted and scaled Fiedler vector components, we find that existing graphs tend to have low *e* values along with *s* values of around 1. When we visualize the graph distributions with *s* and *e* as planar coordinates, we see how this high concentration of existing graphs at the bottom makes it easier for K-means clustering to classify existing graphs into the RNA-like group. As a result, we achieve a significant improvement in K-means clustering accuracy. Only 4 out of 80 existing tree graphs are misclassified, compared to at least 18 out of 79 misclassified with prior variables. For dual graphs, only 2 out of 121 existing are misclassified, compared to at least 29 out of 118 misclassified before. Moreover, these misclassified graphs all have only 1 known RNA structure, while previously, some graphs with large number of RNA structures were also misclassified.

The current K-means misclassified graphs are shown in Figure 10, with corresponding known RNA structures listed. The 4 tree graphs were also misclassified[31] using reduced linear variables, but the 2 dual graphs were correctly classifed before. The current misclassified graphs tend to have a large number of vertices. These motifs tend to have junctions, and thus their *e* values are higher. In K-means clustering, high *e* value graphs are considered non-RNA-like. Because the database of known RNAs has relating few higher-order junctions (five or more-way junctions),[39] clustering based on known structures will inevitably be less accurate for graphs with higher-order junctions.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

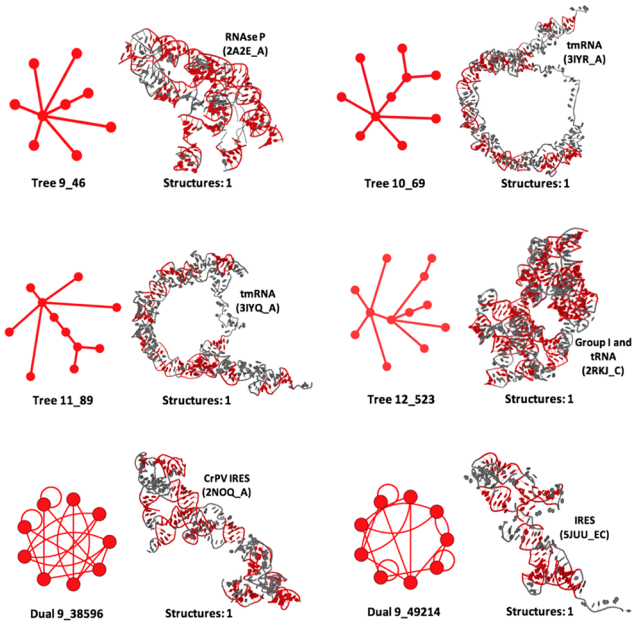


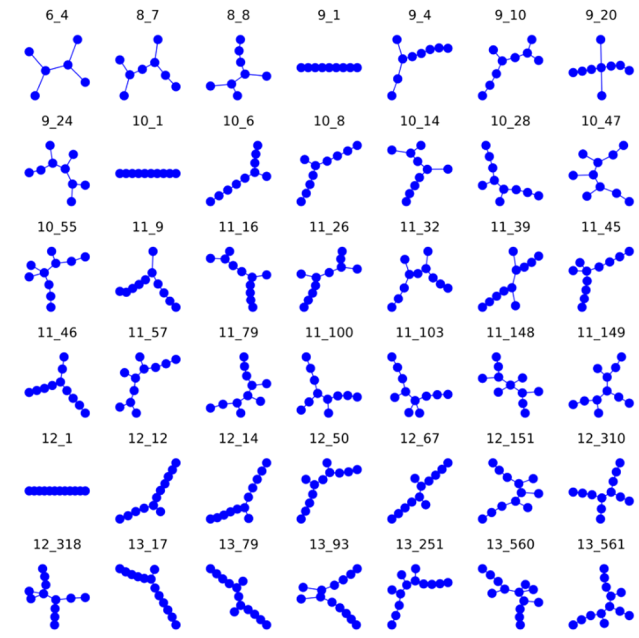
Figure 10. Graphs misclassified as non-RNA-like by K-means using new *s* and *e* features. The corresponding known RNA structures are shown, with stems in red.

