# REPORTER BENCHMARKS GENOMIC REPRESENTATIONS AGAINST REGULATORY ACTIVITY EXPERIMENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Pretraining is increasingly being used to improve the performance of genomic sequence models, but assessing whether pretrained representations broadly capture sequences' gene-regulatory functions remains a challenge. Here, we introduce REPORTER, a set of 22 transfer-learning tasks that we use to evaluate several genomic language models (gLMs) and AlphaGenome—a leading functional-genomics model—as pretrained backbones. Our task data are curated from 14 publicly available massively parallel reporter assay experiments that altogether screened over 2.5 million DNA sequences from across the tree of life for transcriptional, splicing, or UTR-mediated regulatory activity. We find that the most capable gLMs can outperform models trained from scratch on nearly all of our tasks, but fall short of AlphaGenome on most of our animal tasks. Our results provide empirical insight into pretraining, backbone selection, and transfer learning. Looking forward, REPORTER can serve model developers as a standard benchmark and design target.

## 1 INTRODUCTION

Genomic language modeling—large-scale self-supervised learning on genomic sequences—is a promising approach to pretraining that can support diverse genome interpretation tasks (Benegas et al., 2025c). Genomic language models (gLMs) could enable highly accurate prediction of genetic variant effects across molecular, cellular, and clinical phenotypes; facilitate genome engineering; and provide multimodal models with a granular understanding of genomics. Realizing this potential requires rigorous benchmarks of what these models learn and how broadly their representations transfer. However, the diversity of genomic elements, cellular contexts, and downstream tasks has made their evaluation challenging.

Massively parallel reporter assays (MPRAs)—experiments that measure the regulatory activity of up to millions of DNA sequences in parallel—provide a natural testbed for gLMs. Predicting MPRA measurements is itself an important application, as MPRAs are routinely used to probe causal sequence-to-function relationships. Moreover, MPRAs span diverse regulatory mechanisms and experimental conditions, generate large-scale quantitative measurements, and share a common structure that we find allows us to model these assays within a unified framework. These properties make MPRAs particularly well suited for testing the quality and generality of pretrained genomic representations.

Here, we introduce REPORTER (Regulatory Element Prediction with Oligonucleotides and Reporters for Transfer Evaluation of Representations), a suite of 22 transfer-learning tasks curated from 14 MPRA studies. It comprises more than 2.5 million unique sequences from animals, plants, yeast, and prokaryotes, and spans transcription, splicing, and UTR-mediated regulation across diverse experimental systems and conditions. The suite provides a broad evaluation of pretrained genomic representations across regulatory functions, experimental settings, and species.

We apply REPORTER to benchmark 25 gLM backbones, including six species-specific MLM baselines that we pretrained, alongside AlphaGenome, a leading human functional-genomics model recently repurposed as a pretrained backbone (Murphy et al., 2026). The strongest gLMs outperform training from scratch on 21 of 22 tasks, demonstrating that self-supervised sequence pretraining can



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.

yield broadly transferable representations of regulatory function, while substantial variation among gLMs reveals important differences in representation quality. AlphaGenome leads in performance across many tasks—including for species beyond humans—demonstrating that functional-genomics supervision can produce surprisingly transferable representations. We also find that REPORTER distinguishes gLMs with similar performance on existing evaluations, revealing capabilities not captured by conventional benchmarks. Together, our results show that REPORTER provides a systematic framework for measuring the quality and generality of pretrained genomic representations.

## 2 Background

### 2.1 Massively parallel reporter assays

MPRAs have a basic, general design (Fig. 1). First, a library of oligonucleotide sequences (oligos) matching genomic regions of interest is synthesized and inserted into DNA constructs at a fixed position. These constructs contain a simple reporter system for measuring the regulatory activity of each oligo—typically including a promoter, a reporter gene, as well as barcodes that associate individual oligos with their corresponding reporter transcripts. Construct designs include oligos upstream of a reporter gene to measure transcriptional activity, within a UTR to measure post-transcriptional regulation or translation, or within a gene to evaluate splicing (Fig. 1A). Once synthesized, each construct is introduced into a cell, which produces different amounts and forms of reporter RNA depending on the oligo sequence. Sequencing then quantifies each oligo's representation in the input library and its associated reporter output, directly or after sorting cells or transcripts into bins, allowing its regulatory activity to be estimated (Fig. 1B).

The studies we obtained our task data from (Table 1) use two assay designs which differ in the form of their readouts (Fig. 1C). In an *input–output count assay*, each oligo is mapped to an input and an output count, and its activity is defined as the log-ratio of output to input counts. In most assays, these counts measure the oligo's representation in the starting library and the abundance of its associated reporter product, respectively. In splicing assays, they instead count transcripts that skip or include a cassette exon, respectively. By contrast, a *binned count assay* sorts cells or transcripts into ordered bins according to reporter output and counts oligos within each bin. Each oligo therefore yields a vector of counts, from which a scalar activity measure can be derived (e.g., from the expected bin index after normalizing counts within each oligo).

Different MPRA designs can address different questions about regulatory function. For example, screening the same oligo library across experimental systems (e.g., cell lines) or conditions can reveal context-specific activity. MPRAs can also measure genetic variant effects by comparing the activities of paired oligos carrying the reference and alternate alleles in the same flanking context (Fig. 1C). Extending this approach, mutagenesis libraries enable systematic mapping of how mutations influence a genomic region's function by introducing substitutions, insertions, or deletions.

### 2.2 Transfer learning

Developing a comprehensive pretraining pipeline for every single downstream task is impractical. An alternative is to leverage pretrained backbones that extract features useful for a broad range of genome interpretation tasks to infer the effect of variants measured in these assays (Bommasani et al., 2021). One approach—taken by models such as DNABERT-2, Nucleotide Transformer, GPN, Caduceus, Evo 2, and MarinDNA—uses masked language modeling (MLM) or causal language modeling (CLM) to learn which nucleotides are likely in a given sequence context (Zhou et al., 2024; Dalla-Torre et al., 2025; Benegas et al., 2023; Schiff et al., 2024; Brixi et al., 2026; Open Athena, 2026). Because natural selection shapes genomic sequence variation, these objectives can encourage models to learn features associated with selective constraint and biological function. Alternative self-supervised objectives have also been explored. For example, JEPA-DNA (Larey et al., 2026) augments language modeling with a task that predicts the hidden-state embedding of an unmasked sequence from its partially masked counterpart.

A second approach uses supervised learning. Models trained to predict functional genomics assays—such as AlphaGenome (Avsec et al., 2026)—can serve as pretrained backbones (Murphy et al., 2026). Models can also be supervised on records of molecular evolution. For example, PhyloGPN outputs a Markov process describing the evolution of any site in the genome in a manner that

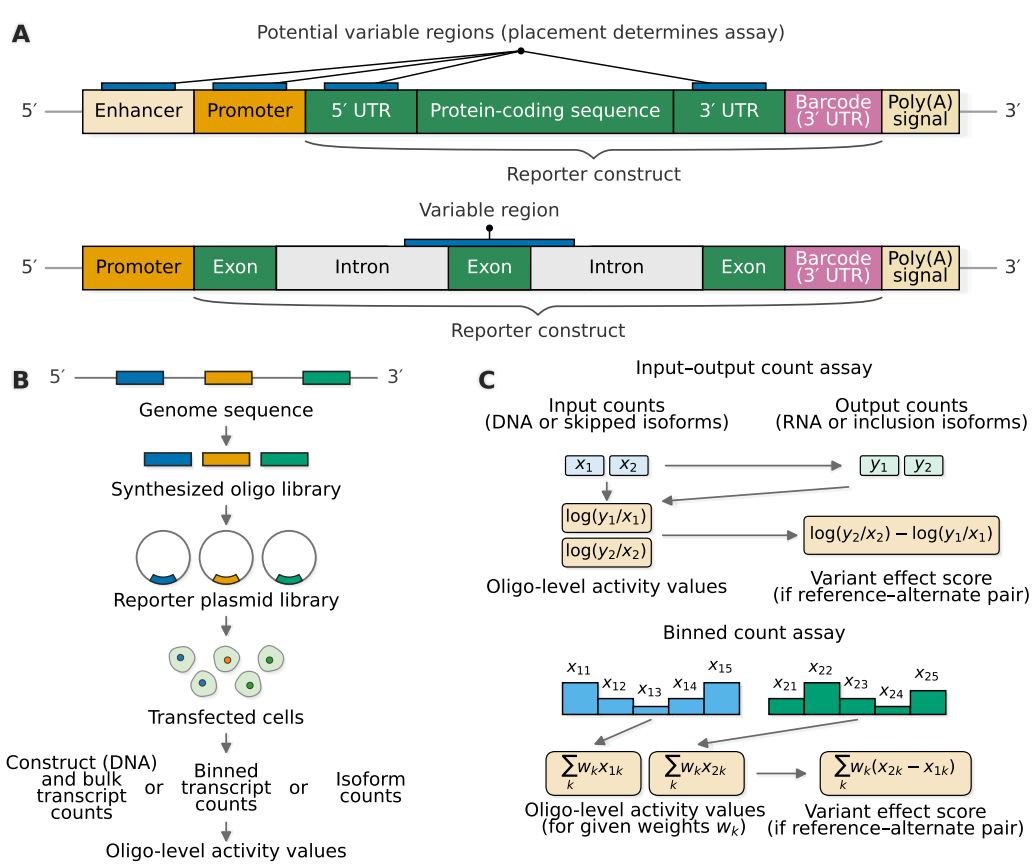


Figure 1: **MPRA designs, experimental workflow, and readouts.** **(A)** Regulatory assays vary oligos at enhancer, promoter, 5′ UTR, or 3′ UTR positions (top). Splicing assays vary a cassette exon and its flanking introns (bottom). **(B)** Genome-derived oligos are synthesized, cloned into plasmids, and introduced into cells. Sequencing measures input DNA and reporter output as bulk transcript, binned, or isoform counts. **(C)** Input–output assays yield log count ratios, and binned assays yield mean ordered-bin indices. Differences between matched reference and alternate activities give variant-effect scores. The schematic omits smoothing and normalization (Section 3.3).

makes it functionally a gLM (Albors et al., 2025). Supervised and self-supervised objectives can also be combined: Boshar et al. (2025) used functional genomics data to post-train a self-supervised model, while Gamba (Consens et al., 2026) incorporates evolutionary information by augmenting language modeling with the auxiliary task of predicting PhyloP scores (Pollard et al., 2010).

After pretraining, models must be adapted to downstream tasks. A simple approach freezes the backbone, extracts its hidden-state embeddings, and trains a lightweight predictor on them. Full fine-tuning instead adapts the representation itself but can be prohibitively expensive. Parameter-efficient fine-tuning provides a middle ground. For example, Low-Rank Adaptation (LoRA) (Hu et al., 2022) freezes the pretrained weights while learning low-rank perturbations to a subset of weight matrices, allowing representations throughout the model to adapt with far fewer trainable parameters.

### 2.3 Previous Work

Existing benchmarks have evaluated pretrained genomic representations across regulatory tasks, but their coverage of reporter assays remains limited. Genomic Benchmarks, GUE, and BEND (Grešová et al., 2023; Zhou et al., 2024; Marin et al., 2024) cover regulatory annotations and chromatin features. GUANinE (Robson and Ioannidis, 2026) includes MPRA data, but its coverage is limited to two synthetic yeast promoter tasks. DNALongBench (Cheng et al., 2025) evaluated long-range epigenomic and transcriptional prediction, and DART-Eval (Patel et al., 2024) evaluated chromatin

accessibility and variant effects, but neither used reporter assay data. Closest to our work, Tang et al. (2025) assessed four gLMs across six DNA- and RNA-regulatory tasks, including lentiMPRA prediction and MPRA-based variant-effect evaluation. These analyses covered two human cell lines, with saturation mutagenesis limited to four regulatory elements, and included frozen and fine-tuned models.

Zero-shot variant-effect prediction (VEP), in which models score variants by comparing the likelihoods they assign to reference and alternate alleles without task-specific training, offers another way to evaluate gLM representations. If language modeling adequately captures evolutionary constraints, the likelihood assigned to an allele should reflect its relative fitness. Zero-shot VEP could therefore serve as a proxy for representational quality, in the same way that perplexity can for language models in NLP (Bommasani et al., 2021). One such evaluation task distinguishes pathogenic ClinVar variants (Landrum et al., 2018) from common gnomAD variants (Chen et al., 2024). Another is to distinguish fine-mapped trait-associated variants from matched controls, as in TraitGym (Benegas et al., 2025b). Despite these tasks being primary metrics for gLMs, we have lacked empirical evidence on whether zero-shot VEP performance predicts how effectively gLMs can adapt to regulatory element interpretation tasks.

# 3 METHODS

## 3.1 DATA PROCESSING

To construct our benchmarks, we obtained and processed data from several published studies (Appendix A.1). To limit distribution shift from pretraining, we included only oligos derived from native genomic sequences or their variants and provided pretrained backbones with enough flanking genomic context to match their pretraining context lengths, subject to the model-specific limits described in Appendix B.2. The assayed oligos range from 50 to 249 bp, with most tasks using 150–200 bp fragments (Table S1). Flanking genomic sequence provides additional context to the backbone, while the prediction target remains the activity of the assayed fragment. We describe sequence selection and genomic-coordinate recovery in Appendix A.1.

We processed each study into a standardized dataset with four groups of columns: sequence and library-design metadata, genomic coordinates for retrieving flanking context, leakage-controlled cluster assignments, and study-specific assay readouts. We group oligos linked by at least $90\%$ sequence identity and keep related constructs together. We assign groups jointly across related tasks, so shared sequences and parent elements receive the same training, validation, or test assignment (Appendix A.1). Assay readouts comprise input–output counts per condition—typically DNA and RNA counts, or excluded and included isoform counts for `human_splicing`—or counts per bin for binned assays. For five studies that released processed activity statistics rather than raw counts, we reconstructed cluster-balanced pseudocounts that preserve the published unsmoothed activity ratios (Appendix A.1). In Appendix A.1, we describe the dataset schema.

## 3.2 TASKS

We condensed our processed datasets into 22 tasks that emulate how a savvy practitioner could use these data to train a model. Each task entails fitting adapters for a backbone on a training set, selecting checkpoints based on validation loss, and evaluating them on a test set. In some cases a task comprises all of the data from a study. In others, we separate a study into several tasks, either because it assays distinct libraries in different organisms or because it contains sub-libraries designed to answer different biological questions. In Table 1, we summarize the tasks, and in Table S1, we report their study origins, oligo lengths, and split sizes.

For studies where readouts were obtained for multiple cell lines or conditions, our tasks combine separate subtasks—one for each cell line or condition. Therefore, each model produces subtask-specific predictions while sharing a common backbone and training procedure across subtasks. Task-level metrics are obtained by aggregating the corresponding subtask-specific metrics.

For each task or subtask, we use a loss function suited to the type of counts reported by the assay. For input–output count assays—which report an input count $x_i$ and an output count $y_i$ for an oligo $i$—we use a negative-binomial excess negative log-likelihood (NLL). We associate each oligo with

Table 1: **Overview of REPORTER tasks.** The code `at` denotes *Arabidopsis*, and `{species}` represents a separate task for each listed species.

| Task ID | Prediction target |
|---|---|
| `human_tx_s26_snp` | Transcriptional effects of fine-mapped complex-trait and eQTL variants. |
| `human_tx_s26_satmut` | Transcriptional effects of saturation mutagenesis across 128 elements. |
| `human_tx_a25` | Transcription driven by ENCODE cCREs in three human cell lines. |
| `human_tx_l25` | Transcriptional effects of psychiatric GWAS variants in neural progenitors. |
| `human_5utr` | Ribosome loading of native human 5′ UTRs and their natural variants. |
| `human_splicing` | Exon inclusion of splice-site variants in five human cell lines. |
| `human_3utr` | Post-transcriptional effects of human 3′ UTR variants across six cell lines. |
| `mouse_tx` | Activation and repression by mouse CRX-bound loci and motif perturbations. |
| `drosophila_enhancer` | Transcription driven by genome-tiling *Drosophila* fragments. |
| `yeast_promoter` | Transcription driven by native *S. cerevisiae* promoters across 18 bins. |
| `at_promoter_j21` | Transcription driven by native *Arabidopsis* promoters. |
| `{species}_promoter` | Transcription driven by native sorghum and maize promoters. |
| `{species}_enhancer` | Transcription driven by accessible regions from *Arabidopsis*, sorghum, tomato, and maize. |
| `at_promoter_v24_native` | Transcriptional effects of sequences flanking *Arabidopsis* core promoters. |
| `at_promoter_v24_gatc` | Transcriptional effects of GATC-motif mutations in *Arabidopsis*. |
| `ecoli_promoter_tss` | Transcription driven by candidate *E. coli* transcription start sites. |
| `ecoli_promoter_scanmut` | Transcriptional effects of scanning mutagenesis in *E. coli* promoters. |
| `prokaryote_promoter` | Transcription driven by sequences from 184 prokaryotic genomes in three hosts. |

a model-predicted activity $\theta_i$ and, for $x_i > 0$, model

$$y_i \mid x_i, \theta_i \sim \mathrm{NB}(x_i, \hat{p}_i), \qquad \hat{p}_i = 1/(1 + e^{\theta_i}), \tag{1}$$

where $x_i$ is the size parameter and $\hat{p}_i$ is the success probability, giving $\mathbb{E}[y_i \mid x_i, \theta_i] = x_i e^{\theta_i}$. For $x_i = 0$ and $y_i > 0$, we define the excess loss by its continuous limit (Appendix A.3). For binned assays, we use a multinomial observation model with a predicted probability for each bin:

$$\boldsymbol{n}_i \mid N_i, \hat{\boldsymbol{\pi}}_i \sim \mathrm{Multinomial}(N_i, \hat{\boldsymbol{\pi}}_i), \qquad \hat{\pi}_{ik} = e^{\theta_{ik}} \Big/ \textstyle\sum_{k'=1}^{K} e^{\theta_{ik'}}, \tag{2}$$

where $n_{ik}$ is the observed read count for oligo $i$ in bin $k$, $\boldsymbol{n}_i$ is its count vector across the $K$ bins, $N_i = \sum_{k=1}^{K} n_{ik}$ is its total read count, and $\theta_{ik}$ is the model's predicted logit for oligo $i$ in bin $k$ (so that $\hat{\pi}_{ik}$ is the predicted probability of a read from oligo $i$ falling in bin $k$).

