CHARACTER-LEVEL TOKENIZATIONS AS POWERFUL INDUCTIVE BIASES FOR RNA FOUNDATIONAL MODELS

Adrián Morales-Pastor[1], **Raquel Vázquez-Reza**[1], **Miłosz Wieczór**[2], **Clàudia Valverde**[1], **Manel Gil-Sorribes**[1], **Bertran Miquel-Oliver**[3], **Álvaro Ciudad**[1,4], **Alexis Molina**[1,4*]

[1]Nostrum Biodiscovery, Barcelona, 08029, Spain
[2]IRB Barcelona, Barcelona, 08028, Spain
[3]Barcelona Supercomputing Center, Barcelona, 08034, Spain
[4]Atlas Labs, Barcelona, Spain
*alexis.molina@nostrumbiodiscovery.com, @theatlaslabs.com

ABSTRACT

RNA plays a critical role in cellular functions and is increasingly targeted for therapeutics, yet its structural complexity poses challenges for computational modeling. While foundational models have transformed protein representation learning, achieving similar success for RNA remains elusive. We introduce ChaRNABERT, a suite of sample- and parameter-efficient RNA foundational models that leverage a learnable tokenization process to achieve superior performance across established benchmarks. We further validate its capabilities on downstream tasks, including RNA-protein and aptamer-protein interaction prediction. The ChaRNABERT-8M model, along with inference code, will be publicly available for academic research, with additional models provided upon request.

1 INTRODUCTION

RNA plays a pivotal role in gene expression, regulation, and therapeutics, making it essential for both biological research and medical advancements. From mRNA vaccines to CRISPR-based gene editing, RNA technologies have revolutionized treatment approaches, yet challenges like stability, delivery, and off-target effects persist (Saw & Song, 2024). Traditional computational tools, such as ViennaRNA for secondary structure prediction (Lorenz et al., 2011) and molecular simulations for dynamic behavior analysis (Sponer et al., 2018), offer valuable insights but often struggle with scalability and resolution. Artificial intelligence (AI) presents a promising alternative, with deep learning models transforming protein science, exemplified by ESM (Lin et al., 2023), while RNA-specific models remain specialized and less generalizable (Penić et al., 2024). To bridge this gap, we introduce ChaRNABERT, a foundational RNA model designed for diverse biological tasks, leveraging Gradient-Based Subword Tokenization (GBST) to dynamically learn biologically meaningful subsequences without relying on a fixed vocabulary (Tay et al., 2022). By training on diverse RNA types, ChaRNABERT generalizes across tasks like aptamer interaction prediction and RNA structure analysis, demonstrating its potential as a scalable, data-driven solution for RNA research and therapeutics.

2 METHODS

ChaRNABERT's architecture. ChaRNABERT (CRB) integrates Gradient-Based Subword Tokenization (GBST) to dynamically learn optimal tokenization patterns for RNA sequences, eliminating the need for a predefined vocabulary. This approach allows the model to identify biologically relevant subsequences while preserving single-nucleotide resolution, key for many downstream applications. GBST operates by applying a depthwise convolution for smoothing, generating candidate subsequence embeddings through overlapping nucleotide spans, and using a scoring network to determine

the most relevant subsequence at each position. A self-attention-inspired probability refinement mechanism ensures that tokenization is contextually optimized across the sequence, making the process fully differentiable and adaptable during training. The model constructs latent subsequence representations by assigning a probability distribution over candidate subsequences and offsets, allowing a soft selection mechanism to emerge. This ensures that RNA sequences are tokenized in a data-driven manner, enabling the model to capture both local and global sequence dependencies. Unlike traditional subword tokenization, GBST allows the model to flexibly segment RNA sequences based on structural and functional relevance. An in-depth explanation of ChaRNABERT can be found in section B.1

Pre-training. We pre-trained CRB models with varying configurations to identify optimal protocols and assess the impact of key factors on performance. Experiments included models with 8, 33, 50, and 150M parameters, datasets featuring non-coding RNA or non-coding and coding RNA types, and diverse masking regimes inspired by masked language modelling and UL2 (Tay et al., 2023). All models were trained for four epochs to evaluate token recycling effects. More details about the pre-training can be found in section C.1

3 Assessing CRB's performance in downstream applications

3.1 BEACON benchmark

Implementing BEACON benchmark. We leverage BEACON (BEnchmArk for COmprehensive RNA tasks and language models) to assess the performance of the CRB models in downstream tasks (Ren et al., 2024a). BEACON is the first comprehensive benchmark designed to evaluate deep learning methods for RNA analysis, encompassing 13 tasks across structural analysis, functional studies, and engineering applications. BEACON evaluates both traditional models such as CNNs (LeCun et al., 1989), ResNets (He et al., 2015), and LSTMs (Hochreiter & Schmidhuber, 1997), as well as RNA language models (RNA LM) like RNA-FM (Chen et al., 2022b) and RNA-BERT (Akiyama & Sakakibara, 2022a). We also added RiNALMo (Penić et al., 2024) to the BEACON benchmark, as we find it an interesting comparison point given its 650M parameter size, larger than any model currently in the comparative study.

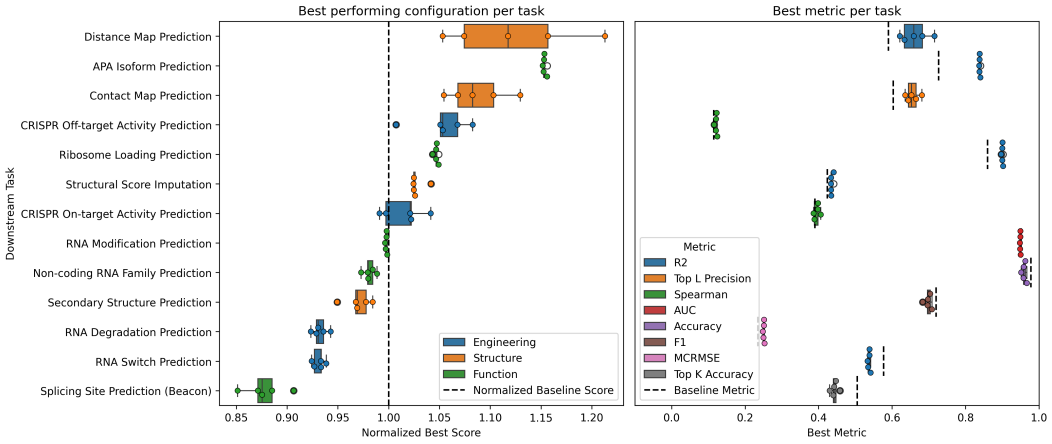

Figure 1: *Left.* Best relative performance achieved by CRB compared to the corresponding baseline (LSTM, CNN, RNA LM or RiNALMo) for each downstream task in the BEACON benchmark. *Right.* Best absolute performance obtained with CRB in the BEACON benchmark.

CRB performs competitively on all tasks compared with the best RNA LM. Figure 1 shows the performance of the best model configuration for each task relative to the goal metric from BEACON or RiNALMo.

In structural prediction tasks, CRB models perform competitively, particularly in mapping spatial relationships critical to RNA's tertiary structure. In Distance Map Prediction, CRB-50M surpasses all

models, effectively capturing RNA spatial patterns. For Structural Score Imputation, CRB-33M and CRB-50M excel in reconstructing structural information despite missing data. However, in Secondary Structure Prediction, RiNALMo slightly outperforms CRB, likely due to its larger parameter count and structure-focused training.

For functional prediction tasks, CRB models excel at identifying biologically significant RNA sequence patterns. In APA Isoform Prediction, CRB-8M outperforms other models, accurately predicting PAS usage ratios. CRB-50M matches RiNALMo's performance in predicting translation efficiency. However, in ncRNA Function Classification and Splice Site Prediction, CRB is surpassed by RiNALMo and Splice-MS510 (Alqassem et al., 2021) respectively.

In engineering tasks, CRB models are high-performing as well. For RNA vaccine degradation prediction, CRB-8M outperforms the best RNA LM reported in BEACON and follows RiNALMo closely. CRB-33M is better than other learned embeddings in CRISPR On-Target Prediction, and matches the best performing model for CRISPR Off-Target Prediction. Other LMs struggle to match CNN and LSTM on this task, however, GBST allows CRB to reach the same score.

CRB achieves competitive performance over a majority of tasks with significantly fewer parameters and no specialized training, demonstrating the efficiency of character-level tokenization. Complete results are in Table 3.

CRB performs consistently better in structural and engineering tasks. Figure 2.left shows the distribution of the best normalized scores grouped by the task categories defined in the BEACON benchmark. Statistical tests confirmed that the mean scores for structural and engineering tasks were significantly greater than 1, indicating that CRB consistently surpasses the reference models in these domains.

Task categories respond differently to model size. We also analyzed how model size influences performance across task categories (Figure 2.right). Functional tasks remain largely unaffected by model size, with performance distributions centered around 1. Structural tasks achieve peak performance at 50 million parameters, with both smaller and larger models showing a decline. In contrast, engineering tasks perform best at 8 million parameters, where increasing model size leads to a gradual performance drop.

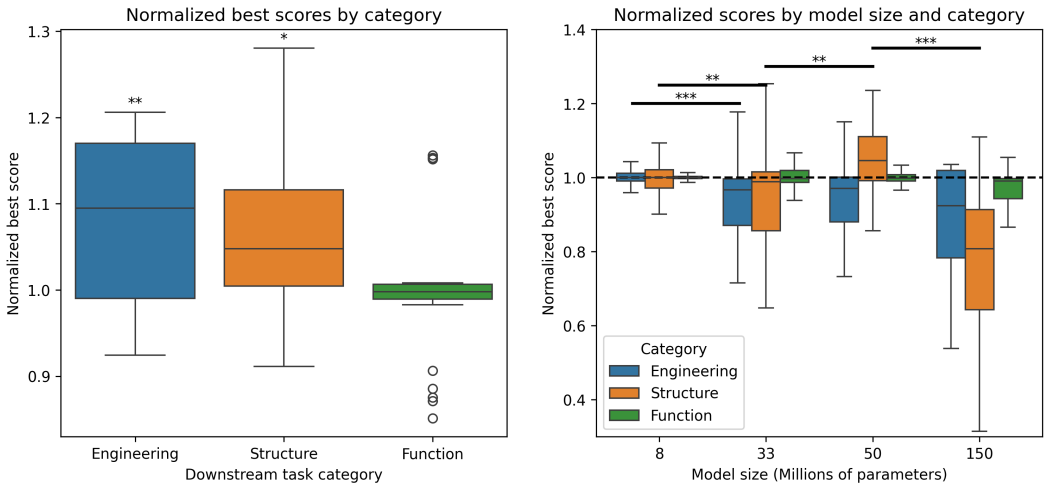

Figure 2: *Left*. Distribution of best, normalized scores for each beacon task grouped by task category. *Right*. Distribution of best, normalized scores for each beacon task grouped by task category and model size.

3.2 Extending the benchmark tasks for RNA

To further assess the capabilities of our foundational language models, we expanded the range of benchmark tasks to gain a deeper understanding of their strengths and limitations. By leveraging

existing databases, we curated new datasets designed to test the models' ability to generalize from sequence data to biologically relevant applications, pushing the boundaries of their performance across diverse downstream tasks.

We introduce a diverse set of downstream tasks, including structure-based predictions such as the categorical distance map task from BEACON and RNA-RNA interaction prediction. Additionally, we incorporate tasks from prior research, such as miRNA target prediction from DeepMirTar (Wen et al., 2018), splice site prediction from RiNALMo (Penić et al., 2024) and RNA-Protein affinity prediction from CoPRA (Han et al., 2025). Beyond these, we implemented multiple RNA molecular interaction tasks, including RNA-binding protein interaction prediction and aptamer-protein binding prediction, using both sequence- and structure-based splits. Finally, we include functional annotation tasks such as genomic segment labeling and RNA half-life prediction. Details on dataset generation, training, and architecture are provided in Appendix C.3.

3.2.1 Training and Baselines

Similarly to BEACON downstream tasks, we explored various model sizes, masking strategies, and dataset compositions. Each task was trained with five replicates to ensure consistency. For benchmarking, we trained three baselines: a CNN, an LSTM, and RiNALMo, allowing a direct comparison between CRB and a state-of-the-art transformer model.

3.2.2 Performance on additional RNA tasks

As shown in Figure 3, CRB matches or surpasses the reference model in most tasks. The only exceptions are splice site prediction and protein-RNA binding affinity prediction. Notably, in categorical distance map prediction, CRB outperforms the reference model by 40%, achieving an F1 score of 0.423. It also shows a strong advantage in genomic segment labeling.

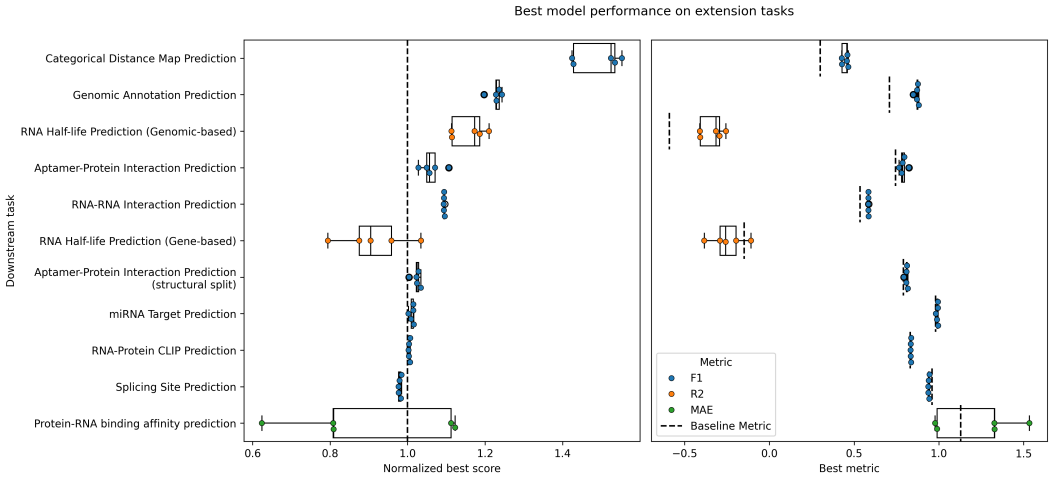

Figure 3: *Left.* Best relative performance achieved by CRB compared to the corresponding baseline (LSTM, CNN, or RiNALMo) for each downstream task in the extension tasks. *Right.* Best absolute performance obtained with CRB in the extension tasks.

However, two tasks remain particularly challenging. The first is RNA half-life prediction, which, as a regression task, saw models failing to surpass the zero baseline. This indicates no improvement over simply predicting the mean of the validation set. Despite exhibiting standard learning curves during training, the models failed to generalize, suggesting that RNA half-life may depend on factors beyond sequence composition alone.

The second challenge is RNA-RNA interaction prediction, where models attempt to predict a binary interaction matrix between two RNA sequences. A naïve baseline that predicts either no interactions or all interactions achieves an F1-score of 0.5, establishing a minimum threshold. The highest-performing model, CRB, achieved a marginally better score of 0.542. Efforts to mitigate overfitting

by progressively unfreezing transformer layers led to degraded validation performance, reinforcing the hypothesis that limited training data is the primary constraint. This conclusion is further supported by CRB's strong performance in secondary structure prediction, a closely related task with more training data.

CRB learns better residue-level representations. Downstream tasks can be categorized based on whether predictions are made at the sequence or residue level. Sequence-level tasks involve making predictions for entire sequences, typically utilizing the CLS token embedding. In contrast, residue-level tasks generate predictions for each residue individually, relying on residue-specific embeddings. As shown in Figure 4, CRB performs significantly better in residue-level tasks than in sequence-level tasks (p-value = 1e-8). This highlights GBST's role in improving residue-level representations while maintaining strong sequence-level performance. Notably, GBST enables a CRB to compete with models 13-80× larger, likely due to the enriched local context in its character-level embeddings. This finding is particularly relevant given the rise of matmul-free transformer architectures (Zhu et al., 2024). While these models achieve strong sequence-level performance with lower computational costs, they struggle with residue-level tasks due to lower-resolution outputs. We speculate that combining such architectures with GBST may improve their performance while maintaining efficiency.

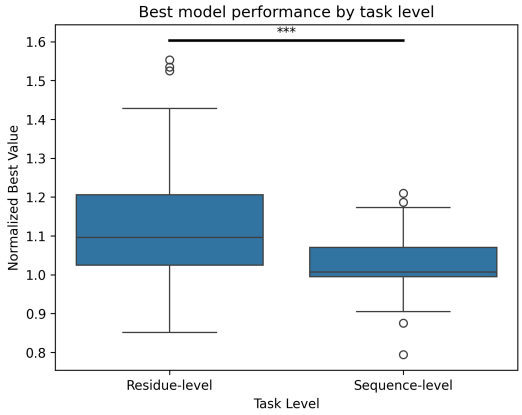

Figure 4: Best performance achieved by CRB relative to the corresponding baseline for all downstream task separating into residue-level and sequence-level tasks.

