regLM: Designing realistic regulatory DNA with autoregressive language models

Avantika Lal, Tommaso Biancalani, Gokcen Eraslan
Biology Research | AI Development,
gRED Computational Sciences, Genentech
lal.avantika@gene.com, eraslan.gokcen@gene.com

Abstract

Designing cis-regulatory DNA elements (CREs) with desired properties is a challenging task with many therapeutic applications. Here, we used autoregressive language models trained on yeast and human putative CREs, in conjunction with supervised sequence-to-function models, to design regulatory DNA with desired patterns of activity. Our framework, regLM, compares favorably to existing CRE design approaches at generating realistic and diverse regulatory DNA, while also providing insights into the cis-regulatory code.

1 Introduction

Cis-regulatory elements (CREs), including promoters and enhancers, are DNA sequences that regulate gene expression. The function of a CRE is influenced by the order, composition and spacing of sequence motifs [1] that bind to transcription factor (TF) proteins. Synthetic CREs with defined properties are needed in therapeutic applications including cell and gene therapy; for example, to maximize activity of a gene in the target cell type. CREs are often designed manually using prior knowledge [2]. Recent studies used directed evolution [3, 4] and gradient-based approaches [5–7] for CRE design, in which supervised models are used to edit sequences iteratively to achieve a desired prediction. However, such approaches are not truly generative and do not necessarily learn the overall sequence distribution of the desired CREs. Instead, they may only optimize specific features with high predictive value. Consequently, the resulting CREs may be out-of-distribution and unrealistic, leading to unpredictable behavior when they are experimentally tested in a cell.

Here, we present regLM, a framework to design synthetic CREs with desired properties, such as high, low or cell type-specific activity, using autoregressive language models in conjunction with supervised models. To our knowledge, this is the first time language modeling has been used for DNA in a generative setting. Synthetic CREs generated by our approach are diverse and realistic, with biological features similar to experimentally validated CREs.

2 Results

2.1 regLM builds on the HyenaDNA framework for CRE generation

Several transformer-based foundation models for DNA have been developed [8–10]. However, these methods, where the authors focused on predictive downstream tasks, are based on masked language modeling which is difficult to use for sequence generation. In contrast, the recent HyenaDNA foundation models are autoregressive models [11] trained on the human genome, and are hence suitable for regulatory element design. regLM builds on the HyenaDNA [11] framework to perform generative modeling of CREs with specific properties. Not only does this take advantage of the high resolution and computational efficiency of HyenaDNA, but the ability to fine-tune pretrained

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

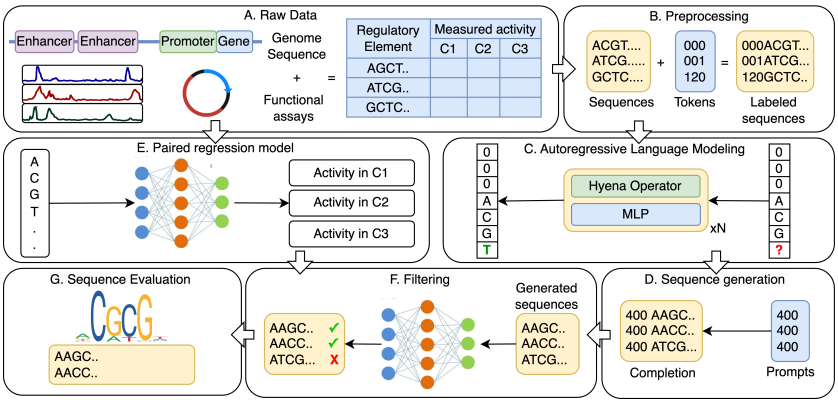

Figure 1: regLM. A, B) DNA sequences are prefixed with a sequence of prompt tokens representing functional labels. C) A HyenaDNA model is trained or fine-tuned to perform next token prediction on the labeled sequences. D) The trained model is prompted to generate sequences with desired properties. E, F) A regression model is used to filter the generated sequences. G) The regulatory content of generated sequences is evaluated.

HyenaDNA models which have already learned genomic features enables design tasks even with little labeled data.

Given a dataset of DNA sequences labeled with their activity (Fig. 1A), we encode the label in a sequence of categorical 'prompt tokens', which is prefixed to the DNA sequence (Fig. 1B). We train or fine-tune a HyenaDNA model to perform next token prediction beginning with the prompt tokens (Fig. 1C). This formulation allows us to use prior knowledge in the model explicitly. Once trained, the language model can be prompted with the sequence of prompt tokens representing a desired function. The model, now conditioned on the prompt tokens, generates a DNA sequence one nucleotide at a time (Fig. 1D). In parallel, we train a sequence-to-activity regression model on the same data (Fig. 1E), and apply it to the generated sequences to select those that best match the desired activity (Fig. 1F). This combined approach allows us to use the regression model as an oracle like previous model-guided approaches, while the language model ensures that the generated sequences have realistic content. Finally, we evaluate the generated sequences as well as the model (Fig. 1G).

2.2 regLM generates yeast promoters of varying strength

We trained regLM on a dataset of 80 base pair (bp) DNA sequences and their measured promoter activities in yeast grown in complex and defined media [12, 4]. Each sequence was prefixed with a two-token label, wherein each token ranged from 0 to 4 and represented promoter activity in one of the media (Fig. 2A). For example, label 00 indicated low activity in both media, while 04 indicated low activity in complex and high in defined media.

The regLM model trained on this dataset (Fig. S1) reached 30.3% mean accuracy on native yeast promoters and 33.2% on the test set, compared to 25% expected by chance. In both datasets, accuracy was higher within known TF binding motifs (Fig. S2A, B; One-sided Mann-Whitney U test native p-value=$1.7\text{x}10^{-37}$ for native promoters, p<10^{-250} for test set), indicating that the model learned to predict functional motifs. Accuracy reduced if we shuffled the labels across sequences (Fig. S2C, D; One-sided Mann-Whitney U test p-value=$6.5\text{x}10^{-47}$ for native promoters, p<10^{-250} for test set), indicating that the model used the information encoded in the prompt tokens. In the test set, accuracy was higher for strong promoters (label 44) (Fig. S2E; One-sided Mann-Whitney U test p-value<10^{-250}), perhaps due to their higher content of predictable TF motifs (Fig. S2F).

We generated promoters of varying strength by prompting the model with labels 00, 11, 22, 33, and 44. Generated sequences were distinct from the training set, having a minimum Hamming distance of 25 bp. A supervised regression model trained on the same data (Fig. S3) was used to discard generated sequences whose predicted activity did not match the prompt. Only 2% of generated sequences were discarded.