### 3.3 Metrics

Although we perform model selection by choosing the lowest validation loss, we represent models' performance using a weighted Pearson correlation, denoted $r_{\mathsf{Global}}$, across oligos in the test set. For tasks and subtasks that use our input–output count loss, we correlate the model-predicted activities with the measured activities $\log[(y_i+1)/(x_i+1)]$, using weights $w_i = (x_i+1)(y_i+1)/(x_i+y_i+2)$ that approximate the inverse posterior variance and reduce the influence of uncertain measurements (see Eq. 5 for a justification). For binned assays, we correlate the predicted and observed expected-bin activities, weighting each oligo by its total read count. For tasks comprising multiple subtasks, we define task-level versions of $r_{\mathsf{Global}}$ by averaging the subtask correlations on the Fisher-$z$ scale, weighting each subtask correlation equally. Across the evaluated backbones, higher $r_{\mathsf{Global}}$ is strongly associated with lower test negative log-likelihood relative to the saturated model, normalized per read (Fig. S4).

For tasks derived from studies designed to measure variant effects, we additionally report correlations assessing whether models resolve these effects. For tasks and subtasks derived from libraries containing matched alleles or designed perturbations, we correlate the measured and predicted differences in activity across eligible allele pairs within each group, which we denote by $r_{\mathsf{Variant}}$. We weight each pair of oligos $i$ and $j$ by $(w_i^{-1} + w_j^{-1})^{-1}$ to reduce the influence of pairs containing an

allele with low read counts. For tasks and subtasks derived from libraries that assay broad sets of mutations within specific genomic windows of interest, we instead correlate the predicted and measured activities within each group of mutations, using the same oligo-level weights as for $r_{\text{Global}}$, and average these correlations on the Fisher-$z$ scale, weighting each group equally, denoting the resulting metric by $r_{\text{MeanWindow}}$. As before, we define task-level versions of $r_{\text{Variant}}$ and $r_{\text{MeanWindow}}$ by averaging the corresponding subtask correlations.

We also assess how well models capture differences in activity across cell lines in tasks derived from multi-cell-line studies. After removing cell-line-specific offsets from predictions and measurements, we compute a weighted correlation across cell lines for each oligo and aggregate these correlations on the Fisher-$z$ scale, denoting the result by $r_{\text{CellLine}}$. In Appendix A.4, we provide the mathematical definitions, weights, and aggregation rules for these correlation metrics.

For `human_tx_s26_satmut`, we distinguish activity differences between HepG2 and K562 across sequence backgrounds from changes caused by individual mutations. For each mutation, we subtract the wild-type HepG2-minus-K562 activity difference from the mutant's difference within the same background. We evaluate how well models rank these changes using a tail-weighted Kendall correlation, denoted $\tau_{\Delta\text{Mutation}}$, that emphasizes large mutation effects. In Appendix A.5, we define this metric, its weighting and aggregation, and the corresponding mutation metrics within each cell line.

### 3.4 Models

We adapt all of the models introduced in Section 2.2 using the pretrained backbones listed in Table S2 with a common convolutional prediction head and LoRA. We train the head and adapters while keeping pretrained weights fixed. We select checkpoints by validation loss and evaluate them on the test set. In Appendix B, we specify the head architecture, optimization settings, adapter targets, and backbone-specific exceptions. Our benchmark measures performance under this shared protocol, which may not reflect each backbone's best attainable performance.

We compare these backbones with a CNN trained from scratch and species-specific MLM baselines, which are pretrained on individual reference genomes and adapted using the same protocol. In Appendix B.2, we describe the baseline architectures and their assignments to tasks.

## 4 Results

We begin with several observations validating our benchmarks and adaptation protocol. First, lower validation excess NLL per read generally accompanies lower test excess NLL per read (Fig. S3). Second, lower test NLL generally corresponds to higher $r_{\text{Global}}$ on the test set (Fig. S4), supporting $r_{\text{Global}}$ as a measure of generalization. Third, despite weaker overall correlations, the test NLL and our auxiliary metrics largely identify the same top-performing models (Fig. S5B), suggesting that these metrics capture aspects of generalization. Finally, LoRA generally improves validation NLL per read over training a prediction head on frozen embeddings (Fig. S5C).

In Fig. 2A, we report $r_{\text{Global}}$ for a representative subset of backbones on each task. In Fig. S6, we show all models. Values vary considerably across tasks due to measurement noise, library composition—more heterogeneous sequences may exhibit greater activity variance and permit higher $r_{\text{Global}}$—and the complexity of the underlying sequence-to-function relationships. We highlight AlphaGenome, Evo 2-7B, and PlantCAD2-Medium because they lead the benchmark on most tasks. AlphaGenome and Evo 2-7B each outperform the from-scratch baseline on 19 of 22 tasks, while PlantCAD2-Medium improves on 6 of 9. Gains are uneven across backbones and tasks. With one designated species-specific MLM baseline per task, these baselines outperform the from-scratch baseline on only 10 of 22 tasks. The stronger results of other gLMs may reflect differences in the scale and curation of pretraining data, as well as model size, although our comparisons do not isolate these factors.

**Functional genomics supervision can yield targeted gains.** Our results suggest that pretraining on task-relevant functional genomics data can markedly improve models' generalization performance. AlphaGenome leads on all five mammalian transcription tasks and ranks second on `human_splicing` and `human_3utr`. AlphaGenome's lead extends beyond mammals to `yeast_`

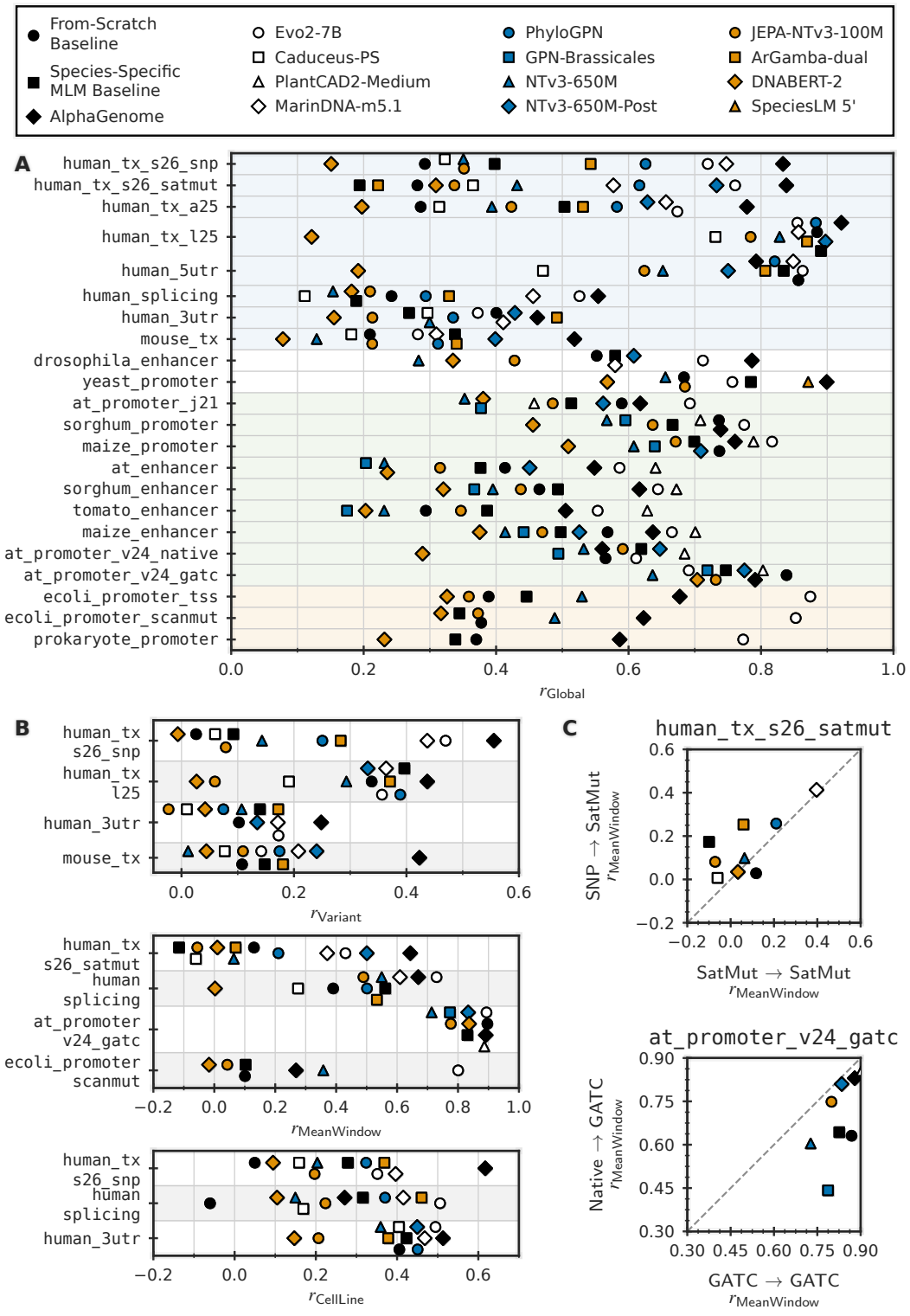


Figure 2: **Benchmark performance across selected backbones.** Metrics are defined in Section 3.3. **(A)** Test-set performance. Row backgrounds are blue for mammals, green for plants, and cream for prokaryotes. **(B)** We report $r_{\text{Variant}}$, $r_{\text{MeanWindow}}$, and $r_{\text{CellLine}}$, with the full model panel in Fig. S7A. **(C)** We compare direct training with transfer from `human_tx_s26_snp` to `human_tx_s26_satmut` (top) and from `at_promoter_v24_native` to `at_promoter_v24_gatc` (bottom), using target-library test $r_{\text{MeanWindow}}$ under matched assay conditions.

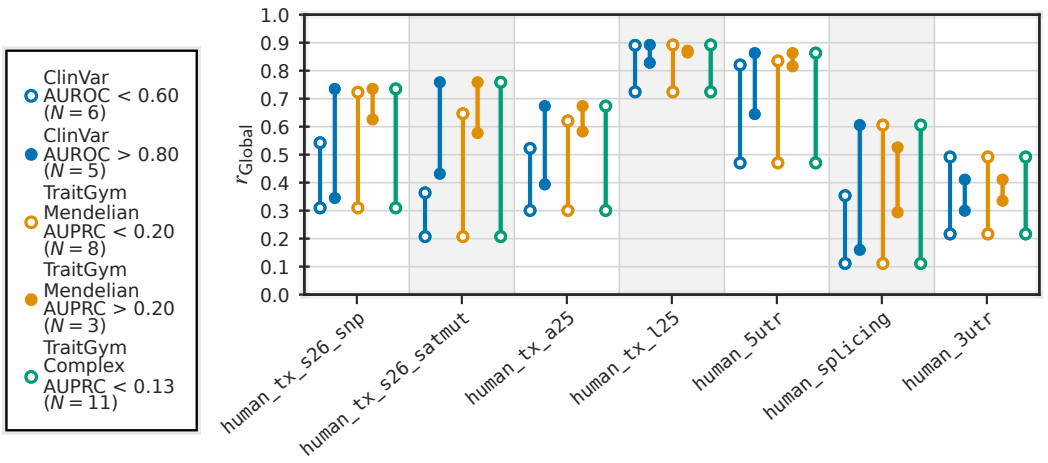


Figure 3: **Transfer performance across zero-shot VEP strata.** We stratify backbones with matched VEP and MPRA results by ClinVar AUROC or TraitGym pooled average precision, labeled AUPRC. Colors identify benchmarks. Open and filled endpoints denote scores below and above the indicated threshold, respectively. Ranges span the minimum to maximum test $r_{\text{Global}}$ within each stratum, and $N$ gives its backbone count.

`promoter` and `drosophila_enhancer`, but not to `ecoli_promoter_tss` or `ecoli_promoter_scanmut`, where Evo 2 performs better (Fig. S6). However, AlphaGenome's advantage is limited on `human_5utr`, consistent with its lack of training targets that directly capture translation initiation or ribosome loading. Similarly, NTv3-650M-Post outperforms NTv3-650M on all 13 tasks for which we evaluated both backbones, supporting the benefit of post-training on functional-genomics data. These results suggest that the value of functional-genomics supervision varies across downstream functions and species, with gains extending beyond mammalian tasks.

**Strong zero-shot VEP indicates strong transfer learning but is not required.** We examined whether the zero-shot VEP performance of gLMs indicates their utility for transfer learning by comparing available zero-shot scores for human-trained gLMs on three human variant sets. The *ClinVar Pathogenic vs. gnomAD Common* set was curated by Benegas et al. (2025a). The two *TraitGym* sets curated by Benegas et al. (2025b) split variants by whether they are predicted to influence a complex or Mendelian trait. We rank substitutions by the log-likelihood ratio of reference to alternate alleles or sequences, using flanking reference-genome context. We summarize the results in Fig. 3 and provide task-specific comparisons in Fig. S8. On ClinVar Pathogenic vs. gnomAD Common, AUROC is generally positively associated with downstream performance, even though this benchmark distinguishes pathogenic from common missense variants and our tasks measure regulatory activity. This correlation may reflect differences in pretraining quality that affect both evaluations, but need not hold universally. For example, a gLM trained exclusively on protein-coding sequences could perform well on this VEP benchmark without transferring well to regulatory tasks. On TraitGym, no backbone with strong zero-shot VEP performance ranks poorly on our benchmark, whereas backbones with near-chance VEP scores span a wide range of transfer performance. Thus, strong TraitGym performance can identify promising backbones, but weak performance does not preclude effective transfer.

**Better-performing backbones are less sensitive to training-set design for *in silico* mutagenesis.** We examined how backbone choice affects robustness to training-library design for *in silico* mutagenesis—a key application of MPRA predictors—by testing whether models trained on oligos from diverse genomic regions generalize to targeted perturbations measured in the same assay. For each backbone, we compared broad- and targeted-library training on the same held-out targeted oligos. We show selected results in Fig. 2C and full results in Fig. S7B. On `human_tx_s26_satmut`, training on `human_tx_s26_snp` yields higher $r_{\text{MeanWindow}}$ than direct training for 13 of 15 paired backbones in HepG2 and K562. Conversely, on `at_promoter_v24_gatc`, training on `at_promoter_v24_native` does not outperform direct training for any of the 11 paired backbones. This

difference may reflect the specificity of the GATC perturbations: training on native promoters may provide less information about the effects of altering this motif than direct training on GATC mutants. Although the preferred library differs between assays, stronger models show smaller performance gaps between library designs in both studies, suggesting that stronger backbones generalize more reliably across library designs.

**Benchmarked models only partially capture cell-line-specific regulatory effects.** Across `human_3utr`, `human_splicing`, and `human_tx_s26_snp`, $r_{\text{Global}}$ and $r_{\text{CellLine}}$ were strongly associated across models, although the magnitude of $r_{\text{CellLine}}$ remained modest (Fig. 2B, Fig. S9A). In `human_tx_s26_satmut`, several models captured activity differences between HepG2 and K562 across sequence backgrounds but struggled to predict mutation-specific changes in these differences (Fig. S9B). Performance on this differential task was generally low and only partly explained by performance on mutation effects within individual cell lines ($\rho = 0.69$–$0.73$). AlphaGenome achieved the highest $\tau_{\Delta\text{Mutation}}$ (0.37), while all other models scored at or below 0.1.

