EVOLUTIONARY CONSTRAINTS AS DATA: AUGMENTING RNA LANGUAGE MODELS WITH PHYLOGENETIC GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning RNA language models is often limited by scarce labelled data. Data augmentation helps, but nucleotide edits may change the biological properties being predicted, making the original labels unreliable. We introduce `PhyloAug`, a plug-in framework that uses phylogenetic information to guide RNA sequence augmentation. Relative evolutionary rates estimated from homologous sequences determine candidate editing sites, and a structure-conditioned RNA language model generates replacement nucleotides. The resulting variants are used to fine-tune existing models without changing their architectures or training objectives. Across seven RNA functional prediction benchmarks and multiple model backbones, `PhyloAug` improves performance over unaugmented fine-tuning in most evaluated settings. Matched augmentation and training budgets isolate the effects of phylogenetic site-selection from structure-conditioned perturbation. Distributional diagnostics show smaller sequence-distribution shifts than less constrained editing policies. Benefits vary across tasks and label budgets, highlighting both the value and limits of evolutionary guidance for improving RNA language model adaptation with limited annotations.

1 INTRODUCTION

Fine-tuning RNA language models (RNA-LMs) for functional prediction often depends on limited labelled data. Such annotations may require experimental measurements of RNA structure or function (Watters et al., 2016; Cao et al., 2024), motivating methods that make better use of existing examples. Data augmentation offers one such approach (Li et al., 2022; Trabucco et al., 2024), but its usefulness depends on what the edits change. A nucleotide substitution may alter a binding site, disrupt a structural interaction, or change molecular activity. Even a biologically plausible variant may have a different label from the original sequence. The challenge is to introduce useful sequence variation while limiting changes to the predicted property.

Existing genomic augmentation methods offer different ways to address this challenge. Random sequence perturbations can improve learning, but their effects depend on the task and the perturbation policy (Lee et al., 2024). `EvoAug` combines evolution-inspired transformations with subsequent training on the original data to reduce unwanted augmentation bias (Lee et al., 2023). For RNA structure prediction, `PriFold` uses covariation-based substitutions to generate structure-compatible variants (Yang et al., 2025a). These approaches motivate a more specific question: *can evolutionary evidence from a training sequence's homologs guide position-specific edits that improve RNA-LM adaptation?* Multiple sequence alignments (MSAs) describe observed variation, while phylogenetic models account for shared ancestry when estimating site-specific evolutionary rates (Yang, 2007). Such estimates provide a potential guide to editing, but their value for a particular prediction task must be established empirically.

We introduce `PhyloAug`, a plug-in augmentation framework for fine-tuning RNA-LMs. `PhyloAug` uses phylogenetic rate estimates to guide *where* to edit and a structure-conditioned RNA-LM to determine *how* to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

replace masked nucleotides. For each training sequence, it retrieves homologous sequences and uses their alignment and phylogenetic tree to estimate relative evolutionary rates at individual positions. It selects positions to mask from sites whose rates fall within a specified interval around the alignment-wide mean. The generator then fills these masked positions using the unmasked nucleotides and the secondary structure predicted from the original input sequence. We assign each generated variant the label of its source example and combine the variants with the original training data for fine-tuning. This incorporates evolutionary information through the training data without changing the target model's architecture or training objective.

We evaluate `PhyloAug` across seven functional RNA prediction benchmarks and multiple prediction backbones. Matched augmentation controls compare alternative site-selection policies under a common generator, while a factorial ablation examines the separate contributions of phylogenetic selection and explicit structure conditioning. Label-budget experiments assess how the benefits change with the number of labelled training examples. The results show improvements in many evaluated settings, with gains depending on the task, target model and augmentation budget. In addition, we complement these prediction experiments with kernel-based distributional diagnostics (Amin et al., 2025), which measure how augmentation changes sequence distributions. These diagnostics provide supporting evidence about the generated data; they do not establish that individual variants preserve their original labels.

Our contributions are threefold:

- **Phylogenetically guided RNA augmentation.** We introduce a reusable augmentation pipeline, combining homolog-derived relative-rate selection with structure-conditioned perturbation for downstream fine-tuning.
- **Analysis of the augmentation components.** We use matched site-selection controls and a factorial ablation to examine the contributions of phylogenetic guidance and explicit structure conditioning, revealing differences between prediction tasks.
- **Evidence on benefits and limitations.** We characterize performance across functional RNA tasks, target backbones and label budgets, using distributional diagnostics to complement downstream predictive utility.

2 Related Work

RNA and genomic data augmentation. Studies of point-mutation augmentation show that its effects depend on the mutation type and genomic context (Lee et al., 2024). `EvoAug` applies evolution-inspired DNA perturbations for regulatory prediction, followed by fine-tuning on the original sequences (Lee et al., 2023). `GPN-MSA` also uses conservation-guided nucleotide replacement during pretraining (Benegas et al., 2025). For RNA secondary-structure prediction, `PriFold` jointly substitutes annotated base pairs using an empirical pairing-frequency distribution (Yang et al., 2025a). These approaches establish precedents for using evolutionary or structural information in augmentation. `PhyloAug` combines phylogenetic rates estimated from each training sequence's homologs with structure-conditioned nucleotide generation to augment labelled data for downstream RNA prediction.

Evolutionary constraint and site selection. Tools such as `phastCons` and `phyloP` use phylogenetic models to assess sequence conservation and evolutionary rate variation (Hubisz et al., 2011). Their scores can inform augmentation policies, but choosing editing sites and replacement nucleotides requires rules. `PhyloAug` uses relative evolutionary rates normalized within each homolog alignment to select candidate editing sites. We treat this selection rule as an augmentation heuristic and evaluate its contribution to downstream prediction; the selected rate interval does not establish evolutionary neutrality or label preservation.

Evolution-aware biological sequence models. Evolutionary information also informs model inputs and training objectives. `MSA Transformer` (Rao et al., 2021), `RNA-MSM` (Zhang et al., 2024), and `GPN-MSA` (Benegas et al., 2025) learn from aligned homologous sequences. `PhyloGPN` incorporates an explicit model of nucleotide evolution into its training loss (Albors et al., 2025), while `CS-Fold` uses phylogenetic evidence of compensatory substitutions to guide RNA secondary-structure prediction (Zhou et al., 2025). Our augmentation pipeline introduces evolutionary information through additional fine-tuning examples without requiring changes to the target predictor's architecture or fine-tuning objective.

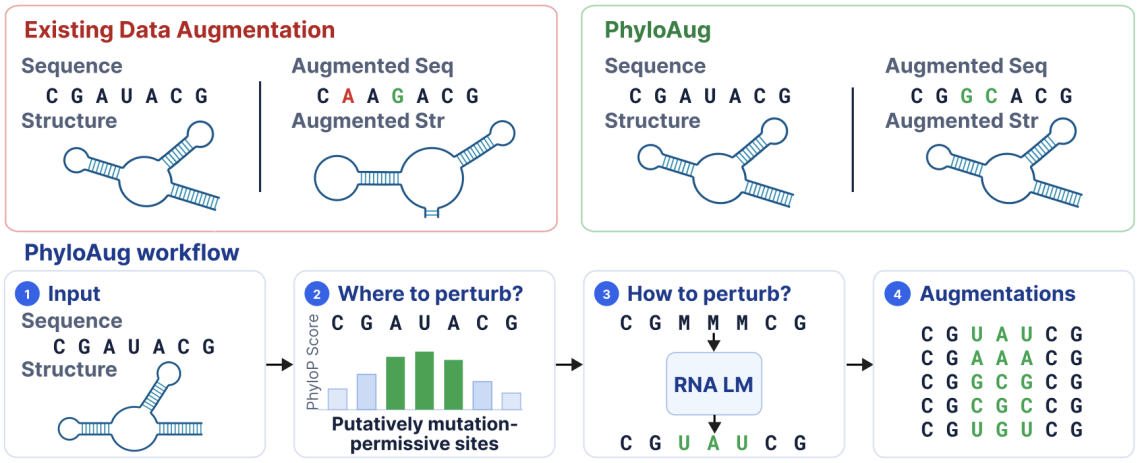


Figure 1: Traditional augmentation may disrupt implicit RNA functional and structural signals. `PhyloAug` mitigates this risk by restricting substitutions based on relative evolutionary rates and structural conditioning.

3 Proposed Method

`PhyloAug` augments training data for RNA-LM fine-tuning by combining phylogenetically guided site selection with structure-conditioned nucleotide generation. For each training sequence, it (1) estimates relative evolutionary rates from aligned homologs and their phylogeny, (2) masks sites within a specified rate interval, and (3) generates replacements conditioned on the remaining nucleotides and secondary structure predicted from the original sequence. This separates *where* to edit from *how* to generate replacements (Figure 1).

3.1 Inputs and Augmentation Policy

Let $\mathbf{x} = (x_1, \ldots, x_L) \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{U}\}^L$ denote a training RNA sequence. Its conditioning structure $\mathbf{s}^\star$ is a position-aligned dot-bracket representation predicted from the original, unmasked sequence using `ViennaRNA` (Lorenz et al., 2011) and held fixed across its augmented samples (Section 3.4).

For each sequence, `PhyloAug` retrieves a homolog set $\mathcal{H} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and estimates relative substitution rates from their alignment and phylogeny (Section 3.2). Let $\mathcal{V} \subseteq \{1, \ldots, L\}$ denote the query positions with available rate estimates, and let r_i denote the estimated rate at position i. We define the candidate editing sites as

$$\mathcal{C} = \{i \in \mathcal{V} : r_{\min} \le r_i \le r_{\max}\}. \tag{1}$$

For each sequence satisfying the homolog and mask-ratio requirements, we generate M augmented samples. For sample $j \in \{1, \ldots, M\}$, we select a mask set $\mathcal{E}^{(j)} \subseteq \mathcal{C}$ using the rules defined in Section 3.3, giving

$$\tilde{x}_i^{(j)} = \begin{cases} \texttt{[MASK]}, & i \in \mathcal{E}^{(j)}, \\ x_i, & \text{otherwise.} \end{cases} \tag{2}$$

The structure-conditioned generator then produces

$$\mathbf{x}'^{(j)} \sim Q_{\theta,\tau}\big(\cdot \mid \tilde{\mathbf{x}}^{(j)}, \mathbf{s}^\star\big), \qquad j \in \{1, \ldots, M\}. \tag{3}$$

where $Q_{\theta,\tau}$ denotes the sampling distribution obtained from the generator with parameters θ after probability thresholding at τ and renormalization (Section 3.4). Only masked positions are resampled; all other nucleotides are copied unchanged. Resampling may recover the original nucleotide, so the number of substitutions can be smaller than the number of masked positions. Original training examples are retained, including those ineligible for augmentation. Validation and test sequences are not augmented.

3.2 Homolog Retrieval and Site-wise Rate Estimation

For each training sequence $\mathbf{x}$, we search the `NCBI nt` database (Sayers et al., 2021) using the `megablast` task in `BLASTN` 2.16.0+ with an E-value threshold of 10^{-5}. We exclude hits with more than 95% identity to the query and apply the query-coverage filter specified in Section 4. We also remove hits matching validation or test sequences under the benchmark-specific exclusion rules in that section. These filters are applied before selecting homologs for alignment and rate estimation. If fewer than $N_{\min}$ hits remain, the original training example is retained without augmentation. Otherwise, we rank the remaining hits by decreasing `BLAST` bit score and retain at most $N_{\max}$ to form the homolog set $\mathcal{H}$.

We align the query and retained homologs, $\{\mathbf{x}\} \cup \mathcal{H}$, using `MAFFT` (Katoh and Standley, 2013) to obtain the multiple sequence alignment $\mathcal{A}$. We then infer a phylogenetic tree $\mathcal{T}$ from $\mathcal{A}$ using `FastTree` (Price et al., 2010). Given $\mathcal{A}$ and $\mathcal{T}$, we estimate site-wise relative substitution rates using `baseml` in `PAML` (Yang, 2007), with a nucleotide substitution model, discrete-gamma rate variation across sites, and empirical-Bayes rate estimation. Rates are normalized within each alignment to a mean of one. The substitution model, inference settings, and normalization procedure are specified in Appendix Q.

We map the estimated alignment-site rates back to query positions, omitting columns containing a gap in the query. Query positions with ambiguous nucleotides or unavailable rate estimates are excluded from editing; the remaining positions form the set $\mathcal{V}$ defined in Section 3.1. The mapped rate at position $i \in \mathcal{V}$ is denoted by r_i. The augmentation policy uses these rates and does not require ancestral sequence reconstruction.

The main protocol uses no RNA-family annotations. We examine an annotation-based family-consistency filter only in separately reported mechanistic analyses; it is not applied to the main functional benchmarks.

3.3 Relative-Rate Site Selection

By default, we select candidate editing sites with relative substitution rates between $r_{\min} = 0.8$ and $r_{\max} = 1.2$. On the normalized scale, this interval retains positions near the alignment-wide mean. The interval is a heuristic; it does not identify selectively neutral sites or guarantee downstream label preservation. The thresholds are fixed before downstream training and not selected using downstream validation performance.

The number of candidate sites varies across sequences. We therefore use fixed masking controls $0 < \rho_{\min} \leq \rho_{\max} \leq 1$, specified in Section 4. If $|\mathcal{C}|/L < \rho_{\min}$, we retain the original training example without generating augmentations. Otherwise, each augmented sample uses a mask set of size $k = \min\left\{|\mathcal{C}|, \lfloor \rho_{\max} L \rfloor\right\}$. For each sample j, we select k positions uniformly without replacement from $\mathcal{C}$ to form $\mathcal{E}^{(j)}$. Selection is performed independently across samples. When $k = |\mathcal{C}|$, all candidate sites are masked.

