Supplementary Data for Identification of Novel RNA Design Candidates by Clustering the Extended RNA-As-Graphs Library

Swati Jain[1], Qiyao Zhu[2], Amiel S.P. Paz[3,4], and Tamar Schlick[1,2,4,*]

[1]Department of Chemistry, New York University, 1021 Silver, 100 Washington Square East, New York, NY 10003, USA
[2]Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA
[3]NYU Shanghai, 1555 Century Avenue, Shanghai 200135, China
[4]NYU-ECNU Center for Computational Chemistry, NYU Shanghai, 3663 Zhongshang Road North, Shanghai 200062, China
[*]Corresponding author: schlick@nyu.edu

S1 Linear and quadratic least square regression for extracting graph features

As mentioned in Subsection 2.3 of the main paper, we refer to two-dimensional points $(1, \lambda_2), (2, \lambda_3), ..., (n-1, \lambda_n)$ as 'eigenvalue points' and points $(1, \lambda_2^2), (2, \lambda_3^2), ..., (n-1, \lambda_n^2)$ as 'squared eigenvalue points'. In this section, we perform both linear and quadratic regression (using functions in Matlab as mentioned in Subsection 2.4.4 in the main paper) on eigenvalue and squared eigenvalue points to determine which is a better fit.

To visualize the linear and quadratic functions fitted to eigenvalue and squared eigenvalue points, graphs with the largest number of vertices n are suitable candidates, as they provide more data points for least squared regression. For a quantitative assessment, we calculate and compare the mean squared errors (MSE) of both the linear and the quadratic function returned by least squared regression. For each vertex number, we take the average MSEs over all graphs. The analysis is done for tree and dual graphs independently.

Figure S1 shows the linear and quadratic fit for eigenvalue and squared eigenvalue points for two representative tree graph topologies. Visually, there is not much difference between linear and quadratic fit for eigenvalue points, but the quadratic curve is clearly a better for for squared eigenvalue points. In Figure S1b, an interesting distribution of eigenvalues can be observed. There are 11 repeated eigenvalues of 1, which results in a poor fit for both linear and quadratic regression. Similar observations are made for every vertex number: the

squared eigenvalues of the some graphs are better approximated using quadratic polynomial regression, while some graphs tends to have many repeated eigenvalues of 1, so that neither linear nor polynomial fits work well.

Figure S2 shows the linear and quadratic fit for eigenvalue and squared eigenvalue points for two representative dual graph topologies. Similar to tree graphs, quadratic curve is clearly a better for for squared eigenvalue points. Therefore, our conclusions for both tree and dual graphs are similar: using quadratic polynomial regression for squared eigenvalue points is a better choice as it leads to no change or an improvement in the fit.

For a quantitative analysis, Tables S1 and S2 show the MSEs for both linear and quadratic regression for tree and dual graphs, respectively. The MSEs for both tree and dual graphs for eigenvalue points does not change much between linear and quadratic regression. However, the quadratic regression obtains significant reduction in MSEs for squared eigenvalue points, also shown in Figure S3.

Based on the above analysis, we decided to use only linear regression for eigenvalue points, combined with either linear or quadratic regression for squared eigenvalue points for extracting features of graph topologies (to be used with clustering algorithms, as described in the main paper).

S2 Analysis of linear dependency of variables

As described in Section 2.3 of the main paper, we derive 4 linear variables (using linear regression) and 5 quadratic variables (using both linear and quadratic regression) to represent each tree and dual graph topology. To analyze the linear dependence of variables, we plot one variable against another.

The four linear variables are:

1. x_1: scaled slope of the fitted line using linear regression for eigenvalue points drawn on a plane.

2. x_2: y-intercept of the fitted line using linear regression for eigenvalue points drawn on a plane.

3. x_3: scaled slope of the fitted line using linear regression for squared eigenvalue points drawn on a plane.

4. x_4: y-intercept of the fitted line using linear regression for squared eigenvalue points drawn on a plane.

The five quadratic variables are:

1. x_1: scaled slope of the fitted line using linear regression for eigenvalue points drawn on a plane.

2. x_2: y-intercept of the fitted line using linear regression for eigenvalue points drawn on a plane.

3. x_3: coefficient a of the fitted polynomial $ax^2 + bx + c$ using quadratic regression for squared eigenvalue points drawn on a plane.

4. x_4: coefficient b of the fitted polynomial $ax^2 + bx + c$ using quadratic regression for squared eigenvalue points drawn on a plane.

5. x_5: coefficient c of the fitted polynomial $ax^2 + bx + c$ using quadratic regression for squared eigenvalue points drawn on a plane.

Figure S4 shows the plots of linear variables for tree graphs that indicate that variables x_1 and x_2 are linearly dependent on each other, so are variables x_3 and x_4. Therefore, there are two linearly independent linear variables for tree graphs. Figure S5 shows the plots of quadratic variables for tree graphs that indicate that variables x_1 and x_2 are linearly dependent on each other. Therefore, there are four linearly independent quadratic variables for tree graphs.

For dual graphs, Figure S6 and Figure S7 shows the plots of linear and quadratic variables, respectively. As can be seen from the figures, all four linear variables and all five quadratic variables are linearly independent.

S3 Clustering algorithms

We use three clustering algorithms, PAM, k-means, and k-NN, to classify all tree and dual graph topologies in our RAG library into two categories/clusters: 'RNA-like' (graph topologies likely to correspond to yet undiscovered RNA structures) and 'non RNA-like' (graph topologies unlikely to correspond to RNA structures). Tree and dual graph topologies are clustered separately. For each clustering techniques, N denotes the total number of graph topologies (2286 for tree graphs and 110,664 for dual graphs, as graphs with 2 vertices are not considered), and $K = 2$ denotes the number of clusters. Of the three clustering techniques, PAM and k-means are unsupervised clustering algorithms as they do not require any training data. In contrast, k-NN requires training data, hence is a supervised clustering algorithm.