## 5 Discussion and Conclusion

The relative performance of pretrained backbones varies substantially across tasks, with no single model leading in every setting. These differences underscore the importance of broad benchmarks for selecting backbones and determining whether improvements generalize across regulatory functions, species, and experimental contexts.

Our findings suggest that improvements in pretraining will come from more effectively utilizing larger and more diverse collections of genome sequences and functional annotations. AlphaGenome's strong performance demonstrates the value of functional-genomics supervision and sets a target for sequence-only pretraining. If the regulatory information needed for these tasks is recoverable from genomic sequence distributions, gLMs could in principle match that performance by scaling the amount and diversity of sequence data they effectively learn from. Even if this gap persists, we expect sequence pretraining to remain central to future models because functional assays cover only part of regulatory biology, with functional-genomics and potentially even MPRA supervision providing complementary information. Whether sequence alone can ultimately yield representations as informative as those learned with functional supervision remains an important open question.

**Designing models for downstream adaptation.** The practical utility of pretrained backbones also depends on how readily they can be adapted to downstream tasks. We offer several recommendations to improve accessibility. First, models should support efficient inference on short inputs, since relatively short contexts appear to suffice in our AlphaGenome encoder experiments (Fig. S10). Second, models should be designed with users' computational resources in mind. Indeed, computational requirements prevented us from evaluating Evo 2-40B. Finally, because MPRA tasks share a common structure, model developers should establish and release validated, reproducible adaptation protocols. Evaluating these recommended protocols within REPORTER would allow comparisons of both backbones and their adaptation procedures.

**Extending REPORTER.** By modeling diverse MPRA readouts within a single framework, we make REPORTER straightforward to extend to new assays. Our count-based formulation also applies to deep mutational scanning assays with analogous readouts, including protein assays represented in ProteinGym (Notin et al., 2023). As more MPRA data become available, REPORTER can grow to cover additional regulatory functions, species, cell types, and experimental conditions.

**Conclusion.** Pretrained genomic backbones can substantially improve regulatory sequence-to-function prediction, but their benefits vary across tasks. By bringing diverse functional assays into a shared evaluation framework, REPORTER provides a basis for measuring and improving the quality and generality of genomic representations.

## REPRODUCIBILITY AND ARTIFACT AVAILABILITY

The source data are publicly available through the publications cited in Table S1. In Section 3 and Appendices A and B, we describe the data processing, objectives, metrics, and model adaptation procedures.

We will make the processed benchmark data and exact training, validation, and test partition assignments publicly available. The submission does not include implementation code or author-trained backbones. Third-party model weights remain subject to their respective distribution terms. No new participant data or biological experiments were collected for this benchmark.

## AI USE STATEMENT

We used generative AI to assist with writing, literature checks, coding, analysis, and mathematical checks. The authors reviewed and revised these contributions and take responsibility for the manuscript.

## REFERENCES


Agarwal V., Inoue F., Schubach M. et al. *Massively parallel characterization of transcriptional regulatory elements. Nature*, 639:411–420, 2025. doi:10.1038/s41586-024-08430-9 (PMID 39814889)

Albors C., Li J.C., Benegas G., Ye C., Song Y.S. *A Phylogenetic Approach to Genomic Language Modeling. Research in Computational Molecular Biology (RECOMB), LNCS*, 2025. arXiv:2503.03773. doi:10.1007/978-3-031-90252-9_7

Avsec Ž., Latysheva N., Cheng J., et al. *Advancing regulatory variant effect prediction with AlphaGenome. Nature*, 649(8099):1206–1218, 2026. doi:10.1038/s41586-025-10014-0

Benegas G., Batra S.S., Song Y.S. *DNA language models are powerful predictors of genome-wide variant effects. Proceedings of the National Academy of Sciences U.S.A.*, 120(44):e2311219120, 2023. doi:10.1073/pnas.2311219120 (PMID 37883436)

Benegas G., Albors C., Aw A.J., Ye C., Song Y.S. *A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. Nature Biotechnology*, 43:1960–1965, 2025. doi:10.1038/s41587-024-02511-w (PMID 39747647)

Benegas G., Eraslan G., Song Y.S. *Benchmarking DNA Sequence Models for Causal Regulatory Variant Prediction in Human Genetics. bioRxiv*, 2025. doi:10.1101/2025.02.11.637758 (PMID 39990426)

Benegas G., Ye C., Albors C., Li J.C., Song Y.S. *Genomic language models: opportunities and challenges. Trends in Genetics*, 41(4):286–302, 2025. doi:10.1016/j.tig.2024.11.013 (PMID 39753409)

Bommasani R., Hudson D.A., Adeli E. et al. *On the Opportunities and Risks of Foundation Models. arXiv preprint arXiv:2108.07258*, 2021. doi:10.48550/arXiv.2108.07258

Boshar S., Evans B., Tang Z., et al. *A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. bioRxiv*, 2025. Preprint. doi:10.64898/2025.12.22.695963

Brixi G., Durrant M.G., Ku J., et al. *Genome modelling and design across all domains of life with Evo 2. Nature*, 652(8112):1349–1361, 2026. doi:10.1038/s41586-026-10176-5 (PMID 41781614)

Chen S., Francioli L.C., Goodrich J.K., Collins R.L., Kanai M. et al. *A genomic mutational constraint map using variation in 76,156 human genomes. Nature*, 625:92–100, 2024. doi:10.1038/s41586-023-06045-0

Cheng W., Song Z., Zhang Y., et al. *DNALongBench: a benchmark suite for long-range DNA prediction tasks. Nature Communications*, 16:10108, 2025. doi:10.1038/s41467-025-65077-4

Consens M.E., et al. *Predicting evolutionary rate as a pretraining task improves genome language model representations. International Conference on Machine Learning (ICML)*, 2026. openreview.net/forum?id=FOIP4Blm4F

Dalla-Torre H., Gonzalez L., Mendoza-Revilla J., et al. *Nucleotide Transformer: building and evaluating robust foundation models for human genomics. Nature Methods*, 22:287–297, 2025. doi:10.1038/s41592-024-02523-z (PMID 39609566)

De Almeida B.P., Reiter F., Pagani M., Stark A. *DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. Nature Genetics*, 54(5):613–624, 2022. doi:10.1038/s41588-022-01048-5 (PMID 35551305)

De Boer C.G., Vaishnav E.D., Sadeh R., Abeyta E.L., Friedman N., Regev A. *Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. Nature Biotechnology*, 38:56–65, 2020. doi:10.1038/s41587-019-0315-8 (PMID 31792407)

Friedman R.Z., Granas D.M., Myers C.A., Corbo J.C., Cohen B.A., White M.A. *Information content differentiates enhancers from silencers in mouse photoreceptors. eLife*, 10:e67403, 2021. doi:10.7554/eLife.67403 (PMID 34486522)

Grešová K., Martinek V., Čechák D., Šimeček P., Alexiou P. *Genomic benchmarks: a collection of datasets for genomic sequence classification. BMC Genomic Data*, 24:25, 2023. doi:10.1186/s12863-023-01123-8

Griesemer D., Xue J.R., Reilly S.K. et al. *Genome-wide functional screen of 3′UTR variants uncovers causal variants for human disease and evolution. Cell*, 184(20):5247–5260, 2021. doi:10.1016/j.cell.2021.08.025 (PMID 34534445)

Hu E.J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Wang L., Chen W. *LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685

Johns N.I. et al. *Metagenomic mining of regulatory elements enables programmable species-selective gene expression. Nature Methods* 15(5):323–329, 2018. doi:10.1038/nmeth.4633

Jores T., Tonnies J., Wrightsman T., Buckler E.S., Cuperus J.T., Fields S., Queitsch C. *Synthetic promoter designs enabled by a comprehensive analysis of plant core promoters. Nature Plants*, 7:842–855, 2021. doi:10.1038/s41477-021-00932-y

Jores T. et al. *Massively parallel characterization and deep learning of enhancers in plant genomes. bioRxiv*, 2026. doi:10.64898/2026.04.26.720828

Karollus A., Hingerl J., Gankin D., Grosshauser M., Klemon K., Gagneur J. *Species-aware DNA language models capture regulatory elements and their evolution. Genome Biology*, 25:83, 2024. doi:10.1186/s13059-024-03221-x (PMID 38566111)

Koplik S.E. et al. *Massively parallel assay of human splice variants reveals cis-regulatory drivers of disease-associated and cell-type-specific splicing regulation (COMPASS). bioRxiv*, 2025. doi:10.1101/2025.10.12.681955

Landrum M.J., Lee J.M., Benson M., Brown G.R., Chao C. et al. *ClinVar: improving access to variant interpretations and supporting evidence. Nucleic Acids Research*, 46(D1):D1062–D1067, 2018. doi:10.1093/nar/gkx1153

Larey A., Dahan E., Bleiweiss A., Kellerman R., Leib G., et al. *JEPA-DNA: Grounding Genomic Foundation Models through Joint-Embedding Predictive Architectures.* arXiv:2602.17162, 2026. doi:10.48550/arXiv.2602.17162

Lee S., McAfee J.C., Lee J. et al. *Massively parallel reporter assay investigates shared genetic variants of eight psychiatric disorders. Cell*, 188(5):1409–1424, 2025. doi:10.1016/j.cell.2024.12.022 (PMID 39848247)

Marin F.I., Teufel F., Horlacher M., Madsen D., Pultz D., Winther O., Boomsma W. *BEND: Benchmarking DNA Language Models on Biologically Meaningful Tasks. International Conference on Learning Representations (ICLR)*, 2024. arXiv:2311.12570

Murphy A., Durán A., Koo P.K. *Adapting AlphaGenome to MPRA data. Genomics x AI Blog*, 20 February 2026. genomicsxai.github.io/blogs/2026-002 doi:10.5281/zenodo.20272900

Notin P., Kollasch A.W., Ritter D., et al. *ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. Advances in Neural Information Processing Systems (NeurIPS)*, 36:64331–64379, 2023. doi:10.52202/075280-2810

Open Athena. *MarinDNA: a 1B standard transformer rivals Evo 2 40B on variant effect prediction.* Technical report, 2026. openathena.ai/blog/marin-dna

Patel A., Singhal A., Wang A., Pampari A., Kasowski M., Kundaje A. *DART-Eval: A Comprehensive DNA Language Model Evaluation Benchmark on Regulatory DNA. Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. arXiv:2412.05430

Pollard K.S., Hubisz M.J., Rosenbloom K.R., Siepel A. *Detection of nonneutral substitution rates on mammalian phylogenies. Genome Research*, 20(1):110–121, 2010. doi:10.1101/gr.097857.109

Robson E.S., Ioannidis N.M. *GUANinE v1.1 reveals complementarity of supervised and genomic language models. Bioinformatics*, 42(Suppl. 2):btag427, 2026. doi:10.1093/bioinformatics/btag427

Sample P.J. et al. *Human 5′UTR design and variant effect prediction from a massively parallel translation assay. Nature Biotechnology*, 37:803–809, 2019. doi:10.1038/s41587-019-0164-5 (PMID 31267113)

Schiff Y., Kao C.-H., Gokaslan A., Dao T., Gu A., Kuleshov V. *Caduceus: Bi-Directional Equivariant Long-Range DNA Sequence Modeling. Proceedings of the 41st International Conference on Machine Learning (ICML), PMLR*, 235:43632–43648, 2024. doi:10.48550/arXiv.2403.03234

Siraj L. et al. *Functional dissection of complex trait variants at single-nucleotide resolution. Nature*, 2026. doi:10.1038/s41586-026-10121-6

Tang Z., Somia N., Yu Y., Koo P.K. *Evaluating the representational power of pre-trained DNA language models for regulatory genomics. Genome Biology*, 26:203, 2025. doi:10.1186/s13059-025-03674-8

Urtecho G., Insigne K.D., Tripp A.D. et al. *Genome-wide functional characterization of Escherichia coli promoters and sequence elements encoding their regulation. eLife* (reviewed preprint), 2023. doi:10.7554/eLife.92558

Voichek Y. et al. *Widespread position-dependent transcriptional regulatory sequences in plants. Nature Genetics*, 56:1062–1072, 2024. doi:10.1038/s41588-024-01907-3

Zhai J., Gokaslan A., Hsu S.-K., Chen S.-P., Liu Z.-Y., Marroquin E., et al. *PlantCAD2: a DNA foundation model for interpreting genomes across flowering plants. bioRxiv*, 2025. Preprint. doi:10.1101/2025.08.27.672609 (PMID 40950025)

Zhou Z., Ji Y., Li W., Dutta P., Davuluri R., Liu H. *DNABERT-2: Efficient Foundation Model and Benchmark for Multi-Species Genome. International Conference on Learning Representations (ICLR)*, 2024. arXiv:2306.15006. doi:10.48550/arXiv.2306.15006

# A Dataset and task details

## A.1 Dataset schema and shared processing

**Dataset schema.** We store the data for each task in a Parquet table, with each record containing an assayed sequence and its genomic anchor. The `seq` field stores the assayed sequence, and `src` identifies its reference assembly. The fields `chrom`, `start`, `end`, and `strand` specify a zero-based, half-open genomic interval from which we retrieve flanking context. For a variant-containing oligo, this interval identifies its parent locus and need not match the assayed sequence. An optional `subset` field identifies a source sub-library or related oligos. The `cluster90` and `shard` fields record the sequence groups used for partitioning, as described below. Orientation augmentation and multiple genomic assignments can produce several rows from one experimental measurement.

Assay-readout columns depend on the experiment. Input–output count assays provide paired columns for each cell line or condition, typically DNA and RNA counts, sometimes sharing one input column across several outputs. For `human_splicing`, the input counts transcripts that skip the exon and the output counts transcripts that include it. Binned assays provide one count column per ordered bin. For five studies that release processed activities rather than raw counts, we construct the pseudocounts described below. We exclude unavailable readouts from training and evaluation.

**Genomic-coordinate recovery.** Pretrained backbones receive the assayed sequence together with enough flanking genomic context to match the context length used during pretraining, subject to the model-specific limits described in Appendix B.2. We therefore retain native sequences and their variants only when we can identify their genomic source or parent locus. We check deposited coordinates against the reference genome, recovering coordinates by exact sequence matching or the study-specific procedures in Appendix A.2 when needed. For a variant-containing oligo, the genomic interval identifies the parent sequence from which we retrieve flanks. The model still receives the assayed variant, including any insertions or deletions, even when its length differs from the parent interval. We exclude constructs without a recoverable parent locus and sequences spanning exon junctions when no single genomic interval represents them.

**Sequence clustering.** Related sequences must remain together when we divide the data into training and evaluation sets. We link sequences that share at least $90\%$ identity and define a cluster as all sequences connected by a chain of these links. A `cluster90` field records cluster membership. A `shard` field identifies larger groups of whole clusters used for split assignment. We also preserve relationships encoded by shared `subset` values, such as paired alleles, mutants of one parent element, or orientation copies of a genomic window. We process tasks with shared cluster or shard identifiers jointly and join records linked by identical assayed sequences or a common parent element across libraries before assigning partitions.

**Pseudocount reconstruction.** Five studies release normalized activities rather than raw read counts. To use these measurements with the common input–output count loss, we construct artificial input and output counts, called pseudocounts. We weight input pseudocounts inversely by sequence-cluster size and set output pseudocounts so that their ratio preserves the reported activity, converting log2 activities to linear ratios where needed. These counts do not recover sequencing depth or experimental uncertainty.

**Training, validation, and test partitions.** We form the benchmark partitions in two stages. First, we order the complete groups defined above by a deterministic hash with a seed of $42$. For tasks prepared jointly, we count records across their combined libraries and assign each group once, targeting $70\%$ of records for training, $10\%$ for validation, and $20\%$ for testing. We reserve a complete group for each partition. Every occurrence of a shared group receives the same assignment across tasks, although the proportions can differ within an individual task.

Second, we subsample within each partition to benchmark size limits of $60$K training records, $20$K validation records, and $15$K test records. We retain all records with the same `subset` value together, or all copies of an identical sequence when that field is absent. If a group does not fit within the remaining allowance, we skip it and continue considering smaller groups. Retained counts therefore stay at or below these limits. Subsampling removes records but never moves them between parti-

tions. In Table S1, we report the retained partition sizes and the full processed-library totals before either stage.

We check that sequence clusters and related constructs stay within one partition, both within tasks and across tasks prepared jointly. For cross-library transfer, we also check that source-training sequences and their parent elements are absent from the target validation and test sets in both directions. The checks use the assayed sequences and parent loci that define the groups, rather than the extended genomic flanks supplied to the backbone.