The lower threshold determines whether a sequence is eligible for augmentation, while the upper threshold caps the number of masked positions. These controls do not impose a minimum number of actual substitutions: the generator may resample the original nucleotides, leaving some or all masked positions unchanged.

3.4 Structure-Conditioned Infilling

For each training sequence $\mathbf{x}$, we predict its minimum-free-energy secondary structure using `ViennaRNA 2.7.2` (Lorenz et al., 2011) before masking. The resulting structure $\mathbf{s}^\star$ is represented in position-aligned dot-bracket notation, where dots denote unpaired nucleotides and matching parentheses denote paired positions. This structure is held fixed across the sequence's augmented samples. Downstream labels and annotated target structures are not used to construct the conditioning input. We use `OmniGenome` (Yang et al., 2025b) as the default structure-conditioned RNA-LM generator. For sample j, its input is $\tilde{\mathbf{x}}^{(j)}$ `[SEP]` $\mathbf{s}^\star$. Only sequence positions in $\mathcal{E}^{(j)}$ are masked; the remaining nucleotides and all structure tokens remain visible. A single forward pass provides replacement probabilities at all masked sequence positions.

For each masked position i, we restrict the generator's output to the nucleotide alphabet $\mathcal{B} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ and renormalize over these bases to obtain $q_i^{(j)}$. We retain nucleotides with probability at least $\tau = 0.05$, defining

$S_i^{(j)} = \{b \in \mathcal{B} : q_i^{(j)}(b) \geq \tau\}$. Re-normalizing over the retained nucleotides gives the sampling distribution

$$\bar{q}_i^{(j)}(b) = \begin{cases} \dfrac{q_i^{(j)}(b)}{\sum_{a \in S_i^{(j)}} q_i^{(j)}(a)}, & b \in S_i^{(j)}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

We independently sample a nucleotide at each masked position from $\bar{q}_i^{(j)}$ and leave unmasked positions unchanged. These steps define the augmentation distribution $Q_{\theta,\tau}$ introduced in Section 3.1. As the original nucleotide may be resampled, masked positions can remain unchanged. Structural conditioning guides the replacement probabilities but does not guarantee that the generated sequence adopts the conditioning structure.

Repeating this procedure produces M augmented samples per eligible training sequence. The choice of M and the comparison protocol are specified in Section 4, with sensitivity to M reported in Appendix H. We compare `ViennaRNA` conditioning with MSA-based `RNAalifold` in Appendix G and examine generator-target interactions through the cross-generator controls in Appendix B.

3.5 Training-Time Integration

We retain all original training examples $(\mathbf{x}, y)$. For each example that meets the homolog and masking requirements, we add M augmented samples $(\mathbf{x}'^{(j)}, y)$, $j \in \{1, \ldots, M\}$, to the training pool. Examples that do not meet these requirements remain in the pool without augmentation. We fine-tune the target RNA-LM on this pool using its existing architecture and task loss.

Reusing the original label assumes that the edits leave the task-relevant property sufficiently unchanged. Downstream labels and task-specific motif annotations are not used for site selection or nucleotide generation. We use structural and evolutionary diagnostics to characterize the augmented data and functional RNA benchmarks to assess their effect on downstream prediction under explicit label budgets. These evaluations do not directly verify label preservation for individual augmented samples. The baselines, training budgets, and hyperparameter-search protocol are specified in Section 4.

4 Experimental Setup

Benchmarks and splits. We evaluate seven RNA prediction tasks: RNA-binding protein interaction (`RBP`), N^6-methyladenosine sites (`m6A`), splice type (`STP`), CRISPR on-target activity (`CRISPR`), guide RNA efficiency (`gRNA`), and mRNA/miRNA subcellular localisation (`mRNA-SL`/`miRNA-SL`). We use official train/validation/test partitions where available (Ren et al., 2024; Wang et al., 2025; You et al., 2026); for `STP`, we construct gene- and species-disjoint splits from `U12DB` (Alioto, 2007). Only training examples are augmented. Homolog hits matching validation or test examples are excluded before rate estimation. Label-budget analyses use stratified $\{1\%, 5\%, 10\%, 20\%, 50\%, 100\%\}$ subsets, with validation and test sets held fixed. Split construction, subset sampling, and homolog-filtering details are provided in (Appendix D).

Target models. We evaluate seven RNA foundation models (FMs), two DNA-pretrained FMs (`HyenaDNA` (Nguyen et al., 2023) and `BiGamba` (Consens et al., 2026)), and three conventional architectures (MLP, CNN, and ResNet). All are evaluated on RNA tasks to assess dependence on the target backbone and its pretraining. Model details, parameter counts, and references are provided in Appendix C.

Augmentation settings and controls. We set $(N_{\min}, N_{\max}) = (5, 50)$ and $(\rho_{\min}, \rho_{\max}) = (0.05, 0.15)$, as defined in Section 3.2. We compare `PhyloAug` with five controls: (1) `NoAug` uses only original training examples; (2) `Duplicate` resamples unmodified examples to match the augmented token budget; (3) `Random` selects mask positions uniformly at the same masking ratio; (4) `MSA-only` selects sites using alignment variability without phylogenetic rate estimation; and (5) `Struct-Infill` uses structure-based site selection without evolutionary rates. Full definitions are in Appendix M. Except for generator comparisons, sequence-modifying policies share the generator, augmentation-count search space, and mask-ratio bounds.

Training and model selection. For augmentation policies, we select $M \in \{1, 2, 4, 6, 8, 10, 12\}$ on the validation split under the same search budget, which sets the number of generated samples per eligible training

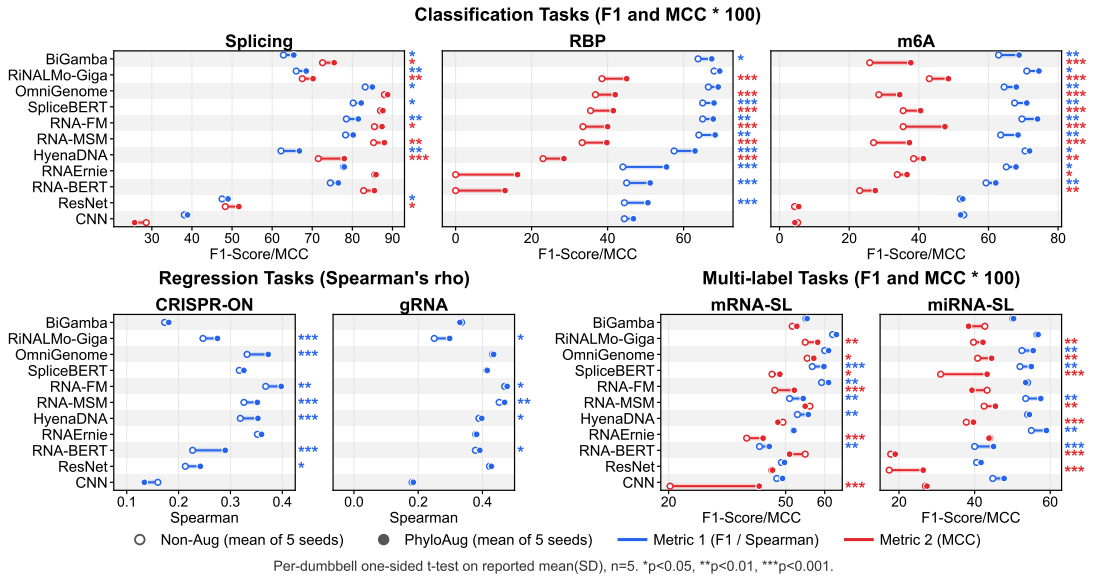


Figure 2: Functional evaluation on seven RNA tasks. Filled circles denote `PhyloAug`, and open circles `NoAug`. Values are means over five seeds. Lines connect the two conditions for each target model and metric. Comparisons follow the training-budget protocol in Section 4, with numerical results in Table 7 and Table 8.

example. Evolutionary thresholds $(r_{\min}, r_{\max})$ are fixed before training, not tuned on validation performance. For each task-model pair, all policies share the same optimization-step and token-budget limits, optimizer, batch size, learning-rate grid, scheduler, and early-stopping. Checkpoints are selected on validation score.

Metrics. We report $F1$ and Matthews correlation coefficient (MCC) for classification tasks and Spearman's ρ for regression, as mean and standard deviation over five seeds. Method comparisons use paired or Welch's t-test with false-discovery-rate adjustment within each figure. Numerical results are reported in Table 7.

5 Experiments

We evaluate whether `PhyloAug` improves RNA prediction across tasks and target models, using augmentation controls and component ablations to assess the contributions of phylogenetic site selection and structure conditioning. We also examine sensitivity to the perturbation generator, label budget, and augmentation count M, and complement functional results with sequence-distribution diagnostics.

5.1 Functional Evaluation

Figure 2 compares training with `PhyloAug` against `NoAug` across seven functional RNA tasks and multiple target models. The direction and magnitude of the changes vary across tasks and models. For a consistent quantitative summary, we consider the seven RNA-LMs and `HyenaDNA`, which have numerical results for all seven tasks in Tables 7 and 8. The comparisons below describe changes in mean performance over five seeds.

On `RBP` and `m6A`, $F1$ and MCC improve for all eight backbones. The mean $F1$ gains across these backbones are 4.32 and 4.00 percentage points, respectively. `STP` shows a more mixed pattern: $F1$ improves for six of eight backbones, whereas MCC improves for five. The two metrics can change in opposite directions. For example, `RNA-BERT` improves from 75.44 to 77.80 in $F1$, while its MCC decreases from 84.33 to 83.66. Thus, the results do not support a uniform benefit across all three tasks grouped as motif-driven prediction.

For the regression tasks, mean Spearman's ρ increases for all eight backbones on `CRISPR` and for seven on `gRNA`. On `CRISPR`, the increases range from 0.0024 for `RNAErnie` to 0.0568 for `RNA-BERT`. On

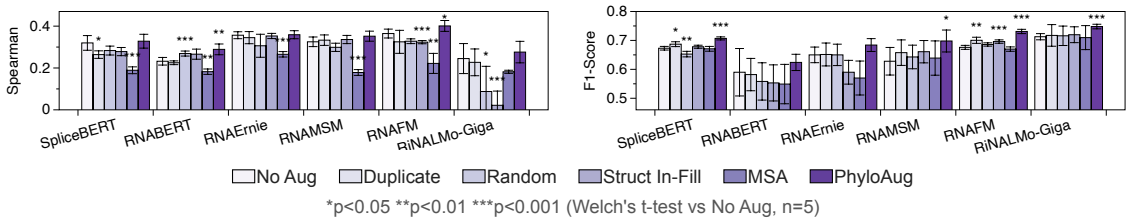


Figure 3: **Matched baseline decomposition.** Results on (a) `CRISPR` and (b) `m6A` under the matched-budget protocol. Full results are in Table 9.

`gRNA`, `SpliceBERT` is the exception, decreasing from 0.4234 to 0.4117. These results show benefits on sequence-to-function prediction, with their magnitude depending on the target model.

For both `mRNA-SL` and `miRNA-SL`, mean $F1$ improves for seven of eight backbones, although MCC changes are more mixed. Some settings show decreases in both metrics: on `miRNA-SL`, `RNA-FM` decreases from 54.44 to 53.40 in $F1$ and from 43.02 to 38.90 in MCC. Augmentation therefore does not improve every model–task combination, and an improvement in one metric need not extend to another.

These comparisons show that `PhyloAug` can improve downstream prediction across multiple tasks and target models. They do not directly establish label preservation for individual augmented sequences or determine which components produce the gains. We therefore examine matched augmentation controls and component ablations to assess the contributions of phylogenetic site selection and structure conditioning.

5.2 Comparison with Augmentation Controls

We compare `PhyloAug` with `NoAug`, `Duplicate`, `Random`, `MSA-only`, and `Struct-Infill` under the training and model-selection protocol in Section 4. These comparisons assess whether `PhyloAug` improves prediction relative to repeated original examples and alternative augmentation policies. Figure 3 shows results on `CRISPR` and `m6A` while Appendix M additionally reports `RBP` and `miRNA-SL`.

On `m6A`, `PhyloAug` achieves the highest mean $F1$ for all six models. Averaged across models, its $F1$ is approximately 0.699, compared with 0.666 for `Duplicate`, the strongest control by average $F1$. On `CRISPR`, `PhyloAug` also exceeds all alternative policies for each model. Its average Spearman's ρ is approximately 0.335, compared with 0.308 for `NoAug` and 0.288 for `Duplicate`. The improvement over `NoAug` varies across models; for `RNAErnie`, both methods score 0.36 at the precision reported in the table.

The additional tasks show both average benefits and model-specific exceptions. On `RBP`, `PhyloAug` has the highest mean $F1$ for all six models, averaging approximately 0.634 compared with 0.595 for `Duplicate`. On `miRNA-SL`, `PhyloAug` also has the highest average $F1$, 0.555 compared with 0.540 for `Duplicate`, but leads on only four of the six models. `Duplicate` performs better for `RNA-BERT` (0.50 versus 0.47) and `RNA-FM` (0.55 versus 0.53). Thus, the average advantage does not extend to every target model.

Our results show that `PhyloAug` performs best on average across the evaluated models, whilst identifying settings where controls may be stronger for individual models. We next decompose `PhyloAug` into phylogenetic site selection and structure-conditioning, to quantify the gains of each component.

