EXTENDED RNA BASE PAIRING NETWORKS IMPRINT SMALL MOLECULE BINDING PREFERENCES

A PREPRINT

Carlos G. Oliver
School of Computer Science
McGill University
Montreal, QC Canada
carlos.gonzalezoliver@mail.mcgill.ca

Vincent Mallet
School of Computer Science
McGill University
Montreal, QC Canada
vincent.mallet96@gmail.com

Roman Sarrazin Gendron
School of Computer Science
McGill University
Montreal, QC Canada
roman.sarrazingendron@mail.mcgill.ca

Vladimir Reinharz
Center for Soft and Living Matter
Institute for Basic Science
Ulsan, South Korea
vreinharz@ibs.re.kr

Nicolas Moitessier
Department of Chemistry
McGill University
Montreal, QC Canada
nicolas.moitessier@mcgill.ca

Jérôme Waldispühl
School of Computer Science
McGill University
Montreal, QC Canada
jeromew@cs.mcgill.ca

July 12, 2019

ABSTRACT

Recent studies have identified small RNA-binding molecules with potential for therapeutic applications including novel antibiotics, antivirals, protein synthesis controllers, and CRISPR activators. As RNA binding data accumulates, machine learning methods are becoming attractive approaches to accelerate the discovery of RNA-targeting drugs. While atomic level representations of molecular systems are regarded as a critical step for physics-based and data-driven drug design, the singularity and hierarchical organization of RNA structures challenges this paradigm. In this work, we present a machine learning framework for assisting in the discovery of small RNA-binding molecules from graphical representations of RNA structures. A key feature of our tool is that it does not rely on manual feature engineering or costly physical simulations. Instead, we extract molecule-binding information directly from known RNA-ligand complexes by combining graph embedding methods with machine learning models. Given only the graph representation of an RNA site as input, our tool is able to reliably predict chemical features, also known as fingerprints, of the observed ligands. The resulting fingerprints can be used to classify RNA sites according to their binding preferences, screen databases for promising ligands, or act as constraints for generating novel ligands. We show consistent performance for across ligand classes in enriching the known binder from a larger ligand library. As a validation, we applied this method to successfully identify binding residues in unbound RNA structures for three different riboswitches. These results also suggest that small molecule binding preferences in RNA can be extracted directly from the nucleotide pairing level.

Code and data are freely available at:
http://jwgitlab.cs.mcgill.ca/cgoliver/rnamigos.

RNA is a key regulator and information carrier found in all forms of life, and the diversity of known functions it drives is growing. Furthermore, recent studies identified small molecules as important non-covalent regulators of RNA function in many cellular pathways [9]. These discoveries contribute to a better understanding of molecular mechanisms regulating biological systems, but also pose RNA molecules as a large class of promising novel drug targets. Indeed, we are already witnessing the development of the first drugs targeting RNAs. Among them, Ribocil, which has recently been uncovered through a phenotypic assay to target the FMN riboswitch is currently undergoing clinical trials as a novel antibiotic [17]. Various other small-molecule mediated RNA control systems are also being proposed [46, 40], including some for CRISPR activation regulation [20] and are likely to play an important role in genetic disorder treatment and synthetic biology. As observed by KD Warner and co-workers [49], only a small fraction of the genome is translated into protein (1.5%) while the vast majority is transcribed into non-coding RNA (70%). This dichotomy suggests that new applications are within reach.

In parallel, the protein-binding drug discovery field is experiencing rapid and significant advances from data-driven artificial intelligence models for tasks such as candidate molecule generation and affinity scoring [14]. As biological evidence accumulates, similar models could accelerate RNA-binding drug discovery and help identify structural patterns governing RNA-ligand interactions. Most importantly, data-driven technologies could also offer an efficient solution to filter the large chemical space of small molecules in a setting where the number of reachable targets is potentially much larger than what has been seen to date.

0.1 Computational Drug Discovery

The problem of computational drug discovery can be viewed from two perspectives: ligand-based and target-based (a.k.a. structure-based).

Ligand-based approaches aim to design or identify compounds with certain properties such as drug-likeness, similarity to known actives, or synthesizability. They are typically used to produce libraries of potential drugs. This field has greatly benefited from recent advances in deep learning technology [11, 15, 39], combined with the large amounts of ligand data available [18, 30]. However, with ligand-based methods, ligand-to-target interactions, which ultimately determine activity, are rarely modelled. In this context, costly upstream and downstream experimental validations are required.

By contrast, structure-based approaches explicitly model a target structure (protein or RNA) and a set of candidate ligands for which the strength of the binding (i.e. affinity) is directly computed. Affinity scores are usually computed using docking techniques [31], using various pose-finding algorithms and scoring schemes have been proposed, including some specialized for RNA [21, 38, 37, 27]. Although potentially more accurate than ligand-based methods, structure-based approaches are severely limited by the computational cost of docking and virtual screening. In particular, the number of candidate ligands largely exceeds the number of molecules that could possibly be processed (10^8 vs 10^{24} possible chemical compounds [12]). In addition, the modelling of interactions between RNAs and small molecules still needs improvement to reach the performance of protein-ligand docking [27, 44].

0.2 RNA structure

RNAs possess multiple levels of structural organization ranging from the secondary structure made, for example, of Watson-Crick (`A-U, C-G`) and Wobble (`G-U`) base pairs found in the structure, to the full tertiary structure modelling the position of all atoms. In a seminal work, Leontis and Westhof expanded the base-pairing nomenclature by identifying 12 different types of base-pairing interactions according to the relative 3D geometry of the participating nucleotides [24, 23]. Among them, the canonical pairs (i.e. `A-U, C-G`) are the most studied class. Notably, they create series of stable stacks that form a scaffold for the full structure [45]. This feature naturally defines the RNA secondary structure level. Non-canonical pairs on the other hand are enriched in loops (i.e. regions without canonical pairs) and create more complex patterns [25, 35]. These interactions fine-tune the specificity of RNA interactions by determining structure at the 3D level [22]. Interestingly, non-canonical pairs were also found to be involved in ligand binding sites [8, 19], which corroborates with further findings showing that some secondary structure motifs can specify ligand binding [6, 48].

These observations lead us to hypothesize that studying RNA structures at the extended base-pairing level (i.e. including non-canonical pairs) holds useful spatial and chemical information about target sites to characterize ligand binding. In practice, this means that a graph using vertices to represent nucleotides and edges to encode base-pairing interactions could offer a signature for RNA ligand binding sites (See **Fig. 1**). This paradigm distinguishes RNA from protein-ligand interactions where surface cavity topologies drive binding preferences [27]. Indeed, graphical representations of RNA base pairing networks have been developed in various tools [41, 42, 35, 7] for their ability to capture RNA-specific interactions in a scalable and interpretable manner.

0.3 Contribution

We show that base pairing networks can be used to automatically predict the binding of small molecules to RNAs. To this end, we propose a new prediction task which aims to bridge the gap between ligand and structure-based approaches. More specifically, we train a machine learning algorithm to use structural patterns in crystal structures of known RNA-ligand complexes to make predictions which allow us to identify potentially active ligands. In machine learning terms, the RNA-ligand complex is treated as an input-output pair where the target structure is the input to the model and the ligand is the output. In order to allow for ligand-based applications, we use molecular fingerprints of ligands as the outputs of our model. These are vector-based representations of chemicals designed for ligand space similarity searches and which can be conveniently handled by machine learning models. The prediction thus serves as a ligand-based tool since it can be used to search for active compounds in the ligand-space. At the same time, we include target information by training the model to produce fingerprints based on known RNA-ligand complexes. Similar methods have been proposed in recent preliminary works [29, 1] for protein binding.

We implement this strategy in `RNAmigos`, a data-driven tool for assisting the RNA-binding drug discovery process. Leveraging a network representation of RNA structures, `RNAmigos` learns structural patterns in known RNA-ligand complexes from crystal structure databases to predict chemical descriptors (i.e. fingerprints) for potential ligands. We demonstrate that the resulting molecular fingerprints serve as effective ligand search tools across different ligand classes, and provide evidence of its effectiveness at identifying binding sites in full RNA riboswitch structures.

1 Prediction Pipeline

In this section we outline the major components of `RNAmigos`. The pipeline begins with the 3D crystal structure of an RNA site as input, (i.e. from the PDB databank) without any ligand information, and outputs a predicted ligand fingerprint. We use the term ‘binding site’ to refer to any RNA region, typically of 5 to 20 nucleotides in size, that has been shown to bind a ligand, or which a user would like to probe for ligand binding. By ligand, we always refer to a small organic compound with potential to bind an RNA site. The full pipeline is illustrated in **Fig. 1** which can be divided into three major modules: binding site representation (**Section 1.1**), fingerprint prediction (**Section 1.2**), and ligand screening (**Section 1.3**). The RNA site is stored internally as a graph, which is then turned into a vector representation (embedding) in the representation phase. In the fingerprint prediction phase, a machine learning model takes the binding site embedding as input to produce a fingerprint vector. Finally, the resulting fingerprint vector is used in a similarity search to screen a library of small molecules and identify potential binders. All steps of the pipeline are fully automated, and the user need only input a target 3D structure or base pairing graph.

