Pretraining strategies for effective promoter-driven gene expression prediction

Aniketh Janardhan Reddy[1*], **Michael H. Herschl**[1*], **Sathvik Kolli**[1], **Amy X. Lu**[1], **Xinyang Geng**[1], **Aviral Kumar**[1], **Patrick D. Hsu**[1], **Sergey Levine**[1] **and Nilah M. Ioannidis**[1]

[*]Equal contributions, [1]University of California, Berkeley

Advances in gene delivery technologies are enabling rapid progress in molecular medicine, but require precise expression of genetic cargo in desired cell types, which is predominantly achieved via a regulatory DNA sequence called a promoter; however, only a handful of cell type-specific promoters are known. Efficiently designing compact promoter sequences with a high density of regulatory information by leveraging machine learning models would therefore be broadly impactful for fundamental research and direct therapeutic applications. However, models of expression from such compact promoter sequences are lacking, despite the recent success of deep learning in modelling expression from endogenous regulatory sequences. Despite the lack of large datasets measuring promoter-driven expression in many cell types, data from a few well-studied cell types or from endogenous gene expression may provide relevant information for transfer learning, which has not yet been explored in this setting. Here, we evaluate a variety of pretraining tasks and transfer strategies for modelling cell type-specific expression from compact promoters and demonstrate the effectiveness of pretraining on existing promoter-driven expression datasets from other cell types. Our approach is broadly applicable for modelling promoter-driven expression in any data-limited cell type of interest, and will enable the use of model-based optimization techniques for promoter design for gene delivery applications. Our code and data are available at https://github.com/anikethjr/promoter_models.

1. Introduction

Gene therapy aims to deliver therapeutic genetic cargo to disease-associated cells and tissues. The expression of therapeutic genes, or transgenes, is controlled by an upstream compact regulatory DNA sequence called a promoter. To effectively treat disease while mitigating off-target side effects, promoters for gene therapy should be optimized for expression only in particular target cell types (differential expression), which requires compact promoter sequences with a high density of regulatory information. Recent advances in single cell-sequencing have illuminated over 400 cell types in the human body (Tabula Sapiens Consortium, 2022), yet only a handful of cell type-specific promoters are known. Existing methods to engineer promoters with cell type specificity rely on manual curation of sequence elements that are known to regulate expression, such as tiling of cis-regulatory elements (CREs) or tandem repeats of transcription factor (TF) binding motifs (Miao et al., 2000; Nissim et al., 2017; Wu et al., 2019). While these approaches have been successful in some cell types, extending promoter design to less studied cell types is a laborious process that can be accelerated by the use of sequence-based models of expression. In particular, deep learning (DL) models that predict cell type-specific promoter-driven expression from compact promoter sequences can be used to prioritize experimental validation of promoters with promising expression profiles, or to design optimal promoters in combination with model-based optimization techniques (Brookes et al., 2019; Linder et al., 2020; Trabucco et al., 2021).

Despite the success of DL in modelling expression from naturally occurring endogenous sequences in the human genome (Agarwal and Shendure, 2020; Avsec et al., 2021), models of expression from compact artificial promoter sequences are lacking. Existing endogenous gene expression models cannot be used directly for this purpose, since endogenous gene expression depends on distal regulatory elements and is not solely driven by the promoter sequence, making such model predictions unrepresentative of promoter-driven expression in a gene therapy setting. Massively parallel reporter assays (MPRAs) measure promoter-driven expression and can be used for model training (e.g. Movva et al. (2019)); however, MPRA expression data are avail-

Corresponding author(s): Aniketh Janardhan Reddy (aniketh@berkeley.edu) and Michael H. Herschl (michael_herschl@berkeley.edu)

able for only a small number of cell types. Thus, we need new modelling strategies for compact promoters that leverage existing data together with a small dataset of promoter-driven expression in a cell type of interest.

In this work, we develop an effective approach to train such models using larger related datasets for pretraining, prior to fine-tuning to predict cell type-specific promoter-driven expression using a small dataset in the cell type(s) of interest. Transfer learning using pretrained models has emerged as one of the most effective ways to model small datasets. Self-supervised tasks such as masked language modelling have been used to pretrain genomic sequence embeddings in recent work (e.g. Ji et al. (2021); Mo et al. (2021); Benegas et al. (2022); Zeng et al. (2023)). Pretraining using task-relevant data can improve the performance of fine-tuned models (Gururangan et al., 2020), while pretraining using irrelevant data can hurt performance (Liu et al., 2022). For our application, there are many datasets that are closely related to promoter-driven expression, including MPRAs and endogenous gene expression datasets, as well as TF-binding data that may help models learn relevant sequence motifs that regulate expression when present in promoters. Here we evaluate the utility of pretraining on such datasets for the task of predicting cell type-specific promoter-driven expression, as measured by a new dataset that we collected in three immune cell lines.