Independent regression models trained on separate data from the language model (Fig. S4) predicted that regLM generates stronger promoters when prompted with higher labels, and that the activity of

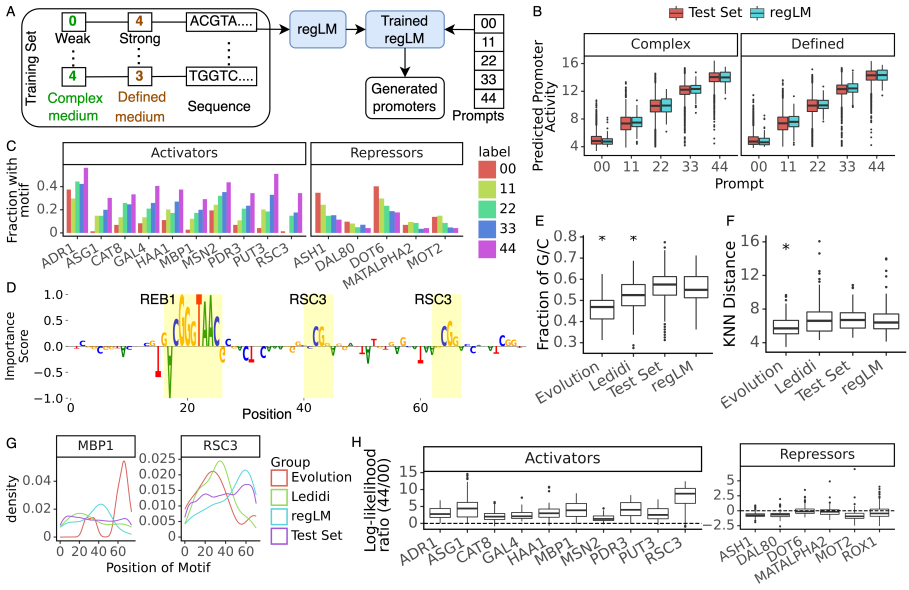

Figure 2: A) Schematic of the experiment. B) Predicted activity of regLM-generated promoters, as well as test set promoters with the same label. C) Fraction of regLM promoters generated from each prompt that contain selected activating and repressing TF motifs. D) Example of a regLM generated strong promoter. The y-axis represents the DeepSHAP per-nucleotide attribution score from an independent regression model, highlighting TF motifs. E) Fraction of G/C bases in strong promoters generated by different methods. F) Average distance of each strong promoter to its 10 nearest neighbors generated by the same method, based on motif content. 'Test Set' represents 100 strong promoters randomly sampled from the test set. G) Distribution of positions of selected activating TF motifs in strong promoters generated by different methods. H) Boxplots showing the ratio between the log-likelihood of the motif sequence given label 44 (high activity) vs. 00 (low activity), for selected TF motifs inserted in random sequences. Asterisks in E and F indicate significant difference from the test set.

regLM promoters matches that of held-out test promoters with the same label (Fig. 2B). Further, regLM produced more sequences containing motifs for activating yeast TFs when prompted with higher labels, while motifs for repressive TFs became less frequent (Fig. 2C, D).

To assess whether regLM-generated promoters are realistic, we compared 100 regLM-generated strong promoters (prompted with 44) with sequences of similar predicted activity (Fig. S5) generated by directed evolution (DE) and the gradient-based method Ledidi [5]. For a fair comparison, we performed DE and Ledidi using the regression model trained on the same dataset as regLM as an oracle. All three sets of generated promoters were compared to experimentally validated strong promoters from the test set. regLM promoters were most similar to test set promoters in GC content (Fig. 2E). We counted the frequency of all 4-mers, as well as yeast TF binding motifs, in all promoters. No 4-mers were differentially abundant (Mann-Whitney U-test adjusted p-value < 0.05) in regLM promoters with respect to test set promoters, compared to 47 4-mers for DE and 29 for Ledidi. When we matched each sequence to its nearest neighbor based on 4-mer frequency, 96% of regLM promoters were matched to a test set promoter, compared to 90% for directed evolution and 83% for Ledidi. Similarly, 93% of regLM-generated promoters had a test set promoter as their nearest neighbor based on motif frequency, compared to 71% for directed evolution and 83% for Ledidi.

We estimated the diversity of motif content in promoters generated by each method, by calculating the average distance between each promoter and its 10 nearest neighbors generated by the same method based on motif frequency. regLM and Ledidi promoters had similar diversity to random samples from the test set, whereas directed evolution produced less diverse sequences (Fig. 2F; Kruskal-Wallis p-value 3.8×10^{-19}, Dunn's post-hoc p-value 1.6×10^{-14} (Directed Evolution vs. regLM)). This indicates that regLM produces functionally diverse sequences.

We also computed the frequencies of pairwise combinations of TF motifs. Out of 2,342 motif pairs that were present in over 5% of any group of promoters, none were differentially abundant (two-sided Fisher's exact test adjusted p-value < 0.01) in regLM promoters with respect to the test set promoters,

compared to 1,964 in directed evolution promoters and 115 in Ledidi promoters. Finally, motifs for activating transcription factors tend to occur in regLM promoters at similar positions as in validated promoters (Fig. 2G, Fig. S6). Together, this evidence demonstrates comprehensively that regLM has learned many aspects of the yeast regulatory code.

2.3 regLM learns species-specific regulatory grammar

To independently assess whether regLM learned regulatory rules of yeast cells, we selected the top motifs for activating and repressing yeast TFs and inserted each motif into 100 random DNA sequences. We used our regLM model to compute the likelihoods of the resulting sequences (P(sequence | label)) given either label 44 (high promoter activity) or 00 (low promoter activity). For each synthetic promoter, we defined a likelihood ratio as follows:

$$OR = P(sequence|label = 44)/P(sequence|label = 00)$$

A positive ratio indicates that the model has learned the motif is more likely to occur in sequences with label 44 than 00, whereas a negative ratio indicates the opposite. We found that sequences containing activating motifs have positive likelihood ratios whereas sequences containing repressive motifs tend to have negative ratios (Fig. 2H). This indicates that regLM has learned relationships between specific motifs and promoter activity.

2.4 regLM generates cell type-specific human enhancers

We applied regLM to a dataset of 200bp human enhancers and their activity in three cell lines (K562, HepG2 and SK-N-SH) [7]. Each sequence was prefixed with 3 prompt tokens, each token ranging from 0 to 4 and representing the measured activity of the enhancer in one of the 3 lines. For example, label 041 indicates that the sequence has low activity in HepG2 cells, high activity in K562 cells, and weak activity in SK-N-SH cells (Fig. 3A). Here, we fine-tuned a pre-existing HyenaDNA model that had already learned from the human genome [11] on the labeled enhancers (Fig. S7). The trained model had a mean per-nucleotide accuracy of 48% on the test set. Despite the rarity of cell type specific enhancers in the training set (Fig. S8; enhancers with label 400, 040 or 004 comprised 0.28% of the training set), accuracy remained high (39.3%) on cell type specific enhancers.