1.1 Binding Site Representation

We represent RNA sites as base pairing graphs and ligands as molecular fingerprints. Binding sites are extracted from a set of all NMR and crystal structures containing a free ligand (excluding metal ions) from the RCSB PDB databank [3], resulting in 2676 structures. The binding site for a given ligand is defined as the set of RNA nucleotides located inside a 5 Å cutoff distance from the nearest ligand atom. The distance cutoff is chosen following *David-Eden et al.* [8] who studied ribosome antibiotic binding sites. We note that at this point, the ligand is removed from the structure so that the graph contains only RNA base-pairing information. we build an RNA interaction graph from these residue selections, which contains nucleotides as nodes with backbone/base pairing interactions as edges. Node and edge annotations are taken from the `CaRNAval` database [41] which maintains fully annotated graphs of all PDBs with Leontis-Westhof and backbone interaction types computed by the software FR3D [43], and the RNA 3D Motif Atlas [35]. Graphs produced by the distance cutoff method often have chain discontinuities, lack structural context, or contain uninformative interactions (e.g. metal ions or long helical regions). To address this issue, we treat the initial graphs with automated post-processing steps and redundancy filters. (see Supplementary Material). We emphasize that all redundant binding sites are removed such that no two binding sites are identical in structure and bind the same ligand. After processing the structures, we obtain a total of 611 final binding site graphs with 202 different ligands. The binding site graphs have an average of 10.14 ±5.52 nodes and 11.44 ±7.23 edges.

In order to make ligand predictions from binding sites graphs with machine learning (ML) models, we build a vector representation, or feature vector, of the RNA graphs. An ML algorithm can then accept RNA graph feature vectors to output another vector which describes a predicted ligand (i.e. fingerprint). One simple method for characterizing non-vector objects (in this case, graphs) as vectors, is to define a features of a graph as distances from itself to a fixed set of representative (a.k.a. prototype) graphs that we denote as $\mathcal{P}$. Since every subsequent graph is compared against the same set of representative graphs, we can encode relative differences between all graphs in the dataset as vectors. This technique is known as graph dissimilarity embedding proposed by *Reisen et. al.* [4]. To measure distances between

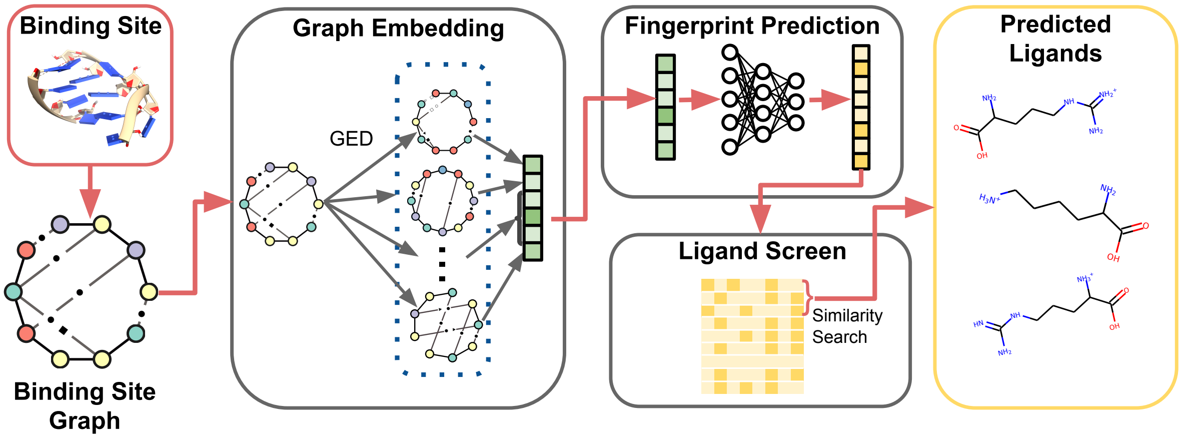


Figure 1: Outline of the RNAmigos pipeline. The user begins by providing either a 3D structure of an RNA site, or a base-pairing graph of the site. In this example, the input graph is drawn using the Leontis-Westhof convention for base-pairing annotation. This example graph contains 4 cis-Watson-Crick ● edges which define the secondary structure, and one cis-Hoogsteen ■ which is a non-canonical base pair. The graph is represented as a real-valued vector of fixed size in the Graph-Representation stage by applying the Graph Edit Distance (GED) graph comparison algorithm. The resulting vector is then passed to a machine learning model in the Fingerprint Prediction module which produces a molecular fingerprint. Finally, the fingerprint is used in a similarity search to identify molecules matching the prediction as candidate ligands for the input site.

graphs, we use the well-known Graph Edit Distance (GED) which is considered to be the gold standard for comparing graphs [13]. More formally, each element of the feature vector $\phi(g) \in \mathbb{R}^{|\mathcal{P}|}$ is given by $\text{GED}(g, p_i)$ for $p_i \in \mathcal{P}$.

A key advantage of this embedding approach is that it does not require manual feature building such as the one proposed in [2] for RNA-magnesium binding prediction. Instead, a graph can be featurized using only a well-defined distance function over graphs. Intuitively, the GED of a pair of graphs $(g, g^{'})$ represents the cost of minimally modifying graph g until it is isomorphic to $g^{'}$ through a predefined set of operations with a given cost. This can also be thought of as a generalization of aligning a pair of sequences, where modifying a node in the graph is equivalent to either deleting it, or substituting it (matching) for a node in the other graph. The more two graphs differ, the more numerous and costly the edit operations (e.g. inserting, deleting, or substituting nodes or edges) will be required, this is in turn reflected in a larger GED value. If two graphs are already identical, then no matching operations are needed and the GED will be 0. Most importantly, we choose this technique because the resulting distances between graphs can be endowed with domain-specific meaning through the appropriate definition of a cost function [34]. The GED cost function assigns a numeric penalty to each graph matching operation according to the degree of distortion the particular operation induces. For example, in our case we may wish to assign a high penalty when matching one edge in g with an edge in $g^{'}$ if the two edges have very different base-pairing type (canonical vs. non-canonical) as this would reflect a strong change in the structure. Our choice of cost function prohibits the matching of backbone edges to base pairing edges which results in more realistic alignments. Furthermore, we distinguish between substitutions across canonical and non-canonical base-pairing classes as these are likely to alter the global structure of the binding pocket. For a description of our cost function, GED implementation, and prototype selection process see Supplemental Material.

1.2 Fingerprint Prediction

The output for our model is a molecular fingerprint predicted from an input binding site. Molecular fingerprints are real fixed-size vectors representing the structural and chemical features of a small molecule such that similar molecules will have similar fingerprints. In this work we use a very common fingerprint implementation known as the MACCS fingerprint [10]. The MACCS fingerprint of a molecule sets the i^{th} bit in a $d = 166$ length binary vector to 1 if chemical property i is present in the compound and 0 otherwise. We use the set of 166 predefined chemical properties from the [33] implementation as a target vector for ML models. We emphasize that the computation of the fingerprint depends only on the chemical composition of the ligand and not on the RNA binding site to which it is bound. Thus, different binding sites may be assigned the same fingerprint if they bind the same or similar ligands. For all ligand (fingerprint) comparisons, we use the widely accepted Tanimoto similarity coefficient or its distance equivalent, the Jaccard distance [5] over bit vectors.

1.3 Ligand Screen

Since the fingerprint is a simplified representation of a real ligand, the final stage of the pipeline is to identify a full ligand. We call this the 'ligand-screening' phase since the predicted fingerprints would most likely be used in a similarity-based ligand search for active ligands. Given a predicted fingerprint $\hat{y} \in [0,1]^d$ for a binding site and a library of potential ligands $\mathcal{L}$, the goal of the screen is to select a ligand that best matches the true ligand. To this end, we compute a normalized rank $S(y, \hat{y})$ for each ligand in library of candidates based on its Jaccard distance to the predicted ligand and note the resulting rank of the true ligand.

$$S(y, \hat{y}) = 1 - \frac{\rho_{y,\hat{y},\mathcal{L}}}{|\mathcal{L}|} \tag{1}$$

Where $\rho_{y,\hat{y},\mathcal{L}}$ is the rank of the true ligand y in $\mathcal{L}$ relative to the prediction $\hat{y}$. A successful predictor will rank the true ligand closest to its prediction (normalized rank close to 0), while a random predictor will result in an average rank of 0.5.