4 Impact of Pretraining Configuration on Downstream Tasks

In this section, we examine how pretraining configuration affects performance across the 23 downstream tasks featured in this study. Specifically, we analyze the impact of token recycling, dataset composition, and masking protocol on model performance. For each pretraining variable, we define a baseline configuration and compare it against models trained with alternative configurations. For every task, we identify the best-performing model trained with the baseline setting and the best-performing model trained with any alternative setting. Performance scores of the alternative models are then expressed relative to their respective baseline models.

Statistical significance is assessed by testing for differences in means between the performance distributions of models trained with the baseline and alternative configurations. For conciseness, detailed analyses of token recycling and dataset composition are provided in Appendix C.4.

4.1 Impact of Masking Protocol on Model Performance

We compare the standard masked language modeling (MLM) approach with UL2 masking, evaluating their effects across the 23 downstream tasks. In this analysis, MLM serves as the baseline configuration, while UL2 is the alternative. In total, 17 out of the 23 tasks used the baseline configuration, while only 6 of them used UL2 masking.

Figure 5 presents the distribution of performance scores normalized relative to the baseline configuration. The results indicate that for most tasks, the masking protocol does not significantly affect performance. However, a few tasks exhibit statistically significant differences. Notably, RNA half-life prediction and APA isoform prediction show a positive shift with UL2 masking, suggesting that a more diverse masking strategy may enhance representation learning in certain functional tasks. Conversely, structural tasks such as secondary structure prediction, distance map prediction, and contact map prediction show a significant performance drop when using UL2, suggesting that MLM may better capture fine-grained structural dependencies in RNA sequences.

These findings suggest that while UL2 masking does not substantially impact most tasks, its effects are task-dependent. Functional tasks may benefit from UL2's increased masking variability, while structural tasks appear to favor the more constrained MLM approach. This pattern aligns with the local nature of the representations obtained by GBST and MLM, which aids tasks that rely heavily on residue-level information. Selecting an appropriate masking strategy should therefore consider the specific nature of the downstream tasks.

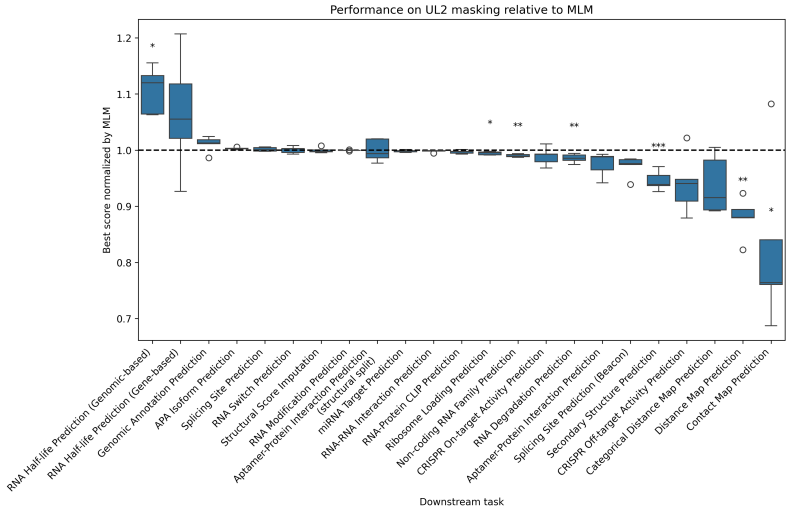

Figure 5: Best performance of CRB using UL2 masking relative to the performance using using MLM masking.

5 Conclusion

In this study, we introduced ChaRNABERT (CRB), an RNA foundational model using learnable character-level tokenization. Across the BEACON benchmark, CRB models achieved competitive or superior performance compared to existing models, often with fewer parameters.

CRB-50M excelled in structural prediction tasks, surpassing larger, specialized models by learning complex features directly from sequences. In functional tasks, such as modeling alternative polyadenylation site usage, CRB models provided critical insights into post-transcriptional modifications. While models like RiNALMo held a slight edge in secondary structure prediction due to task-specific training and larger parameter counts, CRB's generalist approach balances strong performance with computational efficiency. Scaling CRB model parameters offered limited gains across most tasks, suggesting diminishing returns beyond a certain size. However, larger models improved in tasks like non-coding RNA classification and distance prediction, indicating that specific tasks may benefit from targeted scaling. Combining coding and non-coding sequence datasets further enhanced CRB's performance, particularly with parameter scaling. We extended existing benchmarks by introducing downstream tasks absent from the BEACON benchmark. CRB achieved equal or superior performance across all tasks, demonstrating its effectiveness in handling complex challenges. Notably, its strong results in residue-level tasks underscore the potential of GBST in enhancing token representations.

Overall, CRB models consistently delivered strong results, with slight performance gaps in areas dominated by larger, task-specific models. This underscores the importance of rethinking tokenization strategies for biomolecules, as similar sequences can encode diverse functions. The use of GBST as an inductive bias allowed a straightforward BERT architecture to achieve state-of-the-art performance with fewer parameters. We hope this work paves the way for biomolecular language models with adaptable sequence representation and reduced reliance on manual tokenization.

REFERENCES

Manato Akiyama and Yasubumi Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics*, 4 (1):lqac012, 02 2022a.

Manato Akiyama and Yasubumi Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics*, 4 (1):lqac012, 02 2022b.

Israa Alqassem, Yash Sonthalia, Erika Klitzke-Feser, Heejung Shim, and Stefan Canzar. McSplicer: a probabilistic model for estimating splice site usage from RNA-seq data. *Bioinformatics*, 37(14): 2004–2011, 2021.

Ali Askari, Sumedha Kota, Hailey Ferrell, Shriya Swamy, Kayla S Goodman, Christine C Okoro, Isaiah C Spruell Crenshaw, Daniela K Hernandez, Taylor E Oliphant, Akshata A Badrayani, et al. UTexas aptamer database: the collection and long-term preservation of aptamer sequence information. *Nucleic Acids Research*, 52(D1):D351–D359, 2024.

Albi Celaj, Alice Jiexin Gao, Tammy T.Y. Lau, Erle M. Holgersen, Alston Lo, Varun Lodaya, Christopher B. Cole, Robert E. Denroche, Carl Spickett, Omar Wagih, Pedro O. Pinheiro, Parth Vora, Pedrum Mohammadi-Shemirani, Steve Chan, Zach Nussbaum, Xi Zhang, Helen Zhu, Easwaran Ramamurthy, Bhargav Kanuparthi, Michael Iacocca, Diane Ly, Ken Kron, Marta Verby, Kahlin Cheung-Ong, Zvi Shalev, Brandon Vaz, Sakshi Bhargava, Farhan Yusuf, Sharon Samuel, Sabriyeh Alibai, Zahra Baghestani, Xinwen He, Kirsten Krastel, Oladipo Oladapo, Amrudha Mohan, Arathi Shanavas, Magdalena Bugno, Jovanka Bogojeski, Frank Schmitges, Carolyn Kim, Solomon Grant, Rachana Jayaraman, Tehmina Masud, Amit Deshwar, Shreshth Gandhi, and Brendan J. Frey. An RNA foundation model enables discovery of disease mechanisms and candidate therapeutics. *bioRxiv 2023.09.20.558508*, 2023.

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

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

Ke Chen, Thomas Litfin, Jaswinder Singh, Jian Zhan, and Yaoqi Zhou. MARS and RNAcmap3: The master database of all possible RNA sequences integrated with RNAcmap for RNA homology search. *Genomics, Proteomics & Bioinformatics*, 22(1), 2024a.

Ken Chen, Yue Zhou, Maolin Ding, Yu Wang, Zhixiang Ren, and Yuedong Yang. Self-supervised learning on millions of primary RNA sequences from 72 vertebrates improves sequence-based RNA splicing prediction. *Briefings in Bioinformatics*, 2024b.

Yanyi Chu, Dan Yu, Yupeng Li, Kaixuan Huang, Yue Shen, Le Cong, Jason Zhang, and Mengdi Wang. A 5' UTR language model for decoding untranslated regions of mRNA and function predictions. *Nat. Mach. Intell.*, 6(4):449–460, April 2024.

RNAcentral Consortium. RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic Acids Research*, 49(D1):D212–D220, 10 2020.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 23–29 Jul 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Rong Han, Xiaohong Liu, Tong Pan, Jing Xu, Xiaoyu Wang, Wuyang Lan, Zhenyu Li, Zixuan Wang, Jiangning Song, Guangyu Wang, and Ting Chen. CoPRA: Bridging cross-domain pretrained sequence models with complex structures for protein-RNA binding affinity prediction. *arXiv preprint arXiv:2409.03773*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Aleksandra Helwak, Grzegorz Kudla, Tatiana Dudnakova, and David Tollervey. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. *Cell*, 153(3):654–665, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Sheng-Da Hsu, Yu-Ting Tseng, Sirjana Shrestha, Yu-Ling Lin, Anas Khaleel, Chih-Hung Chou, Chao-Fang Chu, Hsi-Yuan Huang, Ching-Min Lin, Shu-Yi Ho, et al. miRTarBase update 2014: an information resource for experimentally validated mirna-target interactions. *Nucleic Acids Research*, 42(D1):D78–D85, 2014.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

Jun-Hao Li, Shun Liu, Hui Zhou, Liang-Hu Qu, and Jian-Hua Yang. starBase v2. 0: decoding miRNA-ceRNA, miRNA-ncRNA and protein-RNA interaction networks from large-scale CLIP-seq data. *Nucleic Acids Research*, 42(D1):D92–D97, 2014.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

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.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

Nuala A O'Leary, Mathew W Wright, J Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, Alexander Astashyn, Azat Badretdin, Yiming Bao, Olga Blinkova, Vyacheslav Brover, Vyacheslav Chetvernin, Jinna Choi, Eric Cox, Olga Ermolaeva, Catherine M Farrell, Tamara Goldfarb, Tripti Gupta, Daniel Haft, Eneida Hatcher, Wratko Hlavina, Vinita S Joardar, Vamsi K Kodali, Wenjun Li, Donna Maglott, Patrick Masterson, Kelly M McGarvey, Michael R Murphy, Kathleen O'Neill, Shashikant Pujar, Sanjida H Rangwala, Daniel Rausch, Lillian D Riddick, Conrad Schoch, Andrei Shkeda, Susan S Storz, Hanzhen Sun, Francoise Thibaud-Nissen, Igor Tolstoy, Raymond E Tully, Anjana R Vatsan, Craig Wallin, David Webb, Wendy Wu, Melissa J Landrum, Avi Kimchi, Tatiana Tatusova, Michael DiCuccio, Paul Kitts, Terence D Murphy, and Kim D Pruitt. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, 44(D1):D733–45, January 2016.

Rafael Josip Penić, Tin Vlašić, Roland G. Huber, Yue Wan, and Mile Šikić. Rinalmo: General-purpose RNA language models can generalize well on structure prediction tasks. *arXiv preprint arXiv:2403.00043*, 2024.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984.

Yuchen Ren, Zhiyuan Chen, Lifeng Qiao, Hongtai Jing, Yuchen Cai, Sheng Xu, Peng Ye, Xinzhu Ma, Siqi Sun, Hongliang Yan, Dong Yuan, Wanli Ouyang, and Xihui Liu. BEACON: Benchmark for comprehensive RNA tasks and language models. *arXiv preprint arXiv:2406.10391*, 2024a.

Zilin Ren, Lili Jiang, Yaxin Di, Dufei Zhang, Jianli Gong, Jianting Gong, Qiwei Jiang, Zhiguo Fu, Pingping Sun, Bo Zhou, and Ming Ni. CodonBERT: a BERT-based architecture tailored for codon optimization using the cross-attention mechanism. *Bioinformatics*, 40(7):btae330, 05 2024b.

Phei Er Saw and Erwei Song. Advancements in clinical RNA therapeutics: Present developments and prospective outlooks. *Cell Reports Medicine*, 5(5), May 2024.

Yaiza Serrano, Alvaro Ciudad, and Alexis Molina. Are protein language models compute optimal? In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery*, 2024.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Reed S. Sorenson, Malia J. Deshotel, Katrina Johnson, Frederick R. Adler, and Leslie E. Sieburth. *Arabidopsis* mRNA decay landscape arises from specialized RNA decay substrates, decapping-mediated feedback, and redundancy. *Proceedings of the National Academy of Sciences*, 115(7): E1485–E1494, 2018.

Jiri Sponer, Giovanni Bussi, Miroslav Krepl, Pavel Banáš, Sandro Bottaro, Richard A Cunha, Alejandro Gil-Ley, Giovanni Pinamonti, Simón Poblete, Petr Jurecka, et al. RNA structural dynamics as captured by molecular simulations: a comprehensive overview. *Chemical Reviews*, 118(8):4177–4338, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), March 2024.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*, 2022.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2023.

Craig Tuerk and Larry Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science*, 249(4968):505–510, 1990.

Eric L Van Nostrand, Peter Freese, Gabriel A Pratt, Xiaofeng Wang, Xintao Wei, Rui Xiao, Steven M Blue, Jia-Yu Chen, Neal AL Cody, Daniel Dominguez, et al. A large-scale binding and functional map of human RNA-binding proteins. *Nature*, 583(7818):711–719, 2020.

Xi Wang, Ruichu Gu, Zhiyuan Chen, Yongge Li, Xiaohong Ji, Guolin Ke, and Han Wen. UNI-RNA: Universal pre-trained models revolutionize RNA research. *bioRxiv 2023.07.11.548588*, 2023.

Ming Wen, Peisheng Cong, Zhimin Zhang, Hongmei Lu, and Tonghua Li. DeepMirTar: a deep-learning approach for predicting human miRNA targets. *Bioinformatics*, 34(22):3781–3787, 06 2018.

Yu-Cheng T. Yang, Chao Di, Boqin Hu, Meifeng Zhou, Yifang Liu, Nanxi Song, Yang Li, Jumpei Umetsu, and Zhi John Lu. CLIPdb: a CLIP-seq database for protein-RNA interactions. *BMC Genomics*, 16(1):51, Feb 2015.

Yuning Yang, Gen Li, Kuan Pang, Wuxinhao Cao, Zhaolei Zhang, and Xiangtao Li. Deciphering 3'UTR mediated gene regulation using interpretable deep representation learning. *Adv. Sci. (Weinh.)*, pp. e2407013, August 2024.

Yikun Zhang, Mei Lang, Jiuhong Jiang, Zhiqiang Gao, Fan Xu, Thomas Litfin, Ke Chen, Jaswinder Singh, Xiansong Huang, Guoli Song, Yonghong Tian, Jian Zhan, Jie Chen, and Yaoqi Zhou. Multiple sequence alignment-based RNA language model and its application to structural inference. *Nucleic Acids Research*, 52(1):e3–e3, 11 2023.

Weihao Zhao, Shang Zhang, Yumin Zhu, Xiaochen Xi, Pengfei Bao, Ziyuan Ma, Thomas H Kapral, Shuyuan Chen, Bojan Zagrovic, Yucheng T Yang, et al. POSTAR3: an updated platform for exploring post-transcriptional regulation coordinated by RNA-binding proteins. *Nucleic Acids Research*, 50(D1):D287–D294, 2022.

Rui-Jie Zhu, Yu Zhang, Ethan Sifferman, Tyler Sheaves, Yiqiao Wang, Dustin Richmond, Peng Zhou, and Jason K. Eshraghian. Scalable matmul-free language modeling. *arXiv preprint arXiv:2406.02528*, 2024.

A Related Work

The rapid advancement of RNA language models has tried to mirror the transformative impact of language models in protein science, aiming to decode the ”language” of RNA sequences. These models endeavor to capture the underlying patterns, structural motifs, and functional elements inherent in RNA, thereby facilitating breakthroughs in structure prediction, functional annotation, and therapeutic design. The progression of these models reflects a concerted effort to overcome the unique challenges posed by RNA's structural diversity and functional versatility.

Early pioneers in this field, such as RNA-FM (Chen et al., 2022a) and RNABERT (Akiyama & Sakakibara, 2022b), laid the foundational groundwork for RNA language modeling. RNA-FM was one of the first general-purpose models designed for non-coding RNA (ncRNA) sequences. RNA-FM, a 100M parameter model, was trained on a previous RNAcentral release, encompassing 23 million samples. This model demonstrated the potential of language models to learn directly from RNA sequences, enabling tasks like secondary structure prediction and functional annotation. Concurrently, RNABERT emerged with a focus on structural alignment and clustering of ncRNA. By incorporating partial multiple sequence alignments from RNAcentral and the Rfam 14.3 dataset, totaling over 762 thousand sequences, RNABERT leveraged evolutionary information to enhance its ability to discern structural similarities among RNA molecules. This integration of evolutionary data marked a step toward understanding RNA structure-function relationships with a special focus over clustering and alignment.