S3.1 Partitioning Around Medoids (PAM)

The PAM algorithm to divide N points into K clusters works as follows:

1. Of N data points, randomly choose K points as medoids.

2. Assign each of the $N-K$ data points to the closest medoid using the euclidian distances between the graph features extracted as described in Subsection 2.3 of the main paper.

3. Compute the cost function as the sum of the euclidian distances of each point from their corresponding medoid.

4. For each iteration of the PAM algorithm:

    - For each medoid m and non-medoid o:

(a) Let o be the new medoid and m be the non-medoid. Repeat steps 2 and 3.

(b) If the cost function decreased, keep the new medoid and the associated clusters.

(c) If the cost function increased, restore the medoid at m and non-medoid at o.

After several iterations, the algorithm should converge to an optimal solution for cluster medoids.

S3.2 K-means clustering

The K-means algorithm to divide N points into K clusters works as follows:

1. Choose K initial cluster centers. The cluster centers can be chosen as any K data points or randomly initialized.

2. Assign each of the N data points to the closest center using the euclidian distances between the graph features extracted as described in Subsection 2.3 of the main paper.

3. Compute the new cluster centers as the average of the data points in the corresponding cluster.

4. Repeat steps 2 and 3 until cluster centers do not change.

S3.3 k-Nearest-Neighbors (k-NN)

The k-NN clustering algorithm divides N points into K clusters, using the cluster identities of its k nearest neighbors. For our purposes, $K = 2$: RNA-like and non-RNA-like. We use the set of existing RNA topologies as training data for the RNA-like cluster and then randomly select an equal number of hypothetical graph topologies as part of the training data for the non-RNA-like cluster.

For each point not in the training set, get the clusters for its k nearest neighbors from the training set. The point is then assigned to whichever cluster has the highest nearest neighbor count.

S4 RNA structural dataset details

RNA 3D structures available on or before August 31, 2018 on the Protein Data Bank (PDB) were downloaded, which included multiple files for parts of large structures. The 4042 downloaded RNA structures were separated into 9019 structure files or "Integrated Functional Elements" (i.e., single chains or multiple strongly base-paired chains) and grouped into "equivalence classes" (based on RNA molecule type, its sequence, structure, and species) available on the Bowling Green State University (BGSU) RNA site (http://rna.bgsu.edu/rna3dhub), version 3.37, August 31, 2018 [1]. Same rules were used to separate chains in RNA structures missing from the BGSU RNA dataset but present in the list download from the PDB. Equivalence classes were combined manually if necessary. Only standard RNA residues (A, U, G, C) and modified RNA residues (listed as "RNA linking") were retained in the PDB

files, and all ligand and water molecules were removed, along with any protein or DNA residues. Residues in structures with insertion codes (residue numbers with letters) were renumbered. To avoid counting duplicate IFEs within the same PDB file that belong to the same equivalence class, only the IFE with the highest number of residues were retained.

For the retained structure files, base pairs were identified using three different 2D structure annotation programs: RNAView [2], MC-Annotate [3], and DSSR [4]. Canonical base pairs (AU WC Saenger class XX, GC WC Saenger class XIX, and GC wobble Saenger class XXVIII, [5]) reported by at least two annotation programs were considered to create a consensus RNA 2D structure. All pseudoknots were removed (as tree graphs cannot represent pseudoknots), as well as structures with no or single/isolated base pairs or only one vertex (as they won't have any associated adjacency matrix or tree graph ID). The remaining 4,488 RNA structure files were used for further study.

Table S1: Mean Squared Errors (MSE) using linear and quadratic polynomial regressions for eigenvalue and squared eigenvalue points for tree graphs.

Number of Vertices	Mean Squared Error			
	Eigenvalue points		Squared eigenvalue points	
	Linear	Quadratic	Linear	Quadratic
3	0.000	0.000	0.000	0.000
4	0.250	0.000	6.694	0.000
5	0.441	0.068	17.570	2.485
6	0.505	0.130	26.684	5.922
7	0.559	0.179	35.973	10.304
8	0.567	0.188	41.899	13.094
9	0.565	0.192	46.704	15.754
10	0.550	0.184	48.555	16.782
11	0.539	0.176	50.352	17.771
12	0.526	0.167	50.937	18.034
13	0.515	0.159	51.339	18.233

Table S2: Mean Squared Errors (MSE) using linear and quadratic polynomial regressions for eigenvalue and squared eigenvalue points for dual graphs.

Number of Vertices	Mean Squared Error			
	Eigenvalue points		Squared eigenvalue points	
	Linear	Quadratic	Linear	Quadratic
3	0.000	0.000	0.000	0.000
4	0.248	0.000	21.774	0.000
5	0.235	0.114	25.653	7.695
6	0.213	0.127	27.316	8.778
7	0.173	0.111	26.394	7.882
8	0.150	0.100	25.956	7.218
9	0.127	0.087	25.061	6.321

Table S3: Eleven tree graph topologies that were removed from the list of existing tree graph topologies.

Missing Topologies	Associated PDB IDs	Reason for Removal
6_4	Several structures	All superseded
9_4	3IZD_A	Assigned different topology
9_20	1L9A_A, 1MFQ_A, 2GO5_A, and 2J37_A	Assigned different topology
11_56	1U9S_A	Assigned different topology
11_207	3DHS_A	Assigned different topology
11_216	2RKJ_C	Assigned different topology
12_286	3ZEX_E	4V8M (supersedes 3ZEX)
12_392	3BO2_BCDE	3BO2 (only chain B)
13_181	3BO3_CDB	3BO3 (chains C,D removed)
13_1021	1GRZ_B	1GRZ (chain A instead of B)
13_1047	1U6B_CDB	1U6B (chains C,D removed)

Table S4: PAM and K-means clustering accuracy. Shown for all tree and dual graph topologies with 3 or more vertices are the motifs classified as RNA-like and non RNA-like by unsupervised clustering algorithms PAM and K-means using full linear and quadratic variables. Also shown are the number and percentage of existing tree and dual graph topologies correctly classified as RNA-like.