5.3 Contributions of Site Selection and Structure Conditioning

We conduct a 2×2 ablation on `m6A` and `CRISPR`, crossing site selection (random or phylogenetic) with generator conditioning (masked sequence only or masked sequence plus predicted secondary structure). The full configuration combines phylogenetic selection with structure conditioning. All four conditions use the same frozen generator, training examples, augmentation volume, per-example mask counts, decoding rule, and downstream training budgets. Runs are paired by target model and seed, using five seeds per model. Within each site-selection condition, the same mask sets are reused with and without structure conditioning. Table 1 summarizes the resulting performance differences across 12 target models.

Table 1: **Phylogenetic site selection and structure conditioning on `m6A` and `CRISPR`.** Mean paired differences across 12 downstream models; parentheses denote models improved.

Task		Phylo $vs.$ Rand. (seq.)	Phylo $vs.$ Rand. (struct.)	Struct. $vs.$ Seq. (Rand.)	Struct. $vs.$ Seq. (Phylo)	Full $vs.$ Rand. (seq.)
m6A	Δ	$+0.0328$ $(11/12)$	$+0.0331$ $(12/12)$	$+0.0072$ $(9/12)$	$+0.0075$ $(9/12)$	$+0.0403$ $(12/12)$
	SD	0.0195	0.0161	0.0070	0.0057	0.0224
CRISPR	Δ	$+0.0154$ $(10/12)$	$+0.0151$ $(10/12)$	$+0.0173$ $(10/12)$	$+0.0169$ $(11/12)$	$+0.0323$ $(11/12)$
	SD	0.0184	0.0135	0.0172	0.0176	0.0222

On m6A, changing the site-selection rule produces a larger average improvement than adding structure conditioning. Replacing random selection with phylogenetic selection increases mean $F1$ by approximately 0.033 under either conditioning setting. The mean difference is positive for 11 of 12 models without structure conditioning and all 12 with structure conditioning. Adding structure conditioning increases mean $F1$ by approximately 0.007 under either selector, with positive differences for nine of 12 models in each case. The full configuration improves over random sequence-only augmentation by 0.0403 $F1$, equivalent to 4.03 percentage points, averaged across models; all 12 models have positive mean differences.

On CRISPR, the two components produce similar average improvements. Phylogenetic selection increases Spearman's ρ by approximately 0.015 under either conditioning setting, with positive mean differences for ten of 12 models. Structure conditioning increases ρ by approximately 0.017, benefiting ten models under random selection and 11 under phylogenetic selection. The full configuration improves over random sequence-only augmentation by an average of 0.0323 in ρ, with positive mean differences for 11 of 12 models.

Across these two benchmarks, both components provide positive average improvements when the other component is held fixed. Their relative contributions differ: site selection has the larger average effect on m6A, whereas both effects are similar on CRISPR. These results support combining phylogenetic site selection with structure conditioning, while showing that the benefit of each component varies across tasks and models.

5.4 Label-Budget Scaling

We next examine how the benefit of PhyloAug changes by varying the amount of labeled training data. On m6A prediction, we train each backbone on stratified label subsets containing $\{1\%, 5\%, 10\%, 20\%, 50\%, 100\%\}$ of the training set, while keeping validation and test sets fixed. The same subsets are used for augmented and non-augmented runs. Unless otherwise stated, PhyloAug uses the same augmentation budget and matched training protocol defined in Section 4.

Across label budgets, PhyloAug improves average performance over training without augmentation, with gains increasing from 1.16 F1 points at 1% data to 3.40 at 100%. Notably, PhyloAug with only 20% labelled data reaches 67.27 F1, exceeding the 65.98 from non-augmented fine-tuning. This suggests that phylogenetically constrained augmentation improves label-budget scaling. Raw results are reported in Table 10.

These results do not support a simple low-data-only story: the smallest label budgets show smaller gains, while improvements persist at higher data fractions. A plausible interpretation is that extremely small subsets provide too little task signal for augmented variants to be useful, whereas moderate or large label budgets allow PhyloAug to increase diversity without overwhelming the decision boundary. The largest gains occur for RNA-MSM, suggesting that augmentation benefits vary across backbones. We treat this as a representative label-budget analysis and evaluate task-level controls in the preceding matched-baseline experiments.

5.5 Distributional Diagnostics

Downstream performance alone can be misleading when evaluation tasks reward the same structural or evolutionary priors used to construct augmentations. We therefore complement functional evaluation with

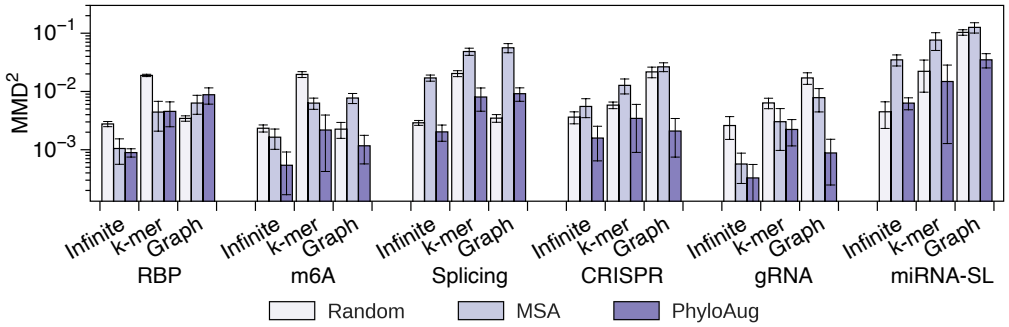


Figure 4: **Distributional diagnostics.** MMD between augmented and held-out samples under edit-aware (Infinite), k-mer, and structure-aware graph kernels. Lower indicates closer distributional alignment.

a label-free diagnostic of augmentation-induced distribution shift. Let D_π denote the augmented samples produced by policy π, and let D_{ref} denote the held-out validation split used only for distributional diagnostics. We measure the discrepancy between the corresponding distributions $\mathbb{P}_\pi$ and $\mathbb{P}_{\text{ref}}$ using maximum mean discrepancy (MMD) (Gretton et al., 2012):

$$\text{MMD}_k^2(\mathbb{P}_\pi, \mathbb{P}_{\text{ref}}) = \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim\mathbb{P}_\pi}[k(\mathbf{x},\mathbf{x}')] + \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\mathbb{P}_{\text{ref}}}[k(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\pi,\mathbf{y}\sim\mathbb{P}_{\text{ref}}}[k(\mathbf{x},\mathbf{y})]. \tag{5}$$

For characteristic kernels, MMD is zero if and only if the two distributions match. In our setting, the kernel specifies the notion of sequence similarity being tested. Our primary diagnostic uses an edit-aware infinite-spectrum sequence kernel inspired by biological sequence kernels with theoretical guarantees (Amin et al., 2025). As auxiliary diagnostics, we also report k-mer spectrum kernels and structure-aware graph kernels (Borgwardt et al., 2005) to test whether trends persist under alternative similarity notions. Where labels are available, we compute MMD in a label-stratified manner to mitigate class-imbalance effects.

We compare `PhyloAug` with matched augmentation policies using the same sample size, mask-ratio bounds, augmentation budget, and perturbation generator when applicable, excluding `DUPLICATE` and `STRUCT-INFILL` as their induced distributions overlap with those of other baselines. MMD is estimated by repeated subsampling, and we report mean and standard deviation across runs. We interpret MMD as a diagnostic of distributional alignment, not as a standalone measure of label preservation, diversity, or downstream utility. Implementation details are provided in Appendix E.

Figure 4 shows that `PhyloAug` yields lower MMD than less constrained augmentation policies on most benchmarks under the primary edit-aware kernel. Similar trends are observed under the auxiliary kernels, although some inconsistencies arise on `RBP` and `STP`, likely due to their limited resolution in capturing fine-grained sequence variations (Amin et al., 2025). This suggests that phylogenetically guided site selection reduces augmentation-induced distribution shift relative to baselines that perturb positions less selectively.

6 Conclusions

We introduced `PhyloAug`, a data-centric augmentation framework that converts homolog-derived evolutionary rates into a position-specific edit policy for RNA FM fine-tuning. Across functional RNA benchmarks, matched controls, cross-generator diagnostics, label-budget scaling, and distributional analyses, `PhyloAug` improves fine-tuning most reliably when label-preserving variation is available. The results show that evolutionary priors can be injected through data rather than target-architecture changes, while also revealing limitations: performance depends on homolog quality, generator choice, augmentation budget, and the validity of label preservation.

REFERENCES

Kyle E. Watters, Angela M Yu, Eric J. Strobel, Alex H. Settle, and Julius B. Lucks. Characterizing RNA structures in vitro and in vivo with selective 2′-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Methods*, 103:34–48, 2016. doi: https://doi.org/10.1016/j.ymeth.2016.04.002. Advances in RNA Structure Determination.

Xinang Cao, Yueying Zhang, Yiliang Ding, and Yue Wan. Identification of RNA structures and their roles in RNA functions. *Nat. Rev. Mol. Cell Biol.*, 25:784–801, 2024. doi: 10.1038/s41580-024-00748-6.

Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90, 2022. doi: https://doi.org/10.1016/j.aiopen.2022.03.001. URL https://www.sciencedirect.com/science/article/pii/S2666651022000080.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR'24: Proc. of the 2024 International Conference on Learning Representations*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ZWzUA9zeAg.

Hyunjung Lee, Utku Ozbulak, Homin Park, Stephen Depuydt, Wesley De Neve, and Joris Vankerschaver. Assessing the reliability of point mutation as data augmentation for deep learning with genomic data. *BMC Bioinform.*, 25(170), 2024. doi: 10.1186/s12859-024-05787-6.

Nicholas Keone Lee, Ziqi Tang, Shushan Toneyan, and Peter K. Koo. EvoAug: improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations. *Genome Biol.*, 24(105), 2023. doi: 10.1186/s13059-023-02941-w.

Chenchen Yang, Hao Wu, Tao Shen, Kai Zou, and Siqi Sun. PriFold: Biological priors improve RNA secondary structure predictions. In *AAAI'25: Proc. of Thirty-Ninth AAAI Conference on Artificial Intelligence*, pages 950–958. AAAI Press, 2025a. doi: 10.1609/AAAI.V39I1.32080. URL https://doi.org/10.1609/aaai.v39i1.32080.

Ziheng Yang. PAML 4: Phylogenetic analysis by maximum likelihood. *Mol. Biol. Evol.*, 24(8):1586–1591, 08 2007. doi: 10.1093/molbev/msm088. URL https://doi.org/10.1093/molbev/msm088.

Alan Nawzad Amin, Debora S. Marks, and Eli N. Weinstein. Biological sequence kernels with guaranteed flexibility. *J. Mach. Learn. Res.*, 26:216:1–216:63, 2025. URL https://jmlr.org/papers/v26/23-0455.html.

Gonzalo Benegas, Carlos Albors, Alan J. Aw, Chengzhong Ye, and Yun S. Song. A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nat. Biotechnol.*, 43: 1960–1965, 2025. doi: 10.1038/s41587-024-02511-w.

Melissa J. Hubisz, Katherine S. Pollard, and Adam Siepel. PHAST and RPHAST: phylogenetic analysis with space/time models. *Brief Bioinform.*, 12(1):41–51, 01 2011. doi: 10.1093/bib/bbq072. URL https://doi.org/10.1093/bib/bbq072.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. In *ICML'21: Proc. of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

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 Res.*, 52 (1), 01 2024. doi: 10.1093/nar/gkad1031. URL https://doi.org/10.1093/nar/gkad1031.

Carlos Albors, Jianan Canal Li, Gonzalo Benegas, Chengzhong Ye, and Yun S. Song. A phylogenetic approach to genomic language modeling. In *RECOMB'25: Proc. of the 2025 International Conference on Research in Computational Molecular Biology*, volume 15647 of *Lecture Notes in Computer Science*, pages 99–117. Springer, 2025. doi: 10.1007/978-3-031-90252-9_7. URL https://link.springer.com/chapter/10.1007/978-3-031-90252-9_7.

Jiren Zhou, Jiajia Xu, Jiayu Wen, and Brian John Parker. CS-Fold: Advancing RNA structure predictions through phylogenetic modelling of compensatory mutations in deep neural networks. *bioRxiv*, 2025. doi: 10.1101/2025.04.27.650904. URL https://www.biorxiv.org/content/early/2025/04/28/2025.04.27.650904.

Ronny Lorenz, Stephan H. Bernhart, Höner zu Siederdissen Christian, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA package 2.0. *Algorithms Mol. Biol.*, 6(26), 2011. doi: 10.1186/1748-7188-6-26.

Eric W. Sayers, Mark Cavanaugh, Karen Clark, Kim D. Pruitt, Conrad L. Schoch, Stephen T. Sherry, and Ilene Karsch-Mizrachi. GenBank. *Nucleic Acids Res.*, 49(D1):D92–D96, 01 2021. doi: 10.1093/nar/gkaa1023. URL https://doi.org/10.1093/nar/gkaa1023.

Kazutaka Katoh and Daron M. Standley. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Mol. Biol. Evol.*, 30(4):772–780, 04 2013. ISSN 0737-4038. doi: 10.1093/molbev/mst010. URL https://doi.org/10.1093/molbev/mst010.

Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PLOS One*, 5(3):e9490, 03 2010. doi: 10.1371/journal.pone.0009490. URL https://doi.org/10.1371/journal.pone.0009490.

Heng Yang, Renzhi Chen, and Ke Li. Bridging sequence-structure alignment in RNA foundation models. In *AAAI'25: Proc. of Thirty-Ninth AAAI Conference on Artificial Intelligence*, pages 21929–21937. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I20.35500. URL https://doi.org/10.1609/aaai.v39i20.35500.