Building on these foundations, models like UNI-RNA (Wang et al., 2023) sought to scale up both in model complexity and dataset size. UNI-RNA featured 400M parameters and was trained on an expansive dataset of 1 billion sequences from RNAcentral, the Nucleotide Collection (nt), and Genome Warehouse (GWH). Aiming to be a universal RNA model, UNI-RNA endeavored to capture a broad spectrum of RNA types and functions, enabling the modeling of very long RNA sequences without truncation.

Application specific models also made significant contributions. RNA-MSM (Zhang et al., 2023) introduced a novel approach by directly utilizing evolutionary information from multiple sequence alignments to model ncRNA sequences, benchmarking a diverse array of dowstream tasks. SpliceBERT (Chen et al., 2024b) addressed the critical aspect of RNA splicing in precursor messenger RNA (pre-mRNA), aiding in the prediction of splice sites and alternative splicing events. These advancements underscored the importance of specialized models in tackling specific biological questions.

Models like CodonBERT, UTR-LM, and 3UTRBERT (Ren et al., 2024b; Chu et al., 2024; Yang et al., 2024) focused on different regions of mRNA, capturing codon usage patterns and post-transcriptional regulation mechanisms mediated by untranslated regions (UTRs). CodonBERT, concentrated exclusively on the coding sequences (CDS) of mRNA, employing codon-level tokenization to capture patterns crucial for gene expression optimization. UTR-LM and 3UTRBERT specialized in the 5' and 3' UTRs, respectively, enhancing our understanding of mRNA expression, translational efficiency, and gene regulation mediated by UTRs.

BigRNA (Celaj et al., 2023) diverged from sequence-based models by integrating genomic context and utilizing thousands of genome-matched datasets. This approach underscored the importance of multi-omics data in capturing the complexity of RNA regulation in different cellular contexts, moving beyond sequence information to include expression patterns and regulatory interactions.

Despite these advancements, a noticeable gap remained when compared to the transformative impact of language models in protein science. Many existing RNA models were specialized or limited in scope, hindering their generalizability and broader applicability. Addressing this challenge, RiNALMo (RNA Integrated Language Model Optimization) (Penić et al., 2024) emerged as a notable milestone in RNA language modeling. RiNALMo was designed to bridge this gap by providing a comprehensive and versatile framework capable of capturing the full complexity of RNA sequences and structures. This model employed a deep transformer-based architecture with attention mechanisms tailored specifically for RNA.

One of the key introductions of RiNALMo was its pre-training strategy. The model was trained on an extensive and diverse dataset that included a wide array of RNA sequences from databases such as RNAcentral, as well as experimentally derived structural data. This multimodal training approach

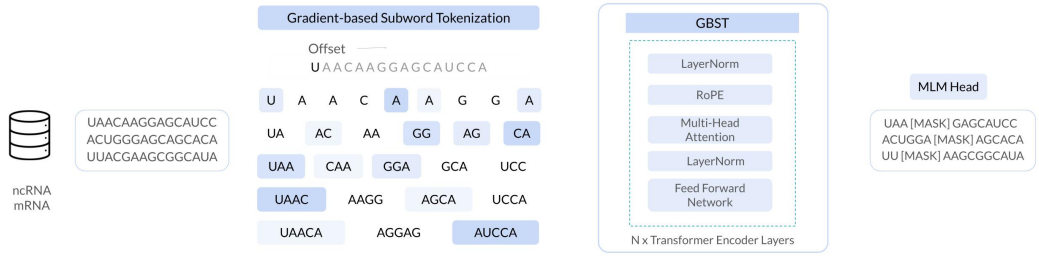

Figure B.6: ChaRNABERT's architecture. We train our models utilizing two datasets: one with exclusively non-coding sequences and another that combines both coding and non-coding sequences. Through Gradient-Based Subword Tokenization (GBST), the model learns optimal tokenization patterns for RNA sequences. ChaRNABERT employs a standard BERT transformer encoder, accommodating an input context of up to 8,190 nucleotides, and is trained using a masked language modeling objective to capture sequence information.

allowed RiNALMo to learn rich representations that encapsulate both linear sequence information and three-dimensional conformations of RNA, bridging the gap between sequence and structure. Moreover, RiNALMo introduced a modification in comparison to standard tokenization methods that went beyond simple nucleotide or codon representations. By utilizing nucleotide-level embeddings, the model could understand folding patterns and motifs crucial for RNA function.

In practical applications, RiNALMo set new benchmarks across multiple RNA-related tasks. It achieved state-of-the-art results in secondary and tertiary structure prediction, surpassing previous models in accuracy and reliability. Additionally, RiNALMo demonstrated exceptional capabilities in predicting RNA-protein and RNA-RNA interactions, key for understanding cellular processes and developing RNA-based therapeutics.

RNA language modeling currently lacks a foundational model that can handle a wide range of tasks through a straightforward token-masking framework without depending on task-specific data or dedicated pre-processing. Such a model would excel at efficiently learning from the intrinsic structure of RNA sequences.

To move away from imposed biases, we propose a new tokenization strategy. Instead of relying on single nucleotides, codons, or static k-mers, which each bring arbitrary assumptions and fixed nucleotide groupings, we introduce a learnable tokenization scheme that adapts to capture sequence details at multiple levels of granularity. Paired with a BERT-like transformer optimized for contextual understanding, this approach achieves competitive or superior performance relative to larger, task-specific RNA models while substantially reducing parameter demands.

B METHODS

B.1 CHARNABERT ARCHITECTURE

The ChaRNABERT (CRB) architecture is designed to be able to capture both fine-grained nucleotide details and broader contextual relationships efficiently, optimized for understanding the complex structures of RNA. At its core, CRB employs a modified Gradient-Based Subword Tokenization (GBST) (Tay et al., 2022), paired with a bidirectional BERT encoder (Devlin et al., 2019) . This combination allows the model to dynamically identify and emphasize biologically relevant subsequences without the constraints of a predefined vocabulary. Simultaneously, it captures the long-range dependencies and bidirectional context crucial for accurately modeling RNA structures and functions (Figure B.6).

B.2 CHARACTER-LEVEL TOKENIZATION

To effectively model subsequence information directly from nucleotide-level inputs, we employ a "soft" subword tokenization approach from character-level inputs. The original idea behind this approach is to allow the model to learn latent subsequence segmentations by dynamically selecting the most appropriate subsequence block at each character position during training procedure.

This key idea is extended through the enumeration of offsets in the sequences in a sliding window manner, as to model the equivalent of open reading frames (ORFs). The learnable combination of both approaches allows us to dynamically select the best tokenization possible for each of the nucleotides and whether or not to take into account the local environment of the sequence.

We also differ from the original implementation in the removal of the downsampling procedure, as single nucleotide resolution is highly desirable for several downstream applications.

B.2.1 CONSTRUCTING CANDIDATE SUBSEQUENCE EMBEDDINGS

Given an input sequence of nucleotides, they are embedded as a tensor $\mathbf{X} \in \mathbb{R}^{L \times d}$, where L is the sequence length and d is the nucleotide embedding dimension. To this individual representation GBST applies a one-dimensional depthwise convolution of kernel size equal to the maximum block size M, that acts as a smoothing operation, encouraging block level representations and allowing model to consider small shifts in the starting positions of blocks and the influence of non-adjacent nucleotides to some extent.

$$\tilde{\mathbf{X}} = \text{1D DWConv}\left(\mathbf{X}\right), \tag{1}$$

Afterwards, we generate candidate subsequence blocks by enumerating contiguous and overlapping spans of nucleotides up to a maximum block size M. For each block size b (where $1 \leq b \leq M$), we construct subsequence blocks $\mathbf{X}_{b,i}$ starting at position i by applying a pooling function over the embeddings of the nucleotides in the block:

$$\tilde{\mathbf{X}}_{b,i} = F\left(\tilde{\mathbf{X}}_{i:i+b}\right), \tag{2}$$

where $F : \mathbb{R}^{b \times d} \rightarrow \mathbb{R}^{d}$ is a non-parametric pooling function, in our case a sum pooling, that aggregates the embeddings within the selected subsequence into a single vector.

This procedure is also repeated for $b - 1$ offsets o of 1 for each of the block sizes in a sliding window manner. For example, for block size of one it obtains single nucleotides representations, whereas for block size two it is able to extract information from each pair of nucleotides starting from position 0 and position 1. This is continued up to the maximum block size M.

$$\tilde{\mathbf{X}}_{b,i,o} = F\left(\tilde{\mathbf{X}}_{i+o:i+b+o}\right), \tag{3}$$

B.2.2 FORMING LATENT SUBSEQUENCE REPRESENTATIONS

To determine the most suitable subsequence block and offset at each nucleotide position, the approach introduces a scoring network $F_R : \mathbb{R}^{d} \rightarrow \mathbb{R}$. This network computes a scalar score $p_{b,i}$ for each candidate $\mathbf{X}_{b,i,o}$, reflecting the model's confidence in selecting that representation:

$$p_{b,i,o} = F_R\left(\tilde{\mathbf{X}}_{b,i,o}\right). \tag{4}$$

We then compute a softmax over the scores for all representation sizes at position i, producing a probability distribution P_i over the candidate blocks and offsets:

$$P_i = \text{softmax}\left(\left[p_{1,i},\, p_{2,i},\, \ldots,\, p_{M,i}\right]\right). \tag{5}$$

This probabilistic weighting allows the model to softly select among the candidate representations based on their scores.

To enhance the model capabilities of capturing global context in the initial representation selection, we decided to incorporate the position-wise score calibration procedure from the original implementation. This layer computes a pseudo self-attention score between the different positions, encouraging the model to learn consensus among representation selection across the entire sequence. Specifically, updates are applied to the representation scores P using a self-attention mechanism without additional projections:

$$\tilde{P} = \text{softmax}\left(PP^{\top}\right)P, \tag{6}$$

where $P \in \mathbb{R}^{L\times M}$ is the matrix of block probabilities, and $\tilde{P}$ is the consensus probability matrix.

The latent subsequence representation at position i is obtained by computing a weighted sum of the candidate representation embeddings, using the probabilities from the modified probability matrix as weights:

$$\hat{\mathbf{X}}_i = \sum_{b=1}^{M}\sum_{o=0}^{M-1} \hat{P}_{b,i,o}\,\tilde{\mathbf{X}}_{b,i,o}. \tag{7}$$

This operation effectively allows the model to learn a soft subsequence segmentation, where each nucleotide position contributes to the final representation based on the likelihood of various block sizes and offsets. This soft selection mechanism ensures that the entire process is differentiable, enabling data-driven changes in the tokenization scheme during the training procedure and end-to-end training of the model.

Overall, by integrating GBST into our model, we leverage the strengths of subword representations while maintaining the flexibility, adaptability and resolution of nucleotide-level processing.

B.3 Bidirectional BERT encoder

As the main architecture we employ a BERT-based model with a few improvements. This model is a transformer encoder that enables bidirectional context learning through a self-attention mechanism and pre-training objectives. Each input token is mapped to an embedding and tokenized, combined with a positional encoding to maintain token order, and passed through multiple layers of the encoder. The core mechanism in each layer is *multi-head self-attention* (multi-heads are omited from all equations for clarity), where for each token i, its attention with all tokens j in the sequence is computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are linear projections of the input X, i.e. token embeddings, and d_k is the dimensionality of the keys/queries.

We introduce a few other common architectural modifications, namely SwiGLU's non-linearities (Shazeer, 2020), Rotary Positional Encodings (ROPE) (Su et al., 2024), Query Key Normalization (QKNorm) (Dehghani et al., 2023) and Flash Attention 2 (Dao, 2024).

$$\text{SwiGLU}(x) = \sigma(xW_1)\odot \text{swish}(xW_2) \tag{9}$$

where the Swish function is defined as:

$$\text{swish}(x) = x\cdot\sigma(x) \tag{10}$$

and σ is the standard sigmoid non-linearity.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{11}$$

SwiGLU non-linearities are a combination of the Swish and Gated Linear Unit (GLU) non-linearities that have shown improved performance over its individual parts or classical functions like ReLU.

We additionally move away from absolute positional encodings and introduce ROPE to our model, which has been shown to increase performance and length generalization capabilities in comparison with absolute and other relative positional encodings. This approach is based upon a rotation mechanism, where positions in the sequence are represented as rotations in the embedding space.

$$f_{\{q,k\}}(x_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \times \\ \times \begin{pmatrix} W^{(11)}_{\{q,k\}} & W^{(12)}_{\{q,k\}} \\ W^{(21)}_{\{q,k\}} & W^{(22)}_{\{q,k\}} \end{pmatrix} \times \begin{pmatrix} x^{(1)}_m \\ x^{(2)}_m \end{pmatrix} \tag{12}$$

with to the following values for θ to add the the long-term decay property between the relative positions to the positional encoding.

$$\theta_i = 10000^{-2i/d_k} \tag{13}$$

Moreover, to reduce the amount of training instabilities and loss spikes we decide to introduce QKNorm, effectively reducing the growth of the attention logits, which we found was a cause of instability in our training procedure. This mechanism applies a LayerNorm (LN) to the output of the Query and Key linear transformations.

$$\text{softmax}\left[\frac{1}{\sqrt{d_k}}\text{LN}(XW^Q)(\text{LN}(XW^K))^T\right] \tag{14}$$

Lastly, we include the hardware-aware Flash Attention 2 algorithm, allowing for an efficient increase of our context window and accelerated computation during training and inference.

C Training and model details

C.1 Pretraining

C.1.1 Masking strategies

Typically, BERT's pre-training uses a masked language model (MLM) objective, where a random subset of tokens is replaced with a special mask token. The model is trained to predict the original tokens based on both preceding and following context, forcing it to encode bidirectional information. The prediction of the masked token is computed as:

$$P(\text{token}_i | X_{\text{masked}}) = \text{softmax}(W_o h_i) \tag{15}$$

where h_i is the hidden state of token i after passing through multiple self-attention layers, and W_o is a learned output projection matrix.

Learning from more complex corruption schemes can enhance a model's ability to capture complex patterns and dependencies, therefore we chose to incorporate the UL2 (Unifying Language Learning) (Tay et al., 2023) paradigm into our training regimen. UL2 is a pre-training framework that unifies various language modeling objectives to create a more versatile and robust language model.

It introduces a novel masking strategy that combines different types of denoising objectives. These include short-span masking (S-denoising), extreme-span masking (X-denoising), and retrieval-augmented masking (R-denoising). S-denoising is similar to BERT's MLM objective, where individual tokens or short spans are randomly masked within the input sequence, and the model learns to predict these masked tokens using bidirectional context. X-denoising involves masking longer contiguous spans of text, which forces the model to understand and reconstruct larger chunks of information, thereby enhancing its ability to handle longer dependencies. R-denoising trains the model in an autoregressive fashion, predicting future tokens based on past context, akin to models like GPT.

UL2 employs a shared g-masked token between the strategies to replace the masked spans, providing a unified way for the model to identify and reconstruct the missing information regardless of the span length. These strategies are selected based on a predefined sampling strategy, exposing the model to the different denoising strategies along the training process.

This process involves preparing the input by selecting a mode (S, X, or R) according to a series of specified probabilities and masking the input text accordingly. The model processes the masked input to generate hidden states for each token, and for the masked positions, it predicts the original tokens using the surrounding context. The training objective is to minimize the cross-entropy loss between the model's predictions and the actual masked tokens across all modes. Therefore the prediction for each masked token i can be computed in the same way as the MLM objective (see Eq.15).

By leveraging UL2 masking, the model benefits from enhanced context understanding, versatility, and improved generalization. First, training on both short and long spans allows the model to comprehend and generate text over varying lengths, improving its understanding of context and long-range dependencies. Second, the combination of bidirectional and autoregressive objectives enables the model to perform well on a wide range of tasks.Last, exposure to different types of denoising tasks helps the model generalize better to unseen data and tasks.

C.1.2 RNA DATASETS

For our study, we employed RNAcentral (Consortium, 2020) as the primary source for non-coding RNA sequences in our training dataset. RNAcentral is an extensive repository that consolidates non-coding RNA data from multiple expert databases, providing a unified and comprehensive resource. The dataset encompasses a diverse range of RNA families, including but not limited to: microRNAs (miRNAs), small nuclear RNAs (snRNAs), small nucleolar RNAs (snoRNAs), transfer RNAs (tRNAs), ribosomal RNAs (rRNAs), long non-coding RNAs (lncRNAs), Piwi-interacting RNAs (piRNAs), and small interfering RNAs (siRNAs). In total, RNAcentral contributed approximately 31 million non-coding RNA sequences to our dataset, offering a rich and varied collection for training our models.

To enable a comprehensive analysis that includes both non-coding and coding sequences, we expanded our dataset by incorporating coding sequences from RefSeq (Reference Sequence) database at the National Center for Biotechnology Information (NCBI) (O'Leary et al., 2016). Specifically, we added 31 million coding sequences to our dataset. This augmentation resulted in a balanced and extensive dataset comprising both non-coding and coding sequences, which is crucial for training robust models capable of distinguishing between the two types.

C.1.3 MODEL SIZES

To assess the impact of model size on performance and explore scalability in RNA sequence analysis, we trained models with parameter counts mostly aligned with the main ESM models, specifically developing models with approximately 8M, 33M, 50M, 100M, and 150M parameters. Details about the architecture can be found in table 1.

