A Comprehensive Library for RNA Structure-Function Modeling

Luis Wyss *
Max Planck Institute of Biochemistry
Munich, Germany
wyss@biochem.mpg.de

Vincent Mallet *
Mines Paris, PSL Research University, CBIO
Paris, France
vincent.mallet@minesparis.psl.eu

Wissam Karroucha
Mines Paris, PSL Research University, CBIO
Paris, France
wissam.karroucha@minesparis.psl.eu

Karsten Borgwardt †
Max Planck Institute of Biochemistry
Munich, Germany
borgwardt@biochem.mpg.de

Carlos Oliver †
Vanderbilt University
Nashville, Tennessee, US
carlos.oliver@vanderbilt.edu

Abstract

The RNA structure-function relationship has recently garnered significant attention within the deep learning community, promising to grow in importance as nucleotide structure models advance. However, the absence of standardized and accessible benchmarks for deep learning on RNA 3D structures has impeded the development of models for RNA functional characteristics.

In this work, we introduce a comprehensive set of benchmarking datasets for RNA structure modeling, designed to address this gap. Our library includes easy data distribution and encoding, splitters, and evaluation methods, providing a robust suite for comparing models. Beyond the proposed tasks, our library is modular and thereby can easily be tailored by researchers to their question at hand. We provide experiments highlighting the ease of use of our library.

Source code: github.com/cgoliver/rnaglib/

1 Introduction

Recent years have witnessed the advent of deep learning methods for structural biology culminating in the award of the Nobel Prize in Chemistry. AlphaFold (Jumper et al., 2021) revolutionized protein structure prediction, equipping the field with millions of new structures. Breakthroughs go beyond structure prediction, notably in protein design (Watson et al., 2023; Dauparas et al., 2022), drug discovery (Schneuing et al., 2024; Corso et al., 2022), or fundamental biology (van Kempen et al., 2022). While it is tempting to attribute the success of these methods to the increase in available structural data caused by AlphaFold, most of the methods are actually not reliant on them. Instead, it seems that these breakthroughs result from progress in training neural encoders modeling protein structures (Jing et al., 2020; Zhang et al., 2022b; Gainza et al., 2020; Wang et al., 2022). This progress is in turn rooted in solid competitions (CASP, CAPRI), and benchmarks (Townshend et al., 2021; Kucera et al., 2023; Zhu et al., 2022; Jamasb et al., 2024; Notin et al., 2023). By setting clear goals, such benchmarks are the foundation for the development of structure encoders. Yet their focus has been on the structure of proteins. If the field of nucleotide structural biology is to advance analogously, similar infrastructure will be required.

[*] equal contribution

[†] equal supervision

Ribonucleic acids (RNAs) are a large family of molecules which support biological functions along every branch of the tree of life. Beyond coding for proteins, non-coding RNAs carry out biological functions by adopting complex 3D folds (Cech & Steitz, 2014) like proteins do and take up diverse roles in cellular function, including gene regulation, RNA processing, and protein synthesis (Statello et al., 2021). However, our understanding of non-coding RNAs and their functions remains limited. This can be largely attributed to the negatively charged nature of RNA backbones, which makes it flexible and limits the availability of high-resolution RNA structures, and imposes significant modeling challenges. Another predominant challenge to a functional understanding of RNA 3D structure lies in the lack of infrastructure for the development and evaluation of function prediction models. In this work, we propose a benchmarking suite to act as this facilitating framework. Our key contributions include:

- seven tasks related to RNA 3D structure that represent various biological challenges. Each task consists of a dataset, a splitting strategy, and an evaluation method, laying the ground for comparable, reproducible model development.
- modular annotators, filters and splitting strategies, both novel and from existing literature, facilitating the addition of new tasks by other researchers. This flexibility ensures that our suite can be continuously updated to accommodate emerging needs and advancements in RNA research.

2 Related Work

2.1 Protein Focused Benchmarking

Classic tasks independently appeared in unrelated papers, such as GO term and EC number prediction (Gligorijević et al., 2021), fold classification (Hou et al., 2018), binding site detection and classification (Gainza et al., 2020) or binding affinity regression (Wang et al., 2005a). To our knowledge, *ATOM3D* (Townshend et al., 2021) was the first systematic benchmark for molecular systems, albeit heavily focused on proteins. Similar, more comprehensive tools were then proposed, such as *ProteinShake* (Kucera et al., 2023), *ProteinWorkshop* (Jamasb et al., 2024) and *TorchDrug* (Zhu et al., 2022), that unify the above tasks and lower the barrier to develop protein structure encoders. *ProteinGym* (Notin et al., 2023) addresses the evaluation of mutation effects, while (Buttenschoen et al., 2024; Kovtun et al., 2024; Durairaj et al., 2024) address protein interactions. These works contain notable efforts to scale datasets using predicted structures and propose strict splitting strategies.

2.2 RNA Structural Datasets

In the realm of RNA 3D structure based modeling infrastructure, three papers propose cleaned datasets with the objective of facilitating machine learning. *RNANet* (Becquey et al., 2021) proposes a dataset joining RNA structures with their corresponding sequence alignment. *RNAsolo* (Adamczyk et al., 2022) provides access to cleaned RNA files in various formats through a web interface. Finally, *RNA3DB* (Szikszai et al., 2024) offers a curated dataset specifically designed for RNA structure prediction models. None of these methods propose benchmark tasks to compare RNA modeling and learning methods. In addition, datasets with splits were independently proposed in several works, pertaining to small-molecules binding sites(Wang et al., 2018), inverse folding (Joshi et al., 2024) or virtual screening (Panei et al., 2022; Carvajal-Patino et al., 2023).

2.3 Structure Based RNA Models

While most deep learning models on RNA focus on secondary structure prediction and sequence-level tasks, some structure-based models have been developed. *RBind* (Wang et al., 2018) proposed learning on RNA structures using residue-graph representations, followed by others integrating sequence features in the learning (Su et al., 2021; Wang et al., 2023). *RNAmigos* (Oliver et al., 2020) proposed incorporating non-canonical interactions in the graph, in conjunction with metric learning pretraining. Finally, *gRNAde* (Joshi et al., 2024) adapted the popular *GVP* (Jing et al., 2021) protein encoder, which uses the atomic position in the message passing algorithm, to RNA.

3 Tools to assemble tasks

Here we introduce methods to ensure our benchmarking suite meets the quality required for biological relevance. We pay particular attention to careful data curation and rigorous splitting.

Data collection and annotation Our data originates from the PDB, where we fetch all RNA containing structures. As a less structurally redundant starting point, we alternatively also rely on the subset proposed in (Leontis & Zirbel, 2012), later referred to as `bgsu`. We annotate each system with RNA-level features such as its resolution, and residue-level features such as their coordinates, the presence of interacting compounds, and the amount of protein atoms in the residues' vicinities.

Given an RNA structure R, we represent it as a residue graph, whose nodes are the residues of R, and edges $\mathcal{E}$ are interactions between residues, such as backbone links. 2D structure, obtained by also using canonical base-pairing in $\mathcal{E}$, explains approximately 70% of the folding energy (Mathews et al., 2004). Using *RNAglib* (Mallet et al., 2022), we additionally compute the 2.5D graph representation (Leontis & Westhof, 2001) where non-canonical interactions are accounted for. We offer support for several graph learning frameworks: *NetworkX* (Hagberg et al., 2008), *DGL* (Wang, 2019) and *PyTorch Geometric* (Fey & Lenssen, 2019), as well as creating different representations (such as voxels or point-clouds) from this initial representation.