(a) Tree Graphs

	All Topologies (Total:2286)		**Existing Topologies (Total:79)**	
	RNA–like	**non RNA–like**	**RNA–like**	**non RNA–like**
Method	**Linear Variables**			
PAM	1645 (71.96%)	641 (28.04%)	61 (77.22%)	18 (22.78%)
K-means	1645 (71.96%)	641 (28.04%)	61 (77.22%)	18 (22.78%)
Method	**Quadratic Variables**			
PAM	1897 (82.98%)	389 (17.02%)	58 (73.42%)	21 (26.58%)
K-means	1890 (82.68%)	396 (17.32%)	58 (73.42%)	21 (26.58%)

(b) Dual Graphs

	All Topologies (Total:110664)		**Existing Topologies (Total:118)**	
	RNA–like	**non RNA–like**	**RNA–like**	**non RNA–like**
Method	**Linear Variables**			
PAM	55250 (49.93%)	55414 (50.07%)	89 (75.42%)	29 (24.58%)
K-means	55257 (49.93%)	55407 (50.07%)	89 (75.42%)	29 (24.58%)
Method	**Quadratic Variables**			
PAM	58335 (52.71%)	52329 (47.29%)	86 (72.88%)	32 (27.12%)
K-means	56994 (51.50%)	53670 (48.50%)	86 (72.88%)	32 (27.12%)

Table S5: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full linear variables using leave-one-out cross validation for tree graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	60.82	67.09	56.96
3	62.66	68.35	59.49
5	58.73	67.09	51.90
7	59.94	69.62	55.06
9	58.73	63.29	53.16
11	58.86	63.29	54.43
13	58.23	63.92	50.00
15	59.68	66.46	50.00
17	60.51	65.82	51.27
19	60.70	65.19	55.70

Table S6: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full quadratic variables using leave-one-out cross validation for tree graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	75.63	82.91	72.78
3	77.28	81.01	75.32
5	78.61	81.65	75.95
7	80.25	84.81	77.22
9	81.08	85.44	79.11
11	79.68	82.28	75.95
13	79.24	82.28	76.58
15	77.66	79.75	74.68
17	76.52	81.01	73.42
19	75.70	78.48	72.78

Table S7: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full linear variables using 10-fold cross validation for tree graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	60.51	66.46	56.33
3	62.78	68.99	58.86
5	59.43	67.09	54.43
7	60.32	67.72	53.16
9	59.43	65.19	51.90
11	58.73	64.56	54.43
13	59.62	65.83	53.80
15	59.62	67.09	53.80
17	60.95	65.19	55.70
19	59.56	63.92	53.80

Table S8: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full quadratic variables using 10-fold cross validation for tree graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	76.08	81.65	72.15
3	76.71	81.01	72.78
5	78.61	84.18	75.32
7	80.32	83.54	76.58
9	80.70	82.91	77.85
11	80.32	82.28	78.48
13	78.23	80.38	76.58
15	77.47	81.01	74.05
17	76.39	81.01	70.25
19	76.27	79.11	74.05

Table S9: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full linear variables using leave-one-out cross validation for dual graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	64.24	66.95	61.02
3	65.55	68.64	60.17
5	66.74	70.34	64.83
7	67.84	72.03	64.41
9	68.64	71.61	66.53
11	68.86	70.76	66.53
13	68.35	71.19	65.25
15	68.60	70.34	66.53
17	68.52	71.61	65.25
19	68.69	72.46	63.98

Table S10: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full quadratic variables using leave-one-out cross validation for dual graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	78.81	82.63	72.46
3	81.10	83.05	79.66
5	80.17	82.20	77.97
7	79.62	82.63	77.97
9	79.41	82.20	76.27
11	78.98	81.78	76.27
13	78.31	80.93	73.73
15	77.75	79.66	75.00
17	77.71	79.24	75.00
19	76.91	78.81	75.42

Table S11: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full linear variables using 10-fold cross validation for dual graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	63.86	65.25	61.86
3	65.59	69.49	61.02
5	66.53	69.49	63.98
7	67.67	71.61	64.41
9	68.01	71.19	65.68
11	68.35	71.19	65.68
13	68.35	72.46	66.10
15	69.19	72.46	66.95
17	68.31	72.03	65.25
19	68.39	71.61	65.68

Table S12: Average, maximum, and minimum accuracy (over 10 trial runs) of k-NN clustering algorithm with full quadratic variables using 10-fold cross validation for dual graphs.

Number of Nearest Neighbors	Average Accuracy	Maximum Accuracy	Minimum Accuracy
1	78.81	81.78	73.31
3	81.06	82.63	78.81
5	80.17	82.63	77.54
7	79.24	82.20	75.85
9	78.81	81.36	75.42
11	78.94	81.36	76.27
13	78.22	80.51	72.46
15	77.50	78.81	75.00
17	77.25	79.24	75.00
19	76.57	78.39	74.15

Table S13: Average accuracy (over 10 trial runs) of k-NN clustering algorithm with reduced linear and quadratic variables using leave-one-out (LOO) and 10-fold cross validation for tree and dual graphs.