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. In *NeurIPS'24: Proc. of the 2024 Advances in Neural Information Processing Systems*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/a8ea503d91320fcfe12cba61c8a6d285-Abstract-Datasets_and_Benchmarks_Track.html.

Hui Wang, Wenjun Lin, Hongwang Xiao, Qiwei Ye, and Yaqing Zhang. RNA-scope: Benchmarking RNA language models for RNA sequence understanding. In *NeurIPS'25 Workshop: Proc. of the NeurIPS Workshop on AI for Science*, 2025. URL https://openreview.net/forum?id=lnV3NXpDiC.

Ningyuan You, Chang Liu, Hai Lin, Sai Wu, Gang Chen, and Ning Shen. Benchmarking pre-trained genomic language models for rna sequence-related predictive applications. *Nat. Commun.*, 17:223, 2026. doi: 10.1038/s41467-025-66899-y.

Tyler S. Alioto. U12DB: a database of orthologous U12-type spliceosomal introns. *Nucleic Acids Res.*, 35: D110–D115, 01 2007. doi: 10.1093/nar/gkl796. URL https://doi.org/10.1093/nar/gkl796.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin W. Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton M. Rabideau, Yoshua Bengio, Stefano Ermon, Christopher Ré, and Stephen Baccus. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. In *NeurIPS'23: Proc. of the 2023 Conference on Neural Information Processing Systems*, 2023.

Micaela Elisa Consens, Kevin K Yang, James Brian Hall, Ashley Mae Conard, Bo Wang, Lorin Crawford, Alan M Moses, and Alex Xijie Lu. Predicting evolutionary rate as a pretraining task improves genome language model representations. In *ICML'26: Proc. of the 2026 International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FOIP4Blm4F.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012. doi: 10.5555/2503308.2188410. URL https://dl.acm.org/doi/10.5555/2503308.2188410.

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56, 2005. doi: 10.1093/bioinformatics/bti1007. URL https://doi.org/10.1093/bioinformatics/bti1007.

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. *Brief. Bioinform.*, 25(3), 05 2024. doi: 10.1093/bib/bbae163. URL https://doi.org/10.1093/bib/bbae163.

Manato Akiyama and Yasubumi Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genom Bioinform.*, 4(1), 03 2022. doi: 10.1093/nargab/lqac012. URL https://doi.org/10.1093/nargab/lqac012.

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 RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. *arXiv*, 2022.

Ning Wang, Jiang Bian, Yuchen Li, Xuhong Li, Shahid Mumtaz, Linghe Kong, and Haoyi Xiong. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nat. Mach. Intell.*, 6:548–557, 2024. doi: 10.1038/s42256-024-00836-4.

Rafael Josip Penić, Tin Vlašić, Roland G. Huber, Yue Wan, and Mileš Šikić. RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks. *Nat. Commun.*, 16:5671, 2025. doi: 10.1038/s41467-025-60872-5.

Shuxian Zou, Tianhua Tao, Sazan Mahbub, Caleb Ellington, Robin Jonathan Algayres, Dian Li, Yonghao Zhuang, Hongyi Wang, Le Song, and Eric P. Xing. A large-scale foundation model for RNA function and structure prediction. In *NeurIPS'24 Workshop: Proc. of the NeurIPS Workshop on AI for New Drug Modalities*, 2024. URL https://openreview.net/forum?id=Gzo3JMPY8w.

Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *J. Mach. Learn. Res.*, 12:2389–2410, 2011. doi: 10.5555/1953048.2021077. URL https://dl.acm.org/doi/10.5555/1953048.2021077.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12:2539–2561, 2011. doi: 10.5555/1953048.2078187. URL https://dl.acm.org/doi/10.5555/1953048.2078187.

Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. GraKeL: A graph kernel library in python. *J. Mach. Learn. Res.*, 21:54:1–54:5, 2020. URL https://www.jmlr.org/papers/v21/18-370.html.

Nancy Ontiveros-Palacios, Emma Cooke, Eric P Nawrocki, Sandra Triebel, Manja Marz, Elena Rivas, Sam Griffiths-Jones, Anton I Petrov, Alex Bateman, and Blake Sweeney. Rfam 15: RNA families database in 2025. *Nucleic Acids Res.*, 53(D1):D258–D267, 01 2025. doi: 10.1093/nar/gkae1023.

Mehdi Saman Booy, Alexander Ilin, and Pekka Orponen. RNA secondary structure prediction with convolutional neural networks. *BMC Bioinform.*, 23(58), 2022. doi: 10.1186/s12859-021-04540-7.

Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat. Commun.*, 10:5407, 2019. doi: 10.1038/s41467-019-13395-9.

Zhen Tan, Yinghan Fu, Gaurav Sharma, and David H. Mathews. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res.*, 45(20):11570–11581, 11 2017. doi: 10.1093/nar/gkx815. URL https://doi.org/10.1093/nar/gkx815.

Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L. Madden. BLAST+: architecture and applications. *BMC Bioinform.*, 10(421), 2009. doi: 10.1186/1471-2105-10-421.

APPENDIX

Disclosure of AI usage In this work, we used generative AI tools to refine and polish some sentences and language within the manuscript. We did not use generative AI to do any of the following: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results.

We have reviewed all AI-assisted work, we checked LLM re-written language and sentences within the paper, and further edited. We did not use LLM's to generate sections of the paper, only to polish minor sentences for clarity and conciseness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

A LIMITATIONS AND BROADER IMPACT

Limitations. `PhyloAug` relies on homolog retrieval, multiple sequence alignment, phylogenetic rate estimation, and structure-conditioned infilling, so its effectiveness depends on the quality and availability of these intermediate signals. Sequences with too few reliable homologs are not augmented, and the neutral-rate interval used for site selection is only an operational proxy for mutation permissiveness rather than a definitive biological criterion. In addition, `PhyloAug` assumes that edits at mutation-permissive sites preserve downstream labels, which may not hold for tasks where function is densely coupled to exact sequence identity. The method also introduces additional computational cost and remains sensitive to the perturbation generator and augmentation budget.

Our label-free diagnostics rely on kernel computations. Although the edit-aware kernel is theoretically well grounded, its computational cost remains high, limiting its applicability to longer sequences. We believe that further investigation into kernel embeddings that are both scalable and theoretically principled is an important direction for future research of data quality.

More broadly, `PhyloAug` may help reduce annotation burden and improve data-efficient adaptation of RNA foundation models in settings where experimental labels are scarce. However, augmented RNA sequences should not be interpreted as experimentally validated functional variants, and downstream predictions may still reflect biases in the retrieved homologs, structural predictions, or pretrained generators. Care should therefore be taken when applying the method to biological design or functional interpretation, especially in high-stakes experimental settings where generated variants require empirical validation.

Broader Impact. As large-scale foundation models increasingly provide general computational priors for biological sequence modelling, many specialised small-data prediction tasks may benefit from stronger computational support. Our method aims to facilitate such tasks through a minimally interventionist augmentation strategy that improves the quality of synthetic training data while preserving distributional and functional consistency and with held-out data. We believe this direction may positively contribute to downstream applications such as RNA therapeutics, functional sequence analysis, and biological discovery in data-limited settings.

At the same time, our framework relies on the assumption that the underlying datasets, particularly the validation distributions used for diagnostics and evaluations, are reasonably reliable and representative. If dataset construction introduces incorrect inductive biases or distributional shifts, our method may neither detect nor mitigate these issues. We therefore recommend applying and analysing the proposed diagnostics and validation-based evaluations under carefully curated and relatively balanced biological tasks, where their distributional or functional assumptions are more likely to hold.

B Cross-Generator Control

We next test whether the gains from PhyloAug depend on the perturbation generator. Here, the generator is the RNA FM used to fill masked positions after phylogenetically guided site selection. We fix the augmentation budget to $M = 6$ for all generator–target pairs, making this a diagnostic of generator effects rather than a search for the optimal generator. We evaluate three representative tasks: m6A prediction, CRISPR on-target activity, and miRNA-SL subcellular localisation, covering motif-driven, sequence-to-function, and distributed contextual regimes.

For each generator–target pair, we report the absolute change relative to the corresponding non-augmented baseline, $\Delta = \text{score}_{\text{aug}} - \text{score}_{\text{base}}$, averaged over five random seeds. The same site-selection policy, masking budget, and training protocol are used across generators.

Figure 5 shows that generator choice substantially affects augmentation quality. If gains were driven primarily by self-reinforcement, diagonal generator–target pairs would be consistently strongest. Instead, several off-diagonal pairs also improve downstream performance, indicating that the benefit of PhyloAug is not reducible to using the same model as generator and target. At the same time, generator quality matters: some generators produce variants that hurt downstream performance under the same site-selection policy and augmentation budget.

These results clarify the scope of the method. PhyloAug is target-architecture-free because the target RNA-FM is not modified, but it is not generator-independent. Phylogenetic site selection defines *where* edits occur, while the generator determines whether the substitutions are useful for fine-tuning.

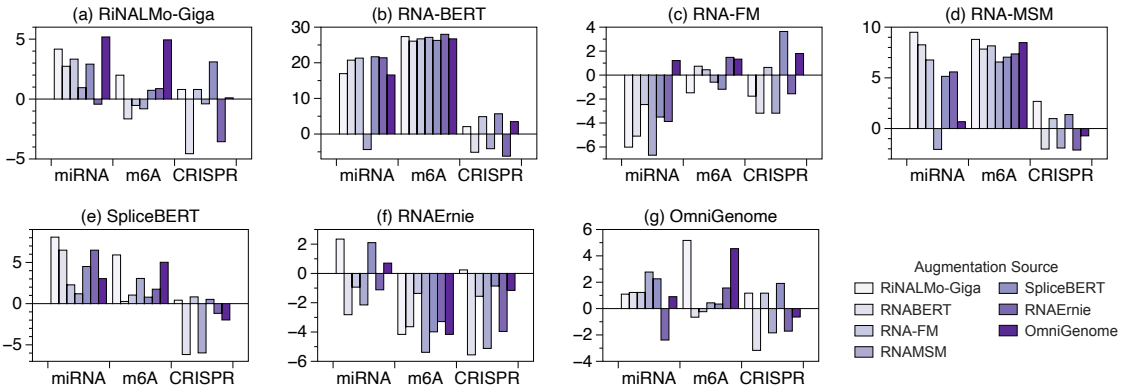


Figure 5: **Cross-model augmentation performance across downstream training models.** Each panel reports the change in performance across miRNA-SL, m6A, and CRISPR tasks for a fixed downstream training model, comparing augmentation sources under the matched-budget protocol.

C Model Details

We provide additional details of model training in Table 2.

Table 2: Summary of sequence foundation models and their pretraining data.

Model	Tok.	Params	Scale	Dataset	Domain
MLP	SNT	0.20M	from scratch	—	task-supervised only
CNN	SNT	2.36M	from scratch	—	task-supervised only
ResNet	SNT	5.32M	from scratch	—	task-supervised only
HyenaDNA-Large (Nguyen et al., 2023)	SNT	54.6M	3.2B tokens	Human GRCh38	Human DNA
SpliceBERT (Chen et al., 2024)	SNT	19M	2M seqs	UCSC pre-mRNA	Vertebrates pre-mRNA
RNA-BERT (Akiyama and Sakakibara, 2022)	SNT	0.5M	4069 families	Rfam	Multi-sp. ncRNA
RNA-MSM (Zhang et al., 2024)	SNT	96M	4069 families	Rfam (MSA)	Multi-sp. ncRNA
RNA-FM (Chen et al., 2022)	SNT	99.5M	23M seqs	RNAcentral	Multi-sp. ncRNA
RNAErnie (Wang et al., 2024)	k-mer	86.7M	23M seqs	RNAcentral	Multi-sp. ncRNA
OmniGenome Yang et al. (2025b)	SNT	186M	54.2B tokens	OneKP	1124 plants
RiNALMo-Giga (Penić et al., 2025)	SNT	650M	36M seqs	RNAcentral	Multi-sp. ncRNA
AIDO.RNA (Zou et al., 2024)	SNT	1.6B	42M seqs	RNAcentral	Multi-sp. ncRNA
Bi-Gamba (Consens et al., 2026)	SNT	7M	240 Mammals	Zoonomia	Mammalian DNA & RNA

D Data Sources and Splits

We additionally provide details of the data used within training, as well as cross-referencing with the papers the data is obtained from.

Label-budget and homolog-filtering protocol. For the m6A label-budget analysis, we construct stratified subsets containing $\{1\%, 5\%, 10\%, 20\%, 50\%, 100\%\}$ of the original training set while keeping the validation and test sets fixed; subsets are nested ($1\% \subset 5\% \subset \cdots \subset 100\%$) and identical across seeds, and the same labelled subsets are used for augmented and non-augmented runs. Homologs are retrieved with `blastn` (megablast) against NCBI `nt` with $E \leq 10^{-5}$, at most 127 target sequences and one HSP per subject, with DUST and soft masking disabled. We retain the aligned subject region of each hit when it is between 20 and 4096 nt, and drop hits whose minimum pairwise Jensen–Shannon divergence to the already-retained hits is below 0.05 to avoid near-duplicate homologs. Retained sets are deduplicated, aligned with MAFFT, and used for tree inference and rate estimation.

Table 3: Functional benchmark sources and split construction.