Structure partitioning and quality filters RNAs in the PDB exhibit a bimodal distribution in the number of residues (see Figures 3a and 3b). Many systems have less than 300 residues, while a few (mostly ribosomal structures) have several thousands. These systems can be split by chain, but also by connected components in the 2.5D graph. This allows independent RNA fragments appearing in the same PDB file to be treated as two systems; the resulting decrease in graph size leads to a reduction in needed compute for downstream steps, such as similarity based splitting.

We implement size and resolution filters, with default cutoff values including systems with a length of 15 to 500 residues and below 4Å resolution. We also provide a protein content filter which removes RNA structures that are heavily structured through protein interaction. This crucial filter has been overlooked in most of existing RNA structural datasets. In addition, we implement the drug-like filter introduced in *Hariboss* (Panei et al., 2022) for small molecules binding to RNA.

Comparing and clustering our data Because biological data points are often related in terms of evolutionary history and topological characteristics, insufficient rigor in splitting can lead to severe data leaks. Not surprisingly, numerous example of data leakage have pestered structural biology learning methods, one famous example being the first *PDBbind* dataset that was shown to incorporate several severe biases (Wang et al., 2005b; Volkov et al., 2022). Efforts like the *PINDER* database (Kovtun et al., 2024) have set a standard in the field for model generalization assessment.

In our work, we implement the computation of sequence-based or structure-based similarity matrices, with *CD-Hit* (Fu et al., 2012) and *US-Align* (Zhang et al., 2022a) respectively. Given a similarity matrix S, computed over a set of RNA molecules $\mathcal{R}$, and a threshold θ, we introduce the matrix S^{θ} defined as $S^{\theta}_{i,j} = \delta\{S_{i,j} \geq \theta\}$, where δ is the indicator function. We propose a clustering algorithm that considers $(\mathcal{R}, S^{\theta})$ as a graph, and returns connected components as clusters. This ensures that any pair of points in different clusters has a maximal similarity of θ.

Redundancy filtering and dataset splitting Starting with clusters, we propose a redundancy removal algorithm that selects the element with the highest resolution for each cluster. In most tasks, we first apply this algorithm at a sequence similarity level threshold of 0.9, then at a structure similarity threshold at 0.8 These stringent thresholds discard copies of systems with minimal variations that are common when the system occurs in multiple slightly different conditions.

In addition, we propose a splitting algorithm that also starts with a clustering, but uses a less conservative threshold (we use 0.5 unless mentioned otherwise). Then, we aggregate those clusters together to form splits of a certain size, optionally following a label balancing secondary objective, using the linear programming *PulP* (Mitchell et al., 2011). This ensures that no leakage happens between our different splits, while performing label stratification.

Task construction Finally, a task is defined as a dataset processed from our database using specific annotations, partitions and filters, with redundancy removing and fixed splits as well as a well-defined evaluation protocol using appropriate metrics. Each task follows this logic; this modular design will help practitioners propose additional benchmark tasks, and lowers the barrier for training models on RNA structure, as illustrated in Figure 2.

4 Tasks on RNA 3D Structure

We introduce a suite of seven tasks exploring various dimensions of RNA structural research, three of which are novel, with the remaining four based on previous research. Datasets and splits used by previous research are accessible within our unified suite where available. For two of these four tasks, we propose an enhanced version, resulting in a total of *nine available datasets*. The tasks are summarized in Table 1 and illustrated in Figure 1.

The tasks fall into three broad categories. Some tasks relate to the functional understanding of RNA, some tasks provide insights into ligand binding for drug discovery, and one task focuses on the design of RNA molecules, specifically inverse folding. We provide a succinct description of our tasks in the following, and a more detailed description in Supplementary Section B.

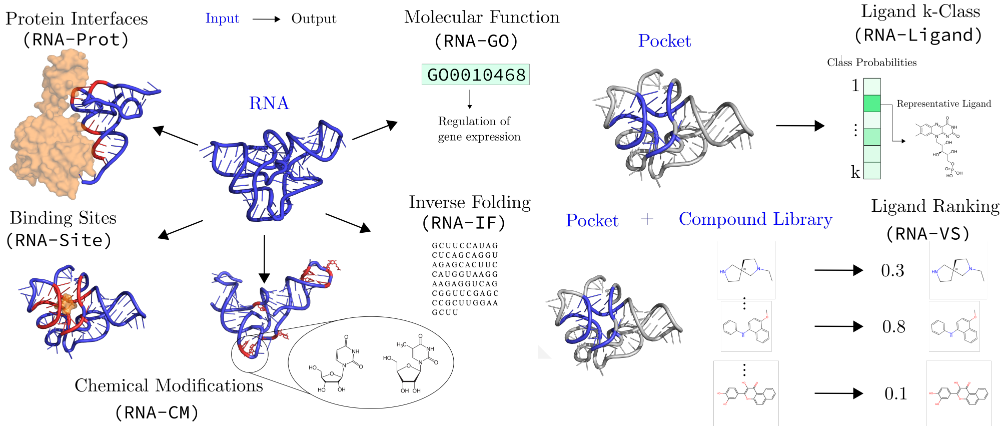


Figure 1: Graphical representation of the seven proposed tasks on RNA structure.

4.1 Function Prediction Tasks

RNA-GO maps structures to their molecular functions using Gene Ontology terms (Ashburner et al., 2000), leveraging manually annotated RNA families from Rfam (Griffiths-Jones et al., 2003; Ontiveros-Palacios et al., 2025). After extracting GO terms, we remove the two most frequent (ribosomal systems and tRNAs) and infrequent (less than 50 occurrences) labels. We group fully correlated GO-terms, resulting in 7 classes. Systems below 15 residues are discarded, yielding 501 systems, which are split on a sequence similarity cutoff following Gligorijević et al. (2021).

RNA-CM detects chemical modifications, which is crucial as over 170 modified nucleotides can alter RNA function (Boccaletto et al., 2022; Roundtree et al., 2017). We propose to detect chemical modifications from subtle perturbations of the canonical RNA backbone geometry. Training a predictor on RNA structures is not the most straightforward way of detecting chemical modifications, but may uncover function-modulating modifications that hitherto went unnoticed. We partition the whole dataset into connected components, apply size filters, and retain systems with modified residues based on PDB annotations. Our default redundancy removal and splitting strategies yield 185 data points.

RNA-Prot identifies protein interaction sites in RNA-Protein functional complexes, which are involved in crucial cellular processes, such as post-transcriptional control (Glisovic et al., 2008). RNA structure is often, but not always, heavily disrupted upon interaction with a protein. Starting from the bgsu non-redundant dataset, we partition into connected components, apply size and resolution

Table 1: Description of tasks, including learning type and dataset characteristics.

Task	Description	Level - Type	Split
1. RNA-GO	Classify RNAs into one or more frequent function.	RNA; Multi-class (7)	351-75-75
2. RNA-IF *gRNAde*	Find a sequence that folds into this structure.	Residue; Multi-class (4)	1615-607-478 13006-683-316
3. RNA-CM	Predict RNA residues with chemical modifications from geometry.	Residue; Binary	137-29-29
4. RNA-Prot *RBind*	Predict RNA residues interacting with a protein.	Residue; Binary	881-189-189 72
5. RNA-Site *RNASite*	Predict RNA residues interacting with a small-molecule protein.	Residue; Binary	320-68-70 53-6-17
6. RNA-Ligand	Classify binding sites by ligand type.	Pocket; Multi-class (3)	203-44-43
7. RNA-VS	Predict small molecule affinity for RNA binding sites (VS).	Pocket; Regression	304-34-65

filters, and remove ribosomal systems fully encapsulated in complexes. After retaining only RNA-protein interactions and applying default redundancy removal and splitting strategies, we obtain 1251 data points. Proteins are considered binding if residues are present within 8Å the RNA.