Considering that the distribution of RNA ligands appears to be limited to specific sub-regions (see Supplementary Materials), this readout ensures that a classifier does not obtain a good score by simply predicting the average ligand as it would with a simple distance between y and $\hat{y}$. Instead, we ensure that the predictions with high scores are useful for partitioning the region of ligand space spanned by RNA binding compounds. Since there are currently no validated datasets of active and inactive binders for a given RNA site (such as DUDE for protein [32]), we use the set of RNA-binding ligands in the PDB as our ligand library. The target binder y for a given pocket is thus the ligand that was co-crystallized with the input binding site. This setting will allow us to determine whether binding site structure contains a discriminating signal for its co-crystallized ligand.

2 Pipeline Performance

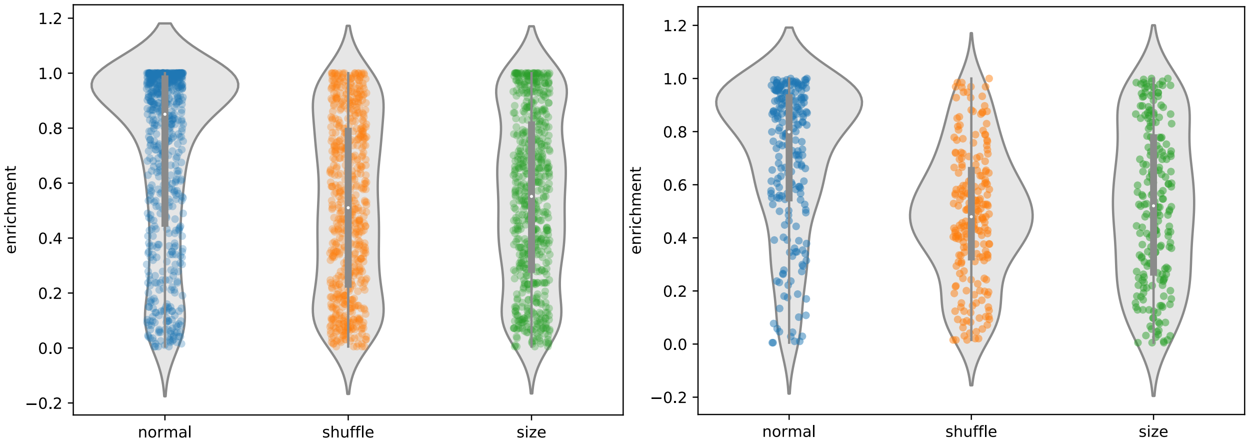
We perform a randomized grid search architectures for three ML algorithms: k-nearest neighbour, random forest, and artificial neural networks to obtain the best fingerprint predictor (see Supplemental Materials for details). As a final performance measure, we split our data into 20 train and validate subsets. For each fold, we train a model on the binding sites in training subset and evaluate on the remaining (validation) sites which were not used for training. We then record the $S(y, \hat{y})$ score for each binding site belonging to the validation set. The model with the highest performance was the neural network which reached an average score of 70% with a median of 80%. In other words, on average, the true ligand was found to be in the 70th percentile (top 30%) of the dataset when using the predicted fingerprint as a query.

For each architecture, we trained two additional classifiers as baselines. The first baseline was trained on a shuffled dataset, that is, the binding sites were assigned random ligand from the set of RNA ligands. This process scrambles specific relationships between binding sites and ligands while keeping the generic distribution of ligands identical. Comparing our learner to this baseline will indicate whether such relationships indeed exist in the data. Our second baseline was trained using the size of the binding site (number of nodes in the binding site graph) as its sole input feature. This acts as a control that ignores binding site structure yet contains some distinguishing information about the graphs. We plot the distribution of scores for each binding site prediction (**Fig. 2**) to confirm that our model clearly outperforms the baselines and demonstrate that the base-pairing networks at binding sites contain useful information for discriminating between potential ligands.

As expected, the size baseline appears to contain a moderate amount of signal with slight enrichment for a small subset of pockets appearing near the top of the distribution. The shuffled baseline has an average score of 50% predicting ligand ranks uniformly at random as expected. However, the score distribution for our best classifier still has a pronounced tail, indicating that this task remains a challenging one. It is also possible that RNA-drug complexes are still underrepresented as this field is relatively new compared to the protein binding drug field.

3 Performance by ligand class

Next, we ask whether the promising performance (70%) can be explained by a small set of ligands, or whether it is able to achieve high scores on a diverse set of ligands. **Fig. 2b** shows the distribution of scores when the score of all pockets binding to the same ligand are averaged and suggests that a significant number of different ligands are scored



(a) Per-binding site predictions.

(b) Per-ligand-grouped predictions.

Figure 2: Performance of RNAmigos on the task of ligand searching from predicted fingerprints derived from RNA binding sites. For each experimental condition (normal, shuffle, random), the distribution of prediction scores is given. Each point corresponds to the prediction score on a left out example during cross-validation. A score near 1 means that the predicted ligand ranks the known ligand as highly similar while 0 means the known ligand is the most dissimilar to the prediction. In **Fig. 2a** each point is the score on a single binding pocket. In **Fig. 2b** the scores for all pockets binding to the same ligand are averaged to produce a single point.

highly. This view treats each ligands as a distinct class, when in reality many ligands will be similar to each other and form ‘families’ or ‘classes’ of ligands. To get a better view of performance, we plot the same prediction scores over ligand types (202 non-identical ligands) against a hierarchical clustering dendrogram of each ligand as shown in **Fig. 3**. Colored-in subtrees indicate groups of ligands that are similar, (i.e., within 0.25 Jaccard distance of each other) which would indicate strong clustering. In this manner, we are able to assess the performance across ‘classes’ of similar ligands. We first observe that successful classifications are not restricted to a single class of ligands and instead show good predictions for diverse ligands. Interestingly, the class that is most consistently predicted accurately corresponds to the aminoglycosides (highlighted in the pink cluster to the right). Aminoglycosides are a class of antibiotics binding to bacterial RNA with well-defined binding sites [47], and are quite abundant in the dataset. Nucleic acid-like compounds, many of which bind riboswitches, also form a large family of binders (green) however results were less consistent than with aminoglycosides. A possible explanation for strong performance on aminoglycosides, apart for the larger number of examples obtained, is that these are typically large polysaccharide-like structures with a large number of interactions with the RNA. On the other hand, riboswitches bind much smaller molecules with a limited number of interactions. As a result, binding site requirements are much more complex and specific with aminoglycosides and the large number of interactions can only be fulfilled by a limited number of molecules. Finally, ligands clustered on the left of the dendrogram show the weakest performance. Since these groups show little branching in the dendrogram, we can conclude that they represent sparsely populated ligand classes for which we have few examples and thus obtaining more data in these regions would improve performance.

4 Binding site identification in unbound riboswitches

As our results suggest, RNA small-molecule binding sites contain structural signatures that are characteristic of certain ligands. In this experiment, we would like to determine whether (i) such signatures can be detected in unbound RNA (since our tool is trained only on bound RNA) and (ii) whether the signature can distinguish the binding site from surrounding structures in the same RNA. We focus on the structure of riboswitches which are a family of small molecule regulated RNA of major therapeutic interest [17], and are the only examples where the same RNA is crystallized in the bound and unbound state. This setting is distinct from previous experiments where we are given a binding site and search for a ligand. Instead, we begin with a ligand and would like to identify the binding site within a full unbound RNA. If a region within the unbound RNA contains structural features characteristic of the riboswitch ligand, we expect RNAmigos fingerprint predictions in these regions will obtain high scores for the known ligand. Indeed, the results shown in **Fig 4** demonstrate that when scanning the structure and obtaining fingerprints with RNAmigos, predictions that best match the true ligand tend to fall within the binding site. Of course, until more bound and unbound structures become available, this remains largely a qualitative experiment. However, these results suggest that the 3D graph