### A.2 Data sources and task-specific processing

In Table S1, we list all 22 REPORTER tasks with citations to their 14 source studies. For each task, we report the oligo length, the sizes of the training, validation, and test partitions, and the full processed-library total.

Below, we describe each source study and the processing choices specific to its tasks.

**The `human_tx_s26_snp` and `human_tx_s26_satmut` tasks.** Siraj et al. (2026) used an MPRA to identify regulatory effects of fine-mapped candidate causal variants from complex traits and tissue-specific eQTLs. The main libraries compared reference and alternate alleles in 200 bp sequence contexts, together with location-matched, annotation-matched, and simulated-trait controls. These controls were selected comparison sequences rather than experimentally established nulls. The study also generated a saturation-mutagenesis library with dense substitutions around selected regulatory variants and a diplotype panel probing nearby variant pairs. Overlapping windows placed a parent variant at the center or an offset position, so an element could contribute several assayed contexts. The SNP libraries were assayed in A549, HCT116, HepG2, K562, and SK-N-SH, and the saturation-mutagenesis library in HepG2 and K562.

We process the SNP and saturation-mutagenesis libraries separately, retaining the released DNA and RNA counts with one readout per cell line. Both libraries use hg19 genomic anchors. Paired alleles share a source locus, and mutagenized constructs retain their parent window, so the grouping reflects the library design. We preserve the assayed sequence even when an insertion or deletion makes its genomic span differ from 200 bp, and retrieve flanks from the adjusted parent span. We partition the two libraries jointly, assigning each shared sequence cluster or shard to the same split in both tasks. This keeps source-training sequence clusters separate from target-validation and target-test clusters for transfer in either direction between `human_tx_s26_snp` and `human_tx_s26_satmut`.

**The `human_tx_a25` task.** Agarwal et al. (2025) used lentiMPRA to test the regulatory activity of human candidate cis-regulatory elements identified through chromatin accessibility and histone modifications. The library contained 200 bp enhancer and promoter sequences, together with positive, negative, shuffled, and CRISPRi controls. Large sub-libraries tested different element sets in HepG2, K562, and WTC11, while a shared library tested elements across all three cell lines. The assay included both orientations and three biological replicates per cell line. Lentiviral delivery integrated the barcoded reporters into the host genome, exposing them to chromatin without recreating their original genomic loci. The released measurements are normalized RNA and DNA barcode rates rather than raw read counts.

We retain elements anchored to hg38 and combine the available readouts from the three cell lines. We reconstruct cluster-balanced paired pseudocounts from the ratio of the released RNA and DNA rates.

**The `human_tx_l25` task.** Lee et al. (2025) used an episomal MPRA to measure the regulatory effects of variants associated with eight psychiatric disorders and compare variants associated with multiple disorders against those associated with one. The library tested 17,841 cross-disorder risk variants as 150 bp elements containing the risk or protective allele. Each element was placed upstream of a minimal promoter and a barcoded reporter, and the library was assayed in human neural progenitors. DNA and RNA barcode counts measured the representation and reporter output of each construct. These data supply the benchmark's neural-progenitor condition and include paired alleles for assessing variant effects.

Table S1: **Study sources, oligo lengths, and sample counts for the 22 REPORTER tasks.** Training, validation, and test counts describe the partitions used in the benchmark. Total is the number of oligo records in the full processed library before caps, so it can exceed the sum of the three displayed splits. Counts include augmented and repeated-locus records, which need not represent independent measurements.

| Task ID | Study of origin | Oligo size (bp) | Dataset size (oligos) | | | |
|---|---|---|---|---|---|---|
| | | | Training | Validation | Test | Total |
| human_tx_s26_snp | Siraj et al. 2026 | 200 | 60,000 | 19,294 | 15,000 | 692,884 |
| human_tx_s26_satmut | Siraj et al. 2026 | 200 | 60,400 | 17,953 | 15,545 | 229,090 |
| human_tx_a25 | Agarwal et al. 2025 | 200 | 60,000 | 19,979 | 15,000 | 416,551 |
| human_tx_l25 | Lee et al. 2025 | 150 | 18,648 | 1,912 | 6,884 | 27,444 |
| human_5utr | Sample et al. 2019 | 50 | 18,118 | 1,982 | 5,930 | 26,030 |
| human_splicing | Koplik et al. 2025 | 161 | 60,496 | 7,934 | 15,084 | 87,537 |
| human_3utr | Griesemer et al. 2021 | 101 | 17,904 | 1,994 | 5,896 | 25,794 |
| mouse_tx | Friedman et al. 2021 | 164 | 6,791 | 1,067 | 1,980 | 9,838 |
| drosophila_enhancer | De Almeida et al. 2022 | 249 | 60,000 | 19,876 | 15,000 | 484,052 |
| yeast_promoter | De Boer et al. 2020 | 59 | 60,000 | 8,011 | 15,000 | 88,436 |
| at_promoter_j21 | Jores et al. 2021 | 170 | 10,870 | 1,095 | 4,349 | 16,314 |
| sorghum_promoter | Jores et al. 2021 | 170 | 17,158 | 1,959 | 5,911 | 25,028 |
| maize_promoter | Jores et al. 2021 | 170 | 21,878 | 2,824 | 8,710 | 33,412 |
| at_enhancer | Jores et al. 2026 | 170 | 60,001 | 10,071 | 15,000 | 105,937 |
| sorghum_enhancer | Jores et al. 2026 | 170 | 44,303 | 6,044 | 14,012 | 64,359 |
| tomato_enhancer | Jores et al. 2026 | 170 | 28,485 | 3,947 | 8,938 | 41,370 |
| maize_enhancer | Jores et al. 2026 | 170 | 60,000 | 8,875 | 15,001 | 89,002 |
| at_promoter_v24_native | Voichek et al. 2024 | 160 | 7,612 | 967 | 3,231 | 11,810 |
| at_promoter_v24_gatc | Voichek et al. 2024 | 160 | 8,718 | 2,242 | 2,310 | 13,270 |
| ecoli_promoter_tss | Urtecho et al. 2023 | 150 | 11,351 | 985 | 5,017 | 17,353 |
| ecoli_promoter_scanmut | Urtecho et al. 2023 | 150 | 36,371 | 5,005 | 11,491 | 52,867 |
| prokaryote_promoter | Johns et al. 2018 | 165 | 20,567 | 2,918 | 6,723 | 30,208 |

We use the released DNA and RNA counts, retaining the reference and alternate members of each measured pair. The deposited hg19 variant position defines the 150 bp genomic window, with the variant at offset 75. Both alleles share this anchor and a `subset` identifier, while their input sequences retain the assayed allele. We use the shared anchor to retrieve genomic flanks and keep related alleles together during partitioning.

**The `human_5utr` task.** Sample et al. (2019) measured how 5′ UTR sequence affects translation to train models of ribosome loading, design synthetic UTRs, and predict variant effects. They cloned 50 nt UTRs upstream of a reporter, transfected the library into HEK293T cells, and separated mRNA into fourteen ribosome-loading fractions by polysome profiling. The released data contain a count for each sequence in each fraction. The study included random and designed libraries alongside native human UTRs and single-nucleotide variants of native UTRs. The genome-derived libraries comprised 41,862 native sequences and 8,738 variant sequences, drawn from the region immediately upstream of the annotated translation start site.

We retain the genome-derived libraries and their fourteen-fraction count vectors, excluding random sequences and engineered controls. We also exclude the `sergii_utrs` subset because its sequences could not be matched exactly to the human transcriptome. To recover genomic context, we match each reference UTR against hg38 while preserving the assay's substituted first transcribed base in the input sequence. UTRs spanning exon junctions cannot be represented by a single contiguous genomic window, so we retain only those whose 50 nt sequence lies within one exon. The retained records carry their genomic interval and source-library identity. The included variant sequences are not organized as matched reference–alternate pairs, so their presence does not define an additional paired-variant dataset.

**The `human_splicing` task.** Koplik et al. (2025) developed COMPASS to measure how sequence variants alter splicing across cell types. The study tested 87,546 sequences spanning 2,096 native human cassette exons in a fixed Citrine–SMN2 minigene. Each 161 bp insert comprised the exon, 20 nt of downstream intron, and a variable amount of upstream intron. The library combined reference constructs, naturally observed variants from Geuvadis, ExAC, and ClinVar, and designed single- and double-substitution constructs. Measurements in HEK293, HeLa, K562, MCF7, and HMC3 counted transcripts that included or skipped the cassette exon, allowing the study to compare variant effects and their dependence on cellular context.

We combine the five cell-line readouts, using skipped-transcript counts as the input and included-transcript counts as the output. Their ratio therefore represents inclusion odds, while percent-spliced-in is the included count divided by the sum of included and skipped counts. We preserve the assayed 161 bp sequence and convert its deposited hg38 interval and strand into a half-open genomic window extending from the upstream intron through the downstream flank. Constructs retain their parent-exon grouping so that the split and within-exon comparisons respect the designed relationships. The processed library contains 87,537 records, with one pair of isoform-count measurements per available cell line.

**The `human_3utr` task.** Griesemer et al. (2021) developed MPRAu to measure how 3′ UTR variants affect post-transcriptional regulation. The library tested 12,173 variants from disease-associated loci, regions under positive selection, rare-variant sets, and common-variant controls. Matched reference and alternate constructs were screened in HEK293FT, HepG2, GM12878, SK-N-SH, K562, and primary HMEC. Plasmid DNA supplied the input counts for each assay, and recovered mRNA supplied the output counts. This design captured regulatory effects mediated by features such as AU-rich elements and microRNA-binding sites.

We use the released counts for all six cell lines and preserve the paired alleles in the processed 101 bp sequences. We anchor the actual oligo window in hg19 rather than recentering it on the variant, because the variant can be off-center in the assayed fragment. Sequences follow the transcript strand, so minus-strand constructs retrieve flanking sequence in the reverse-complement orientation. We exclude 346 oligos spanning exon–exon junctions because their assayed sequence has no single contiguous genomic window. Each variant's remaining alleles retain a shared `subset` identifier for partitioning and paired comparisons.

**The `mouse_tx` task.** Friedman et al. (2021) used an episomal MPRA in mouse retinal explants to distinguish enhancers from silencers among sequences bound by cone-rod homeobox (CRX). They compared native elements with designed perturbations to investigate how transcription-factor motifs contribute to regulatory activity. The library included 4,844 wild-type elements of 164 bp, corresponding mutants, and 150 scrambled controls. The mutant sets comprised 4,167 constructs disrupting CRX sites and 677 targeting sequence shape. They measured RNA counts relative to the barcoded plasmid input with either a basal Rhodopsin promoter or a polylinker replacing it.

We retain all 9,838 records and their DNA and RNA counts for both conditions. Wild types, designed mutants, and scrambled controls share the mm10 coordinates of their parent element and its `subset` identifier. An anchor thus identifies the source of genomic context even when the assayed mutant or scrambled sequence does not match the reference genome. We keep all constructs derived from the same parent element together within clusters and partitions. The scrambled controls remain part of the processed library because they have identifiable genomic parents.

**The `drosophila_enhancer` task.** De Almeida et al. (2022) measured developmental and housekeeping enhancer activity in *Drosophila melanogaster* S2 cells using UMI-STARR-seq. The two screens used distinct core promoters, including the housekeeping promoter from RpS12, and measured reporter RNA relative to input plasmid DNA. The study used these data to train DeepSTARR and investigate how transcription-factor motifs, their flanking sequences, and motif spacing contribute to enhancer activity. The released dataset contains 249 bp genomic windows with log2 activity measurements for both promoter conditions.

We use the released dm3 windows and convert the log2 activities to linear ratios for the cluster-balanced pseudocount construction. We model the two promoter conditions as separate readouts. The authors augmented 242,026 windows with reverse-complement copies carrying the same ac-

tivity labels, producing 484,052 processed records. These copies are computational augmentation rather than independent measurements of the two orientations. We retain the locus key shared by both copies and keep them in the same cluster and partition. Each record retains its dm3 interval and orientation, and we retrieve flanking context in the orientation of the supplied sequence.

**The `yeast_promoter` task.** De Boer et al. (2020) used a large-scale parallel reporter assay to learn how yeast promoter sequence determines expression and whether models trained on random sequences transfer to native promoters. Cells carrying a YFP reporter, normalized to constitutive RFP, were sorted into eighteen fluorescence bins, and sequencing measured each construct's abundance within each bin. Alongside the main random-promoter libraries, a native library tiled *Saccharomyces cerevisiae* promoters into 80 bp fragments across the 480 bp upstream of the transcription start site. These fragments were assayed in the same reporter scaffold under growth in glucose. The deposited native-library data consist of 59 bp sequencing fingerprints with bin counts and no genomic coordinates.

We retain the native library and reconstruct one sequence and eighteen-bin count vector per genomic locus. We align fingerprints to both strands of the yeast reference genome with at most two mismatches, allowing limited sequencing error and strain variation while excluding unmatched reads and non-genomic controls. We then sum bin counts across reads assigned to the same locus and choose the read-weighted majority base at each position as its representative sequence. This removes redundant sequencing-error variants while preserving observed differences from the reference, although those differences can still include alignment or sequencing errors. The resulting 88,436 records retain genomic coordinates and a locus key for clustering and partitioning. The benchmark input is the reconstructed 59 bp fingerprint, while the original reporter insert was 80 bp.

**The `at_promoter_j21` and `{species}_promoter` tasks.** Jores et al. (2021) measured plant core-promoter activity to determine how sequence near the transcription start site influences promoter strength. The libraries contained 170 bp elements spanning $-165$ to $+5$ bp around annotated transcription start sites in *Arabidopsis thaliana*, *Sorghum bicolor*, and *Zea mays*. Barcoded reporters measured RNA abundance relative to plasmid DNA in two heterologous systems, *Nicotiana benthamiana* leaves and maize mesophyll protoplasts. Each system was tested with and without an upstream 35S enhancer. Leaf assays additionally compared light and constant darkness, yielding four leaf conditions and two protoplast conditions. These libraries contain native promoters rather than designed perturbation series.

We process the Arabidopsis, sorghum, and maize libraries separately as `at_promoter_j21`, `sorghum_promoter`, and `maize_promoter`, respectively. We reconstruct the six paired-count readouts from raw barcode reads, summing DNA and RNA counts across replicates for each condition. Light and dark readouts share the corresponding input DNA library. We match each 170 bp sequence against both strands of TAIR10, Sorghum_bicolor_NCBIv3, or B73_RefGen_v4. When an exact match is unavailable, we retain a high-identity window supported by a unique two-sided flank match, preserving interior substitutions in the assayed sequence. For sequences matching multiple loci, we emit one record per locus and divide counts equally among them. We exclude sequences with no recoverable genomic anchor. Because a barcode can match an unrelated subassembly by chance, we check the library pairing against the authors' published promoter activities. The reconstructed log ratios have Spearman correlations of 0.94–0.98 with the published enrichment in the checked leaf and protoplast conditions.

**The `{species}_enhancer` tasks.** Jores et al. (2026) measured enhancer activity in accessible chromatin from *Arabidopsis thaliana*, tomato, maize, and sorghum to learn sequence determinants of plant enhancer strength and support sequence-based enhancer design. They cloned 170 bp elements in both orientations into a barcoded plant STARR-seq reporter. Two heterologous assay systems provided comparable measurements across source species: transiently transformed *Nicotiana benthamiana* leaves and maize mesophyll protoplasts. The leaf assays used a standard light–dark cycle, constant darkness, a warm shift to 35 °C, and a cold shift to 10 °C. The released data provide log2 RNA-to-DNA activity ratios for these four leaf conditions and the protoplast condition.

We process the Arabidopsis, sorghum, tomato, and maize libraries separately as `at_enhancer`, `sorghum_enhancer`, `tomato_enhancer`, and `maize_enhancer`, respectively. We reconstruct cluster-balanced pseudocounts for the five condition-specific readouts. Forward and reverse con-

structs share an element identifier and remain in the same cluster and partition. We verify each sequence against TAIR10, SL3.0, B73_RefGen_v4, or Sorghum_bicolor_NCBIv3, retaining a reconstructing deposited interval or recovering an exact match on either strand. This re-anchoring corrects tomato coordinates from an older assembly. We anchor Arabidopsis Bur-0 elements against the separate Bur-0 assembly GCA_977020685.1. For cross-annotated maize and sorghum elements, we retain deposited orthologous windows that reconstruct the sequence at greater than $90\%$ identity, leaving interior differences in the assayed sequence. All evaluated records have a genomic anchor, including the retained tomato records.