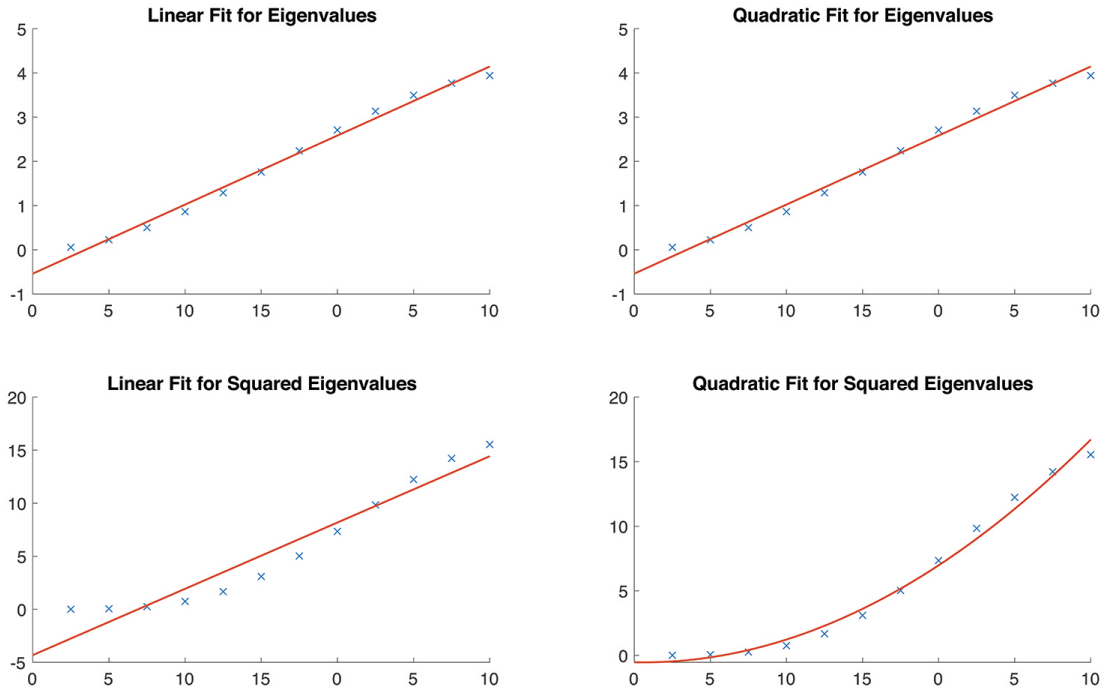
	Average Accuracy (%)			
	Linear variables		**Quadratic variables**	
	LOO	**10-fold**	**LOO**	**10-fold**
Neighbors	**Tree Graphs**			
1	57.22	57.72	75.51	75.19
3	59.87	61.08	77.34	77.53
5	57.85	58.23	78.29	78.54
7	58.67	58.35	79.68	79.94
9	57.15	58.92	80.57	80.13
11	57.60	57.60	80.13	78.92
13	57.60	57.41	78.67	78.92
15	59.05	59.87	78.04	76.84
17	60.06	60.25	76.27	75.57
19	60.70	60.70	75.25	76.33
Neighbors	**Dual Graphs**			
1	61.23	60.68	67.88	68.31
3	62.71	64.28	72.16	71.44
5	65.09	66.36	71.57	71.02
7	66.53	66.95	71.91	72.20
9	67.54	68.18	72.88	73.48
11	67.80	67.63	73.05	73.09
13	68.26	68.35	72.67	72.43
15	68.31	67.84	73.26	73.56
17	68.56	68.77	73.26	73.01
19	69.03	68.56	73.94	73.14

Table S14: Mean squared error for all existing tree graph topologies that were correctly classified (61 of 79) and misclassified (18 of 79) by PAM and K-means clustering using reduced linear variables.

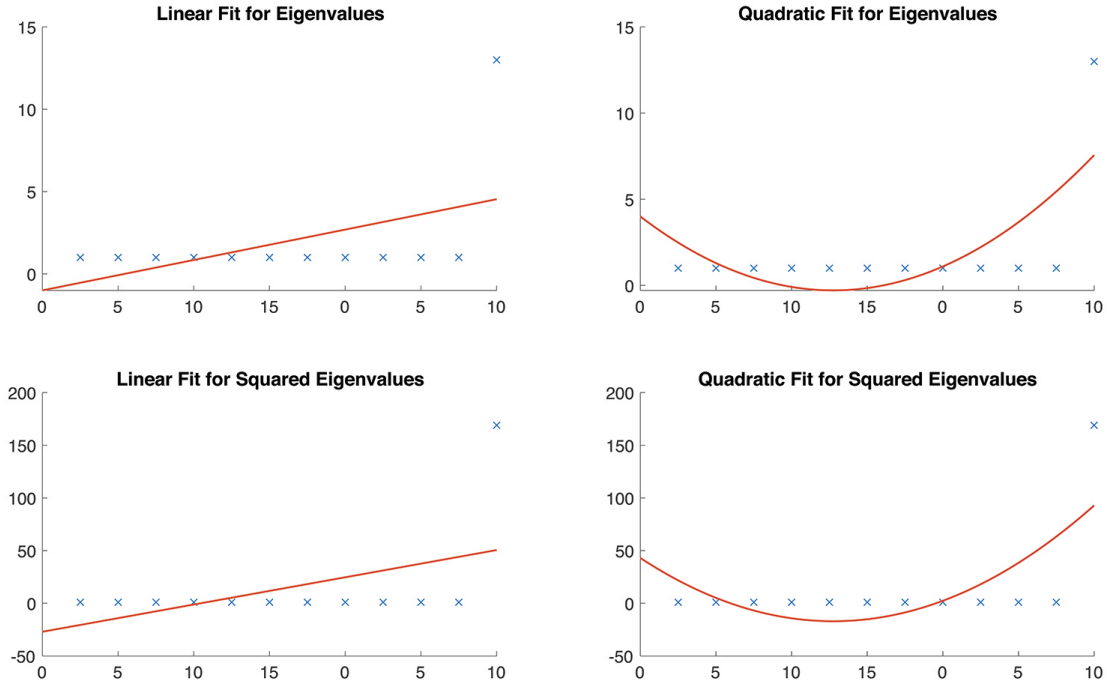
Number of Vertices	Eigenvalue points		Squared eigenvalue points	
	Correctly Classified	Misclassified	Correctly Classified	Misclassified
3	0	–	0	–
4	0.25	–	6.694	–
5	0.061	1.2	4.755	43.2
6	0.062	1.27	5.486	64.94
7	0.182	0.916	12.681	55.7
8	0.136	0.955	11.421	57.201
9	0.114	2.755	11.468	264.87
10	0.109	1.134	11.971	92.447
11	0.128	0.821	12.82	71.858
12	0.188	1.044	16.181	86.022
13	0.197	0.726	22.001	75.838
Average	0.138	1.096	12.366	82.601