We used the trained regLM model to generate enhancers with activity specific to each cell line by prompting it with labels 400 (HepG2-specific), 040 (K562-specific) and 004 (SK-N-SH specific). Generated sequences had a minimum Hamming distance of 28 nucleotides with reference to the training set. Using the regression model trained on the same dataset (Fig. S9), we selected the 100 regLM generated enhancers predicted to be most specific to each cell line.

An independent regression model (Fig. S10) predicted these to have cell type-specific enhancer activity (Fig. 3B). Compared to matched cell type-specific enhancers produced by directed evolution and Ledidi (Fig. S11), regLM enhancers were more likely to have a real enhancer as their nearest neighbor, both in terms of k-mer and motif content (Fig. 3C). Motifs for known cell type-specific TFs occur at higher frequency in regLM enhancers of the appropriate specificity (Fig. 3D). For example, the motif for the erythropoietic TF GATA2 occurs at higher frequency in K562-specific synthetic enhancers (Fig. 3D, E), whereas motifs for the liver-specific HNF4A and 4G factors occur at higher frequency in HepG2-specific synthetic enhancers.

3 Conclusion

Here, we demonstrate that regLM successfully learns the regulatory code of DNA in different species and cell types, and generates diverse, realistic CREs with desired activity. Evaluation of synthetic sequences using prior knowledge shows high concordance between the regulatory rules implemented in the sequences and known regulatory syntax. It should be noted that although regLM generates sequences with properties rarely seen during training, it remains to be seen whether it can extrapolate beyond the training set by generating unseen combinations of motifs. However, it can already generate numerous and diverse sequences which can be filtered by additional models for criteria such as safety and stability. In the future, generated sequences can be experimentally validated to assess their function and safety.

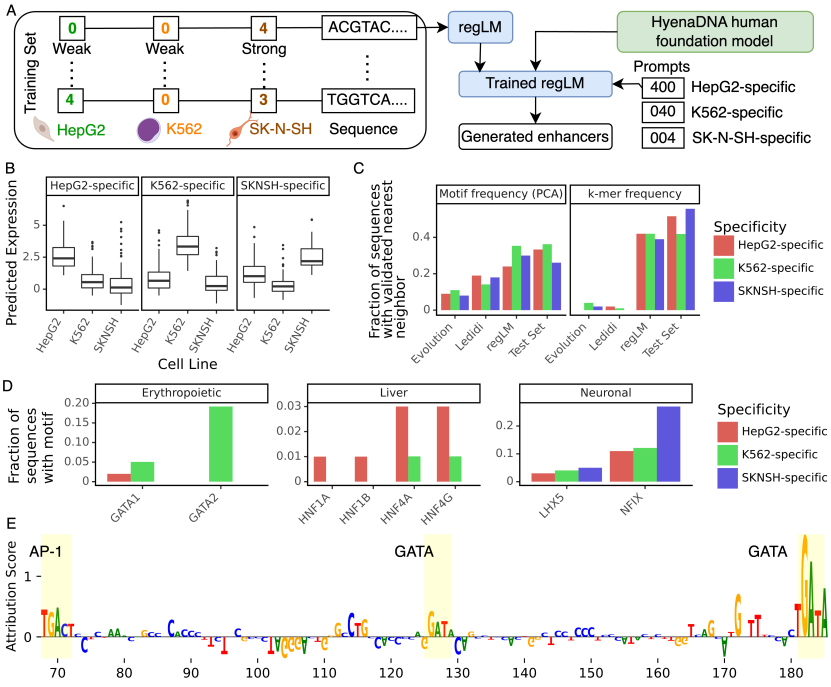

Figure 3: A) Schematic of the experiment. B) Predicted activity of cell type specific human enhancers generated by regLM, in 3 cell lines C) Fraction of generated enhancers whose nearest neighbor based on k-mer content or TF motif content is a real human enhancer from the test set, for different methods. D) Fraction of regLM-generated enhancers containing selected cell type-specific TF motifs. E) Sequence of the top K562-specific regLM-generated enhancer. The y-axis represents the per-nucleotide attribution scores generated by DeepSHAP based on the independent regression model. TF binding motifs are highlighted.

4 Methods

Yeast promoter sequences from [4] were prefixed with two tokens, representing their measured activity in complex and defined media, respectively. Each token ranged from 0-4 with 0 indicating the lowest quintile of activity and 4 the highest. Quintiles were calculated on the training set. The trained regLM model was prompted to generate 150 sequences each with prompts 00, 11, 22, 33, and 44. A regression model trained on the same data was used to predict their expression in both media. We removed regLM generated sequences whose predicted activity in either medium was >2 standard deviations from the mean activity of training set promoters with the same token, and randomly selected 100 generated sequences in each class (00, 11, 22, 33, 44) for evaluation.

Human enhancer sequences from [7] were prefixed with three tokens; a token 0-4 representing its quintile of measured activity in HepG2 cells, a second token 0-4 for K562 cells, and a third token 0-4 for SK-N-SH cells. 0 represents the lowest quintile of activity and 4 represents the highest. We fine-tuned the pretrained HyenaDNA foundation model 'hyenadna-medium-160k-seqlen' [11]. The fine-tuned model was prompted to generate 5000 sequences each with prompts 400 (HepG2-specific), 040 (K562-specific), and 004 (SK-N-SH specific). A regression model trained on the same data was used to select the 100 enhancers predicted to be most specific to each cell type for evaluation. We refer readers to the Appendix for more details on data processing, model training and evaluation.

5 Code Availability

regLM is available at `https://github.com/Genentech/regLM`. Model weights and code to perform the experiments in this paper are linked to from the GitHub repository. Experiments were performed using Python v3.8, PyTorch v1.13.0 and PyTorch Lightning v1.8.2.

References

[1] Patricia J Wittkopp and Gizem Kalay. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nature Reviews Genetics*, 13(1):59–69, 2012.

[2] Oriol Fornes, Tamar V Av-Shalom, Andrea J Korecki, Rachelle A Farkas, David J Arenillas, Anthony Mathelier, Elizabeth M Simpson, and Wyeth W Wasserman. Ontarget: in silico design of minipromoters for targeted delivery of expression. *Nucleic Acids Research*, page gkad375, 2023.

[3] Ibrahim Ihsan Taskiran, Katina I Spanier, Valerie Christiaens, David Mauduit, and Stein Aerts. Cell type directed design of synthetic enhancers. July 2022.