For k-NN classification, our 10-fold cross validation accuracy using the new features increases compared with full linear variables, but the accuracy drops when comparing with full quadratic variables, especially for tree graphs. In our prior work,[31] using all five (independent) quadratic variables increased the accuracy by ∼7% compared to a partial set of variables.

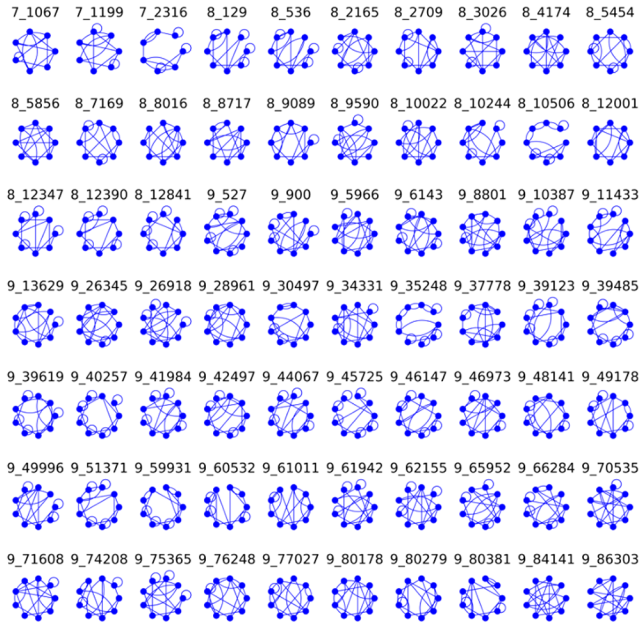
Our proposed scoring model for novel RNA motif selection not only incorporates weights (number of known RNA structures) for existing graphs but also allows setting a threshold for top candidtates. This aspect is particularly attractive for RNA design. We find that our new features work best with the scoring model because the existing graph distributions become more clustered. In Figure 11, we present our top scored RAG hypothetical graphs using our scoring model and *s* and *e* features. The parameters are $\sigma = 1$, $\varepsilon = 5$, $r = 1.5$. For tree graphs, the 42 candidates have scores of ≥90. For dual graphs, the 70 candidates have scores of >99.9. We set a such high threshold for dual graphs because there are 110 546 hypothetical dual graphs up to 9 vertices. As we see from Figure 11, our candidates cover a wide range of number of vertices *n* (most small graphs are existing). Tree graph candidates with both branched structures and highly linear structures appear. This indicates that our algorithm does not have bias for graphs for small *n* or highly linear folds. We hope to explore design of these candidates in future work.

Another potential use of our scoring model is to find similar existing motifs for a given graph. This can be useful for mutation experiments where we seek to change an RNA 2D structure into an alternative, similar graph topology. In our scoring model, scores that a graph receives from all existing motifs are calculated. Existing motifs that are closer to the graph with heavier weights contribute more scores, and these motifs are considered more similar to the given graph. Hence, we can rank the existing motifs in descending order in score.

We can adjust the parameters σ and ε in eq 1 as appropriate (see Methods for more details on parameter value choices). We can also limit the search range to search for existing motifs with vertex number *n* close to that of a given graph. For example, by setting $\sigma = 1$, $\varepsilon = 15$, $r = 1.5$ (we increase ε here to reduce the impact of weights, since we are more interested in graph topology similarity) and limiting *n* to be between 6 and 8, we can perform a motif search for tree graph 7_5. The top 6 similar tree motifs found including 7_5 are shown in order



(a) Tree graph candidates



(b) Dual graph candidates

Figure 11. Top scored RNA-like candidates using new clustering variables *s* and *e*. We show the tree graph candidates with scores of ≥90 and dual graphs with scores of >99.9.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155