Table S15: Mean squared error for all existing dual graph topologies that were correctly classified (89 of 118) and misclassified (29 of 118) by PAM and K-means clustering using reduced linear variables.

Number of Vertices	Eigenvalue points		Squared eigenvalue points	
	Correctly Classified	Misclassified	Correctly Classified	Misclassified
3	0	0	0	0
4	0.203	0.151	21.482	13.997
5	0.141	0.248	25.06	21.253
6	0.2	0.35	30.048	29.451
7	0.216	0.128	41.366	19.894
8	0.141	0.006	30.219	11.164
9	0.142	0.336	31.356	36.116
Average	0.168	0.199	29.668	18.741

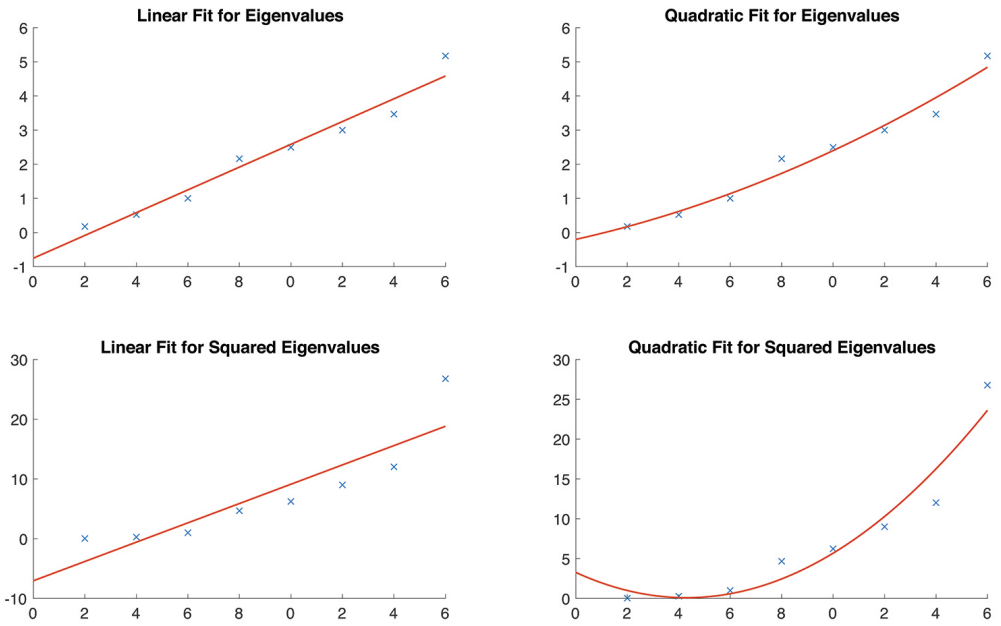


(a) Comparison of linear and quadratic fit for eigenvalue and squared eigenvalue points for tree graph with RAG ID: 13_1.

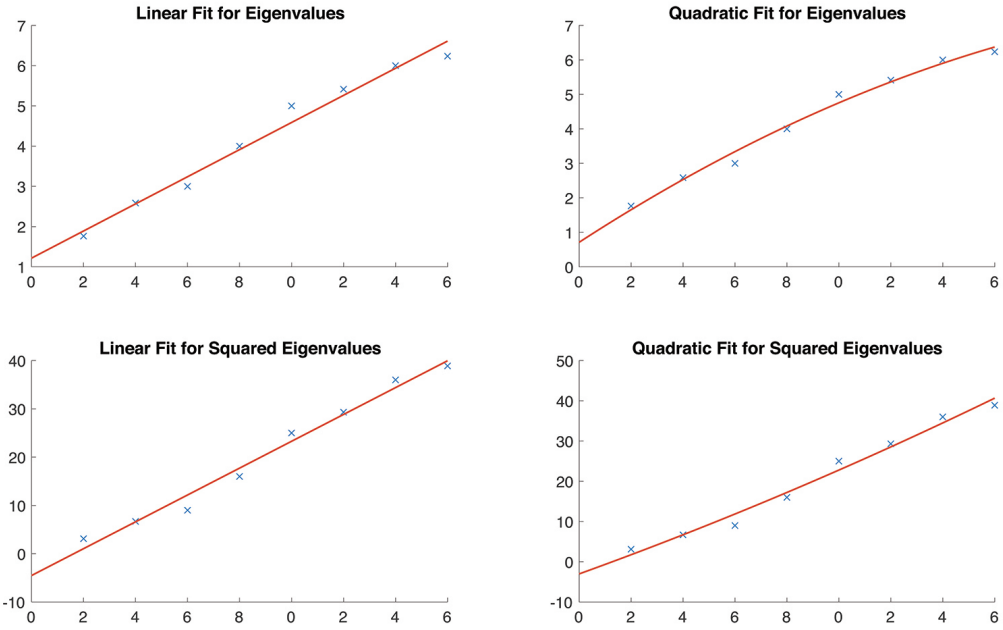


(b) Comparison of linear and quadratic fit for eigenvalue and squared eigenvalue points for tree graph with RAG ID: 13_1301.