The main contribution of this work is the identification of a novel pretraining-based approach to effectively train a model that predicts cell type-specific promoter-driven gene expression in target cell types. We systematically benchmark several pretraining datasets, as well as various pretraining, transfer, and joint learning methods for this task to identify the best approach. We find that pretraining on existing promoter-driven expression data from MPRAs in other cell types, followed by fine-tuning on a new promoter-driven gene expression dataset in the target cell types leads to the best predictions. Pretraining improves prediction performance by $\mathbf{6-12\%}$ in all three experimentally-validated cell types. Our approach is broadly applicable to any cell type-specific promoter-driven expression dataset and can help design promoters for gene therapy that are optimized for expression in the therapeutic target cell type, thereby reducing potential off-target side effects.

2. Motivation for promoter design

In this section, we briefly motivate the need for novel methods for compact promoter design. While many gene delivery approaches have been developed, including both viral and non-viral delivery, nearly all approaches have limitations on the length of DNA that can be delivered, making it difficult to include extensive regulatory sequences with the genetic cargo. The delivery methods themselves also typically lack complete specificity to the target cell or tissue type (Sayed et al., 2022), which can lead to dangerous side-effects if the genetic cargo is expressed in the wrong cell type. One approach to both decrease the size and increase the specificity of the therapeutic DNA is to engineer novel cell type-specific compact promoters.

Promoters are DNA sequences that drive gene expression by recruiting RNA polymerase to initiate transcription of an adjacent gene. They are included before a transgene as part of the therapeutic DNA. Traditionally, promoters for gene therapy are derived from viruses (Montaño-Samaniego et al., 2020). Although these promoters are capable of driving high expression, they lack cell type-specificity and can lead to deleterious immune responses and cytotoxicity (Shirley et al., 2020). They are also often silenced through epigenetic mechanisms once delivered to a cell (Brooks et al., 2004). While endogenous cell type-specific promoters are common in the human genome, their specificity is often conferred by regulatory elements, such as enhancers, located outside of the promoter and farther from the gene, which limits the direct usefulness of endogenous promoters in gene therapy applications (Nott et al., 2019). Thus, we need methods to design compact synthetic promoters with cell type specificity that are scalable to any cell type of interest.

3. Existing gene expression predictors

Endogenous gene expression is a complex process that is regulated by multiple DNA sequence features, including CREs, TF-binding motifs, and epigenetic modifications. Before the advent of DL, most sequence-to-expression models extracted handcrafted sequence features such as counts of known TF-binding motifs and other short sequence (k-mer) counts within the input sequence (Zrimec et al., 2021). Early applications of DL in genomics used convolutional neural nets (CNNs) with one-hot encoded sequence inputs. For example, Zhou and Troyanskaya (2015) used CNNs to predict various epigenetic modifications and TF-binding sites. More recently, Avsec et al. (2021) showed that using convolutional layers followed by transformer layers improves prediction of endogenous gene expression when compared to convolutional layers alone. Although many of these models achieve high accuracy for endogenous expression, they are not suited to predicting expression from compact promoters used in gene delivery applications because **(1)** unlike endogenous gene expression, control of promoter-driven expression relies on only a short promoter sequence without additional distal regulatory elements, and **(2)** promoter-driven expression utilizes promoter sequences with a much higher information density (density of regulatory sequence motifs) when compared to endogenous promoters.

Models of promoter-driven expression trained using MPRA data have also been developed, which are more directly relevant to the gene delivery setting. For instance, Movva et al. (2019) train a CNN to predict promoter-driven expression in K-562 and HepG2 cells. However, these models cannot be used to directly predict expression in cell types other than those used for training. Since collecting large MPRA datasets for every cell type of interest is infeasible, we need new data-efficient approaches that can be used to train models of cell type-specific promoter-driven expression.

4. Transfer learning methods for improving task performance by leveraging related data

Collecting large datasets that measure promoter-driven expression in multiple cell types is expensive and time-consuming. However, there are several large datasets that provide relevant information for modelling promoter-driven expression. Transfer learning was proposed to effectively model small datasets in these settings by leveraging large relevant datasets. In this work, we explore two main types of transfer learning for the promoter-driven expression prediction task: pretraining followed by linear probing or fine-tuning, and joint training. We explain these techniques in this section.

4.1. Pretraining followed by linear probing or fine-tuning

When DL models are trained from scratch on small datasets, it is difficult for them to learn all task-relevant features, leading to poor performance. However, if there is a large related dataset, training on that dataset prior to training on the small dataset can help the model learn relevant features that are similar between the two datasets. This procedure is called pretraining. The pretrained model can then be further trained on the small dataset to learn which of the features learned during pretraining are relevant for the task at hand and and to modify their weights as needed. This process is data-efficient, as the model has learned most relevant features during pretraining, and generally leads to better prediction performance on the small dataset (e.g. Devlin et al. (2018); Chen et al. (2020)).