Task	Source	Train / Val. / Test	Split / redundancy control
RBP	(Wang et al., 2025)	331,591 / 110,537 / 110,555	$> 80\%$ identity removed
m6A	(You et al., 2026)	39,218 / 9,810 / 12,611	Random 80/20 split & $> 80\%$ identity removed
STP	(Alioto, 2007)	15,957 / 1,851 / 1,861	Gene- and species-disjoint splits & $> 80\%$ identity removed
CRISPR	(Ren et al., 2024)	1,453 / 207 / 416	Redundant entries removed & $> 80\%$ identity removed
gRNA	(Wang et al., 2025)	6,134 / 1,480 / 13,631	Independent test datasets
mRNA-SL	(Wang et al., 2025)	13,838 / 1,730 / 1,730	80% similarity; 8:1:1 split
miRNA-SL	(Wang et al., 2025)	373 / 47 / 118	80% identity filtering

E Details of Distributional Diagnostics for Augmented RNA Data

E.1 MMD Measure with Theoretical Guarantees

For MMD to provide a meaningful measure of distributional discrepancy, the kernel should be characteristic and metrise the underlying distribution space (Sriperumbudur et al., 2011). Under such kernels, a smaller MMD value implies a smaller discrepancy between the corresponding distributions, as compared in Figure 4.

The infinite spectrum kernel satisfies these requirements, as established in (Amin et al., 2025). However, computing similarities under this kernel relies on dynamic programming, leading to substantially higher computational cost as sequence length or the number of samples increases. In contrast, k-mer and graph kernels are substantially more efficient in practice and have been widely adopted in biological sequence analysis and bioengineering applications (Borgwardt et al., 2005). However, their expressiveness is limited, and they are generally neither characteristic nor metrising, making them less theoretically grounded as MMD measures (Amin et al., 2025). In our study, these two kernels serve as auxiliary metrics to verify that observed trends are consistent across different similarity notions.

E.2 MMD Estimation and Subsampling Protocol

We estimate the MMD using empirical samples drawn from the augmented training data and held-out validation data. Given two sample sets $D_\pi = \{\mathbf{x}^{(i)}\}_{i=1}^n \sim \mathbb{P}_\pi$ and $D_{\text{ref}} = \{\mathbf{y}^{(j)}\}_{j=1}^m \sim \mathbb{P}_{\text{ref}}$, we use the

unbiased estimator of squared MMD:

$$\widehat{\mathrm{MMD}}^2 = \frac{1}{n(n-1)} \sum_{i \neq i'} k(\mathbf{x}^{(i)}, \mathbf{x}^{(i)'}) + \frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j)'}) - \frac{2}{nm} \sum_{i,j} k(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}). \quad (6)$$

To balance computational complexity and statistical stability, we adopt a repeated subsampling strategy. For each comparison, we randomly sample a fixed number of sequences from both the augmented training data and the held-out validation set, and compute MMD. This process is repeated five times with fixed validation set and different random seeds, and we report the mean and standard deviation across runs. The held-out validation set is provided by the original benchmark, and augmentation is applied only to the training data to avoid data leakage.

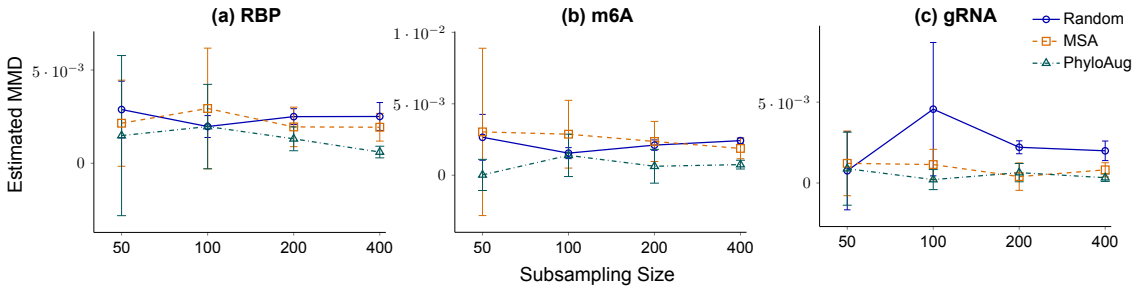


Figure 6: Effect of subsampling size on MMD stability with edit-aware infinite spectrum kernel.

We further analyse the effect of the sampling size on the stability of MMD estimates (Figure 6). We select three prediction tasks (RBP, m6A and gRNA) as they exhibit representative dataset characteristics in terms of sample size and sequence length, and supports sufficiently large-scale subsampling. We observe that the ranking of methods remains largely consistent when the sample size exceeds 200. To balance computational efficiency and ranking stability, we use a sample size of 400 when the validation set is sufficiently large, and otherwise use the full validation set.

E.3 Kernel Implementations

Edit-aware kernel. We adopt an infinite spectrum kernel inspired by prior work (Amin et al., 2025), which is designed to metrise distributions over discrete sequence spaces. This kernel naturally handles variable-length sequences and captures edit operations such as insertions, deletions, and substitutions, providing a biologically meaningful notion of similarity aligned with evolutionary transformations.

In practice, the infinite spectrum kernel can be implemented via a modified local alignment kernel with strict gap constraints. We build upon the original implementation of (Amin et al., 2025) through its open-source package[1]. To improve computational efficiency, we replace the full kernel matrix computation in the original package with a block-wise distributed implementation. Since the dynamic programming is employed in calculating sequence similarity, the complexity grows with sequence length.

Auxiliary kernels. To assess robustness, we additionally evaluate MMD using:

- **k-mer kernel**: measures similarity based on shared subsequence frequencies. For each sequence, we extract all contiguous k-mers and represent the sequence as a vector of k-mer frequencies. The kernel is

[1] https://github.com/AlanNawzadAmin/Kernels-with-guarantees/

then computed as the inner product between k-mer frequency vectors, followed by cosine normalisation to ensure comparability across sequences of different lengths. We use $k = 3$ across all benchmarks.

- **Graph kernel**: inspired by graph-based similarity measures in protein modelling (Borgwardt et al., 2005), we construct an RNA sequence-structure graph by integrating both nucleotide identity and predicted secondary structure from ViennaRNA (Lorenz et al., 2011).
  Each RNA sequence is represented as a graph where nodes correspond to nucleotides, and edges include both sequential adjacency (backbone connections) and base-pair interactions derived from dot-bracket structures. Node labels encode nucleotide types and pairing states, allowing the graph to capture both sequence and structural context.
  We then apply the Weisfeiler–Lehman (WL) subtree kernel (Shervashidze et al., 2011) to compute similarities between graphs implemented by GraKeL (Siglidis et al., 2020). This kernel iteratively refines node representations based on neighbourhood structures, enabling efficient comparison of local sequence-structure patterns. We set the number of WL iterations to 3 across benchmarks.

F Structural and Evolutionary Experiments

In addition to our RNA functional design experiments, we also evaluate PhyloAug's capabilities at improving performance across structural and evolutionary-inspired downstream tasks. These additional experiments are aimed to validate that PhyloAug embeds evolutionary signals, such as covariation and conservation, and preserves structural motifs. We divide these additional tasks into the following:

- **Conserved Site Prediction.** We establish a novel dataset using the Rfam database, which contains the conserved annotations for over 4000 unique RNA families Ontiveros-Palacios et al. (2025). This task is designed to test RNA-FMs understanding of evolutionary invariant nucleotides, which are fundamental to function and structural integrity.
- **Secondary Structure Prediction.** We adopt the three highly appraised secondary structure prediction datasets, ArchiveII, bpRNA and rnastralign, using the same data splits as released in their respective papers Saman Booy et al. (2022); Singh et al. (2019); Tan et al. (2017). This task serves as the benchmark for evaluating model understanding of the inherent RNA secondary structure and folding mechanism.
- **Evolutionary Covariation.** We establish a novel dataset leveraging the Rfam database Ontiveros-Palacios et al. (2025), extracting key sequences where covariation around two base-pairs occur, and does not occur. We ask the model to classify the sequences as "covariation occurs" and "covariation does not occur". This task evaluates whether a model captures the signal of compensatory mutations, a key evolutionary constraint that maintains RNA structure.

Mitigating Data Leakage To prevent homology leakage, we enforce strict split construction. For structural datasets, sequences are first clustered using CD-HIT-EST at 80% identity, with entire clusters assigned to a single split. We further remove residual similarity by performing BLASTN filtering (E-value <= 1e-10) from training/validation to test, discarding any sequence with significant matches. For evolutionary tasks, we use Rfam-family splits, ensuring no cross-split evolutionary relationships.

Experimental Design For each task and model, we compare non-augmented training against PhyloAug using different numbers of augmentations, $M \in 1, 2, 4, 6, 8, 10, 12$, where M denotes the number of augmented variants generated per sequence. The reported augmented performance corresponds to the best-performing value of M, as specified in the main paper. Models are evaluated using F1 score and Matthews Correlation Coefficient, capturing both positive-class accuracy and overall classification robustness. This design tests whether phylogenetically guided augmentation improves RNA model generalisation across conservation, structure, and covariation tasks while controlling for leakage and augmentation scale.

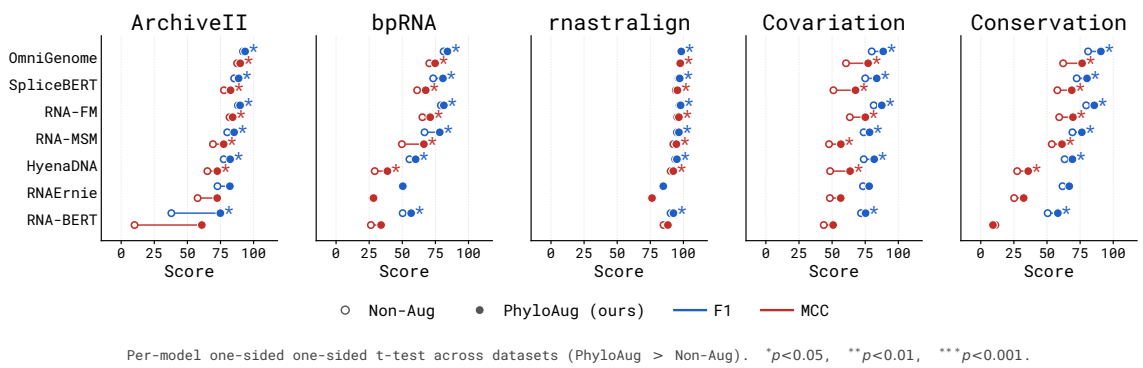


Figure 7: Structural and Evolutionary evaluation across five RNA benchmarks. Each dumbbell compares non-augmented fine-tuning with `PhyloAug` under the matched-budget protocol in Section 4. Full numerical results and statistical tests are reported in Table 11.

Results Across conserved-site prediction, secondary-structure prediction, and evolutionary covariation detection, PhyloAug consistently improves model performance, demonstrating that phylogenetically guided augmentation provides a robust and broadly applicable evolutionary training signal. As summarised in fig. 7, these gains are observed across diverse model families and evaluation settings, with improvements most clearly reflected in MCC, indicating that PhyloAug enhances not only predictive accuracy but also the models' ability to distinguish biologically meaningful positives from negatives. The strongest models, such as OmniGenome, RNA-FM, and SpliceBERT, generally benefit most in absolute terms, suggesting that richer pre-training enables models to exploit the additional evolutionary signal more effectively. Conversely, weaker or more sequence-limited models often show substantial improvements on structurally demanding datasets, indicating that PhyloAug can compensate for missing evolutionary or structural supervision in the original training data.

Taken together, the results show that PhyloAug improves generalisation across both sequence-level and structure-aware RNA tasks. The method is particularly effective when the task depends on evolutionary constraints, such as conserved positions, compensatory mutations, or structurally viable sequence variation. These findings support the central claim that evolutionary reasoning can be introduced into RNA language models through biologically grounded data augmentation, rather than requiring changes to model architecture.

G Structure-Conditioner Bias

An over-reliance on `ViennaRNA`-derived structures in `PhyloAug` could introduce a correlated bias. Thus, we replaced the single-sequence `ViennaRNA` MFE structure with an alignment-informed `RNAalifold` consensus structure and repeated the `m6A` and `CRISPR-ON` evaluations. We fixed examples, homologs, selected sites, and the downstream protocol; only the structure input changed.

Both structure sources retained similar mean gains over NoAug:

Replacing the single–sequence MFE structure with the `RNAalifold` consensus structure changes the mean effect by only +0.0021 F1 on `m6A` and -0.002 Spearman's on `CRISPR-ON`, while mean augmentation gains remained positive at +0.0354 F1 and +0.029 Spearman's, respectively. These results indicate that, in these two short-sequence settings, `PhyloAug`'s downstream benefit does not require reusing the same single-sequence `ViennaRNA` prediction at augmentation time. We deliberately keep our conclusion narrow. This

Table 4: Performance gains and number of positive models.

Metric	m6A	CRISPR-ON
Evaluation metric	F1	Spearman's
`ViennaRNA` gain	+0.0333 (0.0036)	+0.031 (0.080)
`RNAalifold` gain	+0.0354 (0.0037)	+0.029 (0.101)
`ViennaRNA` positive models	10	10
`RNAalifold` positive models	11	9

alignment-informed control does not remove all correlated structural bias nor establish statistical equivalence, but provides an alignment-informed robustness test of the augmentation-time structures.

We also note that predicted structures are less reliable for long or complex RNAs. Six of the seven functional benchmarks have max lengths of at most 149 nt; mRNA-SL is the exception, spanning $186 - 30,609$ nt.

H Augmentation-Budget Sensitivity

We next study the effect of the number of augmented samples generated per original training example. For each task and backbone, we sweep $M \in \{1, 2, 4, 6, 8, 10, 12\}$ while keeping the site-selection rule, mask-ratio bounds, generator, and training budget fixed. The sweep is performed on the validation split; the test set is used only for the final evaluation selected by the validation protocol.