Figure S1

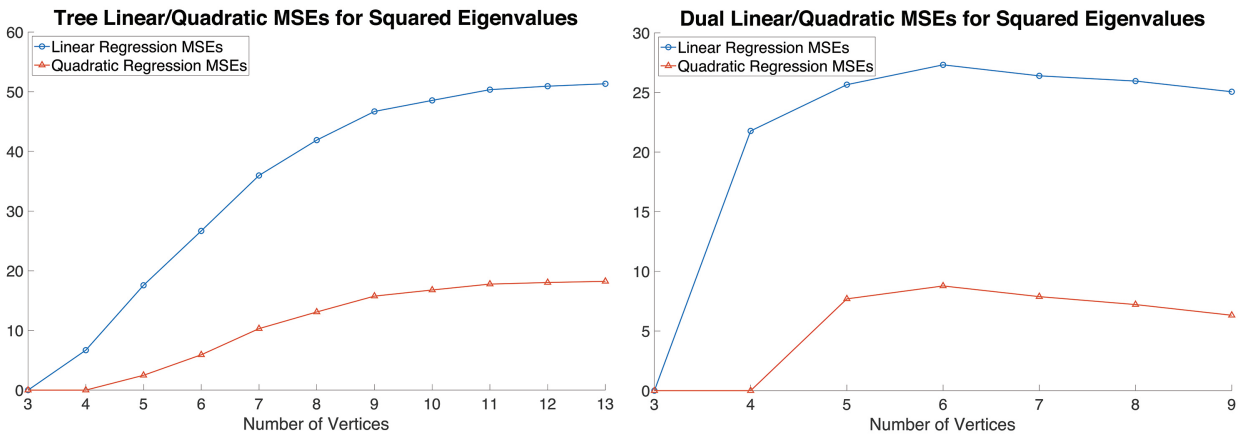


(a) Comparison of linear and quadratic fit for eigenvalue and squared eigenvalue points for dual graph with RAG ID: 9_1.



(b) Comparison of linear and quadratic fit for eigenvalue and squared eigenvalue points for dual graph with RAG ID: 9_92788.

Figure S2



(a) (b)

Figure S3: MSEs using linear and quadratic regressions for (a)tree graph squared eigenvalues (b)dual graph squared eigenvalues

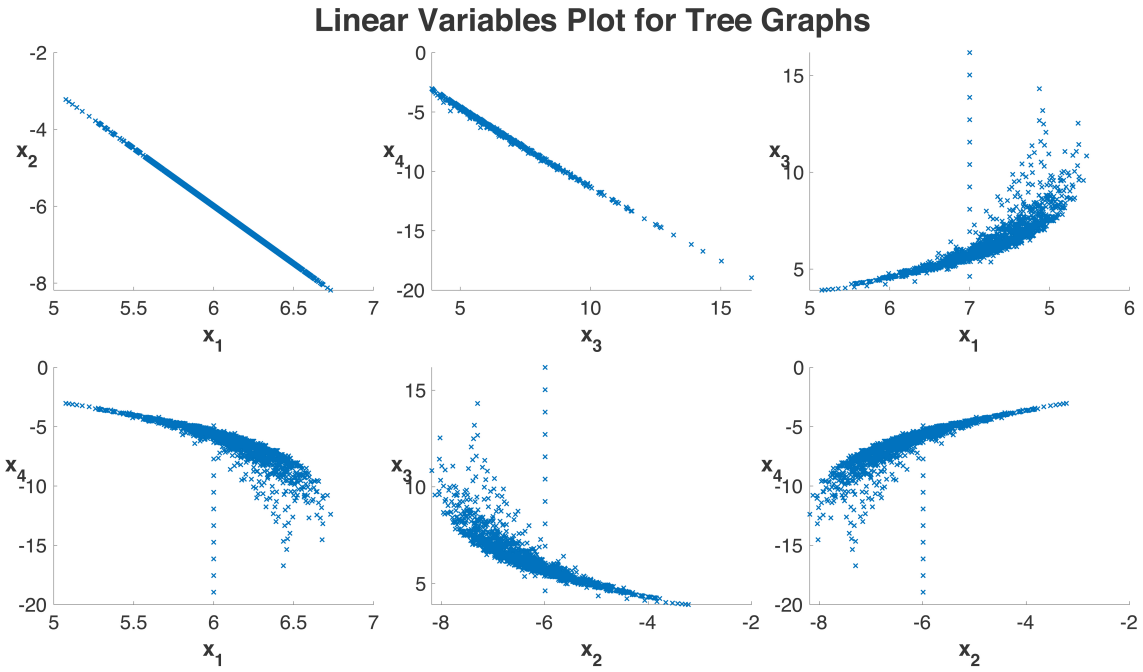


Figure S4: Full linear variables, x_1, x_2, x_3, x_4, plotted against each other for tree graphs. Refer to the text in Section S2 and Materials and Methods in the main paper for definitions of the variables.