Our exploration of scaling effects, both with and without GBST, involved training these models to examine the combined impact of character-level tokenization with the BERT encoder. This investigation not only focuses on MLM/UL2 loss performance but also evaluates the models' effectiveness on downstream tasks and generalization capabilities.

Despite we investigate scaling parametrically (see Section H.3), we chose to train models at these specific sizes as a baseline, even though it may not represent the optimal approach for all scenarios. This decision allows us to better analyze the interaction between model size and GBST. Our objective is to understand how these factors influence not only loss metrics but also broader performance across downstream applications.

For a thorough analysis, all model sizes were trained on two datasets: 31 million non-coding RNA sequences from RNAcentral and the combination of this dataset and the 31 million coding sequences from RefSeq.

All the models are trained in BF16 precision using Distributed Data Parallel with the DeepSpeed (Rasley et al., 2020) and ZeRO (Rajbhandari et al., 2020) frameworks for maximum optimization of computational resources.

Table 1: Dimensions by model parameters of ChaRNABERT models.

Parameters	**num_layers**	**d_model**	**num_heads**
8M	6	320	20
33M	12	480	20
50M	15	500	20
100M	23	600	20
150M	30	640	20

	Values			
Model size	8	33	50	150
Masking strategy	MLM		UL2	
Dataset	Non-coding		Coding-Non coding	
Token recycling	1	2	3	4

C.2 Implementing BEACON benchmark

For the downstream tasks in the BEACON RNA benchmark, we adhered to the methodology outlined in the benchmark's GitHub repository. All tasks used the same dataset, except for splice site prediction, which was also implemented following the RiNALMo framework. This alignment ensured direct comparison with a potentially stronger baseline by using RiNALMo's dataset and prediction objective. For all other tasks, we applied the preprocessing protocols from BEACON, with the exception of sequence encoding, which remained consistent with the encoding used during pretraining.

For task-specific modules, we primarily used BEACON's prediction heads. However, for structure-related tasks such as contact and distance map prediction, we adopted RiNALMo's secondary structure prediction head. This choice enabled direct comparison with a strong reference model while leveraging a validated architecture for structural prediction. To minimize confounding effects, task-specific modules were kept minimalistic, ensuring that performance differences primarily reflected the quality of the foundational model's embeddings.

Evaluation metrics followed those of the respective reference models. In most cases, we used BEACON's benchmark metrics, except for splice site prediction, where we employed RiNALMo's F1 score instead of top-k precision to maintain consistency with its task framing.

To ensure convergence, we implemented early stopping based on a smoothed validation metric with a patience threshold of 10,000 steps. The smoothed metric was computed using an exponential moving average (α = 0.1), with the threshold chosen based on the distribution of step counts where the highest validation score was maintained before further improvement. This approach prevented premature stopping while ensuring optimal performance.

In classification-based tasks, class imbalance posed challenges in learning meaningful patterns. Where necessary, we adjusted the loss function by weighting samples inversely to their frequency, prioritizing underrepresented positive samples and improving performance in imbalanced datasets.

We fine-tuned models of different sizes, dataset compositions and masking strategies, starting from the checkpoint corresponding to the first to the fourth epoch of pre-training. For a more fair comparison, we add RiNALMo's 650M model to the BEACON benchmark.

For easier representation of the downstream task performance of CRB, we decided to show the evaluation metrics as a relative score computed using the reference score from BEACON or RiNALMo. For downstream tasks where the performance metric had to be maximized, the relative score was computed as the ratio between the best score obtained by CRB and the baseline score. For tasks where the metric aligns with the optimization objective (i.e., ranging from 0 to $+\infty$, where 0 is the

Table 2: Summary of BEACON Benchmark Tasks, Categories, and Metrics

Abbreviation	Full Task Name	Task Category	Metric (Abbreviation)
SSP	Secondary Structure Prediction	Structure	F1 Score (F1)
CMP	Contact Map Prediction	Structure	Precision at Length (P@L)
DMP	Distance Map Prediction	Structure	Coefficient of Determination (R^2)
SSI	Structural Score Imputation	Structure	Coefficient of Determination (R^2)
SPL	Splice Site Prediction	Functional	Accuracy at K (ACC@K)
APA	APA Isoform Prediction	Functional	Coefficient of Determination (R^2)
NcRNA	Non-coding RNA Function Classification	Functional	Accuracy (ACC)
MRL	Mean Ribosome Loading	Functional	Coefficient of Determination (R^2)
Modif	RNA Modification Prediction	Functional	Area Under Curve (AUC)
VDP	Vaccine Degradation Prediction	Engineering	Mean Columnwise Root Mean Square Error (MCRMSE)
PRS	Prediction of RiboSwitches	Functional	Coefficient of Determination (R^2)
CRI-On	CRISPR On-Target Prediction	Engineering	Spearman Correlation (SC)
CRI-Off	CRISPR Off-Target Prediction	Engineering	Spearman Correlation (SC)

optimal value), the relative score improvement was computed as the difference in scores relative to the baseline.

Table 3: Tasks and their associated metrics, with model performance values across six different models. Baseline corresponds to the best performing LSTM, CNN, or ResNet as stated in BEACON. RNA LM corresponds to the best performing RNA language model in BEACON. We report performance of ChaRNABERT model of sizes 8M, 33M, and 50M parameters. Average performances and standard deviations were computed over five independent runs. **Bold** measures correspond to the best performing model(s) for the given tasks under a Bonferroni-corrected t-test (p-value < 0.05).

Task	**SSP**	**CMP**	**DMP**	**SSI**	**SPL**
Metric	F1	P@LR	R^2	F1	R^2
Baseline	0.59 ± 0.00	0.60 ± 0.01	0.45 ± 0.00	0.38 ± 0.00	0.36 ± 0.18
RNA LM	0.69 ± 0.01	0.60 ± 0.07	0.56 ± 0.00	0.42 ± 0.00	$\mathbf{0.50 \pm 0.49}$
RiNALMo	$\mathbf{0.72 \pm 0.01}$	0.49 ± 0.06	0.59 ± 0.04	0.39 ± 0.01	0.09 ± 0.04
CRB-8M	0.63 ± 0.03	0.54 ± 0.03	0.64 ± 0.03	0.43 ± 0.00	0.44 ± 0.01
CRB-33M	0.64 ± 0.01	0.56 ± 0.02	0.61 ± 0.04	$\mathbf{0.44 \pm 0.00}$	0.42 ± 0.02
CRB-50M	0.70 ± 0.01	$\mathbf{0.66 \pm 0.02}$	$\mathbf{0.66 \pm 0.04}$	0.43 ± 0.00	0.43 ± 0.02

Task	**APA**	**NcRNA**	**Modif**	**MRL**
Metric	R^2	ACC	AUC	R^2
Baseline	0.67 ± 0.01	0.89 ± 0.00	$\mathbf{0.95 \pm 0.01}$	0.84 ± 0.00
RNA LM	0.73 ± 0.85	0.97 ± 0.00	$\mathbf{0.95 \pm 0.00}$	0.85 ± 0.00
RiNALMo	0.82 ± 0.01	$\mathbf{0.98 \pm 0.01}$	0.76 ± 0.09	0.86 ± 0.01
CRB-8M	$\mathbf{0.84 \pm 0.00}$	0.96 ± 0.00	$\mathbf{0.95 \pm 0.00}$	0.87 ± 0.01
CRB-33M	0.82 ± 0.00	0.96 ± 0.01	$\mathbf{0.95 \pm 0.00}$	$\mathbf{0.90 \pm 0.00}$
CRB-50M	0.83 ± 0.00	0.96 ± 0.01	$\mathbf{0.95 \pm 0.00}$	0.89 ± 0.02

Task	**VDP**	**PRS**	**CRI-On**	**CRI-Off**
Metric	MCRMSE	R^2	SC	SC
Baseline	0.33 ± 0.00	0.55 ± 0.01	0.27 ± 0.01	$\mathbf{0.12 \pm 0.00}$
RNA LM	0.31 ± 0.00	$\mathbf{0.58 \pm 0.01}$	0.35 ± 0.00	0.05 ± 0.01
RiNALMo	$\mathbf{0.23 \pm 0.01}$	0.47 ± 0.02	0.39 ± 0.07	0.01 ± 0.04
CRB-8M	0.25 ± 0.00	0.54 ± 0.01	0.39 ± 0.00	$\mathbf{0.12 \pm 0.00}$
CRB-33M	0.25 ± 0.00	0.52 ± 0.01	$\mathbf{0.40 \pm 0.01}$	0.11 ± 0.01
CRB-50M	0.25 ± 0.00	0.50 ± 0.00	0.39 ± 0.00	$\mathbf{0.12 \pm 0.01}$

C.2.1 IMPACT OF TOKEN RECYCLING IN PERFORMANCE

Examining the effect of token recycling during pretraining, we observe that its impact is highly task-dependent. As a result, no significant overall effect is observed across the three task groups defined in the BEACON benchmark (Figure C.7).

At the individual task level, we find diverse behaviors. In some cases, such as the BEACON version of splice site prediction, peak performance is achieved after a single epoch of pretraining, with additional dataset traversals negatively affecting performance. In many other tasks, model performance remains unaffected by token recycling, at least up to four dataset passes. Conversely, certain tasks benefit from increased exposure to the pretraining dataset, with CRB achieving better results as training progresses.

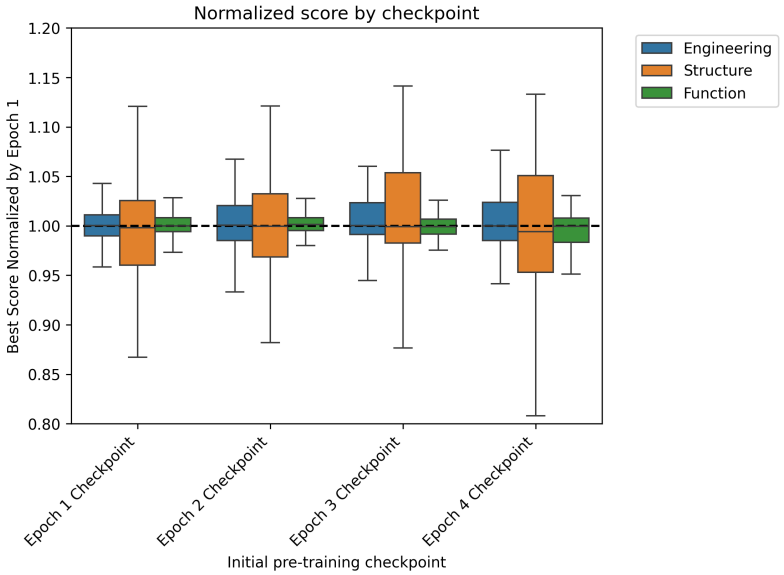

Figure C.7: Best performance achieved by CRB relative to the checkpoint 1 for each downstream task grouped by category.

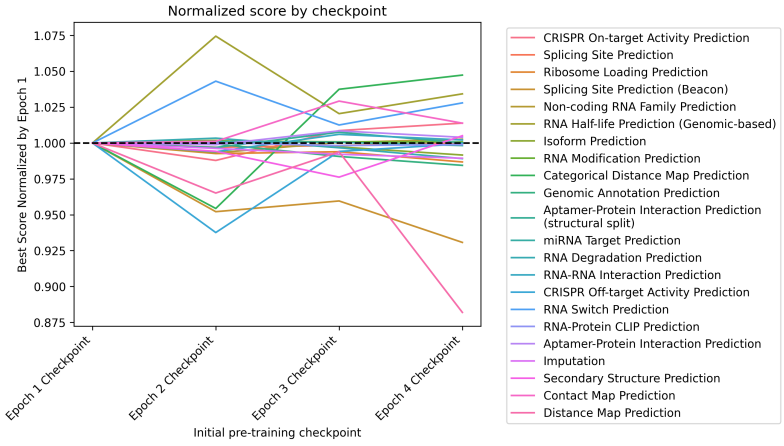

Figure C.8: Best performance achieved by CRB relative to the checkpoint 1 for each downstream task.

C.2.2 IMPACT OF DATASET COMPOSITION ON MODEL PERFORMANCE

In this study, we pretrained the foundational model using two versions of the dataset: one containing only non-coding RNA sequences and another extending this dataset with coding RNA sequences. This approach allowed us to investigate which tasks benefit from pretraining on both RNA types.

Figure C.23 presents the distribution of evaluation scores for CRB configurations pretrained on the coding and non-coding dataset, expressed relative to scores obtained with the same configurations trained only on the non-coding dataset. While some distributions deviate slightly from 1, only one task—isoform prediction—exhibits a statistically significant difference. Although its mean score remains close to 1, the low variance allows us to confidently confirm the deviation.

Among the tasks that benefit the most from pretraining with coding RNA, we identify the Beacon version of splice site prediction, RNA half-life prediction, and CRISPR-related tasks (on-target and off-target activity prediction). Interestingly, all four tasks are closely related to mRNA, suggesting that exposure to coding RNA during pretraining provides a mild but consistent advantage in related downstream tasks.

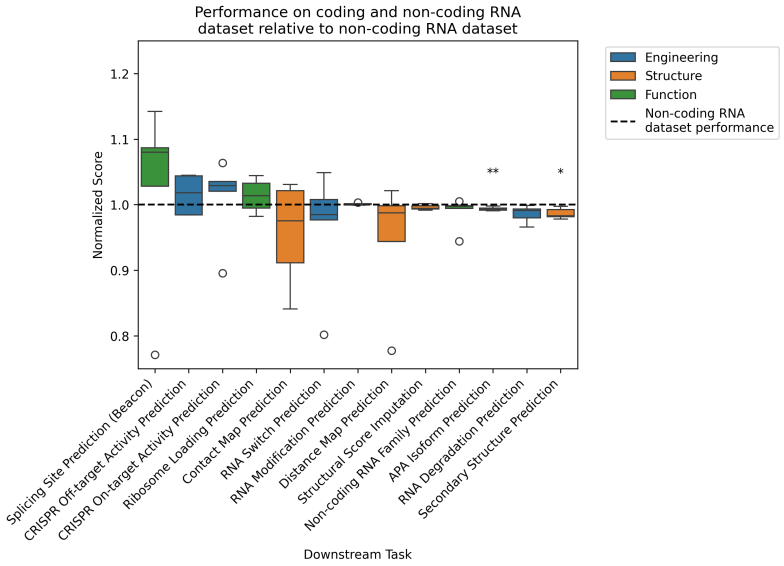

Figure C.9: Best performance of CRB using the non-coding and coding RNA dataset relative to the performance using only the non-coding RNA dataset.

C.2.3 IMPACT OF MASKING PROTOCOL ON MODEL PERFORMANCE

We also investigated how the masking protocol used during pretraining influences model performance on downstream tasks. Figure 5 presents the distribution of scores for models trained with UL2 masking, expressed relative to their counterparts trained with standard MLM masking.

Overall, very few tasks exhibit a statistically significant deviation from 1, indicating that the choice of masking method is generally not a decisive factor in downstream performance. Even in cases where differences are significant, the absolute impact remains small. For example, in the distance map prediction task, models pretrained with UL2 masking perform significantly worse than those using MLM. However, their relative scores remain above 90%, suggesting that while the difference is statistically significant, its practical impact is minimal.

Conversely, tasks such as RNA switch prediction and half-life prediction show slight improvements with UL2 masking, though these differences are not statistically significant. In some cases, the high variability within score distributions suggests that the effect of masking interacts with other factors, making it difficult to isolate its precise contribution.

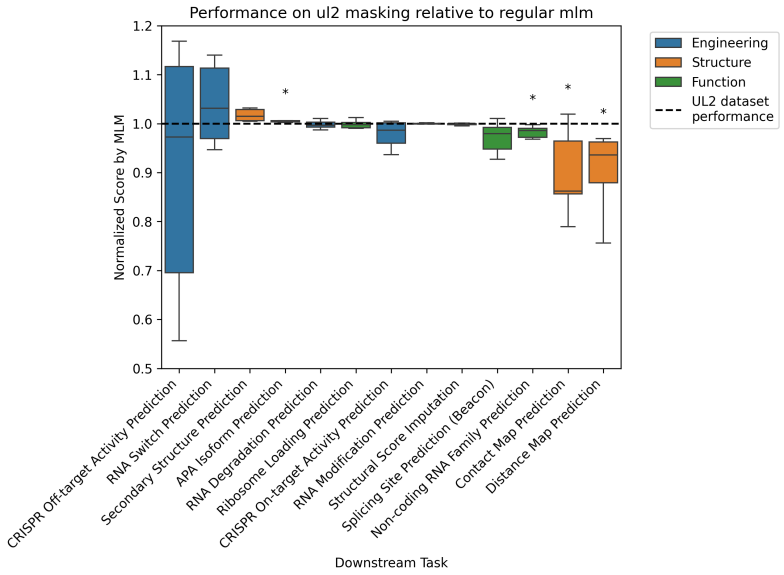

Figure C.10: Best performance of CRB using UL2 masking relative to the performance using using MLM masking.