**The `at_promoter_v24_native` and `at_promoter_v24_gatc` tasks.** Voichek et al. (2024) used an MPRA to determine where proximal plant regulatory elements act relative to a transcription start site and whether a downstream GATC motif promotes transcription. The discovery pool, OP1, assayed $160$ bp Arabidopsis fragments from positions upstream and downstream of genes. Its sixteen steady-state readouts combined four assay species, Arabidopsis, tomato, maize, and *Nicotiana benthamiana*, with two reporter backbones and two reporter positions. A follow-up assay in Arabidopsis protoplasts measured newly synthesized RNA after 5-ethynyl-uridine labeling and total RNA, adding two readouts of the same fragments to distinguish transcriptional effects from RNA stability. The OP2 pool tested mutated derivatives of OP1 fragments, including motif deletions, substitutions, local mutagenesis, and added GATC copies, under two reporter backbones in Arabidopsis protoplasts. These pools provide log2 expression measurements.

We process OP1 native fragments and OP2 GATC mutants separately, retaining eighteen readouts for OP1 and two for OP2. We reconstruct cluster-balanced pseudocounts. Missing measurements remain missing rather than becoming zeros. We first match OP1 sequences to both strands of TAIR10 and assign each OP2 derivative its parent's genomic window. For accession differences involving indels, we match the two $30$ bp ends independently. Remaining fragments are anchored by a central $30$ bp match and chained $25$ bp seeds, with the recovered span checked by edit distance. This second procedure places another $1{,}070$ fragments and reduces positional bias from failures near sequence ends. We exclude non-genomic controls and the $175$ of $11{,}985$ OP1 fragments that remain unplaced. We assign each native fragment and its GATC-mutant derivatives to the same partition using their shared parent locus. This keeps parent elements separate between training and evaluation for transfer between `at_promoter_v24_native` and `at_promoter_v24_gatc`.

**The `ecoli_promoter_tss` and `ecoli_promoter_scanmut` tasks.** Urtecho et al. (2023) used a chromosomally integrated reporter to identify active promoters across the *Escherichia coli* genome and locate the sequences contributing to their activity. The TSS library represented candidate transcription start sites as $150$ bp oligos spanning $-120$ to $+30$ bp, together with positive and negative controls. A separate scanning-mutagenesis library perturbed $2{,}057$ active promoters by dinucleotide-shuffling consecutive $10$ bp windows at $5$ bp intervals. Both libraries were assayed in K-12 MG1655 grown in LB medium using a barcoded sfGFP reporter. The released promoter activities derive from RNA barcode abundance normalized to DNA and fitted to the controls.

We process genome-anchored TSS sequences and the scanning-mutagenesis library separately, retaining one LB-condition readout per library. We reconstruct cluster-balanced paired pseudocounts from the normalized activities. The deposited TSS-derived coordinates do not reconstruct the oligos in the MG1655 reference U00096.3, so we recover each TSS window by exact sequence matching. For scanning mutants, we match the unperturbed genomic flanks around the scrambled region and retain the assayed scramble in the input sequence. All mutants of a promoter share its parent window and `subset` identifier. Every evaluated record has a genomic anchor, which identifies the parent context for a scrambled construct.

**The `prokaryote_promoter` task.** Johns et al. (2018) tested how regulatory sequences from diverse prokaryotic genomes function in different host species. The library drew $165$ bp elements immediately upstream of annotated start codons from $184$ genomes, comprising $169$ bacterial and $15$ archaeal sources. The authors assayed the barcoded library in *E. coli*, *B. subtilis*, and *P. aeruginosa*, measuring sfGFP mRNA counts relative to plasmid DNA. The study also measured protein expression by fluorescence sorting, but full-library translation measurements are not deposited. The available transcription data therefore support a comparison of the same regulatory sequences across three prokaryotic hosts.

We use the released DNA and RNA counts directly, with a separate readout for each host. We record the source genome of each sequence and recover its genomic anchor by exact matching against the source contig, correcting deposited coordinates that do not reconstruct the oligo. The reference collection contains 272 source contigs, and every evaluated record has a genomic anchor. Identical assayed sequences can occur at distinct loci across the source genomes, with different flanking context. We average their per-host counts across the corresponding loci and retain one record per source locus, preserving provenance and total count mass across those records. These repeated sequence records are not independent sequence observations, and preserving count mass does not make their likelihood contribution identical to scoring a single pooled record.

### A.3 Training tasks

For each backbone, we fit a separate model to each benchmark task. In tasks measured across several cell lines or conditions, each readout defines a subtask. These subtasks share the backbone, any adapters, and the hidden layers of the prediction head, with a separate output for each subtask. We train them jointly, averaging each oligo's losses over its observed subtasks and then over oligos. Missing readouts are masked, so an oligo contributes only to the subtasks for which it was measured.

**Input–output count assays.** Let $i$ index the $M$ oligos and $c$ index cell lines or conditions. For each oligo, let $\mathcal{C}_i$ contain the subtasks with finite paired counts $x_{ic}, y_{ic} \geq 0$ and a total $N_{ic} = x_{ic} + y_{ic} > 0$. The head predicts the output fractions $\hat{\pi}_{ic} = (1 + e^{-\theta_{ic}})^{-1}$. The joint excess NLL is

$$\begin{aligned} \mathcal{L} &= \frac{1}{M} \sum_{i=1}^{M} \ell_i, \qquad \ell_i = \frac{\sum_{c \in \mathcal{C}_i} \ell_{ic}}{\max(1, |\mathcal{C}_i|)}, \\ \ell_{ic} &= x_{ic} \log \frac{x_{ic}}{N_{ic}(1 - \hat{\pi}_{ic})} + y_{ic} \log \frac{y_{ic}}{N_{ic} \hat{\pi}_{ic}}. \end{aligned} \tag{3}$$

The per-subtask loss $\ell_{ic}$ subtracts the NLL of the saturated predictor $\pi_{ic} = y_{ic}/N_{ic}$. A term with zero observed count contributes zero, so both $x_{ic} = 0, y_{ic} > 0$ and $x_{ic} > 0, y_{ic} = 0$ remain eligible. An oligo with no observed subtasks contributes zero. Averaging within each oligo prevents oligos measured in more conditions from receiving greater weight solely because they have more readouts. The expression also applies to nonnegative fractional pseudocounts. For $x_{ic} > 0$, it equals negative-binomial excess NLL with a size parameter $x_{ic}$ and a predicted mean $\hat{\mu}_{ic} = x_{ic} e^{\theta_{ic}}$. The saturated subtraction removes a prediction-independent term and preserves the fitted optimum for the same eligible observations.

**Binned-count assays.** Each binned task predicts a single distribution over its $K$ ordered bins. Let $n_{ik}$ be the count in bin $k$, $N_i = \sum_k n_{ik}$, and $q_{ik} = n_{ik}/N_i$ for $N_i > 0$. The head produces one logit $\theta_{ik}$ per bin, transformed into the probabilities $\hat{\pi}_{ik} = e^{\theta_{ik}} / \sum_{k'=1}^{K} e^{\theta_{ik'}}$. The multinomial excess NLL is

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \ell_i, \qquad \ell_i = \sum_{k=1}^{K} n_{ik} \log \frac{q_{ik}}{\hat{\pi}_{ik}}. \tag{4}$$

Zero-count terms contribute zero, and we set $\ell_i = 0$ for records with no reads. The bins are categories of one jointly normalized outcome. Training and checkpoint selection use the oligo-averaged loss $\mathcal{L}$, while the reported NLL per read uses the read-depth normalization described in Appendix B.1.

### A.4 Correlation metrics and aggregation

**Activity scale and precision.** For paired-count assays, the measured activity is $m_i = \log[(y_i + 1)/(x_i + 1)]$ and the predicted activity is $\hat{m}_i = \theta_i$. To quantify measurement uncertainty, we use a likelihood proportional to $\pi_i^{y_i}(1 - \pi_i)^{x_i}$ and a uniform $\mathrm{Beta}(1, 1)$ prior on the output fraction $\pi_i$. The posterior is $\pi_i \mid x_i, y_i \sim \mathrm{Beta}(y_i + 1, x_i + 1)$. Its induced density on the latent log activity $\log[\pi_i/(1 - \pi_i)]$ has the mode $m_i$. A Laplace approximation gives

$$\mathrm{Var}\left( \log \frac{\pi_i}{1 - \pi_i} \,\middle|\, x_i, y_i \right) \approx \frac{1}{x_i + 1} + \frac{1}{y_i + 1}, \qquad w_i = \frac{(x_i + 1)(y_i + 1)}{x_i + y_i + 2}. \tag{5}$$

Thus, $w_i$ approximates the inverse posterior variance of the latent activity, reducing the influence of uncertain measurements. This approximation captures counting uncertainty but does not model biological replicate variability. The added ones stabilize zero counts but make the transformed activity sensitive to a common rescaling of both counts. Reconstructed pseudocounts therefore preserve the published unsmoothed ratio, while correlation scoring applies the same smoothing rule used for raw-count assays. Rate correlations require positive input counts. For `human_splicing`, $x_i$ and $y_i$ are the excluded and included isoform counts, so $m_i$ denotes the smoothed log inclusion odds. Correlations for this task require positive total depth and admit either zero-count boundary.

For binned assays, the predicted and observed activities are the expected ordered-bin indices under $\hat{\boldsymbol{\pi}}_i$ and $\boldsymbol{q}_i$. Each record's correlation weight is its total read count $N_i$.

**Global correlation.** Within one subtask, $r_{\mathsf{Global}}$ is the weighted Pearson correlation across eligible oligos:

$$r_w(\hat{m}, m) = \frac{\sum_i w_i(\hat{m}_i - \bar{\hat{m}}_w)(m_i - \bar{m}_w)}{\sqrt{\sum_i w_i(\hat{m}_i - \bar{\hat{m}}_w)^2 \sum_i w_i(m_i - \bar{m}_w)^2}}, \qquad \bar{m}_w = \frac{\sum_i w_i m_i}{\sum_i w_i}. \tag{6}$$

The predicted weighted mean is defined analogously. We omit undefined correlations, including those with fewer than two eligible observations or constant predicted or measured activities.

**Variant correlation.** Among records with the same `subset` value, the scorer sums observed counts for repeated records of an allele or perturbation and forms differences between distinct measured alleles. For alleles $i$ and $j$, the predicted and measured differences are $\hat{m}_i - \hat{m}_j$ and $m_i - m_j$, with the pair weight $(w_i^{-1} + w_j^{-1})^{-1}$. The weighted Pearson correlation across these pairs is $r_{\mathsf{Variant}}$. The reported scorer includes all measured allele pairs in a group, which can include more than two alleles in designed-perturbation libraries, and omits pairs with exactly zero measured difference. Thus, this metric does not always reduce to one reference–alternate comparison per locus. Pair counts need not represent independent biological loci.

**Mean window correlation.** We compute $r_w$ separately within each parent element, cassette exon, or perturbation group and combine the defined group correlations by an equal-weighted Fisher-$z$ mean. This gives $r_{\mathsf{MeanWindow}}$. Eligibility is evaluated separately for each condition. Groups require at least two measured members by default, at least thirty for `human_splicing`, and at least four for `at_promoter_v24_gatc`. This metric removes between-group offsets and measures relative activity differences within groups.

**Cell-line correlation.** We retain oligos measured in every cell line included in an analysis. We use six cell lines for `human_3utr`, five for `human_splicing`, and HCT116, HepG2, K562, and SK-N-SH for `human_tx_s26_snp`. An exploratory analysis of `human_tx_a25` uses HepG2, K562, and WTC11. For predictions and measurements separately, we subtract each cell line's activity offset, estimated with equal total weight per sequence cluster and precision weighting within a cluster. We then compute a precision-weighted Pearson correlation across cell lines for each oligo and combine the defined correlations with an equal-weighted Fisher-$z$ mean to obtain $r_{\mathsf{CellLine}}$. We exclude `human_tx_s26_satmut`, which has only two cell lines, from this profile metric.

**Aggregation.** We combine defined correlations using the equally weighted Fisher-$z$ average:

$$\bar{r} = \tanh\left[\frac{1}{M}\sum_{c=1}^{M} \operatorname{arctanh}(r_c)\right], \tag{7}$$

where $M$ is the number of defined correlations being averaged and $c$ indexes the contributing conditions, groups, or oligos, depending on the metric. We omit undefined correlations from both the sum and $M$ and report no correlation when $M = 0$. Precision weights operate within a correlation. The outer average gives each contributing correlation equal weight, irrespective of sample size or read depth. The excess NLL per read is averaged arithmetically across defined conditions. Comparisons across backbone rankings use Spearman correlation, which is distinct from the Pearson correlations used to score assay predictions.

### A.5 CELL-LINE DIFFERENCES WITHIN SATURATION-MUTAGENIZED ELEMENTS

For `human_tx_s26_satmut`, we distinguish differences between regulatory elements from the effects of individual substitutions. Its aligned test subset contains 15,511 oligos from 11 parent elements and 26 sequence backgrounds. Here a sequence background is a parent element with a fixed haplotype, distinct from a model backbone. We use the smoothed observed log activities $m_i$ and the predicted log activities $\hat{m}_i$. For each sequence background, we average the HepG2-minus-K562 activity difference across its oligos. The Spearman correlation between predicted and observed background means is $\rho_{\Delta\text{Background}}$. This measures differences across backgrounds, without requiring that a model predict individual mutation effects.

For a substitution $i$ within a sequence background, we instead evaluate the difference between its effects in the two cell lines:

$$s_i = (m_{i,\text{HepG2}} - m_{\text{WT,HepG2}}) - (m_{i,\text{K562}} - m_{\text{WT,K562}}), \tag{8}$$

where WT is the wild-type oligo for that background. The predicted effects $\hat{s}_i$ use the corresponding predicted activities. We compare $\hat{s}_i$ with $s_i$ using a weighted Kendall correlation within each background. To emphasize large effects without ranking importance from the same measurement being scored, we match substitutions by position and alternate base across the other haplotypes of the same parent element. We average their observed effects and rank substitutions by decreasing absolute effect, starting at rank $q_i = 0$. The weighted Kendall statistic uses additive pair weights $1/(q_i + 1) + 1/(q_j + 1)$, with the tie normalization implemented by `scipy.stats.weightedtau`. A background contributes only if at least 100 substitutions have matched measurements. We average the resulting correlations within each parent element and then equally across elements to obtain $\tau_{\Delta\text{Mutation}}$.

The within-cell-line scores $\tau_{\text{HepG2}}$ and $\tau_{\text{K562}}$ use the same procedure on each cell line's mutation effect relative to its wild type. The partner oligos have different haplotype backgrounds, so their agreement is not a technical-replicate noise ceiling. We report point estimates without uncertainty estimates from repeated training seeds. In Fig. S9, we distinguish this 12-model cohort from the larger model cohort evaluated using $r_{\text{CellLine}}$.

## B MODEL ADAPTATION AND EVALUATION

### B.1 TRAINING, MODEL SELECTION, AND EXCEPTIONS

We adapt backbones for at most 30 epochs using AdamW with no weight decay, a cosine learning-rate schedule, and a warm-up fraction of 0.0333, approximately one epoch. Microbatch sizes depend on the backbone and available device memory, with gradient accumulation targeting an effective batch size of 64. We evaluate the validation loss after each epoch and use early stopping with a patience of 8 and a minimum improvement of $10^{-3}$. We select the checkpoint with the lowest validation loss. For cross-library transfer, we use the target task's validation set for model selection. The default learning rates are $3 \times 10^{-4}$ for adapted backbones and $10^{-3}$ for the from-scratch baseline, with a seed of 42.

**Loss normalization and model selection.** Checkpoint selection and the reported NLL use different normalizations. For checkpoint selection, the validation loss averages each oligo's excess loss over its observed conditions and then over oligos. The reported excess NLL per read instead uses $\sum_i \ell_{ic} / \sum_i N_{ic}$ within each condition $c$, then averages the condition scores equally. For a binned task, the same ratio uses its single loss $\ell_i$ and total bin count $N_i$. Consequently, the epoch minimizing the reported validation NLL per read need not be the checkpoint selected by the training loss. We apply the same selection rule to frozen-embedding and LoRA runs and report per-read NLL at each run's selected checkpoint.