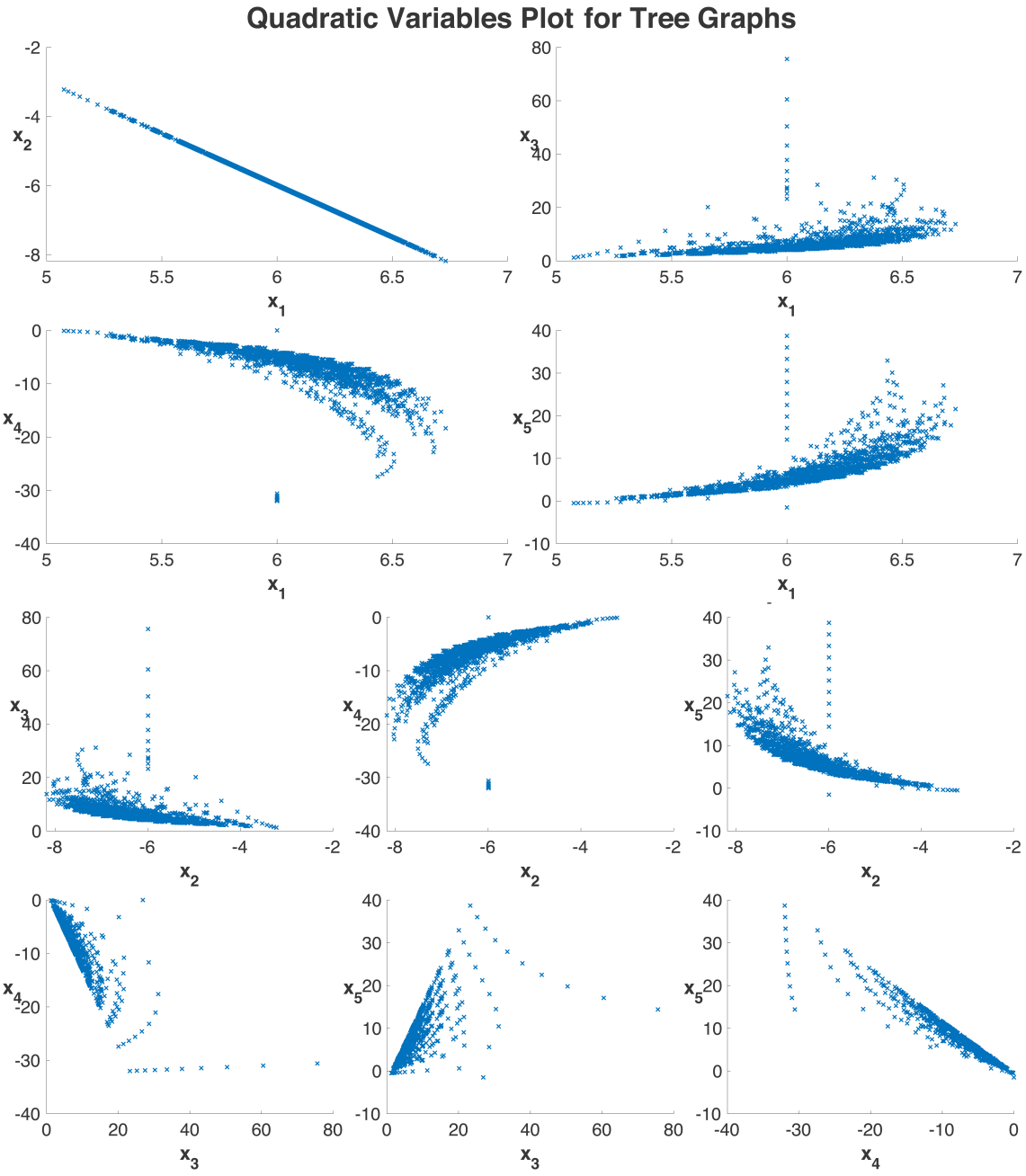


Figure S5: Full quadratic variables, x_1, x_2, x_3, x_4, x_5, plotted against each other for tree graphs. Refer to the text in Section S2 and Materials and Methods in the main paper for definitions of the variables.

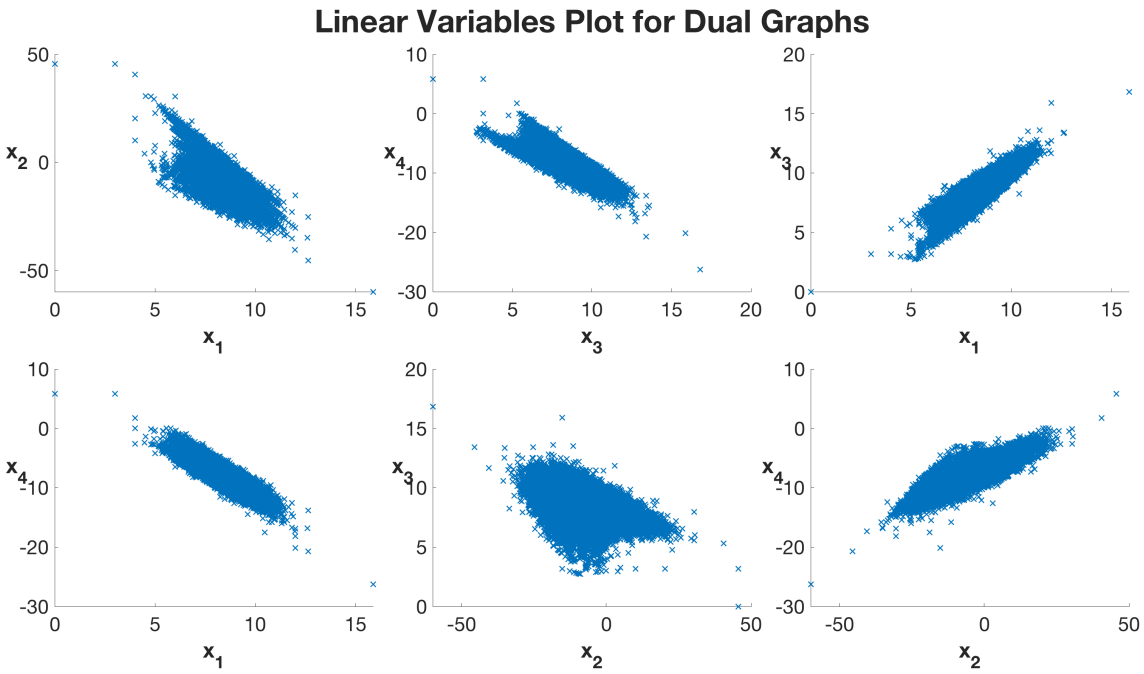


Figure S6: Full linear variables, x_1, x_2, x_3, x_4, plotted against each other for dual graphs. Refer to the text in Section S2 and Materials and Methods in the main paper for definitions of the variables.