C.2.4 ANALYSIS OF INTERACTION BETWEEN TRAINING VARIABLES

To further investigate the impact of training variables on model performance, we examined multiple factors simultaneously, including dataset composition, masking protocol, and checkpoint selection, in relation to model size.

Figure C.11 illustrates the interaction between model size and pretraining dataset composition. In most cases, the two variables exhibit no interaction, as models pretrained on different datasets behave similarly across sizes. However, a few tasks show distinct responses to model size depending on the dataset used.

The most pronounced differences appear in CRISPR off-target activity prediction and RNA half-life prediction. In the former, pretraining on coding and non-coding RNA introduces a negative interaction with model size: while increasing model size has little effect on models trained with non-coding RNA, performance declines in models trained on the combined dataset. Conversely, in RNA half-life prediction, we observe a performance drop when increasing model size from 33M to 50M parameters, but only for models trained on coding and non-coding RNA—this effect is absent in models trained exclusively on non-coding sequences.

For most other tasks, the interaction between dataset composition and model size remains minimal, with no clear trend.

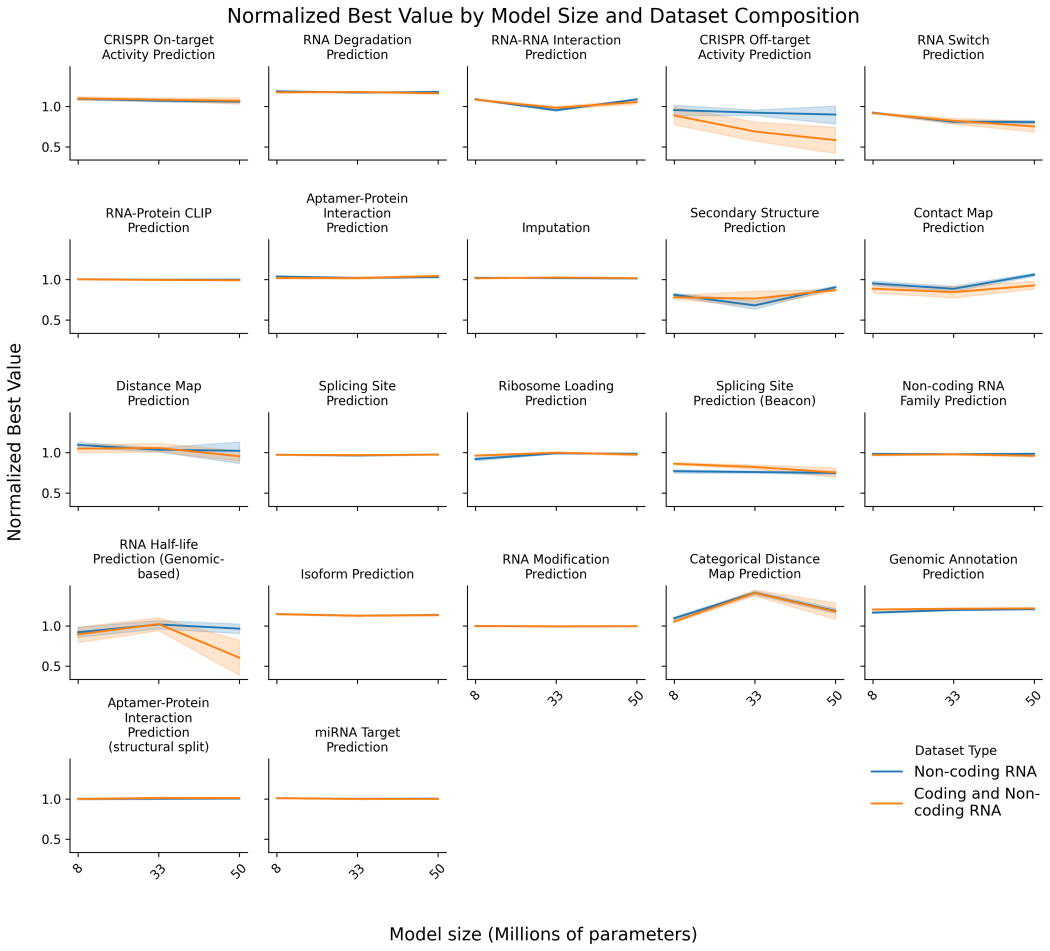

Figure C.11: Model performance by model size and dataset composition.

When analyzing the interaction between the masking regime and model size, we observe stronger interactions across more tasks compared to dataset composition. While CRISPR off-target prediction and RNA half-life prediction continue to show notable interactions, additional effects emerge in contact map prediction and distance map prediction. In both cases, performance at 33M parameters remains similar across masking regimes, but divergence appears in smaller and larger models. Notably, the MLM masking protocol consistently outperforms UL2 in these two tasks.

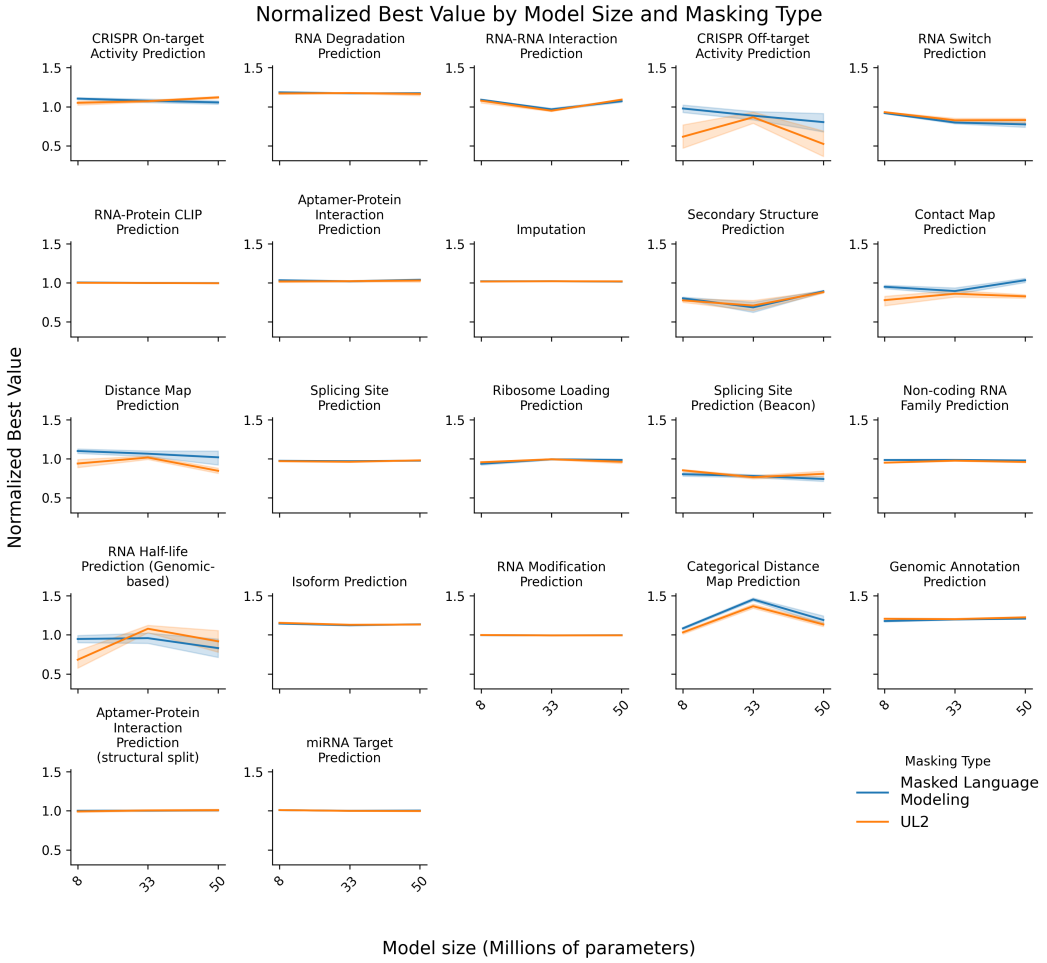

Figure C.12: Model performance by model size and masking type.

We also examined the interaction between token recycling during pretraining and model size. As with previous analyses, CRISPR off-target prediction and RNA half-life prediction exhibit the strongest interactions. However, we also observe significant interactions in distance map prediction, secondary structure prediction, and categorical distance map prediction. Interestingly, token recycling appears to interact most strongly with model size in structural feature prediction tasks. This interaction is not necessarily linear, as seen in distance map prediction, where a sharp drop in performance occurs when comparing a 50M model pretrained for 3 versus 4 epochs.

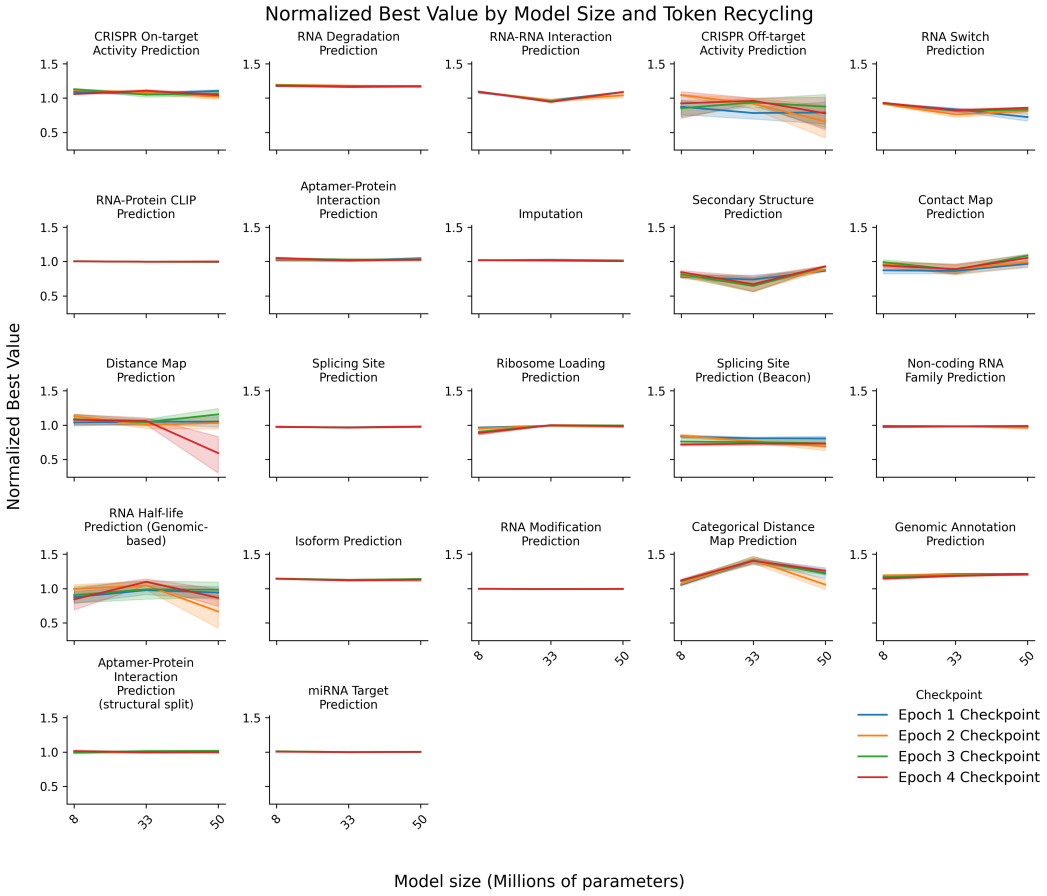

Figure C.13: Model performance by model size and token recycling.

C.3 Extending RNA downstream tasks

C.3.1 Predicting aptamer-protein interactions

An RNA aptamer is a short, single-stranded nucleic acid that can selectively bind to a specific target molecule, such as proteins, small molecules, or even entire cells. Aptamers are typically identified through a process known as SELEX (Systematic Evolution of Ligands by Exponential enrichment) (Tuerk & Gold, 1990), where a large, randomized library of sequences is screened to find those that bind strongly and specifically to the desired target. Their specificity makes them highly useful in diagnostics, therapeutics and research.

Dataset

We leverage the University of Texas Aptamer Database (TAD) (Askari et al., 2024) to curate a dataset of 2310 pairs of aptamer-protein interactions. The database comprises 1443 aptamer sequences and 561 different protein sequences. Since TAD only encompasses positive examples, we generated negative samples dataset by leveraging sequence dissimilarity as the primary criterion. Specifically,

we ensured that the protein sequences selected for negative pairing did not share more than 50% sequence similarity with any of the positive interaction partners. We then split in training and testing sets by ensuring a maximum of 50% identity in sequence similarity between training and testing aptamer sequences. We then split in 2 different training and testing sets by ensuring a maximum of 50% identity in sequence similarity and secondary structure similarity between training and testing aptamers.

Model Architecture and Training

We extracted protein sequence representations using ESM-650M which we used statically. In order to compress the information of the whole representation into a single dimensional tensor we used 10 blocks of residual convolution followed by an adaptive average pooling. The aptamer sequence was processed by CRB and the class token was used for the following steps. Finally, the protein and RNA-derived tensors are concatenated and passed through a linear layer. We used a binary cross entropy as the loss function and the F1 as main performance metric following the choices performed in the BEACON benchmark.

Figure C.15 shows the performance of different CRB configurations grouped by pretraining dataset and model size. Despite no statistically significant difference is found between any of the groups, a trend is observed. Model performance tends to increase with model size, suggesting that for this task in particular may be beneficial to further increase the size of the model. Interestingly, the CRB architecture with 50 million parameters and the coding and non-coding RNA pre-training dataset already surpasses RiNALMo's performance, suggesting that GBST greatly compensates the need for parameters while still allowing model improvement by increasing them.

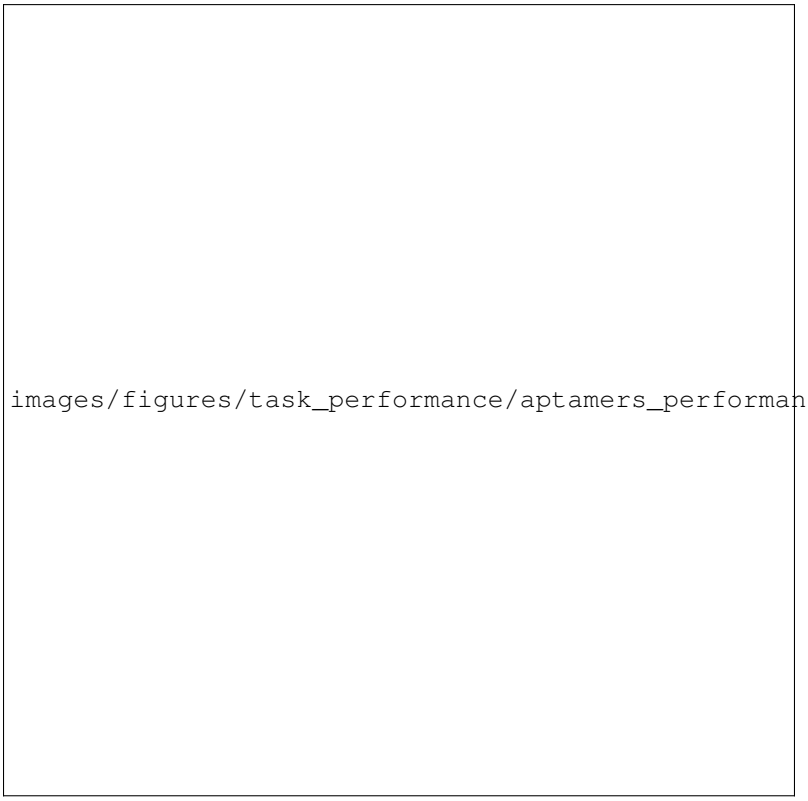

Figure C.14: Performance of different CRB configurations on the aptamer-protein interaction task.

Table 3: F1 performance in the TAD datasets. *Sequence 0.5* corresponds to model performance in the TAD test set by sequence identity split, *Structure 0.5* corresponds to the performance with the structural similarity split. **Bold** measures indicate best model for a dateset under a t-test with p-value < 0.05.