Figure 8 shows that performance is not monotonic in M. Moderate augmentation budgets can improve performance, but larger budgets often reduce gains and can lead to negative transfer. The optimal M varies across both tasks and backbones: `m6A` benefits from moderate budgets, `miRNA-SL` shows higher model-dependent variability, and `gRNA` is more sensitive to over-augmentation. These patterns support treating M as a controlled hyperparameter rather than fixing a universal augmentation count.

In the main experiments, we therefore select M using the same validation search space for all augmentation policies. This keeps the augmentation-budget search matched across methods while avoiding a single fixed M that may favour one task, model, or augmentation policy.

I Comparing Estimated Site-Rates through PAML with PhyloP

Curated whole-genome alignments and PhyloP tracks provide consistent, orthology-aware evolutionary estimates when reliable coordinates and appropriate reference alignments are available. BLAST/PAML provides broader coverage for coordinate-free sequences and unsupported organisms, although homolog quality and paralogy require explicit filtering. In this experiment, we investigate the effect that normalised evolutionary rates may have over rates mapped across the whole-genome. We therefore conducted a preliminary coordinate-aware comparison on m6A and CRISPR-ON, replacing BLAST/PAML-derived site-rate estimates with coordinate-mapped PhyloP scores. We used the Zoonomia 241-mammalian Cactus alignment (2020v2) and mapped scores to query nucleotides using hg38 coordinates and strand, achieving >99.9% nucleotide-level coverage. We restricted both conditions to the common mapped subset: 61,644/61,661 m6A and 2,075/2,076 CRISPR-ON examples; unmapped nucleotides were excluded. These raw mapping numbers include the test data - which was excluded from PAML and PhyloP experiments.

We did not impose a separate PhyloP cutoff. For each retained sequence, the PAML condition determined eligibility and the realised mask count after applying its prespecified near-neutral interval around $r = 1$ and the common mask-ratio controls. PhyloP then selected the same number of positions ranked closest to 0,

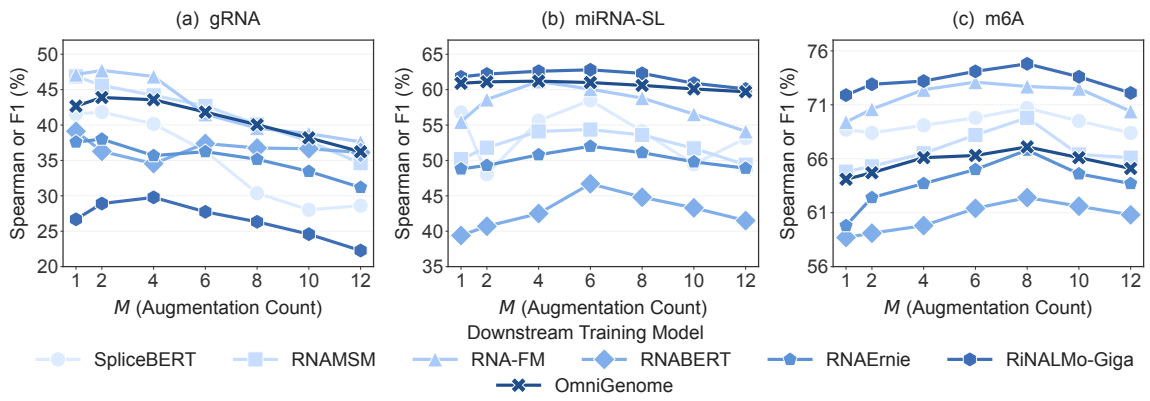


Figure 8: **Augmentation-budget sensitivity.** Mean F1-Score or Spearman ρ across five random seeds, post fine-tuning on the augmentation budget M. Performance is non-monotonic in M, and the best budget varies across tasks and backbones.

corresponding to the least evidence of conservation or acceleration. Thus, within the common mapped subset, eligible examples and realised per-sequence mask counts were identical, whereas the selected positions could differ. This constitutes a per-sequence rank-matched comparison of the two site-selection signals. The downstream models and training protocol were fixed, with $M = 6$ variants generated per eligible example.

Task/Metric	**Model family**	**Mean Δ** (PhyloP *vs.* PAML)	**PhyloP / Within tolerance / PAML**	**Δ Range** (PhyloP *vs.* PAML)
m6A ($F1$-Score)	DL baselines	-0.0049	$1/0/2$	-0.0139 to $+0.0016$
	DNA FMs	-0.0048	$0/0/2$	-0.0051 to -0.0046
	RNA FMs	$+0.0082$	$5/0/2$	-0.0162 to $+0.0297$
	Overall	$+0.0028$	$6/0/6$	-0.0162 to $+0.0297$
CRISPR (Spearman's ρ)	DL baselines	$+0.0205$	$2/1/0$	-0.0002 to $+0.0442$
	DNA FMs	-0.0052	$0/1/1$	-0.0109 to $+0.0005$
	RNA FMs	$+0.0014$	$2/3/2$	-0.0199 to $+0.0201$
	Overall	$+0.0051$	$4/5/3$	-0.0199 to $+0.0442$

Here, each model-level Δ is the difference between the corresponding five-seed means; family and overall entries are unweighted means across models. "Within tolerance" denotes the descriptive thresholds $|\Delta| < 0.0005$ for m6A and $|\Delta| < 0.005$ for CRISPR-ON, applied to unrounded model-level differences; it is not a statistical equivalence claim.

Neither evolutionary signal is uniformly superior. The overall mean differences are small: $+0.0028$ F1 on m6A and $+0.0051$ Spearman's ρ on CRISPR-ON, while effects vary across model families. These preliminary results therefore support PhyloP as a viable coordinate-aware site-selection signal, rather than demonstrating its superiority over BLAST/PAML. We view the approaches as complementary: coordinate-mapped PhyloP is natural when reliable reference coordinates and alignments exist, whereas BLAST/PAML remains applicable to coordinate-free sequences and unsupported organisms.

J Comparison with PriFold

PriFold (Yang et al., 2025a) is a complete RNA secondary-structure prediction framework; here we compare only its covariation-based augmentation operator with `PhyloAug`Both methods generate structure-compatible RNA variants using evolutionary information, but differ in their augmentation policies. PriFold uses annotated contact maps to identify paired nucleotides and substitutes base pairs according to fixed empirical transition probabilities. `PhyloAug` instead uses homolog-derived phylogenetic rates to identify mutation-permissive sites and a structure-conditioned infiller to generate replacements.

As PriFold requires annotated contact maps, applying it to our functional benchmarks would require predicted or unavailable target structures, introducing an additional confounder. We therefore restrict the comparison to the ArchiveII, RNAStrAlign, and bpRNA structure-prediction datasets. PriFold performs augmentation online, selecting a training example with probability 0.1 and replacing each annotated base pair with probability 0.3. For a matched-volume comparison, we apply its published substitution rule offline with replacement probability 0.3 to generate one variant per training sequence. `PhyloAug` follows the same offline $1\times$ augmentation volume. We use identical downstream models, splits, five seeds, and optimisation budgets.

Table 5: Comparison of PriFold and PhyloAug against non-augmented fine-tuning across RNA structure prediction datasets. Positive Δ indicates improved performance for the first method in each comparison. Win/Tie/Loss counts compare PhyloAug against PriFold.

Dataset	**Δ PriFold vs. NoAug**	**Δ PhyloAug vs. NoAug**	**Δ PhyloAug vs. PriFold**	**Win/Tie/Loss**
ArchiveII	$+0.023$	$+0.033$	$+0.010$	$8/1/0$
RNAStrAlign	$+0.006$	$+0.008$	$+0.002$	$4/3/2$
bpRNA	$+0.012$	$+0.026$	$+0.013$	$8/1/0$
Overall	$+0.014$	$+0.022$	$+0.008$	$20/5/2$

Entries first average each model's performance over five seeds and then average the model-level differences across the nine downstream backbones. Ties denote equal five-seed model means after rounding to three decimal places.

Across the 27 model–dataset comparisons, `PhyloAug` is higher in 20, tied in 5, and lower in 2, with an average advantage of $+0.008$ F1 over the PriFold-style offline operator. This result supports `PhyloAug`'s augmentation operator under matched synthetic-sequence volume; it is not a comparison against PriFold's complete prediction architecture or original online training schedule. The $1\times$ protocol also does not match realised edit distance or online augmentation diversity.

K Existing Assets and Licenses

We use existing datasets, pretrained models, and scientific software only for research benchmarking and reproducibility. We cite the original creators in the main paper and list licenses or usage terms in table 6. When an asset did not provide a standard SPDX-style license, we report the available usage terms stated by the asset provider.

Table 6: Existing assets used in this work and their licenses or usage terms.

Asset	Role in this work	License or usage terms
NCBI GenBank / Nucleotide	Homolog retrieval	NCBI places no restrictions on GenBank data; submitter IP may apply.
Rfam	RNA families, conservation, covariation	CC0.
RNAcentral	Pretraining/source corpus for RNA FMs	CC0.
UCSC Genome Browser data	SpliceBERT source data	Raw table/data files freely available; UCSC software has separate terms.
U12DB	Splice-type benchmark construction	GNU GPL.
BEACON	RNA benchmark tasks	Apache-2.0.
RNAscope	RNA benchmark tasks	CC BY 4.0 for the OpenReview paper; dataset terms should be checked with release.
ArchiveII	Secondary-structure benchmark	AGPL-3.0-or-later when accessed via MultiMolecule.
bpRNA	Secondary-structure benchmark	AGPL-3.0-or-later when accessed via MultiMolecule.
RNAStrAlign	Secondary-structure benchmark	AGPL-3.0-or-later via MultiMolecule; CC BY 4.0 for the Figshare release.
BLAST+	Homolog search	Public domain.
MAFFT	Multiple-sequence alignment	BSD-style license for core code; extensions may have separate terms.
FastTree	Phylogenetic tree estimation	GPL-3.0.
PAML	Site-wise rate estimation	GPL-3.0.
ViennaRNA	Secondary-structure prediction	Custom ViennaRNA terms permitting research, educational, and commercial use with redistribution restrictions.
CD-HIT-EST	Sequence clustering/leakage filtering	GPL / GPL-2.0-or-later.
GraKeL	Graph-kernel diagnostics	BSD-3-Clause.
Biological sequence kernels code	MMD diagnostics	MIT.
HyenaDNA	Target model	Apache-2.0.
SpliceBERT	Target model	BSD-3-Clause.
RNA-BERT	Target model	No explicit public license identified in the official repository; used for academic benchmarking and not redistributed.
RNA-MSM	Target model	MIT.
RNAErnie	Target model	MIT.
RNA-FM	Target model	MIT.
OmniGenome	Generator / target model	MIT.
RiNALMo-Giga	Target model	Apache-2.0 code; CC BY 4.0 model parameters.

L FUNCTIONAL EVALUATION RAW RESULTS

We report best functional results for all models below. AIDO.RNA converged on a subset of the main functional benchmarks, so we report successful runs for completeness, but exclude it from ablation studies because task-dependent NaN failures would make matched comparisons non-uniform. NA denotes pre-augmentation results, and PA denotes PhyloAug results.

Functional evaluation results. Table 7 reports the mean performance of each model before augmentation (NA) and after PhyloAug (PA) across classification and localization tasks. Overall, PhyloAug improves performance across most model–task pairs, with the clearest gains on the local motif-driven tasks. On RBP, all models improve in F1, with especially large gains for RNAErnie, RNA-BERT, and HyenaDNA. MCC also increases for all models with non-zero baseline MCC values, indicating that the F1 gains are not simply due to class-imbalance effects. A similar pattern is observed on m6A, where every model improves in both F1 and MCC; the largest MCC gains occur for RNA-FM and RNA-MSM. SpliceType shows a more mixed pattern: some already-strong models change only modestly, while HyenaDNA and RiNALMo-Giga benefit substantially from PhyloAug. For the context-dependent localization tasks, PhyloAug generally improves F1 on mRNA and miRNA localization, although MCC improvements are more model-dependent. In particular, miRNA localization shows strong gains for RNA-BERT, SpliceBERT, RNA-MSM, OmniGenome, and RiNALMo-Giga, while RNA-FM decreases under PA.

Sequence-to-function results. Table 8 reports Spearman's ρ for the sequence-to-function tasks. On `CRISPR`, `PhyloAug` improves Spearman performance for all models. The largest absolute gain is observed for RNA-BERT, followed by OmniGenome, RNA-FM, RNA-MSM, RiNALMo-Giga, and HyenaDNA. On `gRNA`, the improvements are smaller in magnitude, but several models still show consistent gains, including RNA-FM, RNA-MSM, HyenaDNA, and RiNALMo-Giga. SpliceBERT is the main exception on `gRNA`, where `PhyloAug` decreases performance relative to the non-augmented baseline. Overall, the sequence-to-function results suggest that `PhyloAug` is especially beneficial for CRISPR-ON and provides smaller but still measurable benefits for several `gRNA` models.

Table 7: Performance on classification and localisation tasks. Values are means over five seeds.