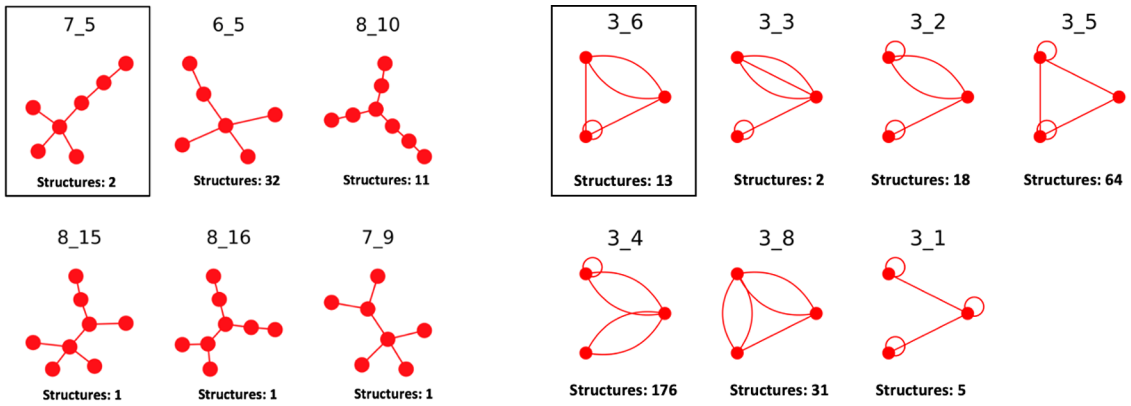


Figure 12. Similar existing motifs for (a) tree graph 7_5 and (b) dual graph 3_6. The similarity ranking goes from left to right.

from left to right in Figure 12a. For each motif, the number of corresponding known RNA structures is written below.

Using the same parameter values and limiting n to be 3, we perform a motif search for dual graph 3_6. The existing motifs are listed in Figure 12b in descending order of similarity from left to right, and their numbers of corresponding known RNA structures are given below. We see that motifs with more known RNAs are not always ranked the highest. In our recent work on the SARS-CoV-2 RNA, we defined minimal mutations that transform a FSE pseudoknot (Figure 2b) with dual graph 3_6 to 5 of the other existing motifs.[29] The top 3 motifs 3_3, 3_2, and 3_5 require only 2 minimal mutations, while motifs 3_8 and 3_1 require 4 mutations. Here, in agreement, we see higher rankings for 3_3, 3_2, and 3_5 than the other two motifs.

One area for improvement is to systematically find optimal parameters and threshold for our scoring model. By combining our new feature selection and scoring model, we can identify top RAG graph candidates for novel RNA design and help identify similar motifs to define mutation targets.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpcb.0c10685.

Appendices with the following titles: Details of Calculations; Hall's Minimization Using Fiedler Vectors; Fiedler Vector Observations; Fiedler Values and Compactness; Slope s Derivation (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Tamar Schlick − *Courant Institute of Mathematical Sciences, New York University, New York 10012, United States; Department of Chemistry, New York University, New York 10003, United States; NYU-ECNU Center for Computational Chemistry, NYU Shanghai, Shanghai 200062, P. R. China;* orcid.org/0000-0002-2392-2062; Email: schlick@nyu.edu

Author

Qiyao Zhu − *Courant Institute of Mathematical Sciences, New York University, New York 10012, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcb.0c10685

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This article is dedicated to Ruth Nussinov for her pioneering contributions in nucleic acids using computational biology. We thank Dr. Swati Jain for her helpful discussions and comments. This work has been supported by the National Institute of General Medical Sciences, National Institutes of Health (NIH) Grant R35GM122562 and by a RAPID Award from the National Science Foundation, Division of Mathematical Sciences, Award 2030377 to T.S. Research in this article was supported (in part) by Philip Morris USA Inc. and Philip Morris International. The funding institutes did not have any say in the design of the study, analysis of the results, or the decision to publish.