Model	Sequence 0.5	Structure 0.5
LSTM	0.724 ± 0.014	0.783 ± 0.019
CNN	0.647 ± 0.024	0.762 ± 0.016
RiNALMo	0.744 ± 0.015	0.790 ± 0.008
ChaRNABERT	$\mathbf{0.791 \pm 0.018}$	$\mathbf{0.808 \pm 0.009}$

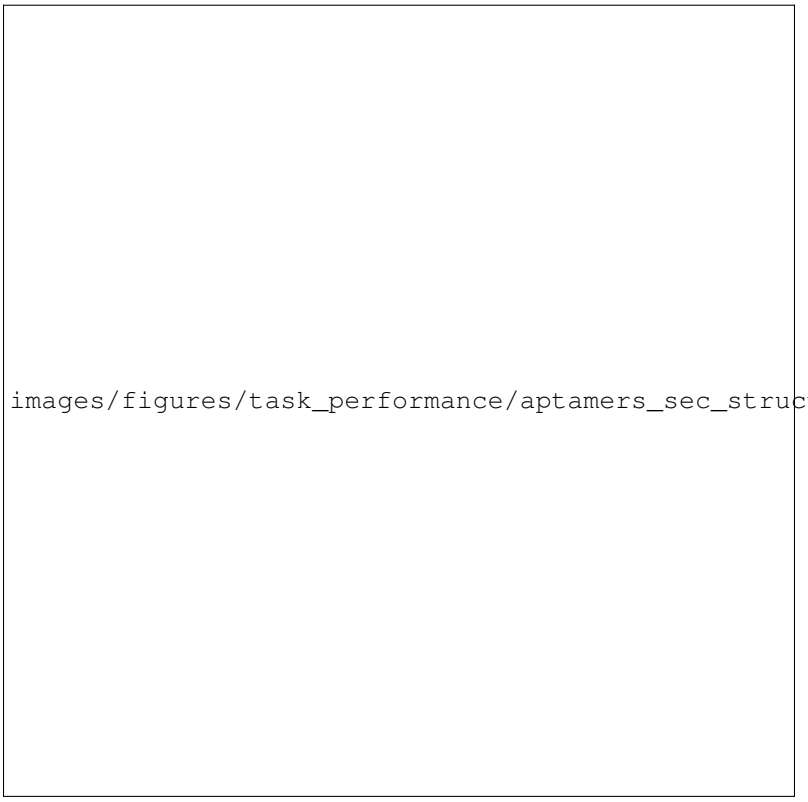

Figure C.15: Performance of different CRB configurations on the aptamer-protein interaction task.

C.3.2 RNA-RNA INTERACTION PREDICTION

In this downstream task, we aimed to evaluate the capability of CRB in predicting interaction patterns between different RNA chains. We formulated this as a classification problem, where the input consisted of two RNA sequences, and the output was a binary interaction matrix.

Dataset

We constructed our dataset using the ENCORI database (Li et al., 2014), which contains experimentally validated intermolecular RNA-RNA contacts. To ensure data reliability, we first filtered out the lowest-confidence entries corresponding to single-read interactions. From the remaining dataset, we sampled genomic sequences that included the entire identified interacting region, along with flanking genomic context of length 175–225 nucleotides, drawn from a uniform distribution. While overlapping genomic fragments were permitted in the training set, we explicitly removed them from the test set to ensure a fair evaluation.

The objective was to predict the base-pairing (interaction) matrix between the two sequences, capturing both non-continuous pairings and non-canonical base pairs, with a minimum requirement of nine base pairs forming the complementary region.

Model Architecture and Training

We employed binary cross-entropy as the loss function and the F1-score as the primary evaluation metric. The task-specific module consisted of an outer concatenation of the embeddings of the two sequences, followed by a linear layer, a ResNet with 64 blocks, and a 2D convolutional layer.

This task proved particularly challenging, as the best-performing language models exhibited only modest improvements over a naïve baseline that labeled all samples as the majority class. The best-performing model, CRB, achieved an F1-score of 0.542, statistically outperforming RiNALMo, which obtained 0.532. Both models significantly surpassed the baseline performances of LSTM and CNN architectures. The results are summarized in Table 4.

Interestingly, the highest F1-score was obtained by freezing the foundational model weights and training only the task-specific module. In contrast, unfreezing even a single transformer layer led to overfitting on the training set and a corresponding drop in validation performance.

Table 4: F1 performance in the ENCORI derived dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Performance
LSTM	0.406 ± 0.003
CNN	0.458 ± 0.008
RiNALMo	0.532 ± 0.015
ChaRNABERT	$\mathbf{0.542 \pm 0.016}$

C.3.3 RNA-RNA-Binding Protein interaction prediction

We designed this downstream task to asses the ability of CRB to capture patterns related to rna-protein interaction. We framed this task as a multiclass classification where the models had to detect interactions among 5 RNA-binding proteins. More details about the dataset and the implementation of the task can be found at section C.3.3.

Dataset

The CLIP database identifies experimentally validated protein-RNA interactions from several experimental sources (Yang et al., 2015; Van Nostrand et al., 2020; Zhao et al., 2022). Here, we used it to generate a dataset of pairs of genomic-context RNA sequences that share a binding site for either identical or different RNA-binding proteins (RBPs), forcing the predictive model to look for possibly degenerate motifs that repeat in both sequences. To keep the difficulty of the task manageable, we selected five proteins with distinct and sufficiently different position weight matrices and abundant hits: CSTF2T, HNRNPM, KHSRP, SF3B1 and U2AF2, and chose the high-resolution eCLIP dataset as a reference. Since between any two protein-specific datasets ca. 2% of binding sites were found to overlap, we filtered the data to only extract genomic-context windows with a unique RBP binding site. Here, each sequence length was drawn from the uniform distribution between 200 and 250 nucleotides.

Model Architecture and Training

We approached this task as a single-label, multi-class classification problem, mapping each sequence to one of five RNA-interacting proteins. The task-specific head consisted of a simple linear layer that mapped the class token to a tensor of size five.

All CRB configurations achieved F1 scores above 80, highlighting the model's effectiveness in capturing sequence patterns associated with recognition by these proteins. Figure C.16 presents boxplots for various CRB configurations, grouped by model size and dataset type used in pretraining. We observe that there are statistically significant differences between non-coding RNA and coding and non-coding RNA datasets for the sizes of 33 and 50 million parameters. This shows that the expansion of the pretraining dataset with coding-RNA sequences is detrimental for performance in

this downstream task. When comparing model configurations with different sizes, the only statistically significant difference appears between the 33M model pretrained with the non-coding RNA dataset and the 50M model trained with the coding and non-coding RNA dataset. With the amount of data available, it is not possible to determine how model size affects models pretrained with the same dataset. However, the general tendency indicates that increasing model size is also detrimental to performance in this downstream task.

Table 5 shows the performance of CRB along with other reference models in our study. Comparing CRB with other models, we observe that it significantly outperforms LSTM and CNN baselines, highlighting the advantages of a more complex architecture for enhanced representation in this task. CRB also surpasses RiNALMo by a statistically significant margin, demonstrating the benefits of using GBST as the tokenizer, enabling an 8M parameter model to outperform a 650M model.

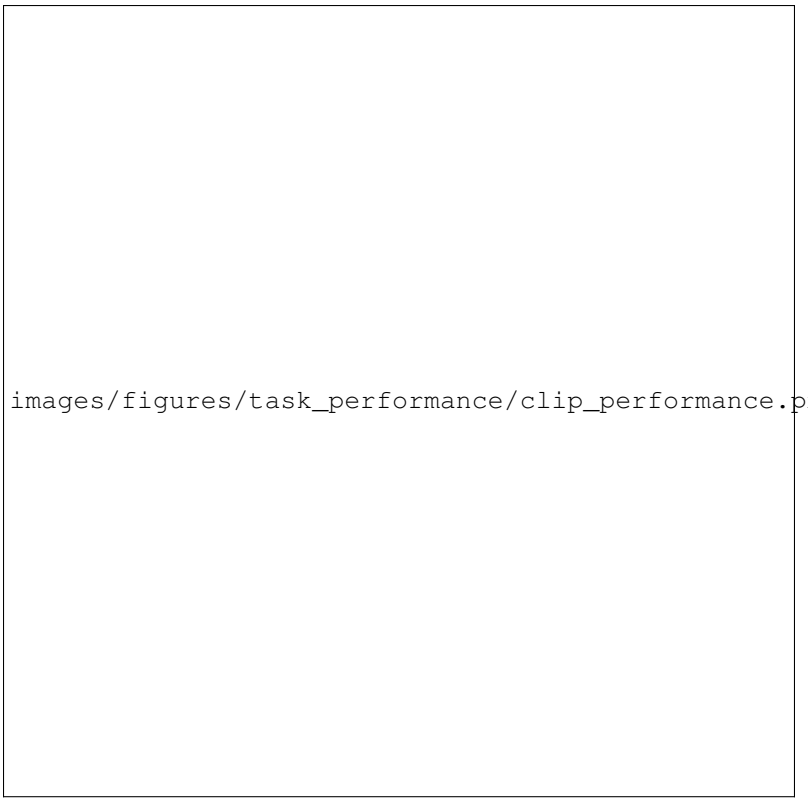

Figure C.16: Performance of different CRB configurations on the RNA-RBP interaction task.

Table 5: F1 performance in the CLIP derived dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Performance
LSTM	0.719 ± 0.006
CNN	0.770 ± 0.054
RiNALMo	0.831 ± 0.001
ChaRNABERT	$\mathbf{0.833 \pm 0.015}$

C.3.4 HALF LIFE PREDICTION

Dataset

A genome-wide database of mRNA decay rates in *A. thaliana* provided us with quantitative data on natural degradation in cytosolic environment for a large number of genetic sequences (Sorenson et al., 2018). The database includes four datasets corresponding to the wild-type organism and lines with partially or completely abolished function of XRN4, the main exonuclease responsible for mRNA degradation *A. thaliana*. We used this database to build a dataset of decay rates by fitting an exponential curve to each individual time series. To filter out low-quality or ambiguous data points, fitted values were kept only if all four standard uncertainties of the fitted decay rate were below 20% of the mean. Two sub-datasets were produced, one containing the coding sequence (CDS) only, and another featuring the full mRNA transcript.

Model Architecture and Training

The task specific module in this case consists only in linear layer mapping tensors from the embedding dimension of the foundational model to one.

Table 6: F1 performance in the *Arabidopsis Thaliana* mRNA half-life derived dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Genomic	Gene
LSTM	-0.756 $\pm$ 0.088	-0.509 $\pm$ 0.088
CNN	-0.612 $\pm$ 0.120	-0.645 $\pm$ 0.120
RiNALMo	-0.5906 $\pm$ 0.375	**-0.148 $\pm$ 0.375**
ChaRNABERT	**-0.378 $\pm$ 0.068**	-0.247 $\pm$ 0.103

The results from this task contrast with the performance seen on the degradation task in the BEACON Benchmark, particularly with the Open Vaccine dataset, which was designed with precise, experimentally controlled degradation conditions. We hypothesize that this structured setup in Open Vaccine allows RNA LM embeddings and downstream models to effectively capture degradation patterns. In contrast, the *A. thaliana* genome-wide dataset, collected under natural cytosolic conditions, introduces significant biological variability and noise, making it harder for the model to isolate meaningful signals. This underscores that even with rich embeddings, performance on degradation tasks relies heavily on the quality of gathered experimental data, as demonstrated in BEACON. By comparison, genome-wide datasets with inherent variability demand more sophisticated strategies to manage noise and improve predictive accuracy.

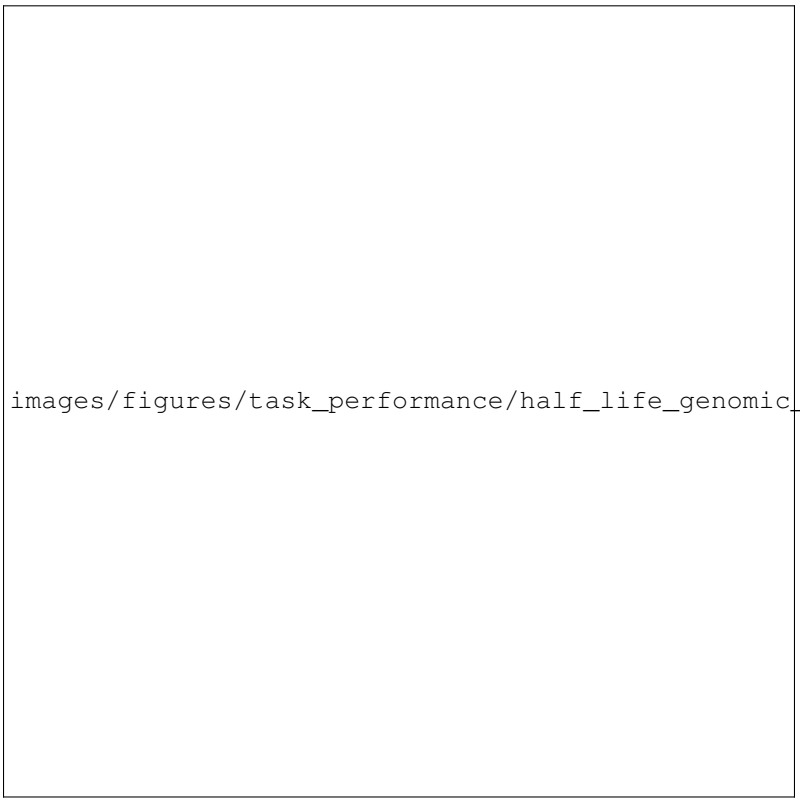

Figure C.17: Performance of different CRB configurations on the RNA half-life (genomic) prediction task.

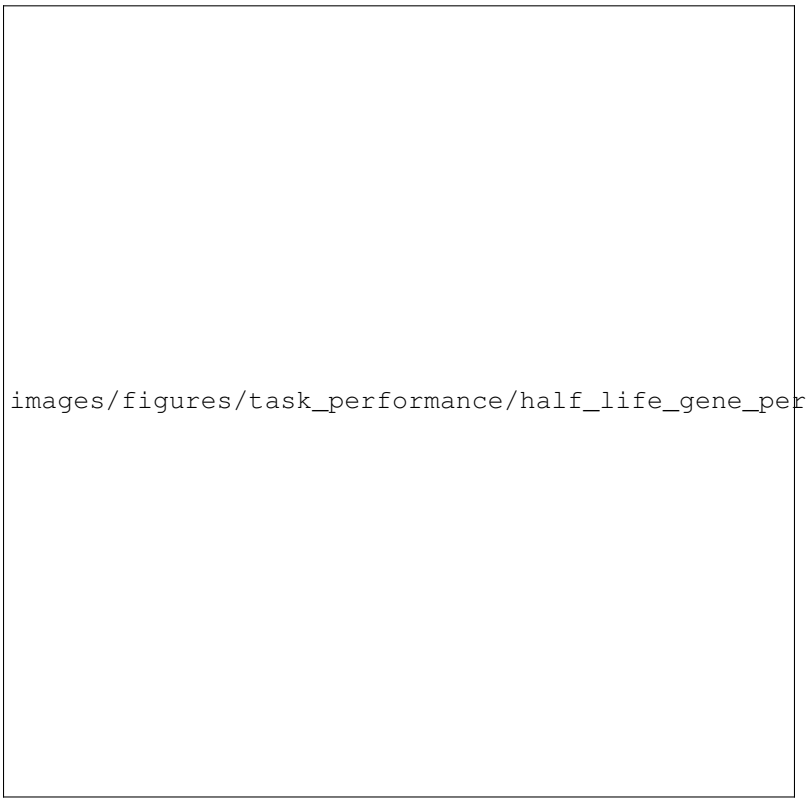

Figure C.18: Performance of different CRB configurations on the RNA half-life (gene) prediction task.

C.3.5 CATEGORICAL DISTANCE MAP

This task serves as the categorical counterpart to the distance map task featured in BEACON. It was designed to simplify the continuous nature of the original task, making the prediction process more manageable. By discretizing the output space, this approach aims to reduce the prediction error relative to the ground truth, ultimately improving the quality of distance predictions for downstream applications such as inverse folding models.

Dataset

The dataset was derived by discretizing the original BEACON dataset into categorical bins. The continuous range of distances from 0 to 1 was divided into 10 equal-length categories, transforming the regression problem into a classification task. This reformulation enables a more structured learning process, reducing sensitivity to minor variations in distance values.

Model Architecture and Training

The task-specific module used in this task was identical to that of the original distance map task, with the exception of the final layer, which was modified to output 10 categories instead of a single continuous value. Binary cross-entropy with logits was selected as the loss function, and the F1-score was used as the primary evaluation metric.