[4] Eeshit Dhaval Vaishnav, Carl G de Boer, Jennifer Molinet, Moran Yassour, Lin Fan, Xian Adiconis, Dawn A Thompson, Joshua Z Levin, Francisco A Cubillos, and Aviv Regev. The evolution, evolvability and engineering of gene regulatory DNA. *Nature*, 603(7901):455–463, March 2022.

[5] Jacob Schreiber and Yang Young Lu. Ledidi: Designing genomic edits that induce functional activity. May 2020.

[6] Johannes Linder and Georg Seelig. Fast activation maximization for molecular sequence design. *BMC Bioinformatics*, 22(1):510, October 2021.

[7] S J Gosai, R I Castro, N Fuentes, J C Butts, S Kales, R R Noche, K Mouri, P C Sabeti, S K Reilly, and R Tewhey. Machine-guided design of synthetic cell type-specific cis -regulatory elements. *bioRxiv*, August 2023.

[8] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, August 2021.

[9] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. March 2023.

[10] Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. DNA language models are powerful zero-shot predictors of genome-wide variant effects. April 2023.

[11] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A Baccus, and Chris Ré. HyenaDNA: Long-Range genomic sequence modeling at single nucleotide resolution. *ArXiv*, June 2023.

[12] Carl G de Boer, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nat. Biotechnol.*, 38(1):56–65, January 2020.

A Supplementary Methods

A.1 Training regLM models

HyenaDNA[11] is a decoder-only, sequence-to-sequence architecture consisting of a stack of blocks. Each block comprises a Hyena operator [14], followed by normalization and a feed-forward neural network. For human enhancers, we fine-tuned the pre-trained foundation model 'hyenadna-medium-160k-seqlen'. This model has 6.55 million parameters and is trained to perform next token prediction on the human genome. For yeast promoters, we trained from scratch a HyenaDNA model with the same architecture as 'hyenadna-medium-160k-seqlen'. HyenaDNA models were trained or fine-tuned for 100 epochs on 1 NVIDIA A100 GPU using the AdamW optimizer with a batch size of 128 and cross-entropy loss. The learning rate was $3x10^{-4}$ for yeast promoters and $6x10^{-4}$ for human enhancers. Validation loss was computed every 2000 steps and the model with lowest validation loss was saved.

A.2 Data Sources

Yeast promoter data from [4] was downloaded from `https://zenodo.org/record/4436477`. The data consisted of randomly generated DNA sequences of approximately 80 bp length and their measured promoter activity in yeast cells, based on reporter gene expression in a Gigantically Parallel Reporter Assay (GPRA). Measurements for 31,349,363 sequences were available in complex medium, and measurements for 21,037,407 sequences were available in defined medium.

Human Massively Parallel Reporter Assay (MPRA) data was downloaded from the supplementary material of [7]. This dataset contains 798,064 enhancer sequences from the human genome, of approximately 200 bp length, along with their measured enhancer activity in three human cell lines: HepG2 (a liver carcinoma cell line), K562 (an erythroleukemia cell line), and SK-N-SH (a neuroblastoma cell line).

A.3 Data Processing

A.3.1 Yeast promoters

We removed the constant sequences flanking each promoter sequence and selected promoter sequences that were 80 bp long and contained no N characters. This left measurements for 23,414,517 sequences in the complex medium and measurements for 16,799,784 sequences in the defined medium.

We then split the dataset into 7,533,156 sequences whose activity was measured in both media, and sequences whose activity was measured in only one medium (15,881,361 sequences measured only in complex medium and 9,266,628 sequences measured only in defined medium).

Taking the 7,533,156 sequences with measured activity in both media, we randomly split them into training (7,483,156 sequences), validation (50,000 sequences), and test (50,000 sequences) sets. We calculated the quintiles (five equally sized bins) of measured activity levels in complex medium and defined medium separately based on the training set. We assigned each sequence in the training set a token 0-4 based on its quintile of activity in the complex medium, and a second token 0-4 based on its quintile of activity in the defined medium. 0 indicates that the sequence belongs to the lowest quintile and 4 indicates that the sequence belongs to the highest quintile, which we describe as a strong promoter. Thus, each sequence was assigned a label consisting of two categorical tokens. For example, label 00 means a sequence that is in the 1st (lowest) quintile of activity in both media. 40 means a sequence that is in the 5th (highest) quintile of activity in complex medium but the 1st in defined medium.

Sequences in the validation and test sets, as well as native yeast promoters, were also assigned labels based on the quintiles calculated on the training set. Each DNA sequence was prefixed with its assigned label.

A.3.2 Human enhancers

We first randomly split the dataset, holding out 120,000 enhancer sequences to train an independent regression model and using the remaining 678,062 enhancer sequences to train the regLM model and its paired regression model.

We split the 678,062 enhancers retained for regLM according to their chromosome, using 4,761 enhancers from chromosome 7 for validation, 2,456 enhancers from chromosome 13 for testing, and the remaining 670,845 enhancers from other chromosomes for training. We calculated the quintiles (five equally sized bins) of measured activity levels in each cell line separately based on the training set. We assigned each sequence in the training set a token 0-4 based on its quintile of activity in HepG2 cells, a second token 0-4 based on its quintile of activity in K562 cells, and a third token 0-4 based on its quintile of activity in SK-N-SH cells. 0 indicates that the sequence belongs to the lowest quintile and 4 indicates that the sequence belongs to the highest quintile.

Thus, each sequence was assigned a label consisting of three categorical tokens. For example, label 401 means an enhancer that is in the 5th (highest) quintile of activity in HepG2 cells, the 1st (lowest) quintile of activity in K562 cells, and the 2nd quintile of activity in SK-N-SH cells. Sequences in the validation and test sets were also assigned labels based on the quintiles calculated on the training set. Each DNA sequence was prefixed with its assigned label.

A.4 Regression models

Regression models were trained to take as input a one-hot encoded CRE sequence and predict as output its activity. All regression models were based on the Enformer architecture [15] and were built using the enformer-pytorch package (https://github.com/lucidrains/enformer-pytorch). All regression models were trained for 10 epochs on 1 NVIDIA A100 GPU with the Adam optimizer and data augmentation by reverse complementing training sequences. Performance was measured as the Pearson correlation between measured and predicted CRE activity on the test set. Validation loss was measured after every epoch and the model with lowest validation loss was saved.

For both yeast and human datasets, we trained two regression models. One model was trained on the same sequences as the regLM model (without categorical labels), to use in conjunction with regLM to filter and prioritize the generated sequences, as well as to serve as an oracle model for directed evolution and Ledidi. A second model was trained on independent data that was held out from regLM, kept entirely separate from all generative methods and was used only for independent *in silico* validation of synthetic CREs.