■ REFERENCES

(1) Eddy, S. R. Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.* **2001**, *2*, 919−929.
(2) Breaker, R. R. Riboswitches and the RNA World. *Cold Spring Harbor Perspect. Biol.* **2012**, *4*, a003566.
(3) Holley, R. W.; Apgar, J.; Everett, G. A.; Madison, J. T.; Marquisee, M.; Merrill, S. H.; Penswick, J. R.; Zamir, A. Structure of a Ribonucleic Acid. *Science* **1965**, *147*, 1462−1465.
(4) Guo, P. The emerging field of RNA nanotechnology. *Nat. Nanotechnol.* **2010**, *5*, 833−842.
(5) Que-Gewirth, N. S.; Sullenger, B. A. Gene therapy progress and prospects: RNA aptamers. *Gene Ther.* **2007**, *14*, 283−291.
(6) Hofacker, I. L.; Fontana, W.; Stadler, P. F.; et al. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.* **1994**, *125*, 167−188.
(7) Zuker, M. Computer prediction of RNA structure. *Methods Enzymol.* **1989**, *180*, 262−288.
(8) Rivas, E.; Eddy, S. R. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* **1999**, *285*, 2053−2068.
(9) Zadeh, J. N.; Steenberg, C. D.; Bois, J. S.; et al. NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.* **2011**, *32*, 170−173.
(10) Busch, A.; Backofen, R. INFO-RNA−a fast approach to inverse RNA folding. *Bioinformatics* **2006**, *22*, 1823−1831.
(11) Laing, C.; Schlick, T. Computational approaches to 3D modeling of RNA. *J. Phys.: Condens. Matter* **2010**, *22*, 283101.