Table 7: F1 performance in the categorical distance map dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Performance
LSTM	0.009 ± 0.002
CNN	0.033 ± 0.025
RiNALMo	0.301 ± 0.013
ChaRNABERT	$\mathbf{0.423 \pm 0.010}$

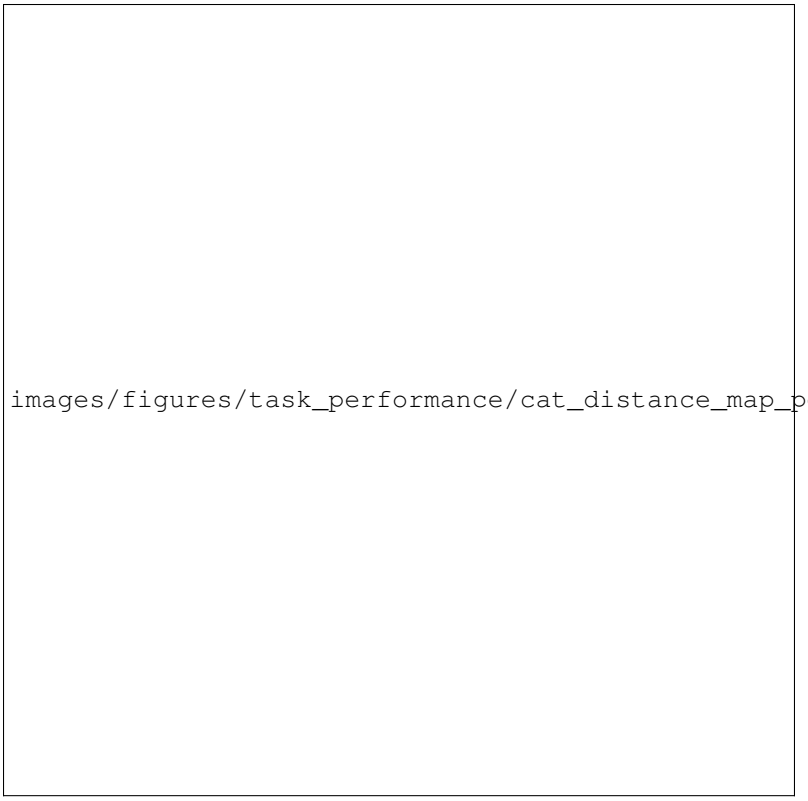

Figure C.19: Performance of different CRB configurations on the categorical distance map prediction task.

C.3.6 Prediction of human miRNA targets (DeepMirTar)

Building on the work of Wen et al. (2018), we evaluated the performance of CRB in predicting human microRNA (miRNA) targets. This task is of significant interest due to its implications in gene regulation, yet it remains challenging due to the limited understanding of the underlying mechanisms. Experimental validation is both time- and resource-intensive, making *in silico* prediction an essential approach.

Dataset

We used the dataset from the DeepMirTar study, which comprises experimentally validated miRNA-target interactions from mirMark (Hsu et al., 2014) and CLASH (Helwak et al., 2013). Following standard preprocessing steps, only interactions within the 3' UTR were considered. Negative samples were generated using a shuffling-based strategy to disrupt seed complementarity (Hsu et al., 2014). Additionally, a PAR-CLIP-derived independent evaluation set was used for further validation. For detailed dataset construction, we refer to Wen et al. (2018).

Model Architecture and Training

The task-specific prediction head utilized the CLS token from CRB's sequence representation, followed by a linear layer mapping embeddings to a single output. To improve efficiency, sequences were batched by length to minimize padding. The loss function was binary cross-entropy with logits, and the primary evaluation metric was the F1-score.

Results and Analysis

Both CRB and RiNALMo significantly outperformed baseline models and the original DeepMirTar model. Although CRB achieved a slightly higher average performance than RiNALMo, the difference was not statistically significant. Both models exhibited near-perfect performance, demonstrating the capability of foundational models to predict miRNA-mRNA interactions with high accuracy.

Table 8: F1 performance in the miRNA target dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Performance
LSTM	0.333 ± 0.002
CNN	0.516 ± 0.0342
DeepMirTar	0.9348 ± 0.037
RiNALMo	$\mathbf{0.980 \pm 0.010}$
ChaRNABERT	$\mathbf{0.990 \pm 0.002}$

Figure C.20: Performance of different CRB configurations on the miRNA target prediction prediction task.

C.3.7 Genomic segment annotation

This task was designed to assess whether CRB can accurately classify different functional elements within a genomic sequence. The ability to automatically annotate genomic data at scale is crucial for

identifying genes and regulatory regions, enabling large-scale genomic screening and accelerating discoveries in genetics and molecular biology. Furthermore, accurate functional annotation is essential for understanding genome organization, disease-associated mutations, and the mechanisms underlying gene expression regulation.

Dataset

The dataset consists of genomic sequences of up to approximately 8,000 bases, each paired with a corresponding class assignment string of the same length. Each sequence begins with a short upstream intergenic region, followed by a structured sequence that includes a 5'-UTR, the first exon, the first intron, the second exon, and subsequent exon-intron pairs. The class assignment labels are as follows: 0 for intergenic regions, 1 for untranslated regions (UTRs) and coding sequences (CDS), and 2 and 3 for exons and introns occurring after the initial UTR/CDS. Notably, all sequences in the dataset contain a single 5'-UTR per parent gene, and at least one coding region (label 2, 3, etc.) begins with the canonical start codon, AUG.

Model Architecture and Training

This task was framed as a residue-level classification problem. The task-specific module consisted of a simple linear layer that mapped each residue embedding to a four-dimensional output tensor, corresponding to the number of classes. To optimize computational efficiency, sequences were grouped into batches based on length, minimizing the number of padding tokens processed during training and improving throughput.

In this downstream task, CRB significantly outperforms both the baselines and RiNALMo, highlighting the impact of subword tokenization in segment labeling. Notably, this is the second task where CRB demonstrates a clear advantage, suggesting that its architectural choices play a crucial role in enhancing performance. This improvement may stem from the ability of GBST to enhance representation power in residue-level tasks, enabling a 33-million-parameter model to surpass a 600-million-parameter counterpart.

Table 9: F1 performance in the genomic classification dataset. **Bold** measures indicate the best model for a dataset under a t-test with p-value < 0.05.

Model	Performance
LSTM	0.665 ± 0.010
CNN	0.470 ± 0.000
RiNALMo	0.709 ± 0.026
ChaRNABERT	$\mathbf{0.866 \pm 0.013}$

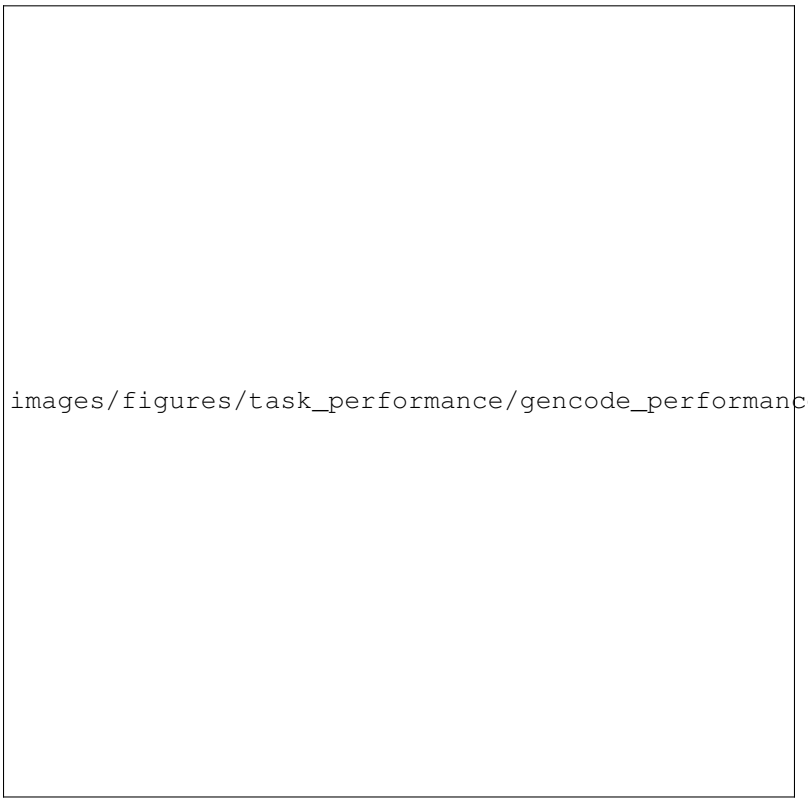

Figure C.21: Performance of different CRB configurations on the genomic segment annotation task.

C.3.8 PROTEIN-RNA BINDING AFFINITY PREDICTION

In this downstream task, we evaluated CRB's ability to predict binding affinity in protein-RNA complexes when mixing RNA LM embedding information and structural learning from complexes. We leverage the CoPRA (Complex structure for Protein-RNA binding Affinity prediction) framework (Han et al., 2025), which integrates protein embeddings, RNA embeddings, and structural information to predict complex-level and node-level interactions. CoPRA enhances interaction understanding by leveraging cross-modal representations and a bi-scope training strategy (combining pre-trained language models with a Co-Former architecture). Additionally, CoPRA introduces the largest protein-RNA binding affinity dataset to date for benchmarking, using 30k protein-RNA complexes for pretraining and 310 protein-RNA complexes with experimental binding affinity data for evaluation.

CoPRA performs competitively in binding affinity prediction by using ESM for protein embeddings, RiNALMo for RNA embeddings, and structural information extracted from protein-RNA complexes, integrated via the Co-Former model. During CoPRA pretraining, proteins and RNA embeddings are frozen, while the Co-Former module is trained.

For comparison, we adopted CoPRA's pre-training and evaluation datasets but substituted RiNALMo (650M) embeddings with CRB (33M) embeddings. We pre-trained on a slightly smaller dataset and obtained comparable results during testing in PRA310 (Table 10) .

Relating to the CoPRA-CRB performance, it yielded slightly lower results, this might indicate that CRB is not able to generalize as well as RiNALMo but the difference in distributions compared to CoPRA-RiNALMo was not statistically significant with p-value > 0.05. Considering that CRB is a much smaller model in terms of parameters and that RiNALMo introduces structural information during training (which CoPRA also introduces), achieving comparable results with no statisticaly significant difference using a model trained only with sequence data is noteworthy.

Analyzing the ablation study of CoPRA-RiNALMo, we observe that removing pre-training results in a decline in performance across all metrics. However, statistical tests (p-value > 0.05) indicate

that the difference between CoPRA-RiNALMo (- Pre-train) and CoPRA-CRB, as well as between CoPRA-RiNALMo and CoPRA-RiNALMo (- Pre-train), is not statistically significant. This suggests that neither pre-training nor RiNALMo embeddings provide a meaningful improvement in predicting protein-RNA binding affinity. Despite RiNALMo incorporating structural information during training, the observed differences do not translate into statistically significant performance gains, questioning the added value of these embeddings for this specific task.

Table 10: Performance comparison of CoPRA models across various metrics.

Model	**Pearson** ↑	**Spearman** ↑	**MAE** ↓	**RMSE** ↓
CoPRA-RiNALMo	0.580 ± 0.033	0.589 ± 0.045	1.129 ± 0.123	1.391 ± 0.142
CoPRA-CRB	0.436 ± 0.110	0.458 ± 0.141	1.231 ± 0.241	1.509 ± 0.259

C.4 Impact of Pretraining Configuration on Downstream Tasks

In this section, we examine how pretraining configuration affects performance across the 23 downstream tasks featured in this study. Specifically, we analyze the impact of token recycling, dataset composition, and masking protocol on model performance. For each pretraining variable, we define a baseline configuration and compare it against models trained with alternative configurations. For every task, we identify the best-performing model trained with the baseline setting and the best-performing model trained with any alternative setting. Performance scores of the alternative models are then expressed relative to their respective baseline models.

Statistical significance is assessed by testing for differences in means between the performance distributions of models trained with the baseline and alternative configurations.

C.4.1 Impact of Token Recycling on Performance

For token recycling, we consider models initialized from the first-epoch checkpoint of pretraining as the baseline. Across all 23 downstream tasks, we find that in 11 cases, the best-performing model was trained from the first-epoch checkpoint. In comparison, checkpoints from epochs 2, 3, and 4 yielded the highest scores in 6, 2, and 4 tasks, respectively. Figure C.22 shows the distribution of performance scores, normalized relative to models trained with the default token recycling configuration.

Extending pretraining beyond the first epoch does not significantly degrade performance and, in certain cases, leads to measurable improvements. However, statistically significant differences are observed in only two tasks. In secondary structure prediction, models trained from the fourth epoch outperform those trained from the first epoch by approximately 5%. In ribosome loading prediction, the best-performing model from the third epoch performs slightly worse than the one trained from the first epoch.

These findings suggest that while pretraining CRB for only one epoch serves as a strong default strategy, additional pretraining can enhance performance in specific tasks. Given that the potential benefits outweigh the risks of performance degradation, extending pretraining may be advisable depending on the nature of the task.

C.4.2 Impact of dataset composition on model performance

Our second analysis revolved around determining the best composition of the pretraining dataset. We pretrained CRB models using two dataset configurations: one composed exclusively of non-coding RNA sequences and another incorporating both coding and non-coding RNA sequences. For this analysis, we selected the non-coding RNA dataset as the baseline configuration and the coding + non-coding RNA dataset as the alternative. Out of the 23 tasks in 14 the best model used the non-coding dataset and 9 used the coding and non-coding dataset.

Figure C.23 presents the scores expressed relative to the baseline configuration. We find that only four tasks exhibit statistically significant differences compared to the baseline. Among these, only two show a substantial deviation from the mean score of the baseline configuration, the BEACON implementation of the splicing site prediction and secondary structure prediction.

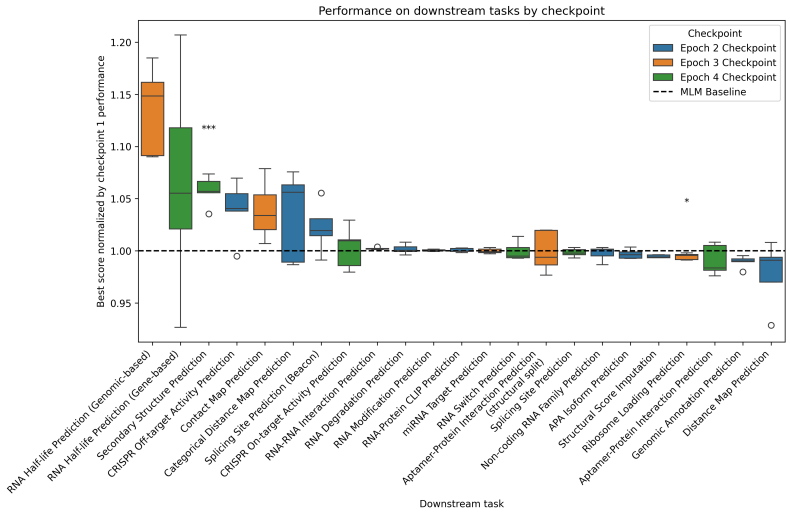

Figure C.22: Best performance achieved by CRB relative to the checkpoint 1 for each downstream task.

Although not statistically significant, the overall trend in Figure C.23 suggests that incorporating coding RNA into the pretraining dataset generally leads to a slight decrease in performance across downstream tasks. However, the presence of coding RNA in the sequences of certain downstream tasks may justify its inclusion in pretraining. This is particularly evident in splicing site prediction and ribosome loading prediction, two tasks that show significant improvements over the baseline configuration and are inherently linked to coding RNA.

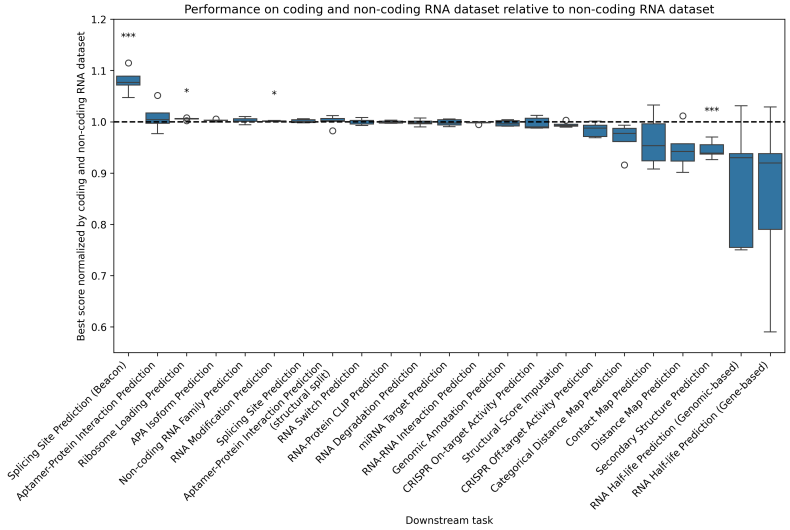

Figure C.23: Best performance of CRB using the non-coding and coding RNA dataset relative to the performance using only the non-coding RNA dataset.

D FLOPS FORWARD PASS COMPUTATION

We follow the protocol of (Hoffmann et al., 2022) Embedding matrices are counted in both FLOPS and parameter counts, while non-linearities, biases and layer normalizations are omitted. Due to the

small differences in both FLOPs and parameter counts of GBST and non-GBST embeddings, they were omitted from the scaling analysis.

Table 11: Forward pass FLOPS computation. The backwards pass is assumed to have twice the amount of FLOPS as the forward pass.