A.4.1 Yeast promoters

Both regression models were Enformer-based models with 3 convolutional blocks followed by 1 transformer encoder layer. The first convolutional block has 384 channels. Models were trained with the Poisson loss function, learning rate 5 x 10-4 and batch size 1024.

The regLM-matched model was trained using the same training, validation and test data used to train regLM. The model has two output heads that predict activity in complex and defined media respectively.

For the independent regression model, the 21,609,084 sequences whose activity was measured only in complex medium and 11,460,087 sequences whose activity was measured only in defined medium were used to train two separate regression models. In each medium, 50,000 randomly chosen sequences were held out for validation and 50,000 were held out for testing. The remaining sequences were used for training. In this case, each model had a single output head.

A.4.2 Human enhancers

For human data, we downloaded the pretrained Enformer model [15], reduced its size by dropping the last 8 transformer encoder layers (leaving 7 convolutional blocks and 3 transformer encoder layers), and added 3 output heads that predict the level of expression for the input sequence in each cell type.

For the regLM-matched regression model, we fine-tuned the model on the same sequences as regLM. For the independent model, we took the 120,000 enhancer sequences that we had held out from regLM, split 896 enhancers from chromosome 7 for validation, 461 enhancers from chromosome 13 for testing, and used the remaining 118,643 enhancers from other chromosomes for training. These models were fine-tuned with MSE loss, learning rate 1x10-4 and batch size 512.

A.5 Generating synthetic CREs using regLM

A.5.1 Yeast promoters

The regLM model trained on yeast promoters was prompted to generate 150 sequences each with labels 00, 11, 22, 33, and 44. Generated promoters were filtered using the regression model trained on the same data. For each medium, we first used the regression model to predict the activity of all sequences in its training set, and computed the mean and standard deviation of predicted activity for training sequences with each class token (0, 1, 2, 3, and 4). We then used the same model to predict the activity of all generated promoters in both media. We discarded generated promoters whose predicted activity in either medium was >2 standard deviations from the mean predicted activity of promoters with the same token in the training set. We performed this procedure separately for complex and defined media. We then randomly selected 100 synthetic promoters of each generated class (00, 11, 22, 33, and 44) for evaluation.

A.5.2 Human Enhancers

The regLM model trained on human enhancers was prompted to generate 5000 sequences each with tokens 400 (HepG2-specific), 040 (K562-specific), and 004 (SK-N-SH specific). Generated enhancers were filtered using a regression model trained on the same data. We estimated the cell type specificity of each sequence as the difference between its predicted activity in the target cell type and its average predicted activity in non-target cell types. Based on this, we selected the 100 most specific regLM-generated enhancers for each cell type.

A.6 Generating synthetic CREs with Directed Evolution and Ledidi

Directed Evolution and Ledidi are methods that iteratively make edits to a starting sequence to maximize a defined objective function using a trained predictive model (the 'oracle').

In Directed Evolution, we generate all possible single-base pair mutations of the starting sequence, predict their activity using the oracle and select the one with highest value of the objective function as the starting sequence for the next iteration. Ledidi is a gradient-based approach that aims to increase the objective function while minimizing the total number of edits made to the starting sequence.

We randomly chose CREs that had been measured to have low activity in all conditions (label 00 for yeast promoters and 000 for human enhancers) as the starting sequences. To ensure a fair comparison to regLM-generated sequences, the regression models trained on the same data as regLM were used as oracles. Directed Evolution and Ledidi were each run multiple times with a different starting sequence each time, to generate diverse synthetic CREs. Directed Evolution was run for 10 iterations. Ledidi was run with the following parameters: max_iter=1000, l=20, lr=3x10-3.

For yeast promoters, we aimed to generate promoters with high activity in both media. The objective function for both Directed Evolution and Ledidi was the mean predicted activity in the two media. Directed Evolution and Ledidi were both run 500 times resulting in 500 synthetic strong promoters from each method.

To compare across methods, each regLM-generated strong promoter was matched to the directed evolution-generated and Ledidi-generated sequences that were closest to it in predicted activity (measured by the mean squared error across both media), resulting in a matched set of 100 putative strong promoters designed by each method. Thus, since the three groups have highly similar predicted activity, we can compare their sequence content to assess which approach gives rise to more biologically realistic sequences while reaching the same objective.

For human enhancers, we aimed to generate enhancers specific to each of the 3 cell lines. The objective function was cell type specificity (defined above). A total of 1500 synthetic enhancers (500 enhancers specific to each cell type) were generated by each method (directed evolution and Ledidi). Each of the 300 regLM-generated enahncers (100 for each cell line) was matched to the directed evolution-generated and Ledidi-generated sequences that had the most similar predicted activity (measured by the mean squared error across all cell lines), resulting in a set of 300 sequences designed by each method.

A.7 *In silico* evaluation of synthetic sequences

A.7.1 GC Content

GC content of each real and synthetic promoter was calculated as the number of G or C bases in the sequence, divided by the total number of bases in the sequence.

A.7.2 K-mer content

All possible DNA sequences of length 4 (4-mers) were enumerated and the frequency of each 4-mer was counted in each real or synthetic promoter. Each sequence was thus represented by a $4\hat{4} = 256$ dimensional vector. Each sequence was matched to its nearest neighbors in this space using the sklearn.neighbors.NearestNeighbors function in scikit-learn. To calculate the fraction of real nearest neighbors, we matched each sequence to its nearest neighbor out of all real and synthetic sequences. For each group of synthetic CREs, we calculated the proportion of sequences whose nearest neighbor was an experimentally validated CRE from the test set. For each 4-mer, we used the two-sided Mann-Whitney U-test to estimate the difference in means between its frequency in experimentally validated test sequences and in synthetic sequences generated by each method. p-values were adjusted with the Bonferroni correction. 4-mers with adjusted p-value < 0.05 were considered differentially abundant.

A.7.3 Transcription factor motif content

Position Weight Matrices (PWMs) for yeast TFs were downloaded from the YeTFaSCo database [16]. PWMs for vertebrate transcription factors were downloaded from the JASPAR 2022 database [17]. To identify

motifs in a given DNA sequence, we one-hot encoded the sequence and scanned it with PWMs using the scipy.signal.convolve2d function. Regions with match scores > 0.8 of the maximum possible match score for the PWM were labeled as motifs. We calculated the frequency of each motif in each real and synthetic promoter. Differentially expressed motifs were calculated using the same procedure as for 4-mers. To compute nearest neighbors and distances more efficiently, PCA was performed on the motif frequency matrix and the top 50 PCs were selected. Thus each sequence was represented by a 50-dimensional vector. Each sequence was matched to its nearest neighbors in this vector space and the proportion of real nearest neighbors for each group of synthetic CREs was calculated using the same procedure as for 4-mers.