**Exceptional learning rate settings.** The from-scratch baseline and NTv3-650M runs on `human_tx_a25` use $10^{-4}$ because the higher rates produced nearly constant predictions. Evo 2-1B and Evo 2-7B also use $10^{-4}$. We specify each run's optimizer settings before test evaluation. If a run produces exactly constant predictions for a validation subtask whose measured activities vary, we allow one restart at $10^{-4}$ when its original learning rate was higher. The restart uses the same seed

Table S2: **Pretrained backbones used in REPORTER.** Parameter counts are approximate. Context size gives the input window for length-preserving backbones and the receptive field for backbones that trim their outputs. Required flanks appear in Appendix B.2. Hugging Face IDs link to the weights. AlphaGenome's second line identifies the converted PyTorch checkpoint used. The in-house MLM weights will be released with the paper. The from-scratch baseline is described in Appendix B.2.

| Model | Parameters (M) | Context size (bp) | Hugging Face ID | Study of origin |
|---|---|---|---|---|
| Human MLM Baseline | 265 | 1,549 | To be released | This work |
| Mouse MLM Baseline | 265 | 1,549 | To be released | This work |
| *Drosophila* MLM Baseline | 66 | 1,549 | To be released | This work |
| *S. cerevisiae* MLM Baseline | 17 | 1,549 | To be released | This work |
| *Arabidopsis* MLM Baseline | 66 | 1,549 | To be released | This work |
| *E. coli* MLM Baseline | 4 | 1,549 | To be released | This work |
| AlphaGenome | 90 | 8,192 | `gtca/alphagenome_pytorch` `model_all_folds.` `safetensors` | Avsec et al. 2026 |
| Evo 2-1B | 1,000 | 8,192 | `arcinstitute/evo2_1b_base` | Brixi et al. 2026 |
| Evo 2-7B | 7,000 | 8,192 | `arcinstitute/evo2_7b_base` | Brixi et al. 2026 |
| Caduceus-PS | <1 | 1,024 | `kuleshov-group/` `caduceus-ps_seqlen-1k_d_` `model-118_n_layer-4_` `lr-8e-3` | Schiff et al. 2024 |
| PlantCAD2-Small | 88 | 8,192 | `kuleshov-group/` `PlantCAD2-Small-l24-d0768` | Zhai et al. 2025 |
| PlantCAD2-Medium | 311 | 8,192 | `kuleshov-group/` `PlantCAD2-Medium-l48-d1024` | Zhai et al. 2025 |
| MarinDNA-m5.1 | 1,100 | 255 | `marin-dna/` `marin-dna-exp135-m5.1` | Open Athena 2026 |
| PhyloGPN | 83 | 481 | `songlab/PhyloGPN` | Albors et al. 2025 |
| GPN-Brassicales | 66 | 512 | `songlab/gpn-brassicales` | Benegas et al. 2023 |
| NTv3-8M | 8 | 32,768 | `InstaDeepAI/NTv3_8M_pre` | Boshar et al. 2025 |
| NTv3-100M | 100 | 32,768 | `InstaDeepAI/NTv3_100M_pre` | Boshar et al. 2025 |
| NTv3-650M | 650 | 32,768 | `InstaDeepAI/NTv3_650M_pre` | Boshar et al. 2025 |
| NTv3-650M-Post | 650 | 32,768 | `InstaDeepAI/NTv3_650M_post` | Boshar et al. 2025 |
| JEPA-NTv3-100M | 100 | 32,768 | `nvidia/NV-JEPA-DNA-NTv3` | Larey et al. 2026 |
| BiGamba-dual | 7 | 2,048 | `micanonsens/` `bigamba-dual-step44000` | Consens et al. 2026 |
| ArGamba-dual | 67 | 2,048 | `micanonsens/` `argamba-dual-step44000` | Consens et al. 2026 |
| DNABERT-2 | 117 | 1,024 | `zhihan1996/DNABERT-2-117M` | Zhou et al. 2024 |
| JEPA-DNABERT-2 | 117 | 1,024 | `nvidia/` `NV-JEPA-DNA-DNABERT2` | Larey et al. 2026 |
| SpeciesLM 5' | 90 | 1,003 | `gagneurlab/SpeciesLM` | Karollus et al. 2024 |
| SpeciesLM 5' (agn) | 89 | 1,003 | `gagneurlab/SpeciesLM` | Karollus et al. 2024 |

and pretrained initialization. We select the attempt using validation diagnostics before evaluating the test set.

**Validation–test comparison.** We evaluate validation and test predictions from the same checkpoint, selected by the lowest validation loss. In Fig. S3, both axes report excess NLL per read at that checkpoint, using the normalization described above.

### B.2 Models

We attach the shared convolutional prediction head in Fig. S2 to each pretrained backbone in Table S2. The head reads nucleotide-aligned hidden states at the assayed oligo positions, while the

backbone can also use the surrounding genomic sequence. We train the head jointly with LoRA adapters and keep pretrained backbone weights fixed. Below, we specify each backbone's adapter targets, representation extraction, tokenization, and genomic context.

For backbones that preserve sequence length, we fill the configured context window with reference sequence around the oligo, dividing the available flank as evenly as possible between the two sides. Valid-convolution backbones instead require a fixed flank equal to their receptive-field radius on each side. Thus, the 1,549 bp receptive field of the species-specific MLM baselines requires 774 bp on either side of the oligo, and PhyloGPN's 481 bp field requires 240 bp per side. These receptive fields differ from the total input length: a 200 bp oligo gives total inputs of 1,748 and 680 bp, respectively.

Our default LoRA rank and scaling parameter are both 16, with no dropout or bias adaptation. Adapter targets and trainable parameter counts vary by architecture, so adaptation budgets differ across backbones.

For NTv3, SpeciesLM, and MarinDNA, we use 4-bit NF4 weights with nested quantization and bfloat16 computation by default. Quantized species-specific MLM baselines use the same weight format for feed-forward linear layers while retaining full-precision convolutions and embeddings. AlphaGenome, PhyloGPN, GPN, DNABERT-2, Caduceus, and the Gamba backbones use no 4-bit weight quantization. The prediction head remains unquantized. The numerical precision for Evo 2 is described below.

**From-Scratch Baseline.** To measure what assay supervision alone can recover, we train a gated dilated CNN from random initialization independently on each task. The model embeds each nucleotide from a six-symbol alphabet, `A`, `C`, `G`, `T`, `N`, and padding, into 128 channels. Its trunk contains six stacks of four blocks. Within each stack, convolutional kernels have widths 11, 5, 5, and 5, with dilations 1, 2, 4, and 8. Each block applies two convolutions, multiplies one output by a SiLU gate computed from the other, and adds the layer-normalized result to the input. A second residual sublayer expands the channels from 128 to 1,024, applies ReLU, projects back to 128, and applies layer normalization. Same padding preserves sequence length throughout. The trunk has approximately 11.45M parameters and a 397 bp receptive field. It receives only the assayed oligo, with padding supplying positions outside that sequence. We attach the common task head in Fig. S2 and jointly train every trunk and head parameter, without LoRA, using a default learning rate of $10^{-3}$. We use the assay-specific losses in Appendix A.3 and the checkpoint-selection and optimization procedures in Appendix B.1.

**Species-Specific MLM Baselines.** The six masked-language-model baselines use residual gated dilated CNNs pretrained on one reference genome each: *E. coli* MG1655, *S. cerevisiae*, *Arabidopsis* TAIR10, *Drosophila* dm6, human hg38, and mouse mm39. They tokenize at single-nucleotide resolution over `A`, `C`, `G`, `T`, `N`, and a padding symbol. We retain the encoder and discard the pretraining output layer. Within each block, we adapt the two gated-convolution projections, `filter_conv` and `gate_conv`, and the two feed-forward linear projections, `layers.0` and `layers.2`. The valid convolutions retain nucleotide resolution while cropping 774 positions per side. Supplying that much genomic flank leaves one output position per assayed nucleotide. We assign the hg38, mm39, dm6, and yeast backbones to the corresponding human, mouse, fly, and yeast tasks. We use the TAIR10 backbone for all plant tasks and the MG1655 backbone for both prokaryotic studies. Thus, the maize, sorghum, tomato, and multispecies prokaryotic comparisons include cross-species transfer by these baselines. Each task contributes one designated MLM backbone to the collapsed family row.

The species-specific MLM baselines use the same gated convolutional family, with six blocks per stack, kernels 11, 5, 5, 5, 5, and 5, and dilations extending to 32. Their shared 1,549 bp receptive field requires 774 bp of genomic flank per side to retain one representation per assayed nucleotide after trimming. Model widths vary by reference genome, as shown in Fig. S1. Downstream adaptation freezes the pretrained weights and trains the adapters and task head, whereas the from-scratch baseline trains all weights on the assay. The comparison therefore measures the practical benefit of these pretrained backbones under the benchmark protocol. Differences in width, depth, context, padding, and trainable parameters prevent isolating pretraining alone.

**AlphaGenome.** We adapt the convolutional encoder of AlphaGenome (Avsec et al., 2026) following the modular-encoder approach of Murphy et al. (2026). We remove the transformer tower and decoder heads and retain the approximately 90M-parameter encoder. We adapt the embedder's `conv1` and the `conv` projections of its weight-standardized blocks. Single-nucleotide inputs are converted to one-hot encodings, with ambiguous and padding bases represented by zero vectors. The encoder output is downsampled by a factor of 128. We repeat each hidden state over its 128 nucleotide positions before the common head gathers the oligo. This alignment does not recover information lost through downsampling. We use an 8,192 bp input window and pretrained weights converted to PyTorch. In Fig. S10, we compare transfer performance across input windows from 512 to 8,192 bp. Results characterize this encoder and adaptation procedure. The full model's transformer tower, decoder, and megabase-scale input are outside this implementation.

**Caduceus and PlantCAD2.** Caduceus-PS (Schiff et al., 2024) and PlantCAD2-Small and -Medium (Zhai et al., 2025) use reverse-complement-equivariant bidirectional Mamba backbones. We remove the masked-language-model head and adapt the state-space projections `in_proj`, `x_proj`, `dt_proj`, and `out_proj` inside each mixer. Single-nucleotide tokenization uses right padding and omits the tokenizer's trailing separator so that input positions remain aligned to nucleotides. The final hidden states concatenate the forward and reverse-complement channels, producing a width of $2d_{\text{model}}$. The task head reads these channels directly. Caduceus-PS uses a 1,024 bp window, and both PlantCAD2 sizes use 8,192 bp. These backbones use unquantized weights.

**DNABERT-2.** DNABERT-2 uses a BERT encoder with ALiBi positional biases and a byte-pair tokenizer (Zhou et al., 2024). We supply 1,024 bp inputs and adapt the fused attention projection `Wqkv`, attention output `dense`, gated feed-forward projection `gated_layers`, and block output `wo`. Because a token can span several bases, we repeat each token's hidden state across the nucleotides it represents. Special tokens contribute no nucleotide positions. We read these aligned states rather than the pooled classification-token output and retain fp32 weights. JEPA-DNABERT-2 (Larey et al., 2026) uses the same architecture, input alignment, and adapter targets, with the released JEPA encoder weights replacing the original encoder weights.

**Evo 2.** We adapt the base releases of Evo 2-1B and Evo 2-7B (Brixi et al., 2026) with 8,192 bp genomic inputs and byte-level tokenization, with one token per nucleotide. Both use the per-position output of the block-10 MLP (`blocks.10.mlp.l3`) as the representation for the task head. We selected this readout using an unsupervised magnitude probe, measuring the per-position L2 norm of each block's MLP output across validation oligos and length-matched random sequences. For Evo 2-1B, block 10 marked the onset of a stable intermediate-layer plateau in activation magnitude. An independent probe of Evo 2-7B showed the same transition at block 10, motivating the shared readout layer across both model sizes. We adapt all three MLP projections (`l1`, `l2`, and `l3`) in blocks 0 through 10, giving 33 adapted projections. Blocks after the readout do not contribute to the prediction. The reported runs use a rank of 16, a scaling parameter of 16, no adapter dropout, and a learning rate of $10^{-4}$. Evo 2 is causal, so the oligo representations use preceding sequence but do not receive information from the right flank.

We adapt Evo 2-1B on GPUs with bfloat16 backbone weights and FP8 input projections, and Evo 2-7B on Trainium with bfloat16 computation. Both use the layer-10 readout and MLP adapter targets described above. For the zero-shot VEP comparison, the Evo 2-7B TraitGym scores use the midtrained release, while the REPORTER and ClinVar scores use the base release, as noted in Fig. S8.

**Gamba.** BiGamba-dual and ArGamba-dual (Consens et al., 2026) use a shared single-nucleotide alphabet and 2,048 bp windows. We read the per-position `representation` output and leave the language-model and evolutionary-rate prediction heads unused. BiGamba's bidirectional Mamba trunk has a hidden width of 256 and no reverse-complement channel doubling. We adapt its state-space projections `in_proj`, `x_proj`, `dt_proj`, and `out_proj`. ArGamba uses a causal Jamba trunk with a hidden width of 512, interleaving Mamba, attention, and mixture-of-experts layers. We adapt the same state-space projections together with the attention projections `q_proj`, `k_proj`, `v_proj`, and `o_proj`, leaving the expert feed-forward layers frozen. Both backbones run without 4-bit quantization. ArGamba's oligo representations can use only preceding sequence, whereas BiGamba's bidirectional representation can use both flanks.

**GPN-Brassicales.** GPN-Brassicales is a character-level convolutional masked language model pretrained on Brassicales genomes (Benegas et al., 2023). We read its per-nucleotide hidden states and adapt the convolution `conv.1` and feed-forward projection `ffn.0` in each encoder block. The trunk preserves sequence length, so we fill a 512 bp window with genomic context around the oligo. It runs without 4-bit quantization.

**MarinDNA.** MarinDNA-m5.1 is an autoregressive single-nucleotide DNA language model on a Qwen3 decoder (Open Athena, 2026). We load the causal language model, discard its language-model head, and adapt the attention projections `q_proj` and `v_proj`. Its 256-token context contains one beginning-of-sequence token and 255 bases. We add that token when encoding the sequence and remove its hidden-state position before passing the nucleotide representations to the task head. For the 50–249 bp oligos evaluated here, the input contains 6–205 bp of total genomic flank. Because the decoder is causal, the assayed bases can use the left flank and preceding oligo bases but not the right flank.

**Nucleotide Transformer v3.** We adapt the 8M, 100M, and 650M pretraining backbones and NTv3-650M-Post (Boshar et al., 2025) together with JEPA-NTv3-100M (Larey et al., 2026). All use 32,768 bp inputs and the released tokenizer, with input lengths aligned to a multiple of 128. LoRA targets the attention projections named `linear` and the feed-forward projections `fc1` and `fc2`. We extract the nucleotide-resolution hidden states for the common head. NTv3-650M-Post additionally requires a species identifier, which we derive from the task's source assembly. Its representation is the post-deconvolution `embedding` output. JEPA-NTv3-100M uses the NTv3-100M architecture with the released JEPA target-encoder weights. The adapter targets and representation extraction match those of the corresponding pretraining backbone.

**PhyloGPN.** PhyloGPN is a reverse-complement-equivariant ByteNet CNN trained on the likelihood of mammalian alignment columns under an F81 model (Albors et al., 2025). We retain the layer-normalized convolutional trunk and discard its pretraining output head. We adapt `conv1`, `conv2`, and `conv3` in each block, excluding `output_conv`, which is unused when extracting trunk representations. The model tokenizes single nucleotides, and forty valid-convolution blocks alternating dilation 1 and 5 give a 481 bp receptive field. We supply 240 bp of genomic flank per side so that the cropped output retains one hidden state per oligo nucleotide. The approximately 83M-parameter trunk runs without 4-bit quantization.