All function prediction tasks evaluate performance with accuracy and AuROC.

4.2 Molecular Design Task

RNA-IF aims to find a sequence that folds into a particular structure. A typical avenue for designing new macromolecules is to start from a coarse tertiary structure, such as a structural motif scaffold, and design a sequence that will fold into it. Finding this sequence is known as Inverse Folding (IF): mapping a known structure to an unknown sequence. IF is a well-established task in protein research with many recent breakthroughs (Watson et al., 2023). It was adapted to RNA with classical models such as Rosetta (Leman et al., 2020a), learning based approaches were pioneered by *gRNAde* (Leman et al., 2020a; Joshi et al., 2024), and others specifically address backbone generation (Anand et al., 2024) or protein-binding RNA (Nori & Jin, 2024).

To build this dataset, we gather all connected components between 15 and 300 residues and remove redundant sequences using a CD-Hit cutoff of 0.99. We cluster on structural similarity at a threshold of 0.9, then split ensuring a maximal similarity of 0.5. We also provide datasets and splits from *gRNAde* (Joshi et al., 2024), that were similarly obtained. Our dataset differ by including chains in interaction with each other, a stricter size cutoff and a duplicates filter, which reduces its size.

4.3 Drug Discovery Tasks

RNA is increasingly recognized as a promising family of targets for novel small molecule therapeutics (Falese et al., 2021; Haga & Phinney, 2023; Disney, 2019; Abulwerdi et al., 2019). Targeting RNA drastically increases the size of the druggable space, providing an alternative for overused protein targets in scenarios where they are insufficient. RNA are also a potential therapeutic avenue in pathologies with no protein targets, such as in triple-negative breast cancer (Xu et al., 2020).

RNA-Site detects drug binding locations by predicting whether residues are within 8Å of ligands. The classical structure-based drug discovery pipeline begins with identifying these binding sites, to then guide small molecule screening through techniques like molecular docking. This task was proposed as a machine learning problem in *Rbind* (Wang et al., 2018). We provide two versions of this dataset. First, we include the established *RNASite* (Su et al., 2021) dataset (76 systems with predefined splits) that has been used by several tools in the field including *RLBind* (Wang et al., 2023). This dataset was created through stringent clustering of an older version of the PDB.

Second, we provide a comprehensive dataset including recent structures. We preprocess as in *RNA-Prot* (partitioning into connected components, applying size and resolution filters) but retain ribosomal systems. We then add two filters: a drug-like filter on the small-molecules, and a protein-content filter that removes systems with $<$ 10 protein atom neighbors to ensure binding is modulated by RNA. After applying our default redundancy removal and splitting strategies, this yields 458 systems.

RNA-Ligand characterizes potential binders of a given binding site. On proteins, the *Masif-Ligand* tasks (Gainza et al., 2020) gathers all binding sites bound to the seven most frequent co-factors, and aims to classify them based on their partner. RNA-Ligand is built analogously, yet we only retain the three most frequent classes of binders in order to keep a sufficient amount of samples per class.

Specifically, we preprocess as in *RNA-Site* to obtain a set of RNA interacting with drug-like small molecules. We extract binding pockets through two rounds of breadth-first search starting from all residues within 8Å of any binder and cluster them on sequence. To identify the most relevant ligand classes, we collect single-ligand clusters, clusters of binding sites that interact with only one type of ligand. We discard all binding events not involving the three most common ligands within these clusters. Then, we split based on structural similarity and evaluate using MCC and AuROC.

RNA-VS ranks potential binders by binding probability, a task coined Virtual Screening. This task is ubiquitous in drug design as it helps select the most promising compounds for further wet-lab assays. The dataset is reproduced from *RNAmigos2* (Carvajal-Patino et al., 2023). Its authors curated a list of binding sites similarly to *RNA-Site* and clustered them using RMScore (Zheng et al., 2019). All binders found for each cluster are retained as positive examples, and a set of drug-like chemical decoys are added as negative partners. Docking scores are computed with rDock (Ruiz-Carmona et al., 2014) on all binding site-small molecule pairs and evaluated with AuROC.

5 Usage

Next, we briefly showcase the use of our framework for a simple access to proposed tasks. In Figure 2, we show how practitioners can access our datasets from Python code, automatically downloading them from Zenodo, choosing a representation (in this example a *Pytorch Geometric* graph) and directly accessing the data in a simple and reproducible fashion.

```python
from rnaglib.tasks import BindingSiteDetection
from rnaglib.representations import GraphRepresentation

task = BindingSiteDetection(root='example', precomputed=True)
task.dataset.add_representation(GraphRepresentation(framework="pyg"))
task.get_split_loaders()

for batch in task.train_dataloader:
    graph = batch["graph"]
    ...
```

Figure 2: Obtaining a machine learning-ready split dataset only requires a few lines of code.

As an example for our anticipated usage of our framework, we studied the impact of using structural information, of using RNA language model as features, and of applying different splitting strategies on the performance of models for two proposed tasks. We find that structure information facilitates learning on RNA, but not the inclusion of RNA language model embeddings. We also find a very moderate impact of using structural splits. The plots and detailed results are found in Supplementary Section C. In addition, we provide baseline results on all seven tasks and contrast simple RGCN implementations using *rnaglib* with published methods. Though our results are preliminary (introduced here to showcase the kind of studies our tool could spark) we believe they highlight significant differences to results obtained in protein modeling and open interesting questions for the field of RNA modeling.

Despite having already introduced seven tasks tying RNA structure and function, due to the rapid advances of the field, we can expect that additional interesting challenges will arise in the near future. Thanks to the modularity of our tool, additional tasks on RNA can be easily integrated in our framework for future releases. This is illustrated in Supplementary Section 8.

6 Discussion

We have introduced a versatile and modular library designed to advance RNA structural analysis. By providing a suite of diverse and well-defined datasets and splits, our library enables robust benchmarking and facilitates the development of new computational models. It also ensures reproducibility and promotes standardized evaluation, fostering confidence in computational findings.

In the future, the development of additional tasks, such as the assessment of various structural models and a greater focus on RNA embeddings, presents exciting opportunities. As the understanding of the dynamic nature of biological macromolecules evolves, so may the preferred representation for RNA. Our tool's built-in 2.5D-graph representations offer extensive adaptability and have been used to develop recent competitive RNA structure-based models (Wang et al., 2024). Nevertheless, the library can easily be extended to incorporate new or improved representations as they emerge.

The number of experimentally solved RNA structures is expected to remain dwarfed by that of protein structures in the foreseeable future. Thus, whether deep learning-based insights into RNA structure-function relationships can match those achieved for proteins will depend on whether it can be shown that large datasets are not required for informative RNA structure-function models. Our library's task module addresses the challenge of limited RNA structural data by enabling the most effective use of available data and can thereby accelerate further advances in RNA structural biology.

References

Fardokht A Abulwerdi, Wenbo Xu, Abeer A Ageeli, Michael J Yonkunas, Gayatri Arun, Hyeyeon Nam, John S Schneekloth Jr, Theodore Kwaku Dayie, David Spector, Nathan Baird, et al. Selective small-molecule targeting of a triple helix encoded by the long noncoding rna, malat1. *ACS chemical biology*, 14(2):223–235, 2019.

Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. Rnasolo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 38(14):3668–3670, 2022.

Tanvir Alam, Mahmut Uludag, Magbubah Essack, Adil Salhi, Haitham Ashoor, John B Hanks, Craig Kapfer, Katsuhiko Mineta, Takashi Gojobori, and Vladimir B Bajic. Farna: knowledgebase of inferred functions of non-coding rna transcripts. *Nucleic acids research*, 45(5):2838–2848, 2017.

Rishabh Anand, Chaitanya K Joshi, Alex Morehead, Arian R Jamasb, Charles Harris, Simon V Mathis, Kieran Didi, Bryan Hooi, and Pietro Liò. Rna-frameflow: Flow matching for de novo 3d rna backbone design. *arXiv preprint arXiv:2406.13839*, 2024.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Louis Becquey, Eric Angel, and Fariza Tahi. Rnanet: an automatically built dual-source dataset integrating homologous sequences and rna structures. *Bioinformatics*, 37(9):1218–1224, 2021.

Pietro Boccaletto, Filip Stefaniak, Angana Ray, Andrea Cappannini, Sunandan Mukherjee, Elżbieta Purta, Małgorzata Kurkowska, Niloofar Shirvanizadeh, Eliana Destefanis, Paula Groza, et al. Modomics: a database of rna modification pathways. 2021 update. *Nucleic acids research*, 50 (D1):D231–D235, 2022.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Juan G Carvajal-Patino, Vincent Mallet, David Becerra, L Fernando Nino, Carlos Oliver, and Jerome Waldispuhl. Semi-supervised learning and large-scale docking data accelerate rna virtual screening. *bioRxiv*, pp. 2023–11, 2023.

Thomas R Cech and Joan A Steitz. The noncoding rna revolution—trashing old rules to forge new ones. *Cell*, 157(1):77–94, 2014.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Matthew D Disney. Targeting rna with small molecules to capture opportunities at the intersection of chemistry, biology, and medicine. *Journal of the American Chemical Society*, 141(17):6776–6790, 2019.

Janani Durairaj, Yusuf Adeshina, Zhonglin Cao, Xuejin Zhang, Vladas Oleinikovas, Thomas Duignan, Zachary McClure, Xavier Robin, Daniel Kovtun, Emanuele Rossi, et al. Plinder: The protein-ligand interactions dataset and evaluation resource. *bioRxiv*, pp. 2024–07, 2024.

James P Falese, Anita Donlic, and Amanda E Hargrove. Targeting rna with small molecules: from fundamental principles towards the clinic. *Chemical Society Reviews*, 50(4):2224–2243, 2021.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Tina Glisovic, Jennifer L Bachorik, Jeongsik Yong, and Gideon Dreyfuss. Rna-binding proteins and post-transcriptional gene regulation. *FEBS letters*, 582(14):1977–1986, 2008.

Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, and Sean R Eddy. Rfam: an rna family database. *Nucleic acids research*, 31(1):439–441, 2003.

Christopher L Haga and Donald G Phinney. Strategies for targeting rna with small molecule drugs. *Expert Opinion on Drug Discovery*, 18(2):135–147, 2023.

Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

Arian R Jamasb, Alex Morehead, Chaitanya K Joshi, Zuobai Zhang, Kieran Didi, Simon Mathis, Charles Harris, Jian Tang, Jianlin Cheng, Pietro Liò, et al. Evaluating representation learning on the protein structure universe. *ArXiv*, pp. arXiv–2406, 2024.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Equivariant graph neural networks for 3d macromolecular structure, 2021. URL `https://arxiv.org/abs/2106.03843`.

Chaitanya K Joshi, Arian R Jamasb, Ramon Viñas, Charles Harris, Simon V Mathis, Alex Morehead, and Pietro Liò. grnade: Geometric deep learning for 3d rna inverse design. *bioRxiv*, pp. 2024–03, 2024.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Daniel Kovtun, Mehmet Akdel, Alexander Goncearenco, Guoqing Zhou, Graham Holt, David Baugher, Dejun Lin, Yusuf Adeshina, Thomas Castiglione, Xiaoyun Wang, et al. Pinder: The protein interaction dataset and evaluation resource. *bioRxiv*, pp. 2024–07, 2024.

Tim Kucera, Carlos Oliver, Dexiong Chen, and Karsten Borgwardt. Proteinshake: Building datasets and benchmarks for deep learning on protein structures. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58277–58289, 2023.

Julia Koehler Leman, Brian D Weitzner, Steven M Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F Alford, Melanie Aprahamian, David Baker, Kyle A Barlow, Patrick Barth, et al. Macromolecular modeling and design in rosetta: recent methods and frameworks. *Nature methods*, 17 (7):665–680, 2020a.

Julia Koehler Leman, Brian D Weitzner, Steven M Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F Alford, Melanie Aprahamian, David Baker, Kyle A Barlow, Patrick Barth, et al. Macromolecular modeling and design in rosetta: recent methods and frameworks. *Nature methods*, 17 (7):665–680, 2020b.

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

Neocles B Leontis and Craig L Zirbel. Nonredundant 3d structure datasets for rna knowledge extraction and benchmarking. *RNA 3D structure analysis and prediction*, pp. 281–298, 2012.

Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6:1–14, 2011.

Vincent Mallet, Carlos Oliver, Jonathan Broadbent, William L Hamilton, and Jérôme Waldispühl. Rnaglib: a python package for rna 2.5 d graphs. *Bioinformatics*, 38(5):1458–1459, 2022.

David H Mathews, Matthew D Disney, Jessica L Childs, Susan J Schroeder, Michael Zuker, and Douglas H Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of rna secondary structure. *Proceedings of the National Academy of Sciences*, 101(19):7287–7292, 2004.

Stuart Mitchell, Michael OSullivan, and Iain Dunning. Pulp: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand*, 65:25, 2011.

Divya Nori and Wengong Jin. Rnaflow: Rna structure and sequence design via inverse folding-based flow matching, 2024. URL `https://arxiv.org/abs/2405.18768`.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379, 2023.

Carlos Oliver, Vincent Mallet, Roman Sarrazin Gendron, Vladimir Reinharz, William L Hamilton, Nicolas Moitessier, and Jérôme Waldispühl. Augmented base pairing networks encode rna-small molecule binding preferences. *Nucleic acids research*, 48(14):7690–7699, 2020.

Nancy Ontiveros-Palacios, Emma Cooke, Eric P Nawrocki, Sandra Triebel, Manja Marz, Elena Rivas, Sam Griffiths-Jones, Anton I Petrov, Alex Bateman, and Blake Sweeney. Rfam 15: Rna families database in 2025. *Nucleic Acids Research*, 53(D1):D258–D267, 2025.

Francesco P Panei, Rachel Torchet, Herve Menager, Paraskevi Gkeka, and Massimiliano Bonomi. Hariboss: a curated database of rna-small molecules structures to aid rational drug design. *Bioinformatics*, 38(17):4185–4193, 2022.

Ian A Roundtree, Molly E Evans, Tao Pan, and Chuan He. Dynamic rna modifications in gene expression regulation. *Cell*, 169(7):1187–1200, 2017.

Sergio Ruiz-Carmona, Daniel Alvarez-Garcia, Nicolas Foloppe, A. Beatriz Garmendia-Doval, Szilveszter Juhos, Peter Schmidtke, Xavier Barril, Roderick E. Hubbard, and S. David Morley. rdock: A fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Computational Biology*, 10:1–8, 2014. ISSN 15537358. doi: 10.1371/journal.pcbi.1003571.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pp. 593–607. Springer, 2018.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

Luisa Statello, Chun-Jie Guo, Ling-Ling Chen, and Maite Huarte. Gene regulation by long non-coding rnas and its biological functions. *Nature reviews Molecular cell biology*, 22(2):96–118, 2021.