Task	Model	$F1_{NA}$	$F1_{PA}$	MCC_{NA}	MCC_{PA}
Local motif-driven tasks					
RBP	RNA-BERT	44.40±0	51.20±4.75	0.00±0	12.90±6.24
	SpliceBERT	64.92±0.92	69.10±1.86	35.12±1.76	41.60±1.45
	RNAErnie	44.40±0	54.60±6.09	0.00±0	16.40±6.23
	RNA-FM	65.76±1.71	68.03±1.58	32.79±2.01	40.64±1.53
	RNA-MSM	65.01±1.29	67.77±2.04	33.20±1.48	39.82±1.76
	OmniGenome	66.45±1.97	68.12±1.60	37.00±1.66	41.28±1.52
	HyenaDNA	58.21±5.25	62.81±3.50	22.82±6.13	29.29±4.73
	RiNALMo-Giga	67.74±1.99	69.84±1.63	38.51±1.98	44.78±1.44
	AIDO.RNA	63.84±1.82	66.19±1.54	32.42±2.98	36.95±2.31
m6A	RNA-BERT	59.53±0.26	62.61±1.25	23.19±4.79	27.48±3.19
	SpliceBERT	66.34±0.63	70.39±0.53	35.90±0.78	41.50±0.57
	RNAErnie	65.11±2.72	67.83±2.16	33.88±0.34	36.61±0.41
	RNA-FM	67.60±0.69	72.58±0.74	35.48±1.32	47.76±1.12
	RNA-MSM	62.93±4.76	68.27±3.79	27.09±9.18	48.98±4.75
	OmniGenome	63.38±2.72	67.44±0.82	27.94±4.58	34.74±1.48
	HyenaDNA	68.45±2.14	72.68±1.26	38.15±3.70	41.43±1.44
	RiNALMo-Giga	71.28±1.07	74.80±0.78	42.88±1.92	48.72±1.35
	AIDO.RNA	68.74±1.82	71.10±0.74	37.62±3.52	41.70±1.41
STP	RNA-BERT	75.44±0.95	77.80±0.86	84.33±0.55	83.66±0.59
	SpliceBERT	81.18±0.34	81.78±0.33	87.96±0.19	87.53±0.21
	RNAErnie	77.47±0.47	77.28±0.53	85.39±0.23	86.19±0.22
	RNA-FM	78.37±0.75	79.69±0.71	86.19±0.63	85.61±0.56
	RNA-MSM	79.06±0.97	79.46±0.94	85.40±0.54	89.17±0.53
	OmniGenome	84.28±0.24	83.64±0.27	89.17±0.31	89.71±0.30
	HyenaDNA	61.34±1.20	67.64±1.01	72.03±2.21	79.82±1.65
	RiNALMo-Giga	65.44±1.38	69.08±1.25	69.11±0.71	71.97±0.67
Context-dependent localisation tasks					
mRNA	RNA-BERT	43.31±4.77	46.19±3.06	53.80±5.9	51.26±5.30
	SpliceBERT	56.82±1.80	59.65±2.52	45.93±15.02	48.04±7.19
	RNAErnie	52.51±0.82	52.65±1.94	39.81±14.3	43.24±9.74
	RNA-FM	59.19±2.18	61.16±2.56	47.38±9.72	51.04±8.74
	RNA-MSM	50.48±2.36	54.62±2.33	56.38±1.31	55.97±1.29
	OmniGenome	60.38±0.71	61.16±0.74	56.02±0.56	57.17±0.85
	HyenaDNA	53.27±0.91	55.21±0.86	49.20±0.67	48.39±0.65
	RiNALMo-Giga	63.07±1.35	62.93±1.27	54.42±1.60	58.12±1.43
miRNA-SL	RNA-BERT	39.56±9.80	46.70±6.48	17.62±17.91	18.91±16.97
	SpliceBERT	51.94±2.75	59.76±3.04	30.88±18.44	43.14±16.59
	RNAErnie	54.88±2.54	58.10±1.50	44.30±2.60	43.40±2.88
	RNA-FM	54.44±1.97	53.40±2.05	43.02±1.88	38.90±5.14
	RNA-MSM	53.84±0.68	57.69±1.00	42.24±1.08	46.11±1.51
	OmniGenome	52.48±2.28	56.17±1.97	40.94±23.40	44.11±16.42
	HyenaDNA	53.62±4.97	54.06±4.97	38.18±15.18	39.97±11.46
	RiNALMo-Giga	55.84±4.14	56.80±3.28	39.72±22.26	41.81±16.33

Table 8: Performance on sequence-to-function tasks. Values are means over five seeds. These tasks are evaluated using Spearman's ρ; zero-valued MCC placeholders are omitted. Δ denotes PA$-$NA. Significance markers on Δ denote one-sided paired t-tests for PA $>$ NA: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Task	Model	ρ_{NA}	ρ_{PA}	$\Delta\rho$	t	p_{one}
Sequence-to-function tasks						
gRNA	RNA-BERT	0.3822±0.02	0.3912±0.02	$+0.0090^{*}$	2.31	4.09×10^{-2}
	SpliceBERT	0.4234±0.03	0.4117±0.04	-0.0117	-5.66	9.98×10^{-1}
	RNAErnie	0.3778±0.02	0.3801±0.02	$+0.0023$	0.55	3.05×10^{-1}
	RNA-FM	0.4690±0.02	0.4772±0.02	$+0.0082^{*}$	2.77	2.51×10^{-2}
	RNA-MSM	0.4556±0.02	0.4687±0.02	$+0.0131^{**}$	4.14	7.19×10^{-3}
	OmniGenome	0.4352±0.02	0.4389±0.02	$+0.0037$	0.90	2.09×10^{-1}
	HyenaDNA	0.3886±0.02	0.3971±0.02	$+0.0085^{*}$	3.57	1.17×10^{-2}
	RiNALMo-Giga	0.2464±0.12	0.2978±0.07	$+0.0514^{*}$	2.30	4.15×10^{-2}
	AIDO.RNA	0.3800±0.02	0.4011±0.02	NA	NA	NA
CRISPR	RNA-BERT	0.2322±0.02	0.2890±0.02	$+0.0568^{***}$	31.77	2.92×10^{-6}
	SpliceBERT	0.3198±0.03	0.3280±0.03	$+0.0082^{***}$	9.66	3.21×10^{-4}
	RNAErnie	0.3566±0.01	0.3590±0.01	$+0.0024$	0.97	1.93×10^{-1}
	RNA-FM	0.3646±0.02	0.4010±0.02	$+0.0364^{***}$	18.23	2.66×10^{-5}
	RNA-MSM	0.3252±0.02	0.3520±0.02	$+0.0268^{**}$	5.46	2.73×10^{-3}
	OmniGenome	0.3338±0.02	0.3771±0.02	$+0.0433^{***}$	11.60	1.58×10^{-4}
	HyenaDNA	0.3184±0.01	0.3479±0.01	$+0.0295^{**}$	7.09	1.05×10^{-3}
	RiNALMo-Giga	0.2450±0.07	0.2760±0.06	$+0.0310^{***}$	10.47	2.35×10^{-4}
	AIDO.RNA	0.2926±0.02	0.3185±0.02	NA	NA	NA

M Full Baseline Results

Shared augmentation setup. All augmented arms share one infiller and one mask-budget rule, so that they differ only in *which* positions are made mutable. The perturbation model is fixed: all replacements are generated with `OmniGenome`. Every selector is then normalised to a common masking budget of 15% of positions: if a selector nominates more sites than the budget, the excess is un-masked uniformly at random. Each augmented training file contains the original sequences plus their k variants, since the augmented file replaces the training set. Downstream fine-tuning uses the same established optimisation and compute across all methods. Because every arm is normalised to the same 15% masking budget, the arms differ only in which positions are nominated, not how many.

NoAug. Fine-tuning on the original training set only, with no additional sequences. This is the reference for every reported difference.

Duplicate. Each original training example is repeated k times, giving exactly the same number of rows, tokens and gradient steps per epoch as the corresponding k-matched augmented arm, while adding no new sequence content. This isolates the effect of the enlarged training budget from the effect of the generated variants.

Random. Mask positions are drawn uniformly at random over all positions of the sequence, at the same 15% masking ratio as `PhyloAug`, and in-filled by the same model. No conservation, structure or motif information is used, and no positions are protected: this is the naive-perturbation control.

MSA-only. Mutable sites are selected from homolog-alignment variability alone, with no phylogenetic rate estimation. Per-column Shannon entropy is computed over the homolog MSA and columns with entropy ≥ 1.0 are declared mutable; an adaptive variant starts from a neutrality probability threshold of 0.8 and relaxes it to 0.6 when fewer than 15% of sites pass, or tightens it to 0.9 when more than 33% pass. No tree is inferred and no substitution model is fitted, so site selection reflects observed column diversity rather than estimated evolutionary rate.

Struct-Infill. Secondary structure used to select sites with no base-pairing: the minimum-free-energy structure is computed with `ViennaRNA` 2.7.2.

Table 9: Performance across augmentation strategies. Values denote mean ± standard deviation over five seeds, rounded to two significant figures.

Task	Model	No Aug	Rand.	Dup.	Struct	MSA	PhyloAug
CRISPR On-Target	SpliceBERT	0.32±0.035	0.28±0.021	0.26±0.018	0.28±0.020	0.19±0.016	0.33±0.033
	RNA-BERT	0.23±0.018	0.27±0.012	0.23±0.0090	0.27±0.024	0.18±0.013	0.29±0.026
	RNAErnie	0.36±0.017	0.31±0.055	0.35±0.029	0.35±0.012	0.27±0.013	0.36±0.019
	RNA-MSM	0.33±0.023	0.30±0.020	0.33±0.025	0.34±0.020	0.18±0.014	0.35±0.024
	RNA-FM	0.36±0.021	0.33±0.011	0.33±0.055	0.32±0.0077	0.22±0.048	0.40±0.026
	RiNALMo-Giga	0.25±0.072	0.088±0.12	0.23±0.064	0.022±0.068	0.18±0.0074	0.28±0.051
miRNA Sub-Local	SpliceBERT	0.52±0.027	0.51±0.018	0.53±0.060	0.48±0.012	0.51±0.028	0.60±0.030
	RNA-BERT	0.40±0.098	0.46±0.041	0.50±0.057	0.44±0.023	0.42±0.021	0.47±0.065
	RNAErnie	0.55±0.025	0.54±0.047	0.56±0.0050	0.54±0.0047	0.54±0.0095	0.58±0.015
	RNA-MSM	0.54±0.0068	0.54±0.011	0.55±0.040	0.48±0.072	0.47±0.051	0.58±0.010
	RNA-FM	0.54±0.020	0.00±0.00	0.55±0.010	0.53±0.040	0.45±0.037	0.53±0.021
	RiNALMo-Giga	0.56±0.041	0.00±0.00	0.55±0.044	0.56±0.047	0.56±0.049	0.57±0.035
m6A	SpliceBERT	0.67±0.0063	0.65±0.0096	0.69±0.0084	0.68±0.0053	0.67±0.0083	0.71±0.0053
	RNA-BERT	0.59±0.082	0.56±0.065	0.58±0.056	0.55±0.062	0.55±0.068	0.62±0.028
	RNAErnie	0.65±0.027	0.65±0.040	0.65±0.037	0.59±0.041	0.57±0.058	0.68±0.022
	RNA-MSM	0.63±0.048	0.64±0.041	0.66±0.044	0.66±0.039	0.64±0.059	0.70±0.038
	RNA-FM	0.68±0.0069	0.69±0.0057	0.70±0.011	0.70±0.0059	0.67±0.0076	0.73±0.0074
	RiNALMo-Giga	0.71±0.011	0.72±0.034	0.72±0.037	0.72±0.027	0.71±0.042	0.75±0.0078
RBP	SpliceBERT	0.6494±0.0092	0.627±0.011	0.690±0.006	0.685±0.016	0.680±0.010	0.6910±0.019
	RNA-BERT	0.444±0.0	0.444±0.000	0.445±0.001	0.445±0.001	0.444±0.000	0.512±0.048
	RNAErnie	0.440±0	0.448±0.006	0.445±0.001	0.444±0.000	0.444±0.000	0.546±0.061
	RNA-MSM	0.6448±0.0129	0.565±0.022	0.669±0.007	0.626±0.011	0.601±0.013	0.677±0.020
	RNA-FM	0.654±0.017	0.568±0.007	0.631±0.007	0.638±0.022	0.612±0.026	0.680±0.016
	RiNALMo-Giga	0.677±0.019	0.645±0.015	0.692±0.014	0.639±0.021	0.662±0.021	0.698±0.016

Comparison with alternative augmentation baselines. Table 9 compares PhyloAug against several alternative augmentation strategies, including random augmentation, duplication, structure in-filling, and MSA-based augmentation. Across CRISPR On-Target and m6A, PhyloAug is consistently the strongest or near-strongest strategy across models. This is particularly clear for m6A, where PhyloAug gives the best score for every listed model. On CRISPR On-Target, MSA-based augmentation generally underperforms, and some alternative augmentations provide only modest or inconsistent gains over the non-augmented baseline. On miRNA Sub-Local, the pattern is more mixed: PhyloAug performs best for several models, including SpliceBERT, RNAErnie, RNA-MSM, and RiNALMo-Giga, but duplicate augmentation is stronger for RNA-BERT and RNA-FM. The zero-valued random augmentation entries for some models indicate collapsed or failed runs, reinforcing that not all augmentation strategies are equally stable. Taken together, these baselines suggest that PhyloAug is not merely benefiting from increased training-set size, but provides a more reliable augmentation signal than random or duplicate-based controls.

Across the raw results, paired statistical tests, sequence-to-function evaluations, and augmentation baselines, PhyloAug provides broad but task-dependent improvements. The strongest and most consistent gains occur on motif-driven classification tasks and CRISPR On-Target prediction, while localization tasks show more

model-specific behaviour. The comparison against random, duplicate, structure-based, and MSA-based baselines further suggests that the observed improvements are specific to the phylogenetically informed augmentation procedure rather than a generic consequence of adding more training examples.