To calculate a metric of diversity in motif content for each group of synthetic CREs, we matched each sequence to its 10 nearest neighbors generated by the same method and calculated the mean of its distances to the 10 neighbors in PCA space.

A.7.4 Selecting biologically relevant TF motifs for interpretation and visualization

For yeast promoters, we ranked motifs by their enrichment in test set sequences with label 44 (strong promoters) compared to 00 (weak promoters). We selected 10 motifs for known activating yeast TFs that are highly enriched in strong promoters, and 6 known repressive motifs that are enriched in weak promoters. For human enhancers, we selected motifs for transcription factors known to be specific to the cell lines of interest based on prior knowledge, and which also are observed in enhancers active in that cell line in our test set. Specifically, these were HNF1A, HNF1B, HNF4A and HNF4B for HepG2 cells, GATA1 and GATA2 for K562 cells, and NFIX and LHX5 for SK-N-SH cells.

A.8 Interpretation of the regLM model trained on yeast promoters

regLM is trained to perform next token prediction, i.e. for each position in a DNA sequence, regLM predicts the probability of all possible bases (A, C, G and T) conditioned on the previous bases as well as the initial label. Thus, we can obtain the likelihood of an observed sequence conditioned on its initial label (P(sequence | label)) as the product of probabilities of the base observed at each position.

To assess whether regLM has learned the function of a given motif, we generated 100 random DNA sequences and inserted the consensus sequence for the motif at the center of each. We prefixed each sequence with label 00 (low activity) and used the trained regLM model to predict the probability of each base in the motif. We calculated the likelihood of the motif conditioned on the sequence being labeled with 00 ($P(sequence|label = 00)$). We then prefixed all 100 sequences with the label 44 (high activity in both media) and repeated the procedure, calculating the likelihood of the motif conditioned on the sequence being labeled with 44 ($P(sequence|label = 44)$).

B Supplementary Figures

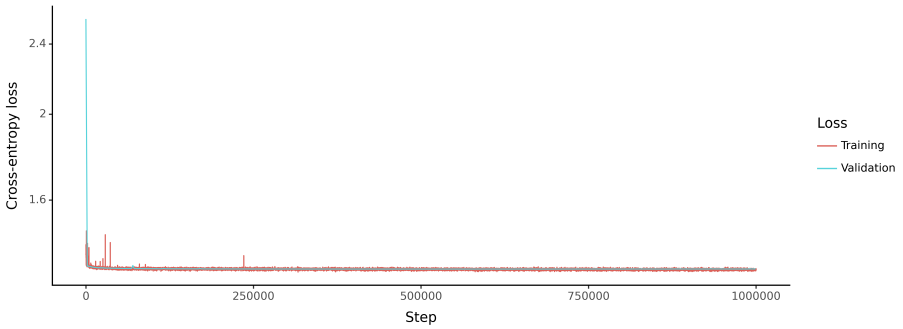

Figure S1: Training curve for yeast regLM model.

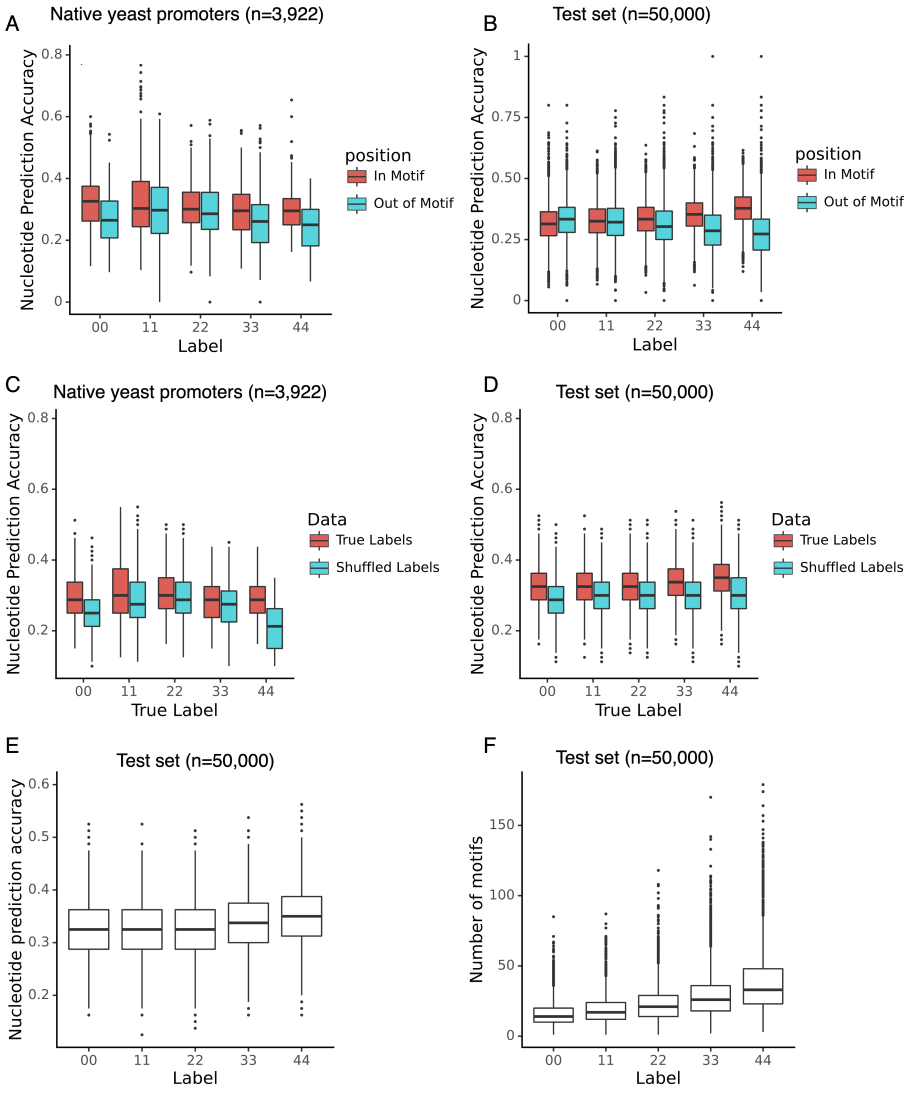

Figure S2: A) Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 3,922 native yeast promoters, for positions within and outside TF motifs B) Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 50,000 sequences in the held-out test set, for positions within and outside TF motifs. Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 3,922 native yeast promoters, before and after shuffling the initial sequence labels. C) Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 50,000 sequences in the held-out test set, separated by their class labels. D) Boxplots showing the number of transcription factor (TF) binding motifs found in 50,000 yeast promoter sequences of the held out test set, separated by the promoter class labels. E) Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 50,000 sequences in the held-out test set, for positions within and outside TF motifs F) Boxplots showing the average per-nucleotide prediction accuracy of the trained yeast regLM model on 50,000 sequences in the held-out test set, before and after shuffling the initial sequence labels. Only sequences with labels 00, 11, 22, 33 and 44 are shown in these plots.