Hong Su, Zhenling Peng, and Jianyi Yang. Recognition of small molecule–rna binding sites using rna sequence and structure. *Bioinformatics*, 37(1):36–42, 2021.

Marcell Szikszai, Marcin Magnus, Siddhant Sanghi, Sachin Kadyan, Nazim Bouatta, and Elena Rivas. Rna3db: A structurally-dissimilar dataset split for training and benchmarking deep learning models for rna structure prediction. *Journal of Molecular Biology*, pp. 168552, 2024. ISSN 0022-2836. doi: https://doi.org/10.1016/j.jmb.2024.168552.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Bozhen Hu, Siyuan Li, Zicheng Liu, and Stan Z Li. Rdesign: hierarchical data-efficient representation learning for tertiary structure-based rna design. *arXiv preprint arXiv:2301.10774*, 2023.

Raphael Townshend, Martin Vögele, Patricia Suriana, Alex Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, Risi Kondor, Russ Altman, and Ron Dror. Atom3d: Tasks on molecules in three dimensions. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks*, volume 1, 2021.

Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pp. 2022–02, 2022.

Mikhail Volkov, Joseph-André Turk, Nicolas Drizard, Nicolas Martin, Brice Hoffmann, Yann Gaston-Mathé, and Didier Rognan. On the frustration to predict binding affinities from protein–ligand structures with deep neural networks. *Journal of medicinal chemistry*, 65(11):7946–7958, 2022.

Junkai Wang, Lijun Quan, Zhi Jin, Hongjie Wu, Xuhao Ma, Xuejiao Wang, Jingxin Xie, Deng Pan, Taoning Chen, Tingfang Wu, et al. Multimodrlbp: A deep learning approach for multi-modal rna-small molecule ligand binding sites prediction. *IEEE Journal of Biomedical and Health Informatics*, 2024.

Kaili Wang, Yiren Jian, Huiwen Wang, Chen Zeng, and Yunjie Zhao. Rbind: computational network method to predict rna binding sites. *Bioinformatics*, 34(18):3131–3136, 2018.

Kaili Wang, Renyi Zhou, Yifan Wu, and Min Li. Rlbind: a deep learning method to predict rna–ligand binding sites. *Briefings in Bioinformatics*, 24(1):bbac486, 2023.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3d graph networks, 2022. URL https://arxiv.org/abs/2207.12600.

Minjie Yu Wang. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR workshop on representation learning on graphs and manifolds*, 2019.

Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005a.

Renxiao Wang, ueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: Methodologies and updates. *Journal of Medicinal Chemistry*, 22, 11 2005b. ISSN 4111–4119. doi: 10.1021/jm048957q.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Fang Wu, Lirong Wu, Dragomir Radev, Jinbo Xu, and Stan Z Li. Integration of pre-trained protein language models into geometric deep learning networks. *Communications Biology*, 6(1):876, 2023.

Juan Xu, Kang-jing Wu, Qiao-jun Jia, and Xian-feng Ding. Roles of mirna and lncrna in triple-negative breast cancer. *Journal of Zhejiang University-science b*, 21(9):673–689, 2020.

Pan Zeng and Qinghua Cui. Rsite2: an efficient computational method to predict the functional sites of noncoding rnas. *Scientific Reports*, 6(1):19016, 2016.

Pan Zeng, Jianwei Li, Wei Ma, and Qinghua Cui. Rsite: a computational method to identify the functional sites of noncoding rnas. *Scientific Reports*, 5(1):9179, 2015.

Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. Us-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature methods*, 19(9): 1109–1115, 2022a.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022b.

Zuobai Zhang, Chuanrui Wang, Minghao Xu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. A systematic study of joint representation learning on protein sequences and structures, 2023. URL https://arxiv.org/abs/2303.06275.

Jinfang Zheng, Juan Xie, Xu Hong, and Shiyong Liu. Rmalign: an rna structural alignment tool based on a novel scoring function rmscore. *BMC genomics*, 20:1–10, 2019.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

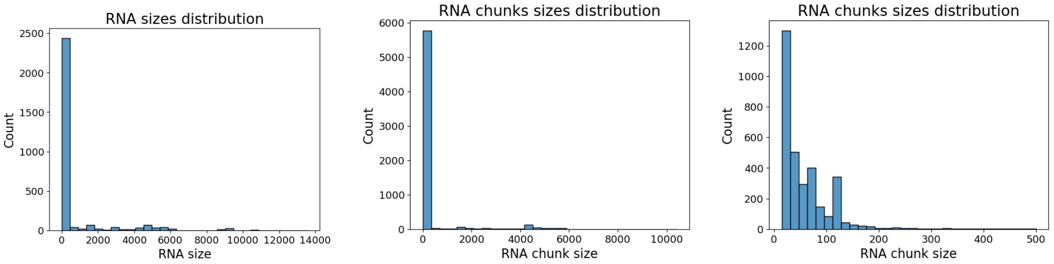
Appendix

A Data and tools to build a task

In this section, we provide additional insights about the tools used to construct our tasks.

As mentioned in section3, RNA molecules present in the PDB files exhibit a bimodal distribution over the number of residues. See distribution of the RNA sizes (defined as the number of residues) of the RNAs from the PDB in Figure 3a. In order to deal with a more biologically relevant scale, we partitioned the raw RNAs from PDB files into connected components in order to use these connected components as samples in our different machine learning tasks, resulting in the size distribution shown in in Figure 3b.

RNA chunks that are too small might not be suited for structure-based machine learning, ones that are too big impact the computational performance of the data loading as well as tasks. Therefore, we filter RNA chunks and only keep those which have between 15 and 300 nucleotides, resulting in the distribution shown in Figure 3c.



(a) Distribution of RNA sizes

(b) Distribution of chunk sizes

(c) Distribution of chunk sizes after filtering

In order to remove redundancy between RNAs or binding pockets too close one from another and perform splitting (ensure that similar binding pockets are both in the train, val or test split set and not in different sets to prevent overfitting), we used two types of similarity metrics: a sequence-based similarity metric on relying on *CD-Hit* (Fu et al., 2012) and a structure-based one relying on USAlign.

The redundancy removal process we applied is the following. We first performed sequence-based clustering based on a similarity threshold above which RNA fragments are clustered together. Then, among each cluster, we select only one sample (by choosing the one having the highest resolution among the cluster). Then we performed structure-based clustering and structure-based redundancy removal following the same procedure. Afterwards, when instantiating the splitters, we perform structure-based clustering with a different threshold to define the clusters which will be required to be grouped either in train, val or test set. In order to calibrate the similarity thresholds to use to perform redundancy removal then splitting, we studied, for each task, the number of clusters generated based on the similarity threshold used for redundancy removal. Since we only select one representative sample per cluster, the number of clusters prefigurates the number of samples we will finally have. We therefore have a tradeoff to do between having a large amount of data and discarding redundant RNAs. Below are the results obtained for RNA-CM task.

The first plot displays the number of clusters obtained with and without removing redundancy based on sequence (and, in the latter, using different redundancy removal thresholds). Here, the thresholds 0.90, 0.80 and 0.70 give the same plot since CD-Hit similarity values are strongly concentrated around 0.5 and 1. We finally chose 0.90.