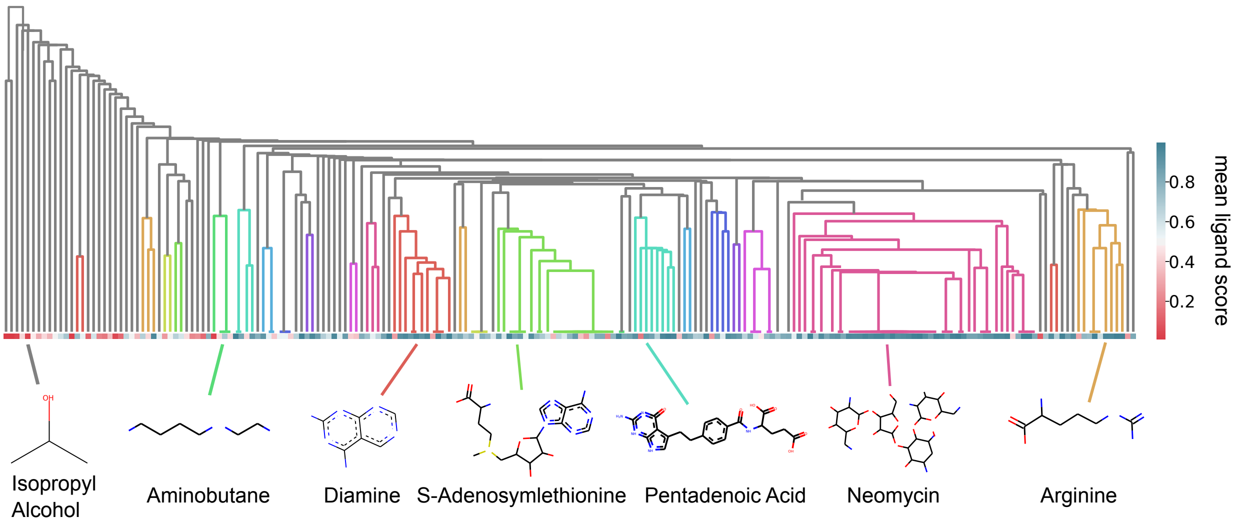


Figure 3: RNAmigos performance by ligand class. Hierarchical clustering dendrogram of the ligands classifying ligand families by similarity. Each cell in the horizontal grid is the average score for binding sites containing a given ligand. Ligand belonging to the same tree are grouped together by the clustering procedure. Colored-in sub-trees denote tight clusters which contain ligands within 0.25 Jaccard distance.

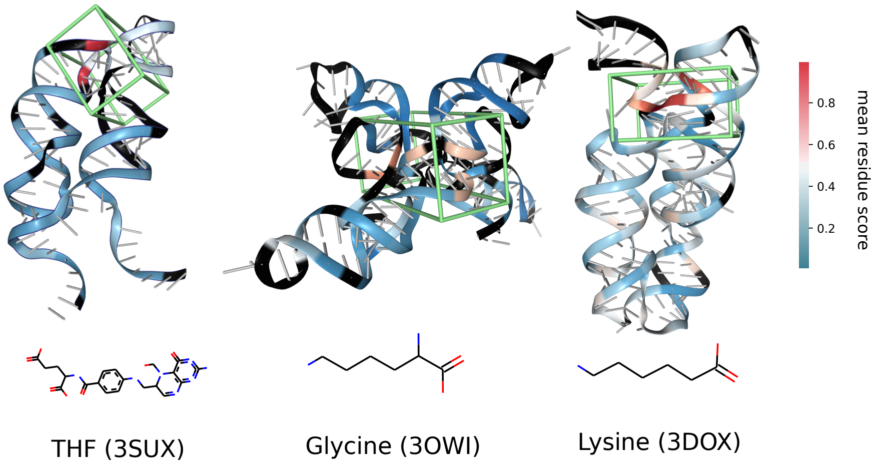


Figure 4: Binding side identification with RNAmigos. For each unbound riboswitch, we sample candidate sites from the full structure and predict a chemical fingerprint using RNAmigos. We score each binding site with its predicted fingerprint and the known riboswitch ligand. We color each residue with the average score over all candidate sites it belongs to. Green boxes highlight the known binding sites. Noticeably, high-score residues always fall within this region. Black residues did not yield any predictions as they were filtered out by our binding site representation step. To determine the location of the binding site in the unbound structure, we aligned the unbound structure to the bound structure using Chimera [36].

information of an RNA-ligand binding site holds a signature which distinguishes the binding site from the rest of the RNA and can persist between bound and unbound states.

5 Conclusion

We have developed a unique computational platform, RNAmigos, to show that extended RNA base pairing networks contain useful ligand-binding information. This finding has potential implications for assisting medicinal chemists in their search of drugs binding to RNA. By presenting graph representations of RNA binding sites to machine learning models, we are able to reliably retrieve the observed ligand from a database search with a good accuracy. Since our prediction is a fingerprint vector (chemical descriptor) and not a simple classification of ligands (i.e directly selecting a single ligand as output), the fingerprint itself can be used as a tool for searching large ligand databases. Next, we showed that the model shows promise for distinguishing the binding site from surrounding unbound structures. Once more data is available, this application can be validated for uses such as binding site identification and off-target prediction (ensuring that a certain compound does not have likely binding sites in undesired targets), where scalability is important as many targets need to be evaluated. While performance was strong across different ligand classes, it is apparent that ligand classes for which data is more abundant received more consistently positive predictions. Therefore, as more examples of RNA-ligand complexes are characterized by experimental and computational techniques, we believe that the performance of our platform will improve. Additional data will also allow for considerations regarding properties desired in medical applications such as synthesizability and drug-likeness [26]. Our choice of graphs for binding site representation reflects this consideration. Graph representations are highly amenable to additional sources of information and can natively hold additional information such as evolutionary or chemical data in their nodes or edges without requiring changes to the pipeline.

Although the signal obtained from a graph-based representation of structures is promising, we acknowledge that it is not the only viable approach. It would be relevant to compare our approach to methods that directly use the atomic-level representation. Comparable data-driven approaches such as [38] are currently unavailable pending a code refactoring, and [2] is no longer being distributed. We hope that this work will motivate explorations of other data-driven methods for RNA-binding molecules with applications in understanding biological control and accelerating RNA drug development.

6 Supplemental Material

6.1 Data Pre-processing

The initial binding site graph includes all nucleotides (and interactions between them) within a user-defined distance from the ligand in the 3D structure. We then apply the following processing steps to produce the final graphs to be trained on:

1. **Expand structural context**. We add all nodes (along with their incident edges) found within 2 hops using Breadth-First traversal. This includes greater structural context in the binding site, as well as eliminates small discontinuities in the chain that resulted from the distance cutoff.
2. **Dangle trim** Starting from the end of each strand (one contiguous set of nodes connected by the backbone) in the binding site graph, we remove nodes as long as they only interact with the backbone. These structures are often known as 'dangles'.
3. **Stem trim** For graph comparison efficiency we remove any terminal Watson-Crick edges that stack on another Watson-Crick edge, and whose participating bases only engage in Watson-Crick interactions. In this manner we shorten long helices found at the terminal ends of the binding site. In the case of multiple stacking edges, all but one are removed.
4. **Loop collapse** Contiguous bases that do not participate in any non-backbone interactions are collapsed into a single special 'loop' node. For example, a large hairpin with no base pairing interactions would be represented as a single node. This is done to reduce the size of the graphs and speed up the GED phase.
5. **Redundant filtering** Any two graph-fingerprint pairs (x, y), (x', y') such that *both* $GED(x, x') = 0$ and $J(y, y') = 0$ are removed as redundant data pairs. Here GED denotes the graph edit distance and $J(., .)$ the Jaccard distance on bit vectors (the distance equivalent of the Tanimoto [5] similarity, often used for comparing fingerprints. In other words, any ligand that occurs more than once is only included if the pocket it binds to is not identical to others with the same ligand.

6.2 Graph Edit Distance Computation

The Graph Edit Distance (GED) between two graphs G and H is defined as follows:

$$GED(G, H) = \min_{(e_1,\ldots,e_k)\in\Upsilon(G,H)} \sum_{i=1}^{k} c(o_i). \tag{2}$$

Where Υ is the set of all edit sequences which transform G into H. Edit operations include: node/edge matching, deletion, and insertion. $c(o)$ is the cost of performing edit operation o and c is known as the cost function.

We implement the A* GED algorithm which is guaranteed to return the sequence of edit operations that minimizes the total cost function. This value corresponds to the distance between G and H. Pseudocode for A*GED can be seen in **Algorithm 1**.