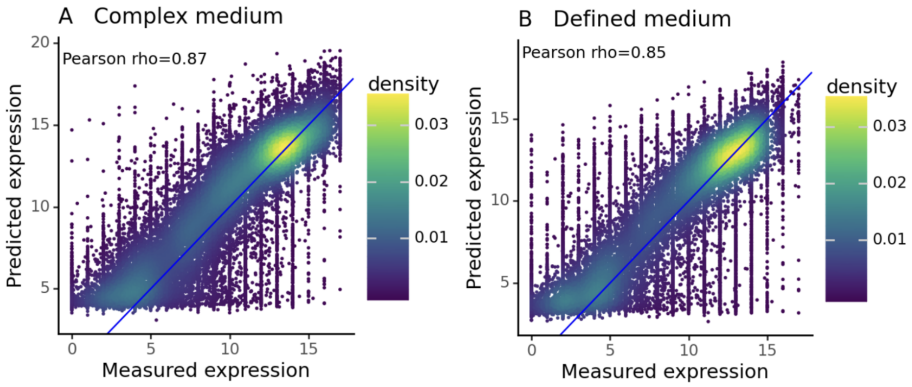

Figure S3: Performance of a supervised regression model trained to predict promoter activity of yeast promoter sequences. The model was trained and tested on the same data as the regLM model. Scatterplots show the measured and predicted activity of 50,000 test set promoters in A) complex medium and B) defined medium.

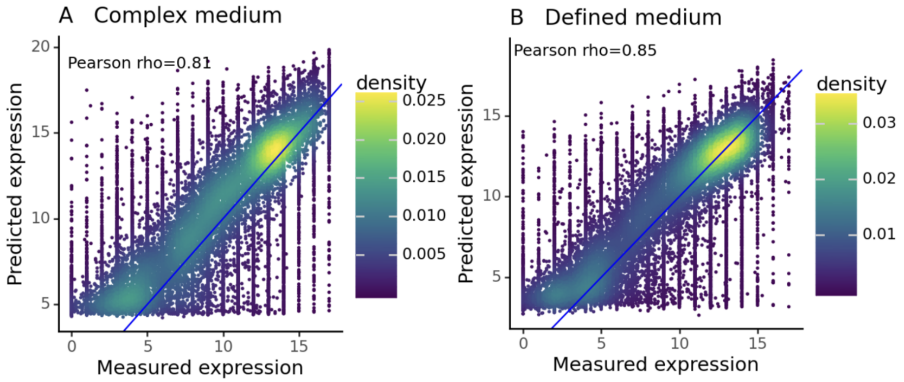

Figure S4: Performance of two supervised regression models trained to predict promoter activity of yeast promoter sequences in complex and defined medium respectively. These models were trained and tested on separate data from the regLM model. Scatterplots show the measured and predicted activity of 50,000 test set promoters each in A) complex medium and B) defined medium.

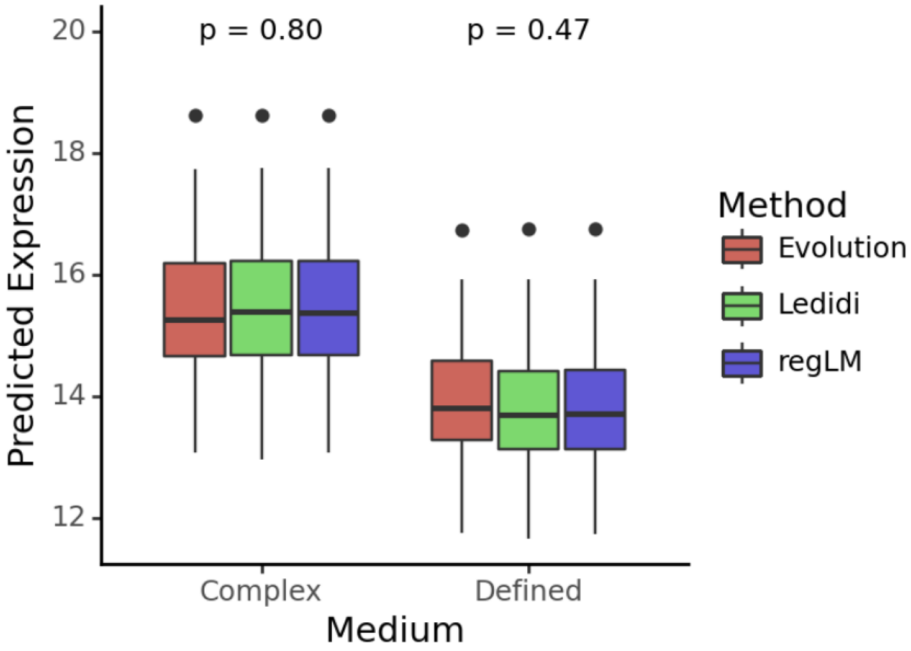

Figure S5: Predicted activity of synthetic strong yeast promoters generated by directed evolution (red), Ledidi (green) and regLM (blue), in complex and defined media. 100 synthetic promoters were generated by each method. The p-value was obtained using the Kruskal-Wallis test.

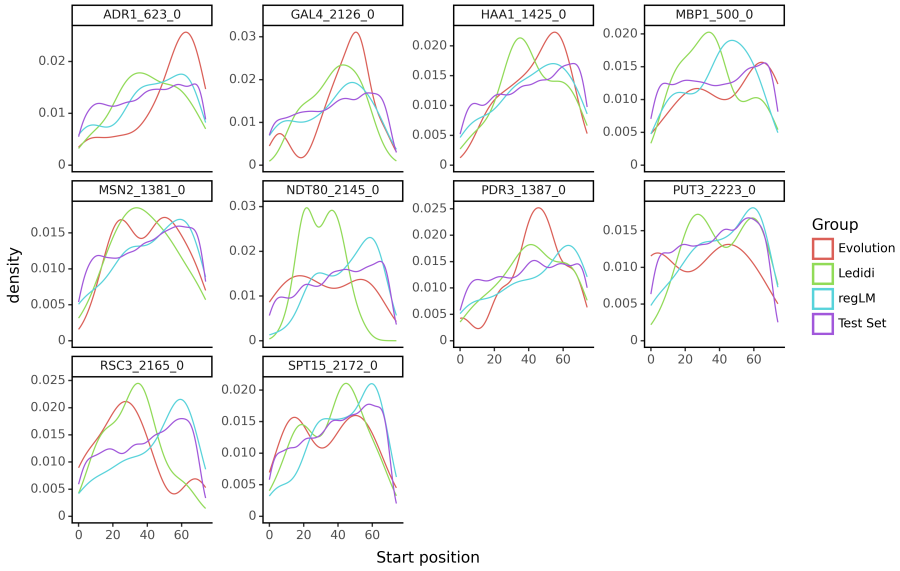

Figure S6: Distribution of positions of 10 activating TF motifs in synthetic strong yeast promoters generated by different methods.