The plot below represents the number of clusters obtained with and without removing redundancy based on structure (and, in the latter, using different redundancy removal thresholds) once a first sequence-based redundancy removal has been performed using a 0.90 threshold.

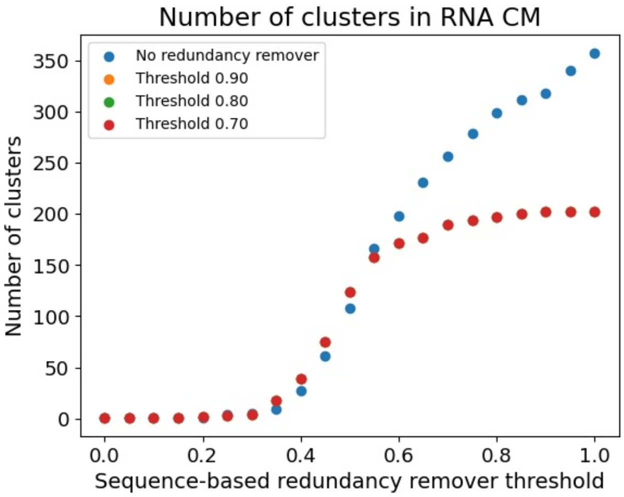


Figure 4: Number of clusters based on the threshold of the sequence-based redundancy remover

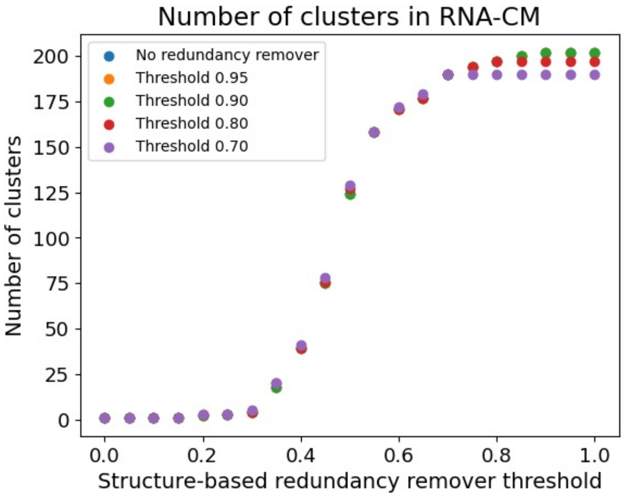


Figure 5: Number of clusters based on the threshold of the structure-based redundancy remover

B DETAILED EXPLANATION OF THE PROPOSED TASKS

Out of space issues, we could not detail the construction and context of our tasks, and propose to do it in the following section.

B.1 RNA-GO: FUNCTION TAGGING

(Definition): This is a multi-label classification task where input RNA R is mapped to a sequence of labels $\mathbf{y} \in \{..\}$ representing the molecular functions of R.

(Context): Effective function prediction models have the capacity to uncover new structure-function connections. The Gene Ontology (GO) (Ashburner et al., 2000) was developed to associate a function to a gene, and thus, indirectly to the RNA or protein it encodes. It resulted in discrete functional categories called *GO terms*. Predicting the GO term from protein structure was proposed as a task in (Gligorijević et al., 2021) and has been regularly used since as a benchmark task. A mapping between RNA sequences and GO terms is available in Rfam (Griffiths-Jones et al., 2003; Ontiveros-Palacios et al., 2025) a database of non-coding RNA families which are manually annotated with GO terms.

(Processing): We extracted all GO terms corresponding to an RNA structure. Then, we filtered these terms based on their frequency, removing the two most frequent labels (ribosomal systems and tRNAs), as well as infrequent ones (less than 50 occurrences). We then grouped together GO-terms that were fully correlated, which resulted in 7 classes. We subset the RNA graph on the RFAM labeled part. Finally, we discarded RNA fragments with less than 15 residues, resulting in 501 systems. We split these systems based on a sequence similarity cutoff, following Gligorijević et al. (2021). Accuracy and AuROC are the task's evaluation metrics.

B.2 RNA-IF: MOLECULAR DESIGN

(Definition): This is a residue-level, classification task where given an RNA structure $R = (\omega, x)$, we mask the residue identity of x (by keeping only the coordinates of the backbone) and aim to learn the mapping $x \rightarrow \omega$.

(Context): One avenue for designing new macromolecules is to first design a coarse grained tertiary structure, for instance scaffolding a structural motif, and then design a sequence that will fold into this given structure. This second step is denoted as Inverse Folding (IF), since it maps a known structure to an unknown sequence. Inverse folding is a well-established task in protein research with many recent breakthroughs (Watson et al., 2023). It has recently been adapted for RNA with classical models such as Rosetta (Leman et al., 2020a). Learning based approaches were pioneered by *gRNAde* (Leman et al., 2020a; Joshi et al., 2024), other papers specifically address backbone generation (Anand et al., 2024) and protein-binding RNA design (Nori & Jin, 2024).

(Processing): To build this dataset, we gather all connected components in our dataset with a size between 15 and 300. Then, we removed identical sequences using a redundancy removal step with CD-Hit with a cutoff of 0.99. Then, we cluster the data on structural similarity with a threshold of 0.9, and split these while ensuring a maximal similarity of 0.5. We also provide datasets and splits from *gRNAde* (Joshi et al., 2024), that were obtained in a similar manner. The key differences are that our dataset include IF for several chains in interaction with each other, a stricter size cutoff and a duplicates filter, leading to a reduced dataset size.

B.3 RNA-CM: DYNAMIC FUNCTION MODULATION

(Definition): This is a residue-level, binary classification task where given an RNA structure R, we aim to predict which, if any, of its residues are chemically modified.

(Context): In addition to the four canonical nucleotides, more than 170 modified nucleotides were found to be integrated in RNA polymers (Boccaletto et al., 2022). Multiple functions of a diverse range of ncRNA have been shown to be directly dependent on such chemical modifications (Roundtree et al., 2017). We propose to detect such chemical modifications from subtle perturbations of the canonical geometry of the RNA backbone. While obtaining the modified RNA structure is not the easiest way to detect such modifications after investigation, it is possible that certain modifications go unnoticed, yet alter the function of an RNA of interest.

(Processing): To build this task, we start from the whole dataset, partition it into connected components and apply our size filter. Then, we filter for systems with modified residues, relying on PDB annotations that flag such modified residues. We apply our default redundancy removal and splitting strategies, which results in 185 data points. Performance is evaluated with accuracy and AuROC.

B.4 RNA-PROT: BIOLOGICAL COMPLEX MODELING

(Definition): This is a residue-level, binary classification task where given an RNA structure R, we aim to predict whether a protein residue is closer than 8Å to any of its residues.

(Context): RNAs and Protein often bind to form a functional complex. Such complexes are involved in crucial cell processes, such as post-transcriptional control of RNAs (Glisovic et al., 2008). RNA structure is often, but not always, heavily disrupted upon interaction with a protein. We expect this task to be an easier version of the RNA-CM task.

(Processing): To build this task, we start from the `bgsu` non-redundant dataset, partition it in connected components and apply our size and resolution filters. We remove systems originating from ribosomes that are fully encapsulated in a complex. Then, we retain only RNA interacting with a protein and apply our default redundancy removal and splitting strategies, resulting in 1251 data points. Performance is evaluated with accuracy and AuROC.

B.5 RNA-SITE: DRUG TARGET DETECTION

(Definition): This is a residue-level, binary classification task where given an RNA structure R, we aim to predict whether a ligand is closer than 8Å to any of its residues.