(12) Pyle, A. M.; Schlick, T. Challenges in RNA structural modeling and design. *J. Mol. Biol.* **2016**, *428*, 733−735.
(13) Schlick, T.; Pyle, A. M. Opportunities and challenges in RNA structural modeling and design. *Biophys. J.* **2017**, *113*, 225−234.
(14) Miao, Z.; Adamiak, R. W.; et al. RNA-Puzzles Round II: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA* **2015**, *21*, 1066−1084.
(15) Miao, Z.; Adamiak, R. W.; et al. RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA* **2017**, *23*, 655−672.
(16) Gan, H. H.; Pasquali, S.; Schlick, T. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic Acids Res.* **2003**, *31*, 2926−2943.
(17) Laing, C.; Schlick, T. Computational approaches to RNA structure prediction, analysis, and design. *Curr. Opin. Struct. Biol.* **2011**, *21*, 306−318.
(18) Dawson, W. K.; Maciejczyk, M.; Jankowska, E. J.; Bujnicki, J. M. Coarse-grained modeling of RNA 3D structure. *Methods* **2016**, *103*, 138−156.
(19) Le, S. Y.; Nussinov, R.; Maizel, J. V. Tree graphs of RNA secondary structures and their comparisons. *Comput. Biomed. Res.* **1989**, *22*, 461−473.
(20) Fontana, W.; Konings, D. A.; Stadler, P. F.; Schuster, P. Statistics of RNA secondary structures. *Biopolymers* **1993**, *33*, 1389−1404.
(21) Benedetti, G.; Morosetti, S. A graph-topological approach to recognition of pattern and similarity in RNA secondary structures. *Biophys. Chem.* **1996**, *59*, 179−184.
(22) Schlick, T. Adventures with RNA Graphs. *Methods* **2018**, *143*, 16−33.
(23) Nussinov, R.; Pieczenik, G.; Griggs, J.; Kleitman, D. Algorithms For Loop Matching. *SIAM J. Appl. Math.* **1978**, *35*, 68−82.
(24) Gan, H. H.; Fera, D.; Zorn, J.; et al. RAG: RNA-As-Graphs database−concepts, analysis, and features. *Bioinformatics* **2004**, *20*, 1285−1291.
(25) Baba, N.; Elmetwaly, S.; Kim, N.; Schlick, T. Predicting large RNA-Like topologies by a knowledge-based clustering approach. *J. Mol. Biol.* **2016**, *428*, 811−821.
(26) Jain, S.; Saju, S.; Petingi, L.; Schlick, T. An extended dual graph library and partitioning algorithm applicable to pseudoknotted RNA structures. *Methods* **2019**, *162−163*, 74−84.
(27) Zhang, K.; Zheludev, I. N.; Hagey, R. J.; Wu, M. T.-P.; Haslecker, R.; Hou, Y. J.; Kretsch, R.; Pintilie, G. D.; Rangan, R.; Kladwang, W.; et al. Cryo-electron Microscopy and Exploratory Antisense Targeting of the 28-kDa Frameshift Stimulation Element from the SARS-CoV-2 RNA Genome. *bioRxiv* **2020**, DOI: 10.1101/2020.07.18.209270.
(28) Kelly, J. A.; Olson, A. N.; Neupane, K.; Munshi, S.; San Emeterio, J.; Pollack, L.; Woodside, M. T.; Dinman, J. D. Structural and functional conservation of the programmed −1 ribosomal frameshift signal of SARS coronavirus 2 (SARS-CoV-2). *J. Biol. Chem.* **2020**, *295*, 10741−10748.
(29) Schlick, T.; Zhu, Q.; Jain, S.; Yan, S. Structure-Altering Mutations of the SARS-CoV-2 Frameshifting RNA Element. *Biophys. J.* **2020**, DOI: 10.1016/j.bpj.2020.10.012.
(30) Schlick, T.; Zhu, Q.; Dey, A.; Jain, S.; Yan, S.; Laederach, A. To knot and not: Multiple conformations of the SARS-CoV-2 frameshifting RNA element. 2021, in preparation.
(31) Jain, S.; Zhu, Q.; Paz, A. S. P.; Schlick, T. Identification of novel RNA design candidates by clustering the extended RNA-As-Graphs library. *Biochim. Biophys. Acta, Gen. Subj.* **2020**, *1864*, 129534.
(32) Jain, S.; Laederach, A.; Ramos, B. V. S.; Schlick, T. A pipeline for computational design of novel RNA-like topologies. *Nucleic Acids Res.* **2018**, *46*, 7040−7051.
(33) Kim, N.; Zheng, Z.; Elmetwaly, S.; Schlick, T. RNA graph partitioning for the discovery of RNA modularity: a novel application of graph partition algorithm to biology. *PLoS One* **2014**, *9*, No. e106074.
(34) Zahran, M.; Bayrak, C. S.; Elmetwaly, S.; Schlick, T. RAG-3D: a search tool for RNA 3D substructures. *Nucleic Acids Res.* **2015**, *43*, 9474−9488.
(35) Jain, S.; Schlick, T. F-RAG: Generating Atomic Models from RNA Graphs using Fragment Assembly. *J. Mol. Biol.* **2017**, *429*, 3587−3605.
(36) Jain, S.; Tao, Y.; Schlick, T. Inverse Folding with RNA-As-Graphs Produces a Large Pool of Candidate Sequences with Target Topologies. *J. Struct. Biol.* **2020**, *209*, 107438.
(37) Siegfried, N. A.; Busan, S.; Rice, G. M.; Nelson, J. A. E.; Weeks, K. M. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nat. Methods* **2014**, *11*, 959−965.
(38) Spielman, D. A.; Teng, S. H. Spectral partitioning works: planar graphs and finite element meshes. *Linear Algebra Its Appl.* **2007**, *421*, 284−305.
(39) Laing, C.; Jung, S.; Iqbal, A.; Schlick, T. Tertiary motifs revealed in analyses of higher order RNA junctions. *J. Mol. Biol.* **2009**, *393*, 67−82.
(40) Jain, S.; Bayrak, C. S.; Petingi, L.; Schlick, T. Dual graph partitioning highlights a small group of pseudoknot-containing RNA submotifs. *Genes* **2018**, *9*, 371.

https://dx.doi.org/10.1021/acs.jpcb.0c10685
J. Phys. Chem. B 2021, 125, 1144−1155