Operation	**FLOPs**
GBST	
Embedding layer	$2 \times$ seq_len $\times$ vocab_size $\times$ d_model
GBST Convolutions	$2 \times$ (max_blocksize^2 $\times$ d_model + d_model^2)
GBST Scoring	$2 \times$ seq_len $\times$ d_model
Attention Layer	
KQV projections	$2 \times 3 \times$ seq_len $\times$ d_model $\times$ (key_size $\times$ num_heads)
Key @ Query	$2 \times$ seq_len $\times$ seq_len $\times$ (key_size $\times$ num_heads)
Softmax	$3 \times$ num_heads $\times$ seq_len $\times$ seq_len
Softmax @ Query reductions	$2 \times$ seq_len $\times$ seq_len $\times$ (key_size $\times$ num_heads)
Output projection	$2 \times$ seq_len $\times$ d_model $\times$ (key_size $\times$ num_heads)
FFN Layer	$2 \times$ seq_len $\times$ (d_model $\times$ ffw_size + d_model $\times$ ffw_size)
Total FLOPs	Embeddings + num_layers $\times$ (Attention Layer + FFN Layer)

E PARAMETER COUNT

We follow the parameter count schema of Kaplan et al. (2020), removing sub-leading terms such layer normalizations and biases.

Table 12: Parameter computation

Operation	**Parameters**
GBST	
Embedding layer	vocab_size x d_model
GBST Convolutions	d_model x max_blocksize + d_model^2
GBST Scoring	d_model
Attention Layer	
KQV projections	3 x d_model x (key_size x num_heads)
Output projection	d_model x (key_size x num_heads))
FFN Layer	2 x (d_model x ffw_size)
Linear Language Head	d_model x vocab_size
Total Parameters	Embeddings + num_layers x (Attention Layer + FFN Layer) + RoBERTa Head

F SCALING LAW

Following the methodology outlined in (Hoffmann et al., 2022), we estimated the parameters (A, B, E, α, β) of the proposed scaling law:

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}, \tag{16}$$

To achieve this, we applied the optimization procedure recommended by (Hoffmann et al., 2022), minimizing the Huber loss (Huber, 1992) to account for the difference between the predicted and

observed logarithmic losses. This was done using the L-BFGS algorithm (Nocedal, 1980), which is well-suited for optimizing smooth, differentiable functions.

$$\min_{A,B,E,\alpha,\beta} \sum_i \text{Huber}\left(\log L(N_i, D_i) - \log L_{\text{observed},i}\right), \tag{17}$$

To solve this optimization problem, we first conducted a grid search to explore a range of initial parameter values. The results from the grid search were then refined using L-BFGS, which efficiently minimized the function.

For the exponents α and β, which define how the loss scales with model size and the number of tokens, we used the following relationships derived from the scaling law:

$$a = \frac{\beta}{\alpha + \beta}, \quad b = \frac{\alpha}{\alpha + \beta},$$

where a and b are constants determined by fitting the parametric model to the observed data.

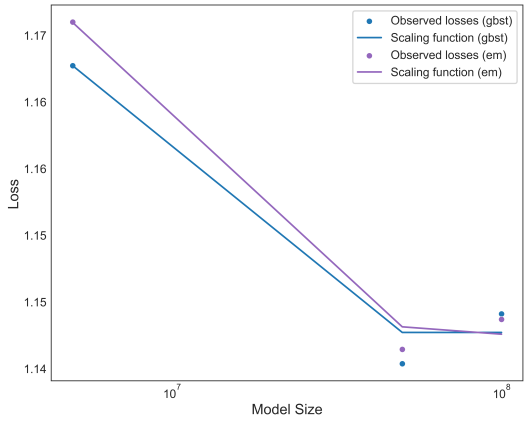

Figure F.24: Parametric Loss function

G CNN, LSTM AND RiNALMo CONFIGURATION AND ARCHITECTURE

We describe the architectures and parameters used in the downstream tasks for the CNN, LSTM and RiNALMo models. All configurations employ the Adam optimizer and follow the strategies described in 3.2. Heads used in the downstreams tasks are the same as the ones used in ChaRNABERT.

CNN architecture:

- **Convolutional Layer 1**: 1D convolution with 64 filters, kernel size of 25, padding of 1, and stride of 5. Includes *batch normalization*, followed by ReLU activation and *max pooling* (filter size of 2, stride of 1).
- **Convolutional Layer 2**: 1D convolution with 128 filters, kernel size of 3, padding of 1, and stride of 1. Includes *batch normalization*, followed by ReLU activation and *max pooling* (filter size of 2, stride of 1).
- **Global Average Pooling**: adaptive pooling to reduce the output dimension to 1 per channel.
- **Fully Connected Layer**: 128 neurons with ReLU activation, followed by dropout at a rate of 50%.

LSTM architecture:

- **LSTM Layer**: bidirectional LSTM with a hidden size of 128, and 2 layers. Configured with batch_first as True.

Table 13: Parameters used for each downstream task in the CNN model

Parameter	CLIP	Aptamers
Learning Rate (lr)	1×10^{-3}	1×10^{-3}
Batch Size	128	128

- **Dropout Layer**: with a rate of 50%
- **Layer Normalization**: with an output dimension of 128 * 2.
- **Fully Connected Layer**: reduces the output to 128 neurons with a linear transformation.

Table 14: Parameters used for each downstream task in the LSTM model

Parameter	CLIP	Aptamers
Learning Rate (lr)	1×10^{-4}	1×10^{-3}
Batch Size	256	64

RiNALMo was used in its pretrained version. For all downstream tasks, except for the ones already done in (Penić et al., 2024), we followed the same strategies applied in ChaRNABERT and described in 3.2.

Table 15: Parameters used for each downstream task in RiNALMo

Parameter	CLIP	Aptamers	CMP	DMP
Learning Rate (lr)	1×10^{-5}	1×10^{-5}	1×10^{-6}	1×10^{-5}
Batch Size	16	8	4	4

Parameter	CRI-On	CRI-Off	VDP	SSI
Learning Rate (lr)	1×10^{-5}	1×10^{-6}	1×10^{-5}	1×10^{-6}
Batch Size	1024	1024	16	128

Parameter	APA	Modif	NcRNA	PRS
Learning Rate (lr)	1×10^{-5}	1×10^{-6}	1×10^{-5}	1×10^{-5}
Batch Size	128	256	32	128

H EXPERIMENTS

In this section, we show several analyses on the factors that influence the performance of ChaRNABERT in order to identify the settings that provide the best performance of the model as well as to identify the optimal trade-off between performance and computational cost.

Advances in large language models (LLMs) have been driven by scaling up parameters, enhancing their applications in natural language processing (NLP), as detailed by the studies of (Kaplan et al., 2020) and (Hoffmann et al., 2022). While scaling studies have also progressed in fields like pLMs (Serrano et al., 2024), with research investigating model size effects, a comprehensive analysis of LLM scaling applied to RNA remains underexplored. Works such as RiNALMo (Penić et al., 2024) have examined performance differences under variations of the parameter count, yet a detailed investigation of RNA-specific scaling laws is still absent. This analysis aims to bridge that gap, using ChaRNABERT as an initial model to guide future large-scale RNA LLM research.

First, we explore how various learning rates and context window sizes affect ChaRNABERT's performance across model sizes, aiming to understand the impact of key hyperparameters. Next, we assess the model's efficiency with datasets of different sizes to evaluate the effects of data scaling.

Finally, we analyze computational efficiency by measuring the impact of increased floating-point operations (FLOPs) on model improvements. Throughout, we follow (Hoffmann et al., 2022)'s scaling principles and compare tokenization strategies, highlighting the performance gains of using GBST over embeddings alone.

H.1 Impact of learning rate and context window

We analyzed the impact of using different learning rates and context window sizes on three models of different sizes: 5 million, 50 million, and 100 million parameters, using a dataset composed of 31 million non-coding sequences extracted from RNACentral (Consortium, 2020).

For each model size, three learning rates were tested, selected on the basis of the number of parameters in the model. Specifically, for the 5M and 50M parameter models, the learning rates tested were 5e-4, 1e-4, and 5e-5. For the 100M parameter model, the learning rates tested were 1e-4, 5e-5, and 1e-5. We choose different learning rates per model size in order to avoid training instabilities. The aforementioned tests permitted an evaluation of the influence of the learning rate on convergence.

In general, it was observed that in smaller models (Table 16), such as the 5 million parameter model, higher learning rates, 5e-4 or 1e-4, achieved a slightly lower loss compared to lower rates like 5e-5. Nevertheless, these gains, while present, were not substantial. In the middle models, such as the 50M parameter model (Table 17), learning rates such as 1e-4 or 5e-5 achieved lower losses compared to higher learning rates like 5e-4 which resulted in high instability. For larger models, such as 100M model (Table 18), higher learning rates, 1e-4 and 5e-5, similarly resulted in a lower loss compared to 1e-5. However, upon examining the convergence curves, lower learning rates helped to avoid instability in mid and large-sized models.

Table 16: Comparison of Final Loss by Learning Rate - Model Size 5M

Model — Learning Rate	**5e-4**	**1e-4**	**5e-5**
EM	0.508	0.580	0.639
GBST	0.498	0.537	0.581

Table 17: Comparison of Final Loss by Learning Rate - Model Size 50M

Model — Learning Rate	**5e-4**	**1e-4**	**5e-5**
EM	3.516	0.437	0.469
GBST	19.52	0.434	0.450

Table 18: Comparison of Final Loss by Learning Rate - Model Size 100M

Model — Learning Rate	**1e-4**	**5e-5**	**1e-5**
EM	0.446	0.465	0.640
GBST	0.454	0.460	0.577

To explore how context window size affects performance, tests were conducted with window sizes of 8192, 4096, 2048 (Figure H.25). For RNA language models, the context window is especially relevant due to the biological nature of the sequences processed by these models. Interactions, such as base pairing and secondary structures, can be scattered throughout the sequence, making it essential that the model is able to capture a context wide enough to identify relevant patterns.

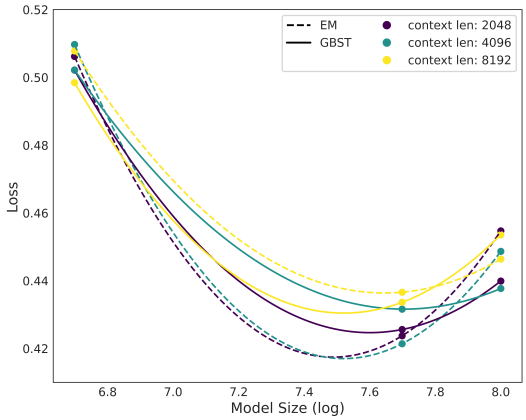

Figure H.25: Performance comparison with different context windows. We present the final exponential moving average (EMA) loss results for two tokenisation methods, EM (dashed lines) and GBST (solid lines), evaluated in different model size configurations, which are plotted in logarithmic scale on the X-axis. Each colour indicates a different context window length (2048, 4096 and 8192).

The results in Figure H.25, reveal a consistent trend for both tokenization strategies: single nucleotide embedding (EM) and GBST. The curves generally follow a U-shaped pattern, indicating that for each sequence size there is an optimal point at which the model reaches its minimum loss. After this point, further increase in model size result in diminishing returns. This saturation point appears to be around 30 to 50 million parameters, beyond which there is no significant improvement in performance, and the loss tends to increase slightly.

One notable aspect of these results is that, despite the existence of some differences between the context window sizes, these are not pronounced enough to conclude that one size is clearly superior to another in terms of performance. Nevertheless, is noteworthy that, in models with EM tokenization, smaller window sizes (2048 and 4096) tend to achieve lower final losses compared to larger windows such as 8192, suggesting that a smaller window might be more effective in certain cases. This indicates that, although the size of the context window has some impact, this seems to be more limited, especially when GBST tokenization is used.

H.2 VARYING TOKEN COUNTS

We also aimed to identify the impact of token count in model performance. We retained the three model sizes, 5M, 50M and 100M parameters, and generated datasets of varying sizes: 15M, 66M, 100M, and 150M sequences which by the average number of tokens per sequence correspond to 2.48B, 10.9B, 16.5B, and 24.8B tokens respectively.

Since the amount of non-coding sequences is limited, we included both coding and non-coding, extracted from the MARS (Chen et al., 2024a) sequence database. The sequences were randomly selected. In Figure H.26, the loss obtained per model size and per dataset for both tokenization methods, GBST and EM, is shown.

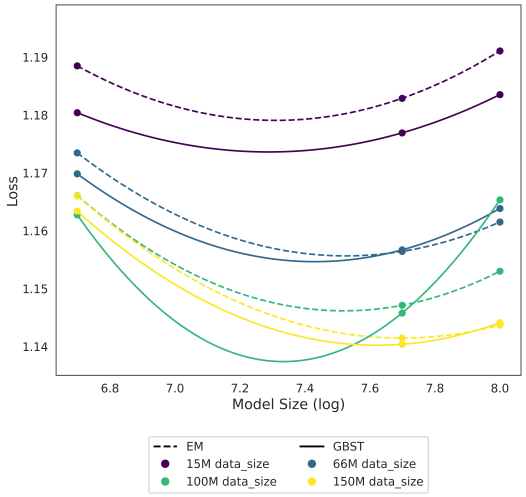

Figure H.26: Performance comparison across data sizes. The figure displays the final EMA loss for two tokenization methods, EM (dashed lines) and GBST (solid lines), evaluated across various model sizes (plotted in logarithmic scale on the X-axis). Different colors represent distinct data sizes (15M, 66M, 100M, and 150M of sequences).

From the results displayed in Figure H.26, it can be observed that the size of the dataset has a visible but not substantial impact on the overall performance of the model. In both GBST and EM tokenization experiments, the losses decreased slightly as the dataset size increased, but these improvements were not substantial enough to drastically improve the performance of the models. Furthermore, for both tokenization techniques, a clear U-shaped trend is still observed in the plots, suggesting that model performance can saturate at around 30M to 50M parameters, regardless of the size of the dataset used.

Based on the analysis conducted, it is clear that while learning rate and context window size have some impact on performance, the factor that most significantly affects ChaRNABERT's performance is the size of the model. Larger models tend to perform better up to a certain point, after which performance gains become marginal, and in some cases, loss even increases slightly. In the following section, we will further explore this by conducting a detailed study on model scaling, with the aim of understanding how the number of parameters influences performance and identifying the optimal scaling strategies for ChaRNABERT.

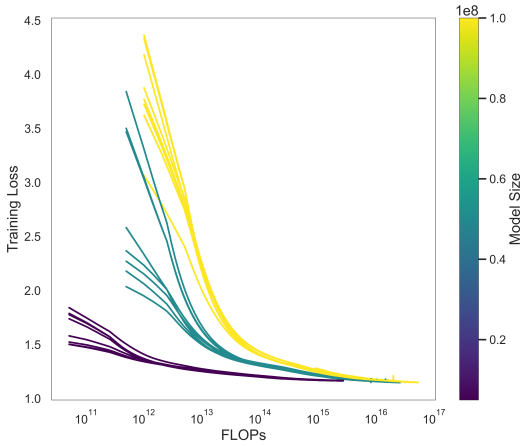

Figure H.27: FLOPs vs. training loss across model sizes. We show the relationship between computational cost (FLOPs, on a logarithmic scale, X-axis) and training loss (Y-axis) for models of varying sizes.

H.3 Increasing model size

We follow Hoffmann et al. (2022) to fit a parametric loss function. Using the data collected during the experimental phases, we fit the power laws to establish the relationships $N_{opt} \propto C^a$ and $D_{opt} \propto C^b$, where N is the model size, D is the number of tokens, and C represents the computational budget in FLOPs[1]. The exponents a and b were determined based on the fitted parameters, of the scaling law:

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} \tag{18}$$

Here, N is the number of model parameters, D is the size of the data set (in tokens), and E captures the natural entropy of the text (ideal loss). The terms with A and B reflect the deviation of the model from the ideal loss, due to the limited size of the model and data. The exponents α and β determine the impact of model and dataset size on the loss[2].

We fit the expression in Eq. 18 following (Hoffmann et al., 2022)[3]. In particular, we utilized the most optimal results obtained for each model size from the experiment described in Section H.2. This analysis was intended to capture the effects of both model size and dataset size on the model performance.

We found that for both GBST and EM, the final loss decreases predictably as the model parameters increase (Figure F.24), following a general trend of improvement with larger models. However, in both cases, improvements plateau or even cease to be significant once a certain parameter threshold is exceeded, particularly around 30 to 50 million parameters.

From the obtained parameters we derived the power-law exponents for model size and token count as a function of compute $N_{opt} \propto C^{0.2279}$ and $D_{opt} \propto C^{0.7720}$. A key finding is that the optimal model size scales sublinearly with the compute budget. Specifically, the relationship $N_{opt} \propto C^{0.2279}$ indicates that the model size grows at a slower rate than the compute budget. In contrast, the optimal number of training tokens scales superlinearly with compute. The relationship $D_{opt} \propto C^{0.7720}$ shows that, as compute grows, the number of training tokens increases more rapidly than the model size.

We observe that model performance improves rapidly with increased compute at first, but after a certain threshold, approximately 10^{16} FLOPs, the improvements in training loss begin to plateau (Figure

[1]FLOPs computation is equivalent to the one defined in Hoffmann et al. (2022) and can be found at Appendix D.

[2]Parameter counts for our models are defined at Appendix E.

[3]Detailed procedure can be found at Appendix F.

H.27). This trend suggests that, while scaling up model size and compute provides significant early gains, we rapidly find a model size as a point where increasing parameter count yields diminishing returns.