Algorithm 1: A* GED

```
Data: Pair of graphs $G$, $H$, cost function $c$ and heuristic $h$. WLOG let $G$ be the smaller of the two graphs.
Result: Minimum cost distance and alignment between two graphs.
$OPEN \leftarrow priorityQueue()$
$V_G \leftarrow G.nodes()$
$V_H \leftarrow H.nodes()$
$v \leftarrow$ first node in G
foreach $v' \in V_H$ do
    $OPEN.add((v_0, v'), c(v_0, v') + h(v_0, v'))$
end
while $OPEN$ do
    $v_{min} \leftarrow OPEN.pop()$
    Let $\mathcal{M}_k \leftarrow$ be partial mapping $\{(v_1, v_1'), .., (v_k, v_k')\}$
    if $|\mathcal{M}_k| = |V_G|$ then
        Mapping complete
        return $\mathcal{M}_k$
    end
    Add nodes at next depth
    foreach $u \in V_H \setminus v_{min}$ do
        $OPEN.add(v_{min} \cup (v_{k+1}, u), c(v_{k+1}, u) + h(v_{k+1}, u))$
    end
end
```

Intuitively, we can think of the process of matching nodes between graphs as forming a path down a tree of all possible matches. A node in the tree would contain a pairing between two nodes, and a path from root to leaf would be a mapping $\mathcal{M}$ which covers all the nodes in the graphs. The algorithm reduces to a minimum cost path search over the tree of all possible edit sequences from G to H. In a Dijkstra fashion, the algorithm processes nodes in order of ascending GED cost (i.e. cost of adding operation e_i under current cost function) plus a lower bound estimate of the cost of processing the remainder of the graph, also known as a heuristic h. The heuristic is used to steer the search toward more promising paths while still guaranteeing optimality. Once the search arrives at a leaf of the tree, a complete node assignment has been achieved.

6.3 Local Edge-Breaking Heuristic

After partial mapping is computed, we estimate a lower bound h using a heuristic. This is done to estimate the minimum amount of editing that is left to be done given the current partial matching. The heuristic thus prioritizes visiting nodes with lower cost bounds which are more likely to be optimal. For a heuristic to be valid, it must always be less than or equal to the actual cost of completing the matching. The most widely used heuristic is proposed by [4] and applies a greedy node assignment on unmapped nodes while ignoring graph structure. In simple terms, for each node, match it to the lowest cost node (ignoring graph structure and whether it was matched already) and do the same for each edge. Add to the lower bound estimate, the difference in number of unmatched nodes and edges between the two graphs. Since repeated substitutions and graph structure are ignored, this is necessarily a lower bound on the true cost. We call this strategy the `global` heuristic. However, because the main contribution to mapping cost in our current cost function comes from edge insertions and deletions, we implement a slightly more detailed heuristic that considers the arrangement of unmapped edges induced by the current mapping and locally anticipates future edge deletions.

More formally, let $\mathcal{M}_t$ be a partial mapping between two graphs at step t in the search. This is a mapping between the nodes of G and H. If graph G had nodes $\{1, 2, 3\}$ and H had nodes $\{\alpha, \beta, \gamma\}$. A partial mapping would could look like $\mathcal{M} = \{(1, \beta), (2, \alpha)\}$. We denote the set of nodes of the mapping belonging to G and H as $\mathcal{M}^G$ and $\mathcal{M}^H$. Finally, let $\mathcal{M}_t(n)$ return the node that n maps to (e.g. $\mathcal{M}_t(1) \to \beta$)

$$h_{\text{local}}(\mathcal{M}_t) = d \sum_{v \in \mathcal{M}^G} |deg(\bar{v}) - deg(\mathcal{M}(\bar{v}))| \tag{3}$$

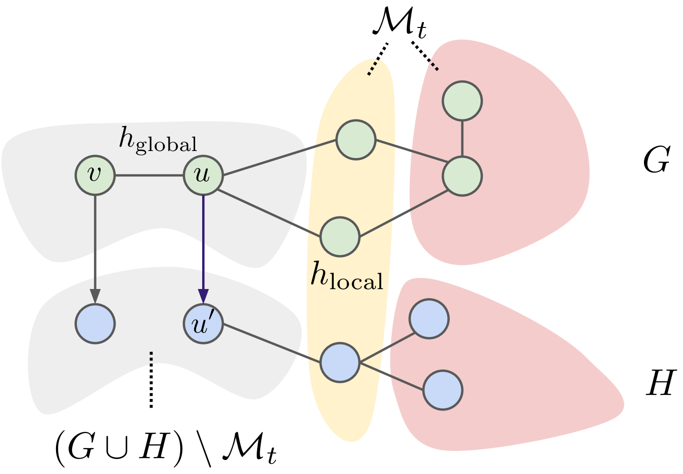


Figure 5: Illustration of local heuristic. We show that the local heuristic operates on a disjoint set of edges compared to the global heuristic which allows us to sum the contributions from the two heuristics.

Where $(\bar{v})$ represents a node v with all neighbors that belong to $\mathcal{M}$ omitted. Thus, when taking the degree of $\bar{v}$ we get the number of unmapped edges incident to v. Hence, if two nodes are mapped to each other, the difference in number of adjacent unmapped edges weighted by the edge deletion cost d necessarily implies a requuired future edge insertion/deletion cost. A key challenge in estimating $h(\mathcal{M})$ is to provide a large enough lower bound with the lowest time complexity (i.e. overhead). Higher lower bounds are closer to the true future cost, yet are usually more expensive to compute. Since the neighbours of a node can be maintained and accessed in constant time with some pre-processing, we can very quickly compute $h(\mathcal{M})$ and steer the search away from edge breaking paths. A benchmark on a sample of our dataset shows that this heuristic provides a modest speed-up in the size region where the majority of our graphs lie **Fig. 6, 7**. However, by summing the local heuristic with the established global heuristic discussed above, we achieve a significant speedup which is especially useful when computing all-to-all graph alignments.

Thus, the new combined heuristic is:

$$h(\mathcal{M}_t) = h_{\text{local}}(\mathcal{M}_t) + h_{\text{global}}(\mathcal{M}_t) \tag{4}$$

We must still guarantee that the combined heuristic is still a lower bound (i.e. it does not exceed the true remaining matching cost). To do so, we enforce that $h_{\text{global}})$ and h_{local} operate on disjoint sets of edges. This guarantees that we are not double counting some edge. More formally, we consider that each edge is a tuple of nodes (u, v). For all $(u, v) \notin \mathcal{M}_t$ (fully unmapped edges) we apply h_{global}. Meanwhile, h_{local} is used on nodes such that *one* of u or v has been mapped. Nodes with both u and v belonging to $\mathcal{M}_t$ are not considered in the heuristic computation since they are already fully mapped and will not contribute to future cost. Since an edge can only be in one of those three categories, and each heuristic is a valid lower bound their sum is also valid. We illustrate this in **Figure. 5**.

We stop GED computation for a pair of pockets after 90 seconds if the search has not terminated. We found that using this cutoff, 83% of pocket comparisons finished before the timeout. For those that exceeded the timeout we take the current best score as the GED. An investigation on the potential distances between the pockets that could not be compared in time showed they were always too different to be of interest for our methods. Computation of the full pairwise distance matrix was parallelized and run on a 20 core Ubuntu 16.04 server with Intel Xeon 2.90 GHz processors.

6.4 Cost Function

Let $\mathcal{L}(.)$ be a function that returns the edge label for a given edge, B denote the set of backbone edge labels, and W the set of canonical base pair labels. We define an RNA cost function over edges as follows:

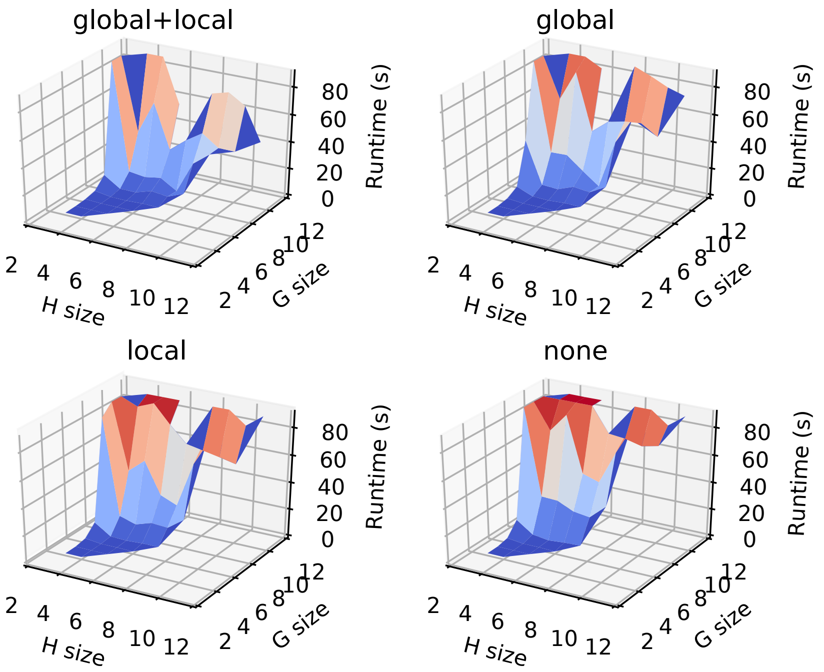


Figure 6: Benchmarking of local edge-breaking heuristic. Each 3D plot shows the runtime for aligning pairs of graphs with different sizes for each heuristic strategies. The X-Y plane shows the size of each graph G and H being aligned and the Z plane is the time in seconds needed to reach a final alignment. We show that by combining the established 'global' heuristic with our proposed 'local' heuristic we are able to gain significant speedups in the region of ~10 node graphs. The color highlights the height of the surface. 'none' is a run that does not use any heurstic lower bounds.