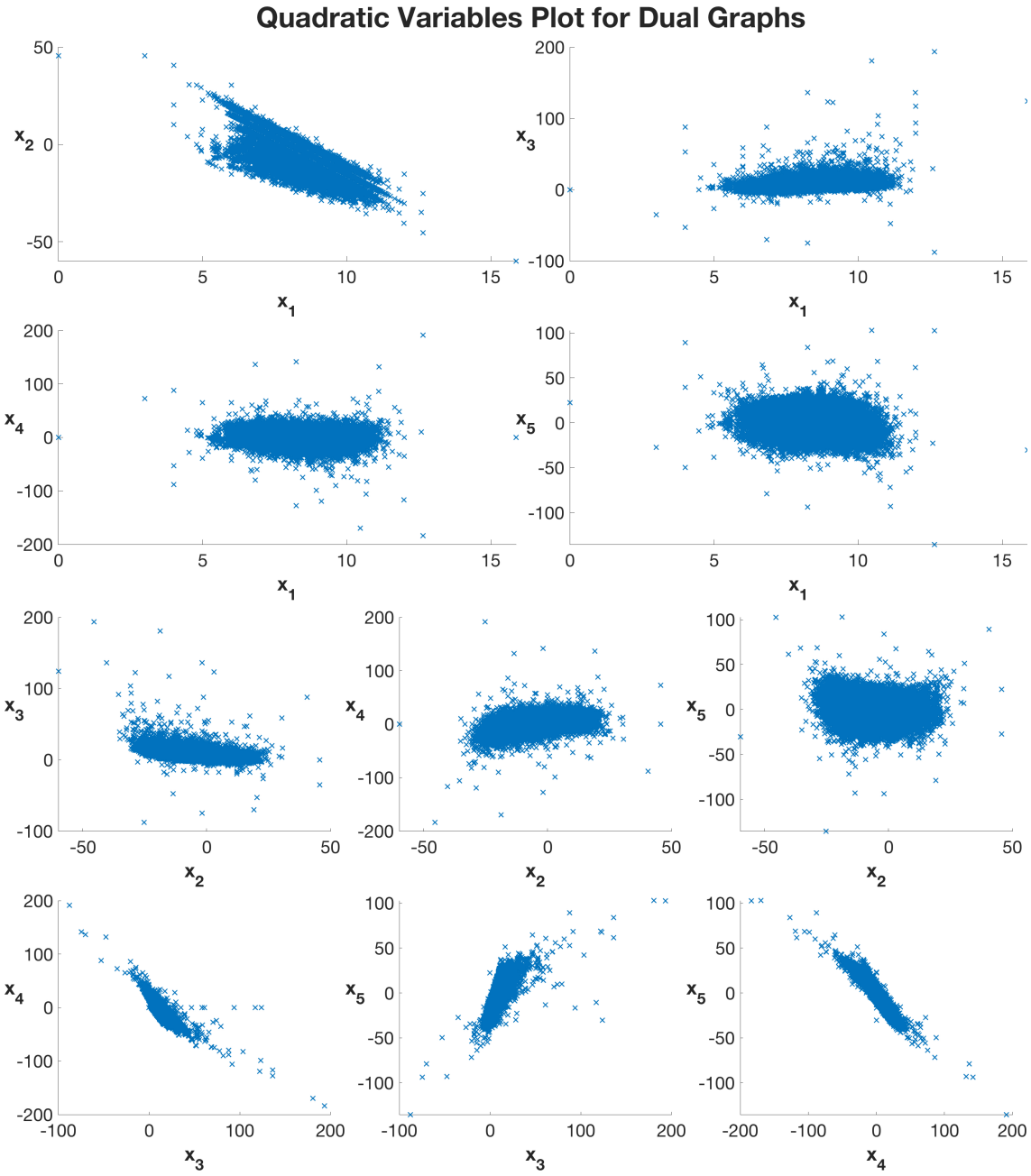


Figure S7: Full quadratic variables, x_1, x_2, x_3, x_4, x_5, plotted against each other for dual graphs. Refer to the text in Section S2 and Materials and Methods in the main paper for definitions of the variables.

References

[1] Leontis, N. B. and Zirbel, C. L. 2012, Nonredundant 3D Structure Datasets for RNA Knowledge Extraction and Benchmarking. In Leontis, N. and Westhof, E., (eds.), *RNA 3D Structure Analysis and Prediction*, pp. 281–298 Springer Berlin Heidelberg Berlin, Heidelberg.

[2] Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H., and Westhof, E. 2003, Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acid Res,* **31**(13), 3450.

[3] Lemieux, S. and Major, F. 2002, RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. *Nucleic Acid Res,* **30**(19), 4250–4263.

[4] Lu, X.-J., Bussemaker, H. J., and Olson, W. K. 2015, DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acid Res,* **43**(21), e142.

[5] Saenger, W. 1984, Forces Stabilizing Associations Between Bases: Hydrogen Bonding and Base Stacking. In *Principles of Nucleic Acid Structure* pp. 116–158 Springer New York New York, NY.