**PlantCAD2.** See Caduceus and PlantCAD2 above.

**SpeciesLM.** The species-aware and species-agnostic SpeciesLM 5′ backbones use DNABERT-style BERT encoders over overlapping stride-1 6-mers (Karollus et al., 2024). We supply 1,003 bp inputs and adapt the attention `query` and `value` projections. For the species-aware backbone, we prepend the *Kazachstania africana* species token used as the authors' proxy for *Saccharomyces*, which was held out of pretraining. The agnostic backbone receives no species token. We discard the framing and species-token hidden states, align each 6-mer state to its starting nucleotide, and repeat the final 6-mer state for the last five nucleotides, which have no starting 6-mer. The common head then reads the resulting nucleotide-aligned oligo representation.

```
KERNELS = [11, 5, 5, 5, 5, 5]  # Species-Specific MLM Baseline
# KERNELS = [11, 5, 5, 5]       # From-Scratch Baseline
NUM_STACKS = 6

def gated_block(x, kernel, dilation, trim):
    filtered = conv_same(x, kernel, dilation)  # filter_conv
    gate = silu(conv_same(x, kernel, dilation)) # gate_conv
    x = x + layer_norm(filtered * gate)
    if trim:
        radius = (kernel - 1) * dilation // 2
        x = x[:, radius:-radius]
    return x + layer_norm(feedforward(x))
    # feedforward: Linear(D, 8*D), ReLU, Linear(8*D, D)

def trunk(tokens, trim):
    x = embed(tokens)                          # (B, L, D)
    for _ in range(NUM_STACKS):
        for i, kernel in enumerate(KERNELS):
            x = gated_block(x, kernel, 2 ** i, trim)
    return x

per_stack = sum((k - 1) * 2 ** i for i, k in enumerate(KERNELS))
receptive_field = NUM_STACKS * per_stack + 1
# 1549 bp for the MLM baseline, 397 bp for the from-scratch baseline
```

Figure S1: **Baseline architectures.** Both architectures use six stacks of gated convolutions and feed-forward sublayers with residual connections and layer normalization. The dilation doubles within each stack and resets between stacks. The species-specific MLM baselines use six blocks per stack, a 1,549 bp receptive field, and a 774 bp crop per side. The embedding widths $D$ are 64 for MG1655, 128 for yeast, 256 for TAIR10 and dm6, and 512 for hg38 and mm39. The from-scratch baseline uses four blocks per stack, a width of 128, a 397 bp receptive field, and no trimming. Both use the head in Fig. S2.

```
import torch
from torch import nn

def make_head(D, L, O):
    return nn.Sequential(
        nn.Conv1d(D, 64, kernel_size=L, stride=1, padding=0),
        nn.Flatten(start_dim=1),
        nn.ReLU(),
        nn.Linear(64, 64), nn.ReLU(),
        nn.Linear(64, 64), nn.Linear(64, O),
    )

def predict(head, hidden, oligo_mask):
    # hidden: (B, S, D), aligned to nucleotide positions
    # oligo_mask: (B, S), with L contiguous oligo positions
    L = head[0].kernel_size[0]
    starts = oligo_mask.long().argmax(dim=1)
    pos = starts[:, None] + torch.arange(L, device=hidden.device)
    index = pos[:, :, None].expand(-1, -1, hidden.shape[-1])
    oligo = hidden.gather(1, index)  # (B, L, D)
    features = oligo.to(head[0].weight.dtype).transpose(1, 2)
    return head(features)  # (B, O), raw assay logits
```

Figure S2: **Shared convolutional prediction head.** Every backbone uses this head, fitted separately for each task. The inputs contain $B$ sequences, $S$ nucleotide positions after any upsampling, and $D$ channels. The mask extracts $L$ oligo positions, and the convolution produces 64 features. Convolutional and linear layers have biases, with no dropout or normalization. The output width $O$ counts cell lines or conditions for input–output count assays, or bins for binned-count assays. The likelihood applies sigmoid or softmax to the logits, respectively (Appendix A.3).

# C SUPPORTING RESULTS

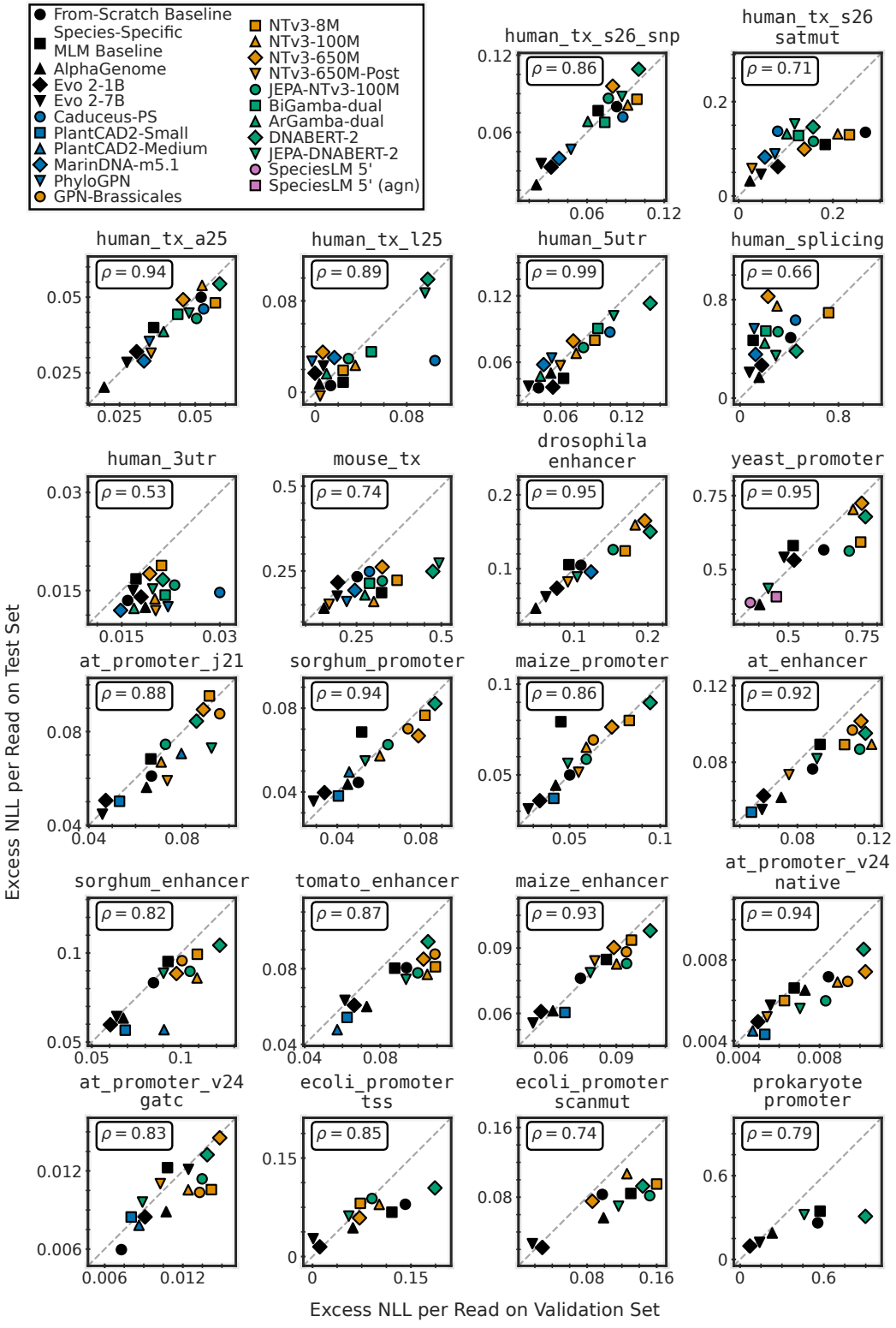


Figure S3: **Validation and test excess NLL per read.** Both splits use the validation-selected checkpoint. Dashed lines mark equality. Boxes give signed Spearman correlations. Markers are displaced to avoid overlap.

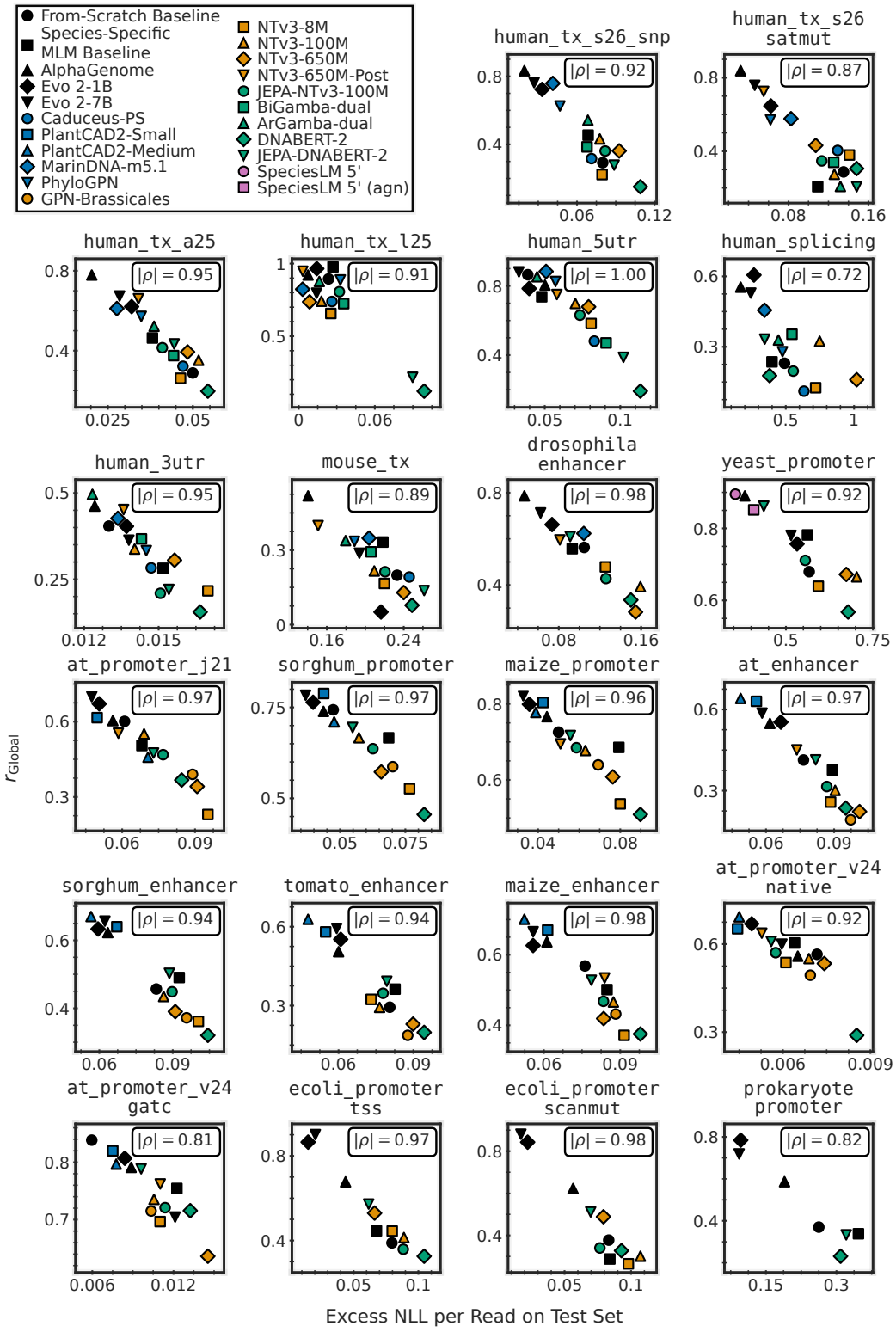


Figure S4: **Test excess NLL per read versus $r_{\text{Global}}$.** Associations are negative for every task. Boxes show Spearman correlation magnitudes across backbones. Markers are displaced to avoid overlap, and statistics use original scores. Subtask aggregation follows Appendix A.4.

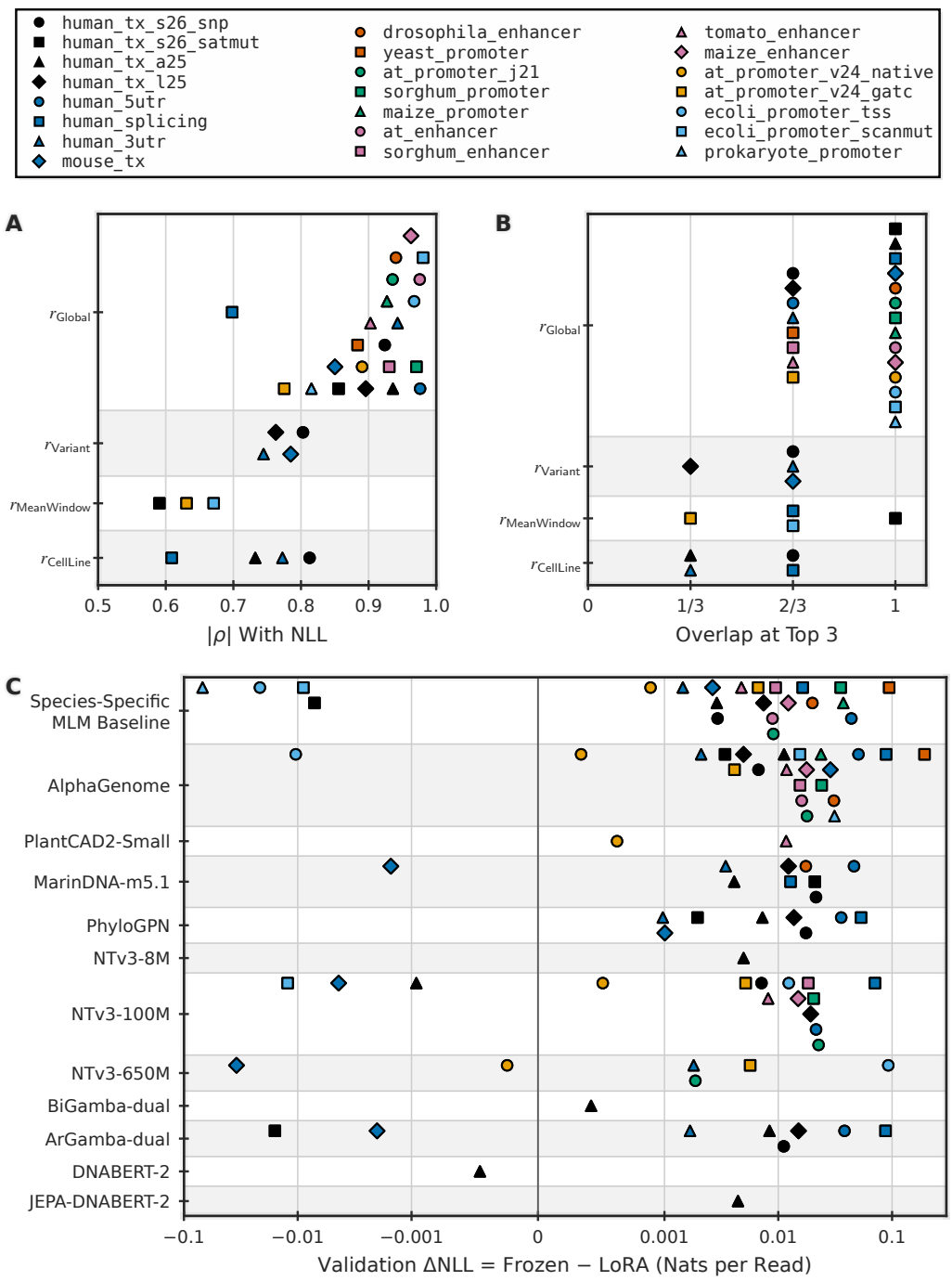


Figure S5: **Metric associations, ranking overlap, and adaptation gains.** **(A)** We show Spearman correlation magnitudes between test NLL and activity correlations across backbones. All signed associations are negative. **(B)** We report top-three backbone overlap using all available models. **(C)** We compare training only the prediction head on frozen backbone embeddings with training the head and LoRA adapters. Each point gives frozen minus LoRA validation NLL per read at each run's validation-selected checkpoint (Appendix B.1). Positive differences favor LoRA (82 of 96 backbone–task pairs), and negative differences favor frozen embeddings. The symmetric logarithmic axis is linear within $\pm 0.001$ nats per read. Designated MLM backbones share one row. Colors and shapes identify tasks.

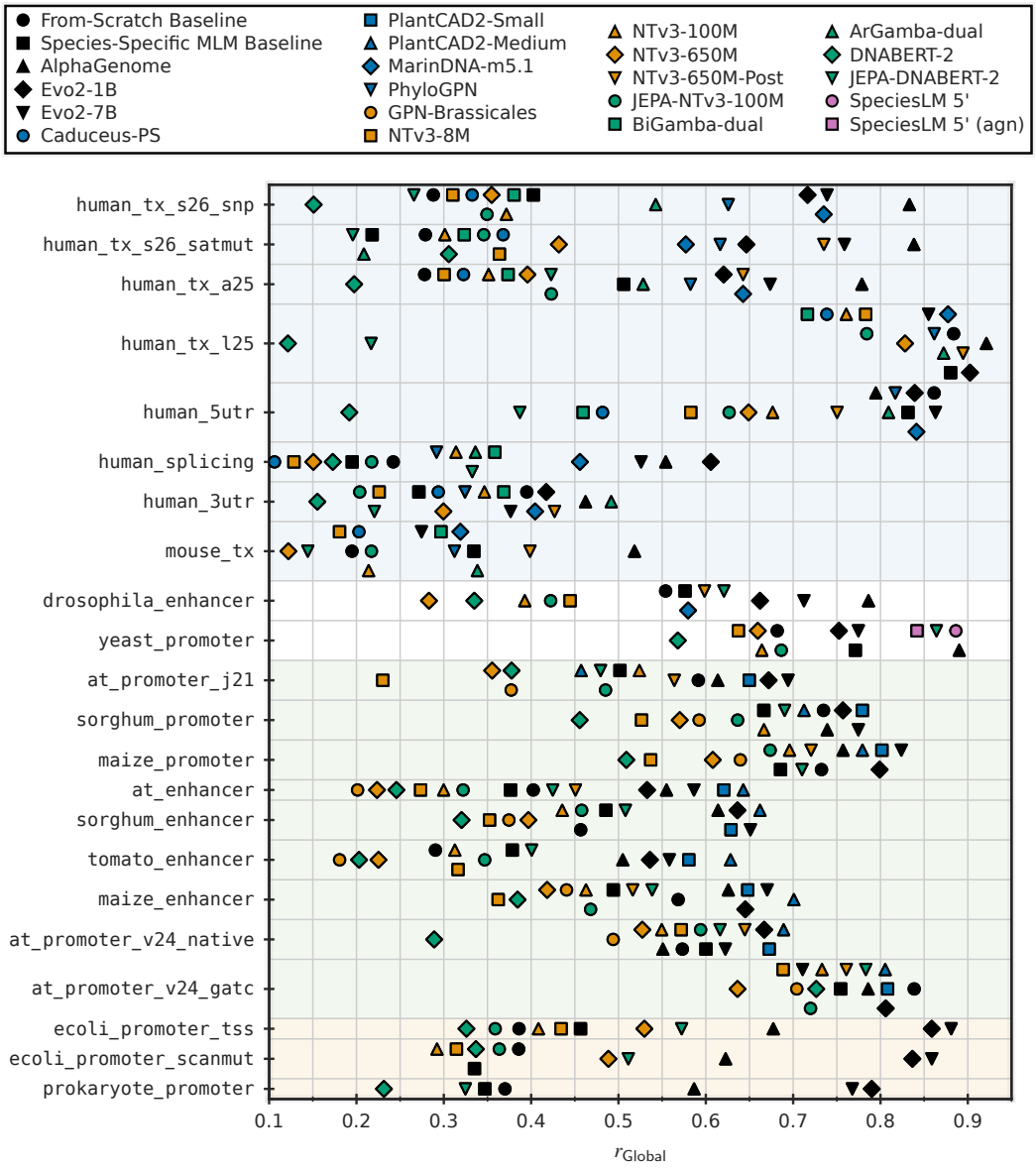


Figure S6: **Available benchmark results for the full model panel.** This expands Fig. 2A. Each point gives a model's test $r_{\text{Global}}$ for one task. Row backgrounds are blue for mammals, green for plants, and cream for prokaryotes. The panel includes the pretrained backbones in Table S2 and the from-scratch baseline wherever a score is available. The species-specific MLM baselines follow the task assignment in Appendix B.2, including the Arabidopsis backbone on all plant tasks and the MG1655 backbone on both prokaryotic studies. Points are vertically offset only to prevent overlap.