$$c(p \rightarrow q) = \begin{cases} 0 & \mathcal{L}(p) = \mathcal{L}(q) \\ \infty & \mathbb{1}_{\mathcal{L}(p) \in B} \neq \mathbb{1}_{\mathcal{L}(q) \in B} \\ s + g(p, q) & \mathcal{L}(p) \neq \mathcal{L}(q) \end{cases}$$

$$g(p \rightarrow q) = \begin{cases} z & \mathbb{1}_{\mathcal{L}(p) \in W} \neq \mathbb{1}_{\mathcal{L}(q) \in W} \\ 0 & \text{else} \end{cases}$$

Our choice of cost function prohibits the matching of backbone edges to base-pairing edges which results in more realistic alignments as well as greatly improves the speed of GED computation. When a substitution results in a mismatch between base pairing edges, we introduce an additional penalty $g(p, q)$ which penalizes mappings that alter canonical interactions known to contribute more to stability and overall structure than non-canonical pairs. Any edge and node deletions incur a cost of d. We use parameters $s = 1, d = 4, z = 2$ to discourage edge deletions and breaking canonical interactions. The cost function proposed here is very simple and can eventually incorporate additional features such as sequence identity, chemical properties, and evolutionary information.

6.5 Prototype Selection

We implement three different strategies for selecting the set of prototype graphs $\mathcal{P}$.

- A random selector simply selects k points at random in the graph space as representatives.
- The `k-centers` strategy attempts to find k centroid graphs (i.e. those that minimize their distance to a given subset of the space) in a strategy similar to the well-known k-means clustering algorithm.

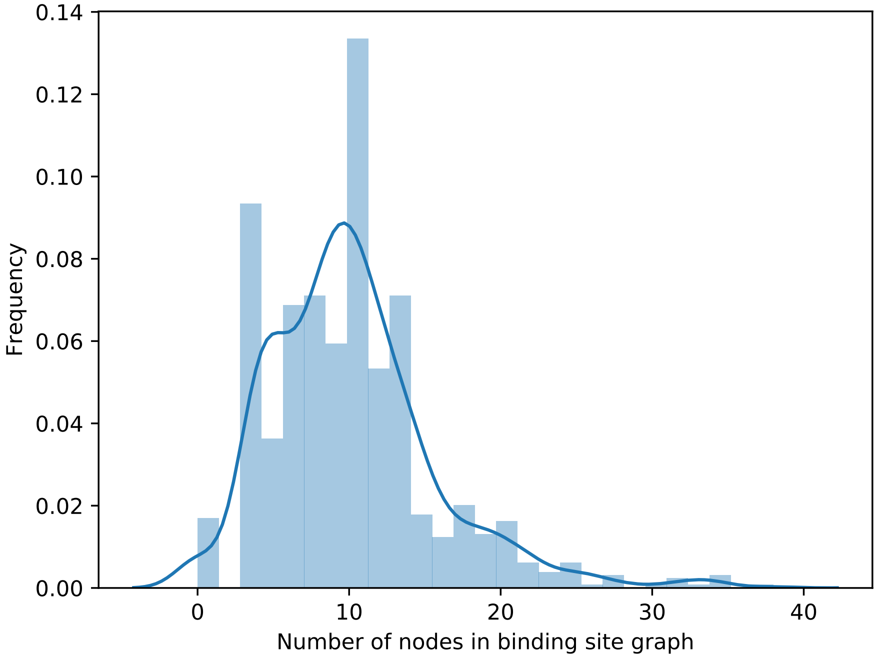


Figure 7: Distribution of number of nodes in binding site graphs

- The `spanning` strategy attempts to build a prototype set that is as diverse as possible by starting with the centroid graph and greedily selecting a prototype whose distance to the currently selected partial set is maximized.

7 Classifier Training

7.1 Hyperparameter Search

Given the combinatorial nature of hyperparameter selection, we tune the fingerprint predictor using a randomized grid search combined with 20-fold cross-validation. We tested three different classifier algorithms: k-nearest neighbors, random forests, and artificial neural networks. Apart from classifier-specific hyperparameters, we tune the embedding method and embedding size which are used to produce the input to the classifiers. We need to take care to not select a prototype graph that belongs to the validation split since this would introduce a bias at evaluation time. To address this, for each training split we remove training indices x from the set of binding site graphs $\mathcal{D}$ to obtain a prototype set $\mathcal{P}$. We then embed the training graphs with respect to the current $\mathcal{P}$ and perform training. At the next training split we recompute $\mathcal{P}$ with the same procedure. The pseudocode is shown in **Algorithm 2**. Below are the parameter settings evaluated in the grid search.

- Embedding size: $\{10, 30, \ldots, 300\}$
- Prototype selection: `k-centers`, `spanning`, `random`
- MLP
  - Hidden layer sizes: $\{(50,), (50, 100), (100, 50), (200,)\}$
  - Activation function: `tanh`, `relu`
- Random Forest
  - Number of trees
- KNN
  - Number of neighbors: $\{1, 3, 5, 7\}$.

```
Algorithm 2: Model training without prototype selection bias.
Data: Set of binding site graphs $\mathcal{D}$
Result: Model accuracy controlled for prototype bias.
for $x, y$ in split($\mathcal{D}, Y$) do
    $\mathcal{P} \leftarrow prototypes(\mathcal{D} \setminus \{x\})$
    $X \leftarrow embed(\mathcal{D}_x, \mathcal{P})$
    $train(X, y)$
end
```

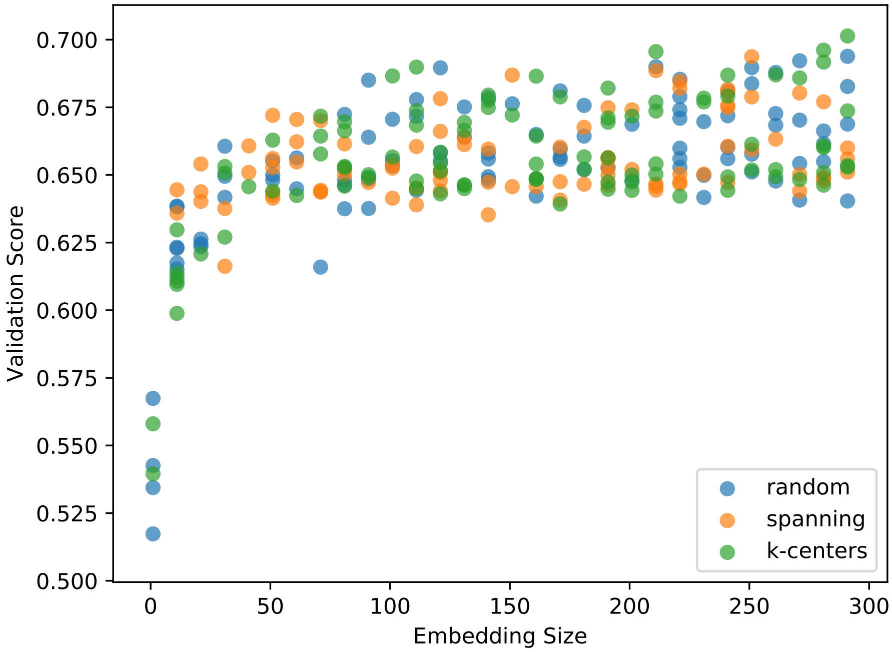


Figure 8: Validation performance with different embedding sizes and techniques

7.2 Model Training Results

In **Fig. 8** we show that the model is able to achieve ligand screening scores near 0.7 as embedding sizes get close to 100. Of course, we note that very small embeddings do not contain enough information to achieve strong performance. However, the accuracy gains for much larger embeddings are quite small after embeddings of size 100. Second, we note that the prototype selection algorithm does not have a discernible effect on the classifier quality. This could be due to the effect of a small dataset which would render most prototype selections beyond a certain size largely equivalent. Finally, we show in **Fig. 9** that neural networks (MLP) achieved the strongest performance (0.67 mean score), while k-NNs show slightly reduced scores (0.65 mean score) which may suggest that the hypothesis of 'similar binding sites bind similar ligands' appears to hold for some binding sites [16].