(Context): The classical flow of a structure-based drug discovery approach starts with the identification of relevant binding sites, which are subparts of the structure likely to interact with ligands, or of particular interest for a phenotypical effect. The structure of the binding site can then be used to condition the quest for small molecule binders, for instance using molecular docking (Ruiz-Carmona et al., 2014). The framing of this problem as a machine learning task for RNA was introduced in *Rbind* (Wang et al., 2018).

(Processing): We provide datasets and predefined splits from *RNASite* (Su et al., 2021), which are also utilized by other tools in the field such as *RLBind* (Wang et al., 2023). This dataset contains 76 systems obtained after applying stringent clustering on an older version of the PDB.

Beyond providing the dataset existing in the literature, we propose a larger, up-to-date dataset. We start by following similar steps as for RNA-Prot (without removal of ribosomal systems). We then include a drug-like filter on the small-molecule side. Finally, we include a protein-content filter, removing systems with more than 10 protein atom neighbors, ensuring that the binding is modulated by RNA only. We use the default redundancy removal and splitting, resulting in 458 systems.

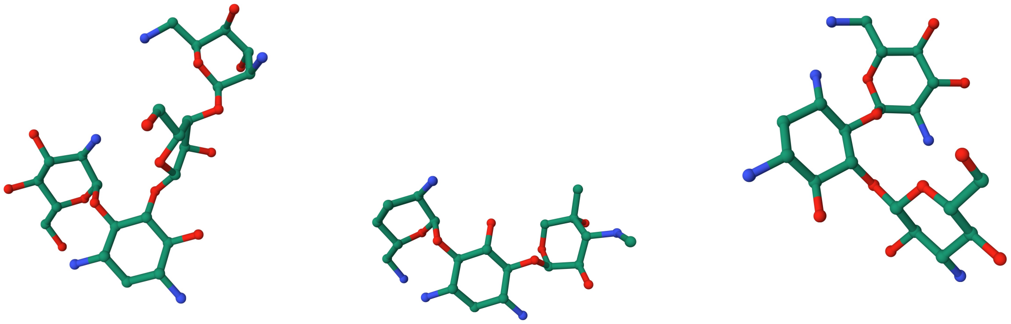
B.6 RNA-LIGAND: POCKET CATEGORIZATION

(Definition): This is a binding site-level, multi-class classification task where the structure of an RNA binding site is classified according to the partner it accommodates.

(Context): Equipped with a binding site, one wants to use its structure to characterize its potential binders. On proteins, the *Masif-Ligand* tasks (Gainza et al., 2020) gathers all binding sites bound to the seven most frequent co-factors, and aims to classify them based on their partner. Inspired by this work we propose the *RNA-Ligand* task. To keep sufficiently many examples per task, we only retained the three most frequent classes. This task is less ambitious than training a molecular docking surrogate and can help understanding the potential modulators of a given binding site.

(Processing): Starting with similar steps as the RNA-Site, we obtain a set of structures that display RNAs in interaction with one or more drug-like small molecules. We then proceed to extracting the context of the binding pocket by seeding two rounds of breadth-first search with all residues closer than 8Å to an atom of the binder. This results in all binding pockets in our database and we group these pockets with a sequence clustering.

Then, our aim is to find the most frequent ligands binding in non-redundant pockets. To that end, we first gather a dataset of binding site clusters that bind to only one ligand. We compute the most frequent ligands among this dataset and retain only the top three: paromomycin (called PAR in the PDB nomenclature), gentamycin C1A (LLL) and aminoglycoside TC007 (8UZ). Their structure are displayed in Figure 6. We then discarded all binding events to other small molecules, and added clusters that bind only one of the top three to our dataset. This dataset is split on structural similarity and evaluated on MCC and AuROC.



(a) Paromycin (PAR)

(b) Gentamycin C1A (LLL)

(c) Aminoglycoside TC007 (8UZ)

Figure 6: Structures of the ligands selected for RNA-Ligand task

B.7 RNA-VS: Drug screening

(Definition): This is a binding site-level regression task. Given the structure of an RNA binding site and a small molecule in graph representation, the goal is to predict the affinity of their binding.

(Context): Beyond classification into fixed categories, virtual screening aims to score compounds based on their affinity to a binding pocket. This task is ubiquitous in drug design as it helps selecting the most promising compounds to be further assayed in the wet-lab. Our last task implements a virtual screening task as introduced in (Carvajal-Patino et al., 2023). Their model is trained to approximate normalized molecular docking scores. Trained models can then be used to rank compounds by their binding likelihood to target sites, hence achieving virtual screening.

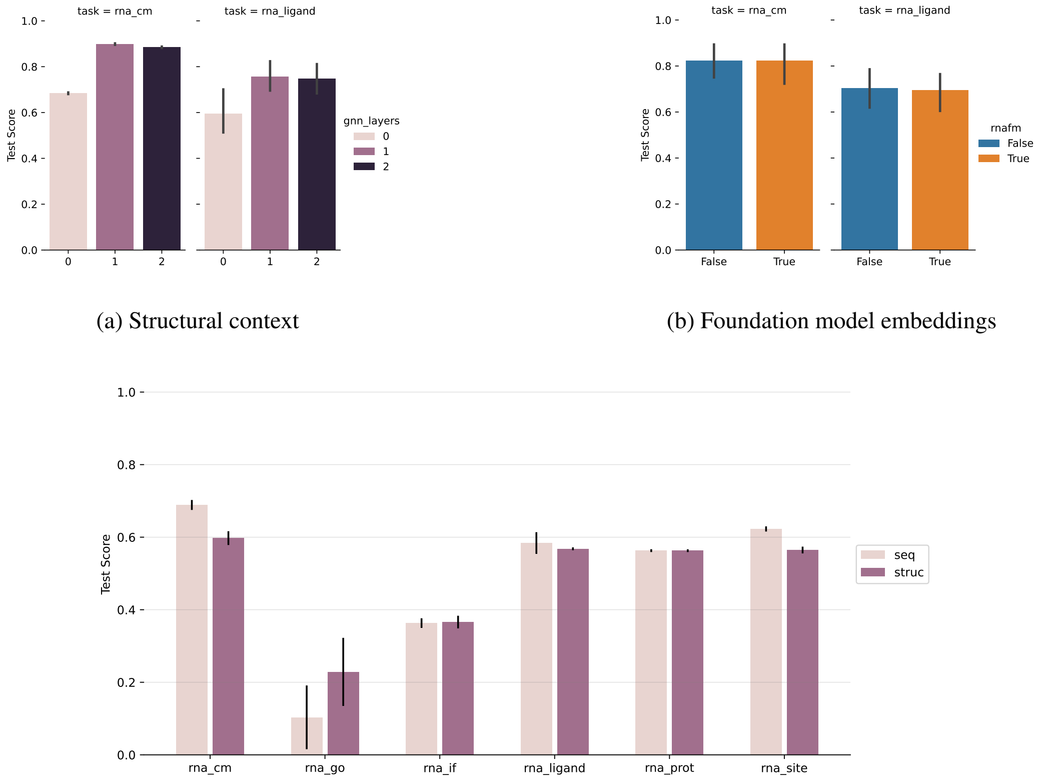
(Processing): The dataset is reproduced from their paper. Authors curated a list of binding sites in a similar fashion to the RNA-Site task and clustered them using RMScore (Zheng et al., 2019). All binders found for each cluster are retained as positive examples, and a set of drug-like chemical decoys are added as negative partners. Molecular docking scores are computed with rDock (Ruiz-Carmona et al., 2014) on all binding site-small molecule pairs and evaluated with AuROC.