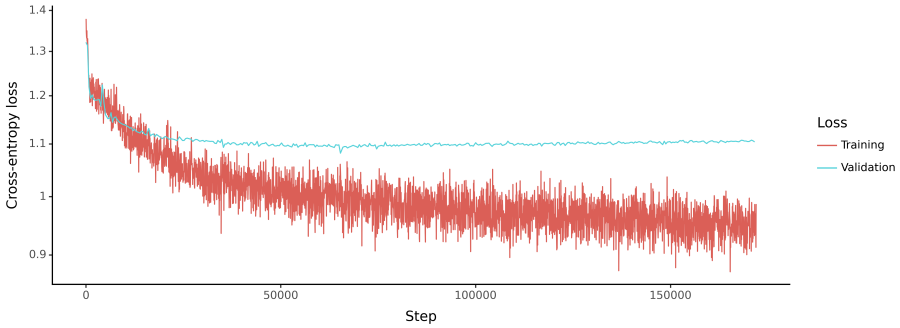

Figure S7: Training curve for regLM model trained on human enhancer sequences.

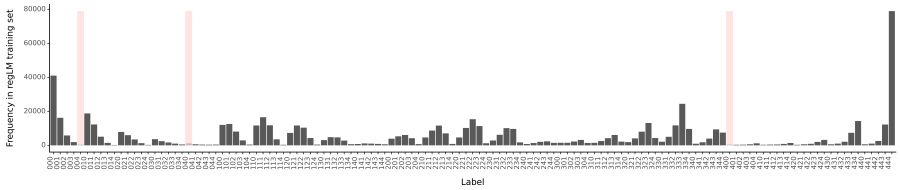

Figure S8: Frequency of labels in the training set for the regLM model trained on human enhancers. Labels 400, 040 and 004 are highlighted.

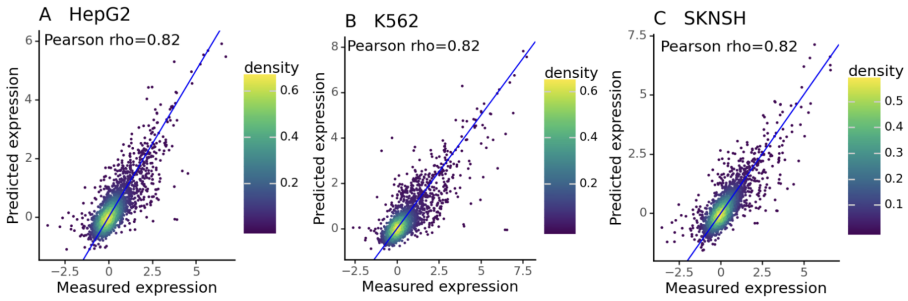

Figure S9: Performance of a supervised regression model trained to predict activity of human enhancer sequences in 3 cell lines. The model was trained and tested on the same data as the regLM model. Scatterplots show the measured and predicted activity of 2,456 test set enhancers.

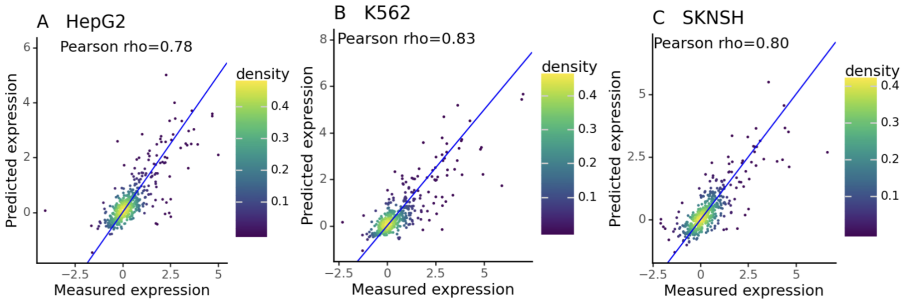

Figure S10: Performance of a supervised regression model trained to predict activity of human enhancer sequences in 3 cell lines. The model was trained and tested on separate data from the regLM model. Scatterplots show the measured and predicted activity of 461 test set enhancers.

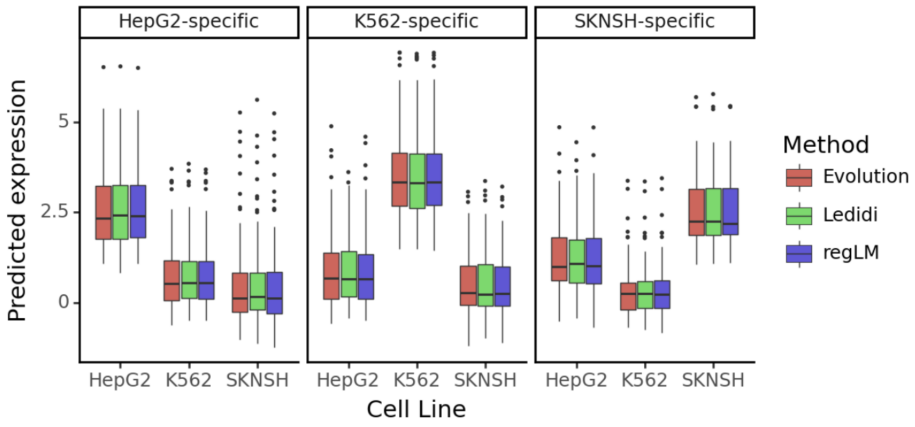

Figure S11: Predicted activity of synthetic cell type-specific enhancers generated by directed evolution, Ledidi and regLM, in 3 cell lines. 100 synthetic enhancers were generated by each method for each cell line.

C Supplementary References

[14] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. arXiv preprint arXiv, February 2023.

[15] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. Nat. Methods, 18(10):1196–1203, October 2021.

[16] Carl G De Boer and Timothy R Hughes. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. Nucleic acids research 40(D1):D169-D179, 2012.

[17] Jaime A Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas et al. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. Nucleic acids research 50(D1):D165-D173, 2022.