7.3 Ligand Space

We can use fingerprint vectors to inspect the diversity and distribution of RNA ligands within the larger chemical space of known ligands. We use a t-SNE [28] embedding to visualize the high dimensional fingerprints in a 2D space while placing similar points close to each other in the reduced representation. Embedding a random sample of 2000 co-crystallized protein ligands and all RNA ligands (**Fig. 10**) shows that while RNA ligands cover a seemingly wide portion of the space, it is clear that certain sub-regions are more populated. This distribution suggests that some chemical properties are preferred for RNA binders. Within these regions we observe clustering of the well-known aminoglycosides (see HYG, CYY in the plot) and riboswitch binding nucleic acid-like ligands (FMN, TPP, GTP).

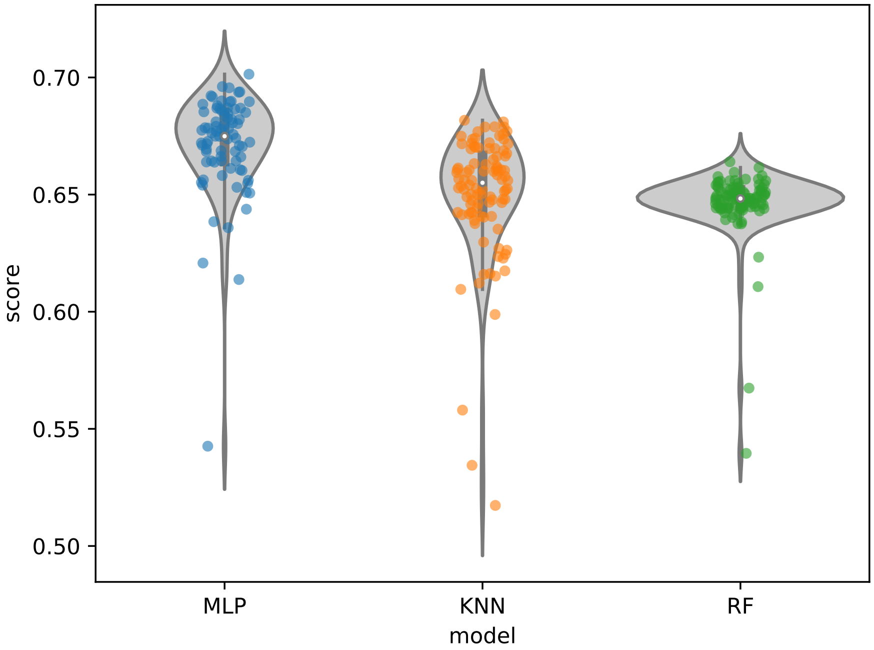


Figure 9: Effect of ML model choice on performance. Each point in the violin is the average performance for each hyperparameter setting with a given ML model. (MLP: Multi-layer perceptron, KNN: k-nearest neighbours, RF: Random Forest)

Despite RNA ligands falling to specific sub-regions of the ligand space, these regions are also densely co-inhabited by protein ligands (blue points) which corroborates similarities reported between protein and RNA binding compounds in other studies [49]. This would further motivate the use of structure-aware ligand-based techniques to better explore the known chemical space. In **Fig. 11** we use the MACCS fingerprints to show, as expected, that the space of known RNA ligands more constrained than for protein ligands. Indeed, the average variance over all dimensions of fingerprints for RNA ligands is 0.34 for protein and 0.28 for RNA and the number of dimensions where RNA fingerprints have higher variance than protein is only 37 out of 166. This finding motivates the use of an enrichment-based scoring function over a direct distance between fingerprints, since a predictor that outputs an average RNA ligand can achieve seemingly high performance in a constrained space.

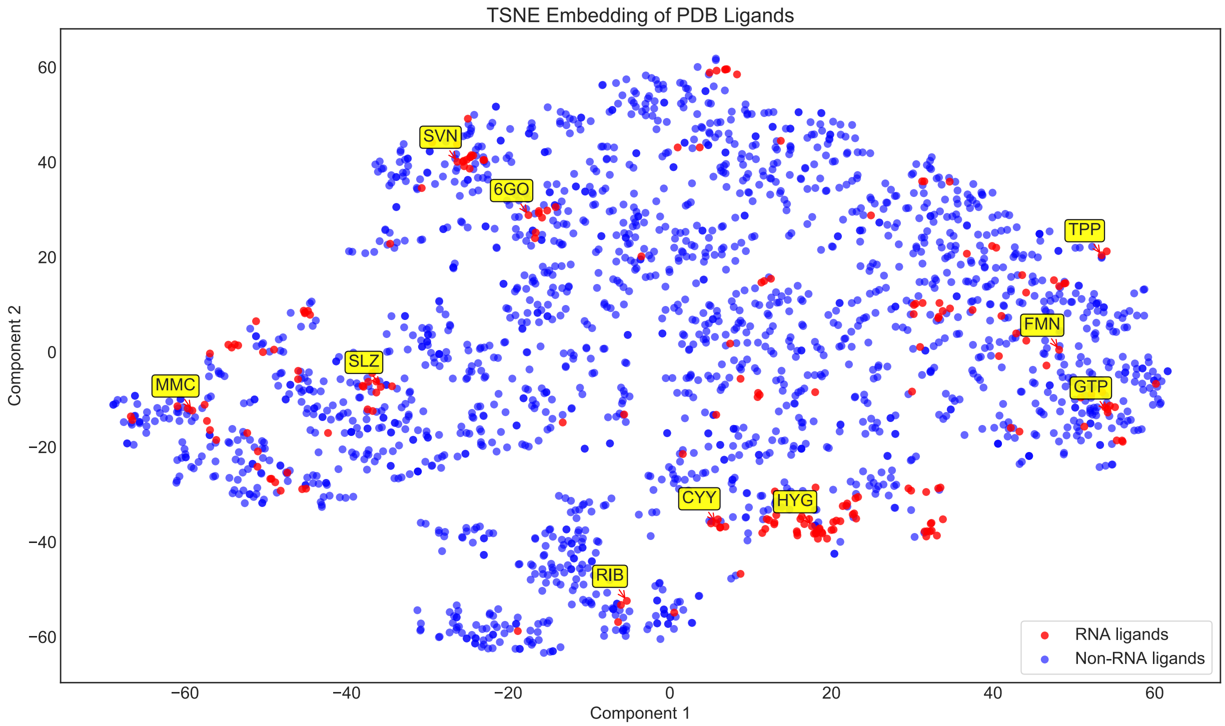


Figure 10: Ligand space visualization with t-SNE. We plot the embedding over ligand fingerprints using Jaccard distance. We sample 2000 ligands found in the RCSB PDB databank (~10% of all ligands in the database) and include all RNA ligands shown in red. For readability, we add the 3-letter code annotations for a random subset of the RNA ligands.

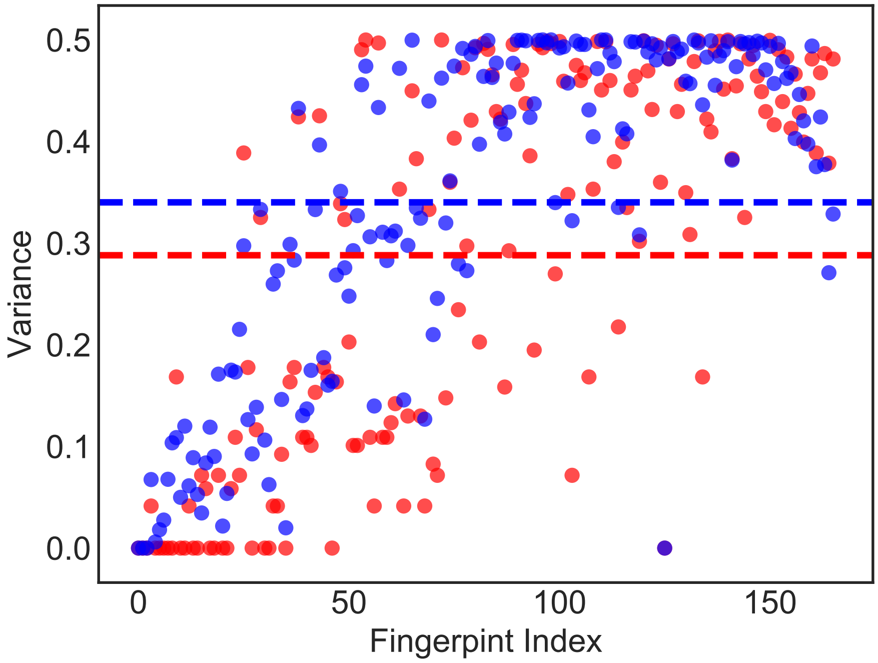


Figure 11: Variance for each fingerprint index over protein and RNA fingerprints. We show in dotted lines the mean variance over all dimensions.