N LABEL-BUDGET SCALING RESULTS

Results. Table 10 reports mean F1 scores for m6A classification across six labelled-data fractions. PhyloAug improves mean performance in 40 of the 42 model–budget comparisons. Averaged across the seven backbones, the improvement increases from 1.16 F1 percentage points at 1% of the training data to 3.40 points at 100%, where average F1 rises from 65.98 to 69.38. With 20% of the labelled training data, PhyloAug achieves an average F1 of 67.27, exceeding the 65.98 obtained by unaugmented fine-tuning with the full training set.

The magnitude of improvement varies across backbones. At 100% data, RNA-MSM shows the largest gain, increasing from 62.93 to 68.24, followed by SpliceBERT, which improves from 67.32 to 71.30. RNA-FM improves at every evaluated fraction and reaches 72.11 with full-data augmentation, compared with 68.52 without augmentation. RiNALMo-Giga achieves the highest absolute F1, reaching 74.80 compared with an unaugmented baseline of 71.28. The two decreases occur at the 5% fraction: SpliceBERT falls from 65.37 to 63.25, and OmniGenome from 64.38 to 63.63. These descriptive results suggest that PhyloAug benefits this benchmark across a range of label budgets, with larger average gains at higher data fractions, while retaining model-specific exceptions.

Table 10: m6A modification classification: mean F1 score (%) across subset sizes.

Aug	Model	1%	5%	10%	20%	50%	100%
Non-Aug	SpliceBERT	61.25	65.37	65.47	66.18	66.45	67.32
	RNA-BERT	52.49	57.90	58.64	58.81	58.88	59.53
	RNAErnie	58.66	59.44	62.68	63.78	64.10	65.60
	RNA-MSM	56.46	58.12	59.34	60.17	61.34	62.93
	RNA-FM	58.19	61.07	63.15	66.21	67.89	68.52
	RiNALMo-Giga	62.12	68.32	70.70	70.90	70.90	71.28
	OmniGenome	60.86	64.38	65.64	65.91	66.27	66.68
PhyloAug	SpliceBERT	61.75	63.25	66.52	68.14	70.05	71.30
	RNA-BERT	55.44	59.18	60.03	60.39	61.15	62.11
	RNAErnie	58.91	61.60	66.19	67.13	67.81	68.60
	RNA-MSM	58.66	61.22	63.52	66.04	67.48	68.24
	RNA-FM	59.11	65.28	67.14	68.68	70.51	72.11
	RiNALMo-Giga	62.86	69.82	71.82	73.11	73.78	74.80
	OmniGenome	61.41	63.63	65.72	67.38	67.76	68.50

O RAW STRUCTURAL AND EVOLUTIONARY RESULTS

Results. We report the raw value results across our evolutionary and structural experiments. Across both tasks, `PhyloAug` generally improves both F1-score and MCC relative to the non-augmented setting. The gains are especially consistent on the evolutionary tasks, where nearly every model improves on both ConsSites and Covariation. Improvements are particularly large for OmniGenome, SpliceBERT, RNA-FM, and HyenaDNA, with PhyloAug often increasing MCC by more than 10 points, indicating better calibrated binary predictions rather than only higher F1.

The structural tasks show a similar trend, though the magnitude of improvement depends on the benchmark and base model. ArchiveII exhibits some of the largest gains, most notably for RNA-BERT, which increases from 37.9 to 74.9 F1 and from 10.1 to 60.9 MCC under PhyloAug. bpRNA also benefits broadly, with strong gains for RNA-MSM, SpliceBERT, HyenaDNA, and RNA-FM. StrAlign is already a high-performing setting for most models, so the absolute improvements are smaller, but PhyloAug still generally yields consistent gains. The main exception is RNAErnie on StrAlign, where performance is essentially unchanged or slightly lower. Overall, these appendix results support the conclusion that evolutionary augmentation is broadly beneficial across RNA sequence-function and structure-related tasks, with the largest impact appearing in settings where the non-augmented baseline leaves more room for improvement.

Table 11: Performance across evolutionary and structural RNA tasks. We report F1-score and MCC under non-augmented and `PhyloAug` settings.

Evolutionary tasks

	Conserved Sites				Covariation			
	Non-Aug		PhyloAug		Non-Aug		PhyloAug	
Model	F1	MCC	F1	MCC	F1	MCC	F1	MCC
RNA-BERT	50.5±2.7	11.0±2.1	58.2±1.8	09.2±1.4	71.8±0.2	43.7±0.5	75.3±1.0	50.8±1.8
SpliceBERT	72.4±1.7	57.8±1.5	80.2±1.0	68.7±0.7	75.0±0.2	51.0±1.0	83.7±1.6	67.7±3.1
RNAErnie	61.7±1.9	25.2±1.2	66.8±1.2	32.4±1.0	73.3±1.6	48.3±2.2	78.0±1.0	56.5±1.9
RNA-FM	79.6±1.5	59.2±1.1	85.7±1.2	69.6±1.0	81.4±0.3	63.4±0.7	87.5±0.8	75.1±1.5
RNA-MSM	69.2±1.3	53.6±0.9	76.3±0.9	61.3±0.8	73.8±0.3	47.7±0.8	78.3±0.6	56.6±1.2
OmniGenome	81.0±1.6	62.2±1.1	90.7±1.3	76.5±1.0	79.9±2.7	60.5±4.9	88.6±0.6	77.2±1.1
HyenaDNA	63.3±1.6	27.5±1.2	69.2±1.4	35.8±1.1	73.9±0.4	48.5±0.9	81.8±1.6	63.6±3.3

Structural tasks

	ArchiveII				bpRNA			
	Non-Aug		PhyloAug		Non-Aug		PhyloAug	
Model	F1	MCC	F1	MCC	F1	MCC	F1	MCC
RNA-BERT	37.9±4.2	10.1±3.8	74.9±0.5	60.9±0.7	50.3±2.5	26.3±3.1	56.7±2.2	34.0±2.8
SpliceBERT	85.3±0.3	77.6±0.5	88.7±0.2	82.6±0.3	73.3±1.3	61.2±1.5	80.7±1.1	67.7±1.6
RNAErnie	72.7±0.1	57.7±0.2	82.1±0.1	72.5±0.2	50.0±0.6	27.7±0.9	50.4±0.5	28.3±0.7
RNA-FM	88.2±0.1	81.8±0.2	89.8±0.1	84.2±0.1	79.1±0.4	65.2±0.8	81.4±0.3	71.1±0.8
RNA-MSM	80.1±0.7	69.3±1.1	85.3±0.8	77.4±1.2	66.8±1.3	49.7±1.4	78.3±1.1	66.3±1.2
OmniGenome	92.0±0.0	87.5±0.0	93.6±0.1	89.9±0.2	81.2±0.5	70.4±0.6	84.1±0.3	74.8±0.4
HyenaDNA	77.4±0.2	65.1±0.3	82.3±0.1	72.6±0.2	55.4±1.1	29.2±1.7	60.1±1.3	38.8±1.6

	StrAlign			
	Non-Aug		PhyloAug	
Model	F1	MCC	F1	MCC
RNA-BERT	90.3±0.4	85.0±0.5	92.5±0.3	88.3±0.3
SpliceBERT	96.4±0.3	94.3±0.3	97.2±0.1	95.5±0.1
RNAErnie	85.0±0.1	76.5±0.2	84.8±0.0	76.3±0.0
RNA-FM	97.1±0.2	95.4±0.3	97.9±0.0	96.6±0.0
RNA-MSM	95.1±0.2	92.4±0.3	96.6±0.1	94.7±0.1
OmniGenome	98.2±0.0	97.4±0.0	98.4±0.1	97.6±0.1
HyenaDNA	93.8±0.1	90.4±0.1	95.1±0.1	92.4±0.1

P EXPERIMENTAL DETAILS

We fine-tune all models for up to 50 epochs using the AdamW optimiser with its parameters set to $\beta 1 = 0.9$ and $\beta 2 = 0.999$, a shared learning rate grid, weight decay of 0.01, bf16 precision, and a batch size of 8. Early stopping is applied with a patience of 5 epochs based on validation performance, and the best validation checkpoint is used for test evaluation. Unless otherwise stated, the same optimiser, batch size, learning rate, early-stopping rule, and training budget are used across non-augmented, baseline-augmented, and `PhyloAug` runs to ensure fair comparison. All results are reported over five random seeds.

For augmentation experiments, the number of augmented variants per eligible training sequence is selected from $M \in \{1, 2, 4, 6, 8, 10, 12\}$ using the validation split, with the same search space applied to all augmentation policies. All experiments were run on four GH200 accelerators from Isambard-AI. Homolog retrieval, alignment, phylogenetic rate estimation, augmentation generation, and downstream fine-tuning were run as separate stages, with intermediate outputs cached where possible to avoid redundant computation.

To ensure practical usability, our pipeline optimises the three most computationally intensive steps: homolog retrieval, phylogenetic rate estimation, and Rfam alignment. We use BLAST+ (Camacho et al., 2009) in megablast or blastn mode (with sequences < 150 nucleotides) for efficient homolog searches, MAFFT (Katoh and Standley, 2013) to create Multiple-Sequence-Alignments from homologous sequences, FASTTREE (Price et al., 2010) to build phylogenetic trees. Our sequences are processed in batches to parallelise site-wise rate estimation with PAML (Yang, 2007), and implement a cache for Rfam covariance models to avoid redundant alignments. These optimisations help reduce the computational burden when augmenting large-scale fine-tuning tasks.

We release the data and code at the anonymous repo: https://anonymous.4open.science/r/PhyloAug

Q PAML IMPLEMENTATION

Software. Site-wise evolutionary rates are estimated with `baseml` from PAML version 4.10.9 (7 May 2025). Alignments are built with MAFFT and tree topologies are inferred with FastTree (`-nt -gtr`); PAML is used only for rate estimation, not for tree search.

Substitution model. We use the JC69 model (`model = 0`: equal base frequencies and a single substitution rate) with among-site rate variation modelled by a discrete gamma distribution of `ncatG` $= 10$ categories. The gamma shape parameter α is estimated by maximum likelihood (`fix_alpha = 0`, initial value $\alpha = 1$). Branch lengths are re-estimated from the alignment rather than taken from the input tree (`fix_blength = 0`), and no molecular clock is assumed (`clock = 0`).

Inference settings. Rates are obtained by empirical Bayes: `RateAncestor = 1` makes `baseml` report the posterior mean rate of each alignment column, which we read from its `rates` output. Columns containing gaps or ambiguous characters are retained and treated as missing data (`cleandata = 0`) rather than deleted, and the alignment is analysed as a single partition (`Mgene = 0`).

Rate normalization and site classification. The discrete-gamma rate distribution is parameterized to have mean one. Candidate editing sites are restricted to the interval $0.8 \leq r \leq 1.2$, following Section 3.3. This interval is an augmentation heuristic and does not establish selective neutrality or preservation of downstream labels.

R COMPUTE DETAILS

Compute resources. Experiments were run on NVIDIA GH200 nodes on Isambard-AI. Each node used an ARM Neoverse-V2 aarch64 CPU with 288 logical CPU cores, configured as four sockets with 72 cores per socket, and provided 856 GiB of system memory with no swap configured. Reproducing the full pipeline requires downloading the NCBI `nt` database, which occupies approximately 500 GB, plus at least 300 GB for pretrained models, benchmark data, cached homologs, alignments, generated augmentations, and intermediate outputs. We therefore recommend at least 1 TB of available storage when retaining all intermediate caches.

Homolog retrieval used BLAST+ with 16 CPU threads. Runtime varied substantially across query sequences depending on sequence length, repetitiveness, database hit density, and whether `megablast` or `blastn` was used. On our benchmark workload, observed throughput was approximately 15 minutes per $1,000$ queries with `megablast` and approximately 60 minutes per $1,000$ queries with `blastn`, corresponding to roughly 4 and 16 allocated CPU-hours per $1,000$ queries, respectively. These values should be interpreted as workload-level averages rather than per-sequence guarantees. `MAFFT` alignment and `FastTree` phylogeny construction were CPU-only and typically completed within minutes after homolog retrieval for most benchmarks. `PAML` rate estimation was also CPU-only and was parallelised across sequences; runtime varied with alignment length and homolog count. Augmentation generation consisted of batched model inference and was substantially cheaper than downstream fine-tuning. Fine-tuning runtime varied by task, sequence length, backbone size, and augmentation budget.

Compute accounting. Compute estimates are reported as allocated resource estimates rather than exact hardware utilisation. CPU-hours are computed as wall-clock time multiplied by the number of allocated CPU threads, and GPU-hours are computed as wall-clock time multiplied by the number of allocated GPUs. Because runtime depends on sequence length, homolog density, model backbone, augmentation budget, and kernel choice, we report approximate ranges and workload-level averages rather than per-sequence guarantees. Where available, estimates were obtained from scheduler logs; otherwise, they were estimated from representative runs using the same resource configuration.

At the stage level, homolog retrieval required approximately 0.25 allocated CPU-hours per $1,000$ `megablast` queries or 16 allocated CPU-hours per $1,000$ `blastn` queries, while augmentation generation and fine-tuning accounted for most GPU-hours; MSA, phylogeny, `PAML`, and MMD costs varied with benchmark size and are included in the aggregate totals below.

Across the reported experiments, we estimate that the full reproducible suite required approximately 250–350 allocated CPU-hours and 300–400 allocated GPU-hours. These totals aggregate preprocessing, augmentation generation, fine-tuning, and distributional diagnostics. Preliminary debugging and failed exploratory runs were not logged with sufficient granularity to estimate separately; this additional compute is therefore not included in the reported total.