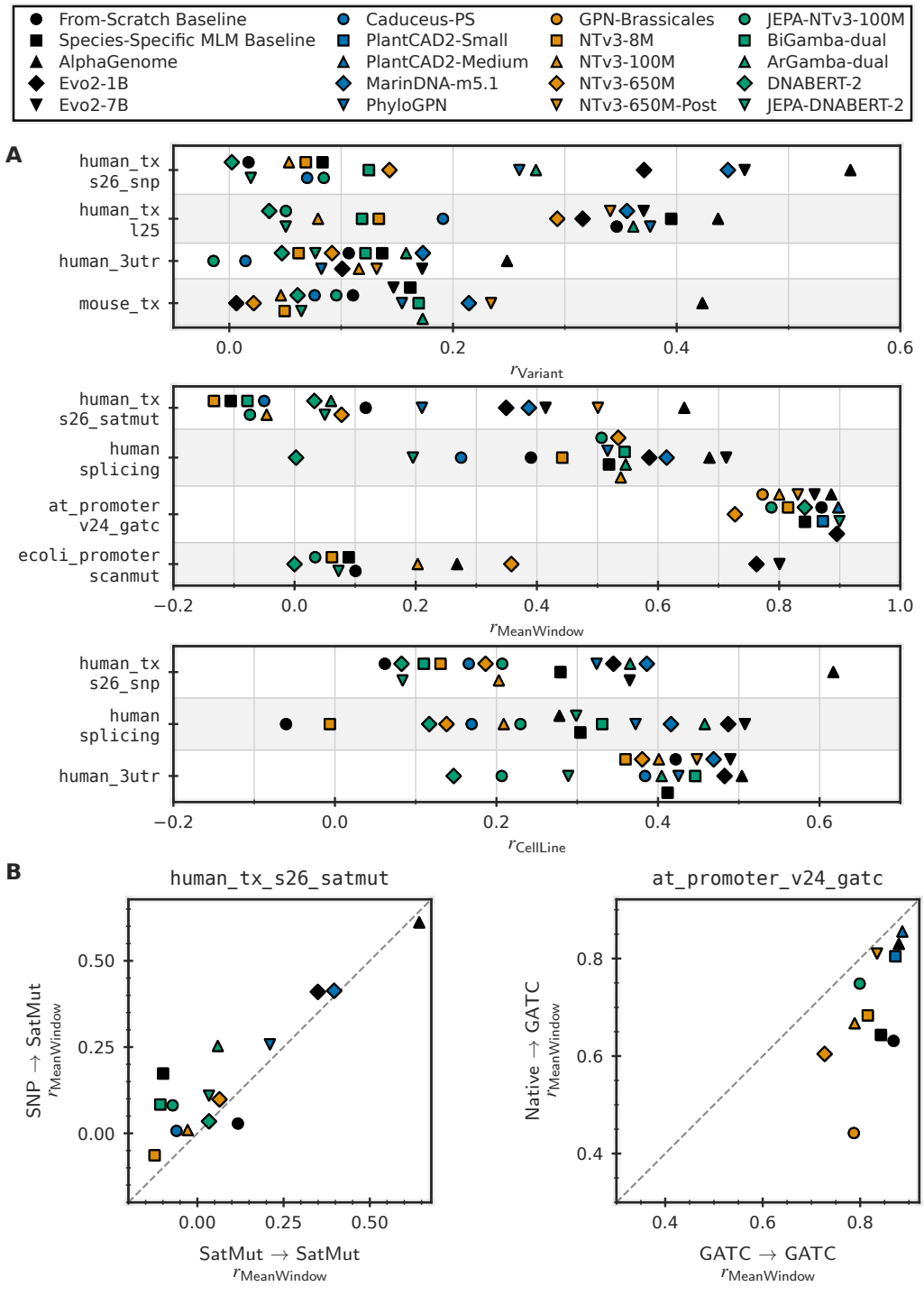


Figure S7: **Auxiliary correlations and cross-library transfer. (A)** We extend Fig. 2B to all available backbones. **(B)** Transfer from `human_tx_s26_snp` to `human_tx_s26_satmut` and from `at_promoter_v24_native` to `at_promoter_v24_gatc` uses target test sets and $r_{\text{MeanWindow}}$. Above-diagonal points favor transfer. Joint partitioning keeps source-training groups separate from target-evaluation groups (Appendix A.1). The species-specific MLM baselines follow Appendix B.2. Both panels share a model legend, with slight offsets separating overlapping markers.

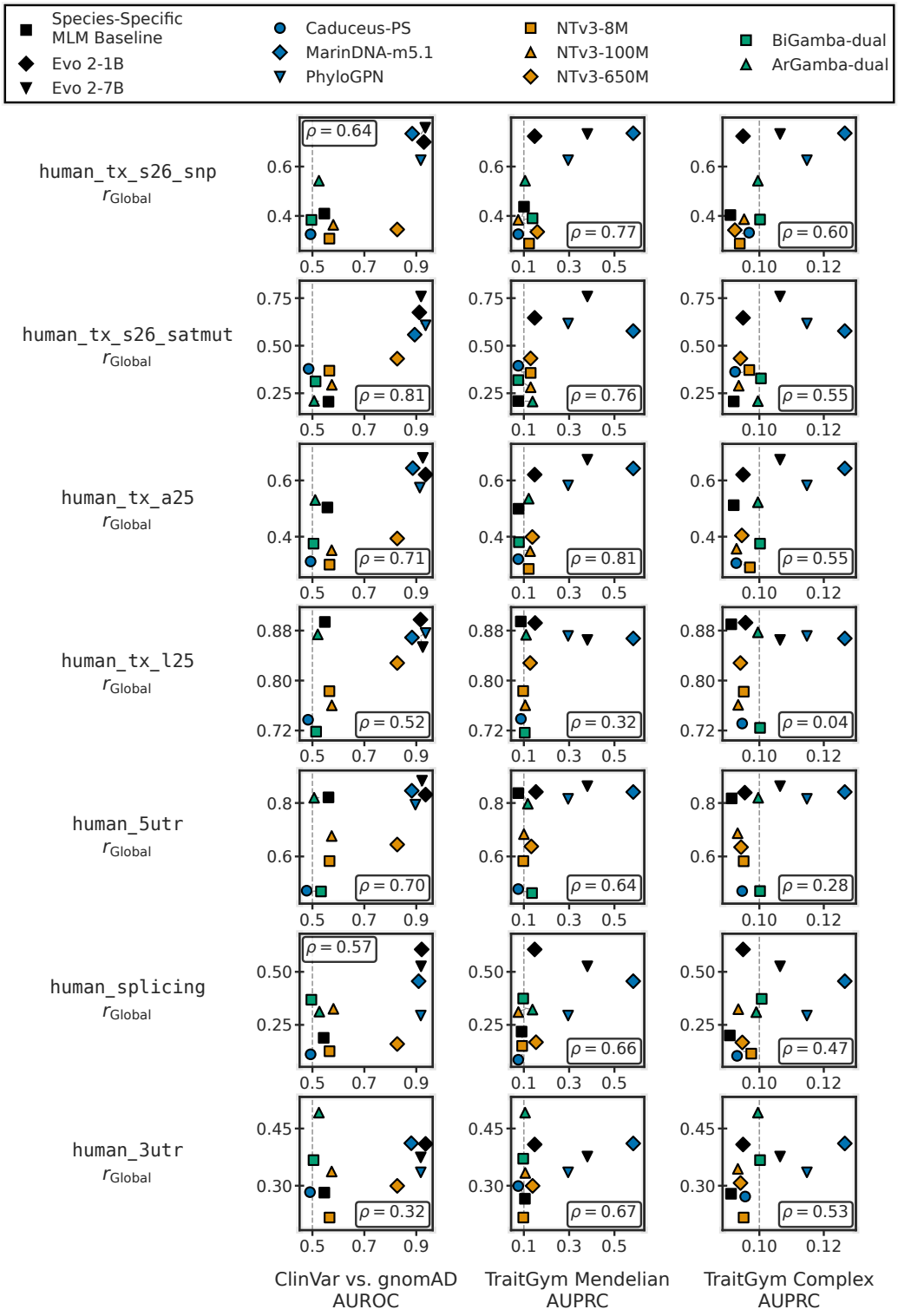


Figure S8: **Zero-shot VEP and MPRA transfer performance.** Columns report AUROC for ClinVar pathogenic versus gnomAD common missense variants, then pooled average precision, labeled AUPRC, for TraitGym Mendelian and Complex. Boxes give Spearman correlations with MPRA test $r_{\text{Global}}$. Gray lines connect displaced markers to the original coordinates used for statistics. Dashed vertical lines mark chance. The species-specific MLM baseline uses the human backbone. Evo 2-7B uses base weights for ClinVar and MPRA and midtrained weights for TraitGym.

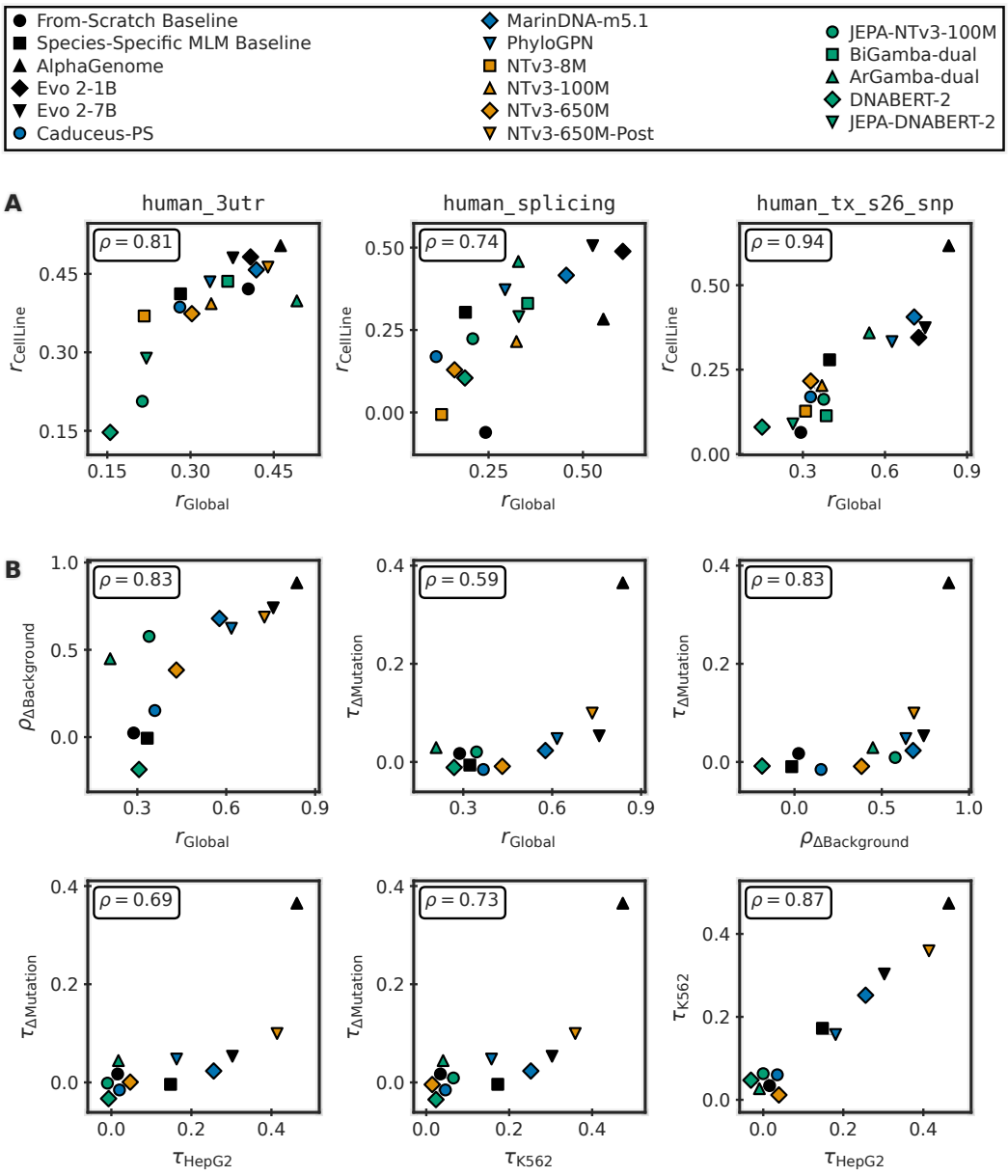


Figure S9: **Overall and cell-line-specific prediction performance.** **(A)** Each panel compares test $r_{\text{Global}}$ with $r_{\text{CellLine}}$ for all available eligible models on the indicated task. **(B)** For the 12 available models on `human_tx_s26_satmut`, we separate differences across sequence backgrounds, measured by $\rho_{\Delta\text{Background}}$, from mutation-specific cell-line differences, measured by $\tau_{\Delta\text{Mutation}}$, and mutation effects within each cell line, measured by $\tau_{\text{HepG2}}$ and $\tau_{\text{K562}}$ (Appendix A.5). Boxed Spearman correlations compare model scores. Displaced markers have thin connectors to their measured positions, and statistics use the original values.

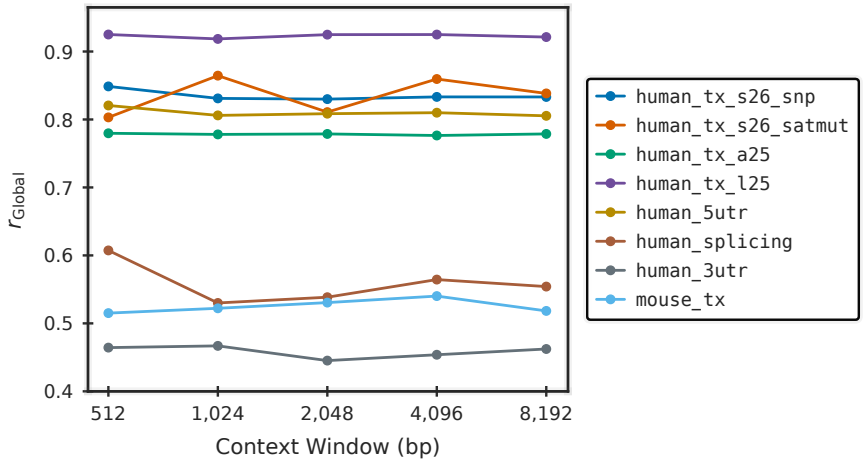


Figure S10: **AlphaGenome encoder transfer across five context-window sizes.** Each encoder is independently adapted using an input window centered on the assayed oligo. Colors distinguish the eight tasks on a shared test $r_{\text{Global}}$ scale. For odd-length oligos, the realized window is one base pair shorter than the nominal power-of-two value shown.