References

[1] Tristan Aumentado-Armstrong. Latent molecular optimization for targeted therapeutic design. *arXiv preprint arXiv:1809.02032*, 2018.

[2] D Rey Banatao, Russ B Altman, and Teri E Klein. Microenvironment analysis and identification of magnesium binding sites in rna. *Nucleic acids research*, 31(15):4450–4460, 2003.

[3] HM Berman, J Westbrook, Z Feng, G Gilliland, TN Bhat, H Weissig, IN Shindyalov, and PE Bourne. The protein data bank nucleic acids research, 28: 235-242. *URL: www. rcsb. org Citation*, 2000.

[4] Horst Bunke and Kaspar Riesen. Graph classification based on dissimilarity space embedding. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 996–1007. Springer, 2008.

[5] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[6] Jessica L Childs-Disney, Tuan Tran, Balayeshwanth R Vummidi, Sai Pradeep Velagapudi, Hafeez S Haniff, Yasumasa Matsumoto, Gogce Crynen, Mark R Southern, Avik Biswas, Zi-Fu Wang, et al. A massively parallel selection of small molecule-rna motif binding partners informs design of an antiviral from sequence. *Chem*, 4(10):2384–2404, 2018.

[7] José Almeida Cruz and Eric Westhof. Sequence-based identification of 3d structural modules in rna with rmdetect. *Nature methods*, 8(6):513, 2011.

[8] Hilda David-Eden, Alexander S Mankin, and Yael Mandel-Gutfreund. Structural signatures of antibiotic binding sites on the ribosome. *Nucleic acids research*, 38(18):5982–5994, 2010.

[9] Anita Donlic and Amanda E Hargrove. Targeting rna in mammalian systems with small molecules. *Wiley Interdisciplinary Reviews: RNA*, 9(4):e1477, 2018.

[10] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[12] Peter Ertl. Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *Journal of chemical information and computer sciences*, 43(2):374–380, 2003.

[13] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[14] Erik Gawehn, Jan A Hiss, and Gisbert Schneider. Deep learning in drug discovery. *Molecular informatics*, 35(1):3–14, 2016.

[15] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[16] V Joachim Haupt, Simone Daminelli, and Michael Schroeder. Drug promiscuity in pdb: protein binding site similarity is key. *PLoS one*, 8(6):e65894, 2013.

[17] John A Howe, Hao Wang, Thierry O Fischmann, Carl J Balibar, Li Xiao, Andrew M Galgoci, Juliana C Malinverni, Todd Mayhood, Artjohn Villafania, Ali Nahvi, et al. Selective small-molecule inhibition of an rna structural element. *Nature*, 526(7575):672, 2015.

[18] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[19] Efrat Kligun and Yael Mandel-Gutfreund. Conformational readout of rna by small ligands. *RNA biology*, 10(6):981–989, 2013.

[20] Kale Kundert, James E Lucas, Kyle E Watters, Christof Fellmann, Andrew H Ng, Benjamin M Heineike, Christina M Fitzsimmons, Benjamin L Oakes, Jiuxin Qu, Neha Prasad, et al. Controlling crispr-cas9 with ligand-activated and ligand-deactivated sgrnas. *Nature communications*, 10(1):2127, 2019.

[21] P Therese Lang, Scott R Brozell, Sudipto Mukherjee, Eric F Pettersen, Elaine C Meng, Veena Thomas, Robert C Rizzo, David A Case, Thomas L James, and Irwin D Kuntz. Dock 6: Combining techniques to model rna–small molecule complexes. *Rna*, 2009.

[22] Neocles B Leontis, Aurelie Lescoute, and Eric Westhof. The building blocks and motifs of rna architecture. *Current opinion in structural biology*, 16(3):279–287, 2006.

[23] Neocles B Leontis and Eric Westhof. Conserved geometrical base-pairing patterns in rna. *Quarterly reviews of biophysics*, 31(4):399–455, 1998.

[24] Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of rna base pairs. *Rna*, 7(4):499–512, 2001.

[25] Neocles B Leontis and Eric Westhof. Analysis of rna motifs. *Current opinion in structural biology*, 13(3):300–308, 2003.

[26] Christopher A Lipinski. Lead-and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, 1(4):337–341, 2004.

[27] Jiaying Luo, Wanlei Wei, Jérôme Waldispühl, and Nicolas Moitessier. Challenges and current status of computational methods for docking small molecules to nucleic acids. *European journal of medicinal chemistry*, 168:414–425, 2019.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[29] Vincent Mallet, Carlos G Oliver, Nicolas Moitessier, and Jerome Waldispuhl. Leveraging binding-site structure for drug discovery with point-cloud methods. *arXiv preprint arXiv:1905.12033*, 2019.

[30] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.

[31] N Moitessier, P Englebienne, D Lee, J Lawandi, Corbeil, and CR. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. *British journal of pharmacology*, 153(S1):S7–S26, 2008.

[32] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.

[33] Noel M O'Boyle, Chris Morley, and Geoffrey R Hutchison. Pybel: a python wrapper for the openbabel cheminformatics toolkit. *Chemistry Central Journal*, 2(1):5, 2008.

[34] Erdem Ozdemir and Cigdem Gunduz-Demir. A hybrid classification model for digital pathology using structural and statistical pattern recognition. *IEEE Transactions on Medical Imaging*, 32(2):474–483, 2012.

[35] Anton I Petrov, Craig L Zirbel, and Neocles B Leontis. Automated classification of rna 3d motifs and the rna 3d motif atlas. *Rna*, 2013.

[36] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. Ucsf chimera?a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[37] Patrick Pfeffer and Holger Gohlke. Drugscorerna knowledge-based scoring function to predict rna- ligand interactions. *Journal of chemical information and modeling*, 47(5):1868–1876, 2007.

[38] Anna Philips, Kaja Milanowska, Grzegorz Łach, and Janusz M Bujnicki. Ligandrna: computational predictor of rna–ligand interactions. *RNA*, 2013.

[39] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

[40] Ely B Porter, Jacob T Polaski, Makenna M Morck, and Robert T Batey. Recurrent rna motifs as scaffolds for genetically encodable small-molecule biosensors. *Nature chemical biology*, 13(3):295, 2017.

[41] Vladimir Reinharz, Antoine Soulé, Eric Westhof, Jérôme Waldispühl, and Alain Denise. Mining for recurrent long-range interactions in rna structures reveals embedded hierarchies in network families. *Nucleic Acids Research*, 46(8):3841–3851, 2018.

[42] Roman Sarrazin-Gendron, Vladimir Reinharz, Carlos G Oliver, Nicolas Moitessier, and Jérôme Waldispühl. Automated, customizable and efficient identification of 3d base pair modules with bayespairing. *Nucleic acids research*, 2019.

[43] Michael Sarver, Craig L Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B Leontis. Fr3d: finding local and composite recurrent structural motifs in rna 3d structures. *Journal of mathematical biology*, 56(1-2):215–252, 2008.

[44] Li-Zhen Sun, Dong Zhang, and Shi-Jie Chen. Theory and modeling of rna structure and interactions with metal ions and small molecules. *Annual review of biophysics*, 46:227–246, 2017.

[45] Ignacio Tinoco Jr and Carlos Bustamante. How rna folds. *Journal of molecular biology*, 293(2):271–281, 1999.

[46] Tyler E Wagner, Jacob R Becraft, Katie Bodner, Brian Teague, Xin Zhang, Amanda Woo, Ely Porter, Bremy Alburquerque, Brian Dobosh, Oliwia Andries, et al. Small-molecule-based regulation of rna-delivered circuits in mammalian cells. *Nature chemical biology*, 14(11):1043, 2018.

[47] Frank Walter, Quentin Vicens, and Eric Westhof. Aminoglycoside–rna interactions. *Current opinion in chemical biology*, 3(6):694–704, 1999.

[48] Zi-Fu Wang, Andrei Ursu, Jessica L Childs-Disney, Rea Guertler, Wang-Yong Yang, Viachaslau Bernat, Suzanne G Rzuczek, Rita Fuerst, Yong-Jie Zhang, Tania F Gendron, et al. The hairpin form of r (g4c2) exp in c9als/ftd is repeat-associated non-atg translated and a target for bioactive small molecules. *Cell chemical biology*, 26(2):179–190, 2019.

[49] Katherine Deigan Warner, Christine E Hajdin, and Kevin M Weeks. Principles for targeting rna with drug-like small molecules. *Nature Reviews Drug Discovery*, 17(8):547, 2018.