C Experiments

We hereby explicitly state that we limit the scope of this work to the creation of a useful deep learning library for RNA 3D structure based modeling. We refrain from suggesting novel model architecture; models discussed in this work uniquely serve the goal of illustrating the practical applicability of our library. Within these limitations, we provide experiments showcasing the utility of the available features.

We quickly investigate three questions pertaining to learning on RNA structures. Does including structural information improve performance on node level prediction tasks? Can including RNA language model node embeddings increase prediction performance? Does splitting according to structural and not just sequence similarity affect performance on our tool's datasets?

To answer these three questions, we use the 2.5D (graph) representation of RNA structure, along with a classical RGCN (Schlichtkrull et al., 2018) model throughout our experiments. We emphasize that we have not optimized our results for performance. Due to space constraints, we provide results for two tasks, *RNA-CM* and *RNA-Ligand* in Figure 7.



(a) Structural context

(b) Foundation model embeddings

(c) Sequence/structure splitting

Figure 7: Benchmark on RNA-CM (Chemical Modification) and RNA-Ligand (Ligand classification) tasks, or all tasks for splitting evaluation. Variance reported across three seeds.

Structural modeling We compute evaluation metrics on our tasks when using models with increasingly large number of layers. A model with zero layers is basically using only the sequence information, while a wider support incorporates its geometric features. As expected, we found that using structure significantly improves the performance of our model on both tasks.

Foundation model embeddings The inclusion of protein language model embeddings have been reported to enhance structural model performances for proteins (Wu et al., 2023; Zhang et al., 2023). We tested this hypothesis for learning on RNA and included *rna-fm* (Chen et al., 2022) foundation model embeddings to our experiments. Surprisingly, we do not observe any significant difference between training a model on data that includes *rna-fm* embeddings at the node level and data that includes nucleotide information only. This might stem from the significant increase in data dimensionality (4 to 644) for which the model lacks data to learn on.

Sequence vs. structure splitting We find minimal effects when splitting the tasks in terms of sequence vs structural similarity (Fig. 7c). While this is an initial result, the contrast to well-reported effects in proteins of structure-based splitting (Kucera et al., 2023; Townshend et al., 2021) could be evidence of qualitatively different sources of signal in the two data types.

In summary, we find that the use of structure helps learning on RNA, but not the inclusion of RNA language model embeddings. We also find a very moderate impact of using structural splits. Even though those results are preliminary ones, primarily introduced here to showcase the kind of studies our tool could spark, we believe they highlight significant differences to the ones obtained in the context of protein modeling and open many interesting questions for the field of RNA modeling. Beyond these results, we provide a simple benchmark of all our seven tasks and a comparison to exisiting published models in tables 2, 3, and 4.

Table 2: Test performance metrics for various RNA-related tasks.

Task	Test F1-Score	Test AUC	Test Global Balanced Accuracy	Test MCC	Test Jaccard
RNA_Ligand	0.2771	0.6751	0.4678		
RNA_CM	0.1957	0.7393	0.6615	0.1695	
RNA_Site	0.3346	0.5929	0.6309	0.3098	
RNA_Prot	0.4545	0.6654	0.6254	0.2469	
RNA_IF	0.3326	0.6201	0.3523*	0.1319	
RNA_VS		0.855			
RNA_GO	0.4074	0.8406	0.7067		0.3167

Hyperparameters used:
RNA_Ligand: n_layers=4, hidden_dim=128, lr=0.00001, dropout=0.5
RNA_CM: n_layers=3, hidden_dim=128, lr=0.001, dropout=0.5
RNA_Site: n_layers=4, hidden_dim=256, lr=0.001, dropout=0.5
RNA_Prot: n_layers=4, hidden_dim=64, lr=0.01, dropout=0.2
RNA_IF: n_layers=3, hidden_dim=128, lr=0.0001, dropout=0.5
RNA_VS: n_layers=3, hidden_dim=64/32, lr=0.001, dropout=0.2
RNA_GO: n_layers=3, hidden_dim=64, lr=0.001, dropout=0.5

* *For RNA_IF, the reported value in "Global Balanced Accuracy" is sequence recovery.*

Table 3: We compare a standard RGCN using the *rnaglib*'s task module with various published results using the TR60/TE18 split. *Note:* Binding site definitions may vary slightly between models.

Methods	MCC	AUC
Rsite2 (Zeng & Cui, 2016)	0.010	0.474
Rsite (Zeng et al., 2015)	0.055	0.496
RBind (Wang et al., 2018)	0.141	0.540
RNAsite_seq (Su et al., 2021)	0.160	0.641
RNAsite_str (Su et al., 2021)	0.185	0.695
RNAsite (Su et al., 2021)	0.186	0.703
rnaglib RNA-Site	0.113	0.606

Table 4: Sequence recovery scores for RNA inverse folding models. We use a standard two layer RGCN part of *rnaglib*'s task module on the dataset and split published by Joshi et al. (2024)

Method	Sequence Recovery
gRNAde (Joshi et al., 2024)	0.568
Rosetta (Leman et al., 2020b)	0.450
RDesign (Tan et al., 2023)	0.430
FARNA (Alam et al., 2017)	0.321
ViennaRNA (Lorenz et al., 2011)	0.269
rnaglib RNA-IF	0.410

D INTRODUCING NEW TASKS

Below, we provide the code necessary to introduce a task, by providing the implementation of **RNA-CM**.

```
# imports
class ChemicalModification(ResidueClassificationTask):
    """Residue-level binary classification task to predict whether
    a given residue is chemically modified.
    """

    input_var = "nt_code"
    target_var = "is_modified"
    name = "rna_cm"

    def __init__(self,
                 root,
                 size_thresholds=(15, 500),
                 **kwargs):
        super().__init__(root=root, size_thresholds=size_thresholds,
            **kwargs)

    @property
    def default_splitter(self):
        return ClusterSplitter(distance_name="USalign")

    def get_task_vars(self):
        return FeaturesComputer(nt_targets=self.target_var,
            nt_features=self.input_var)

    def process(self):
        # Define your transforms
        residue_attribute_filter = ResidueAttributeFilter(
            attribute=self.target_var,
            value_checker=lambda val: val == True)
        connected_components_partition = ConnectedComponentPartition()

        # Run through database, applying our filters
        dataset = RNADataset(debug=self.debug, in_memory=self.in_memory)
        all_rnas = []
        os.makedirs(self.dataset_path, exist_ok=True)
        for rna in tqdm(dataset):
            for rna_cc in connected_components_partition(rna):
                if residue_attribute_filter.forward(rna_cc):
                    if (self.size_thresholds is not None
                        and not self.size_filter.forward(rna_cc)):
                        continue
                    rna = rna_cc["rna"]
                    self.add_rna_to_building_list(all_rnas=all_rnas,
                    rna=rna)
        dataset = self.create_dataset_from_list(all_rnas)
        return dataset

    def post_process(self):
        # Remove sequence redundancy
        cd_hit_computer = CDHitComputer(similarity_threshold=0.9)
        cd_hit_rr = RedundancyRemover(distance_name="cd_hit",
            threshold=0.9)
        self.dataset = cd_hit_computer(self.dataset)
        self.dataset = cd_hit_rr(self.dataset)

        # Remove structural redundancy
        us_align_computer = StructureDistanceComputer(name="USalign")
        us_align_rr = RedundancyRemover(distance_name="USalign",
            threshold=0.8)
        self.dataset = us_align_computer(self.dataset)
        self.dataset = us_align_rr(self.dataset)
```

Figure 8: Code used to implement a new task.