There are two main transfer methods for training on the small dataset after pretraining: linear probing and fine-tuning. Pretrained models generate an embedding of the input before using this embedding to make predictions for the pretraining task. Linear probing freezes all weights of the pretrained model and adds a trainable linear layer that is trained on the small dataset to make predictions for the downstream task of interest using the input embeddings. Fine-tuning not only adds a trainable output linear layer but also allows the weights of the pretrained model to be updated when training on the small dataset. Fine-tuning typically leads to better predictions, but there are some instances where linear probing is better, such as when the small dataset contains inputs that are out-of-distribution for the pretrained model (Kumar et al., 2022).

4.2. Joint training

Another effective method to perform transfer learning is to jointly train a model on multiple related datasets, some which are much larger than the target task. Joint training can be accomplished by having a shared backbone network that outputs embeddings of the inputs. These embeddings are then supplied to task-specific layers that output predictions for all tasks. The motivation behind this approach is that the shared backbone network learns a wide variety of features based on the larger datasets, and these features can then be efficiently utilized by the task-specific layers even for tasks with small training datasets. This method has also been shown to improve prediction performance on the smaller datasets (e.g. Yang et al. (2017)).

4.3. Performing multi-task learning (MTL)

MTL is required to pretrain or jointly train on multiple tasks. We perform MTL using the torchmtl package (Bock, 2020). A common backbone network is used to embed inputs. The embeddings are then supplied to task-specific linear layers that make task-specific predictions. During training, each batch is composed of samples for one task and we cycle through the tasks while sampling batches in an epoch (batch-level round-robin) which has been shown to be effective (Alayrac et al., 2022). Since the losses for each task can be on different scales, we use Kendall et al. (2018)'s method to learn weightings for each task's loss. The weighted sum of losses is then minimized using an optimizer.

5. Our approach

Our primary goal is to develop an approach for training effective predictors of cell type-specific promoter-driven expression that leverages large related datasets using the transfer learning methods described in the previous section. To evaluate various training strategies, we generate a target experimental dataset of cell type-specific promoter-driven expression measured in three cell types, as described in Section 5.1. To model these data, we propose a model architecture that draws from current trends in the field of genomic deep learning, described in Section 5.2. Finally, we identify four large related datasets that can be used for transfer learning to improve our expression predictions, described in Section 5.3. In the Results section, we show the effectiveness of these various training strategies and identify the best strategy for predicting cell type-specific promoter-driven expression in new cell types.

5.1. Target task: cell type-specific promoter-driven expression measured by induced fluorescence levels

We collect a new gene expression dataset that measures promoter-driven expression in 3 immune cell lines: Jurkat, K-562, and THP-1. All models trained using various strategies are ultimately evaluated in terms of their effectiveness in predicting promoter-driven expression in each of these 3 cell lines. This serves as a good proxy for a natural setting where we often want to model a small expression dataset. These specific cell lines are chosen because of their similarity to primary cells, and because promoters designed for these cell types could be useful for treating blood cancers. Although promoter-driven expression is well-studied in K-562 cells, with multiple MPRAs using K-562s (e.g. Ernst et al. (2016); van Arensbergen et al. (2019)), there are no large scale datasets that measure promoter-driven expression in Jurkats and THP-1s. Thus, we measure expression from a set of 20,000 promoters of length 250 base pairs (bp), limited by synthesis constraints similar to a gene therapy setting. We choose our tested promoter sequences using heuristics designed to maximize the number of differentially expressed promoters. Briefly, ∼ 50% of the tested promoters are derived from promoters of differentially expressed endogenous genes (Class I). Another ∼ 40% are designed by tiling known and de-novo motifs that were discovered to be enriched in the promoters of differentially expressed endogenous genes by HOMER (Heinz et al., 2010), a motif detection tool (Class II). The final ∼ 10% of promoters are derived from promoters of highly expressed endogenous genes so that our models can learn features of sequences that lead to high expression across many cell types (Class III).

Each promoter is cloned upstream of a minimal CMV promoter and the enhanced green fluorescent protein (EGFP) reporter gene into a lentiviral vector. The expression induced in each cell line upon transduction is measured by the induced fluorescence levels, and we collect two replicate measurements of fluorescence. We get adequate data from 17,104 promoters. For model training and evalutation, ∼ 70% of these promoters are included in the training set, ∼ 10% in the validation set, and ∼ 20% in the test set. The promoters in each set are stratified by both promoter class and GC content. Our models are trained to simultaneously predict fluorescence levels (average across replicates) in each cell line from the promoter sequence using 3 output heads. More details about the experimental protocol (including how we quantify expression strength) and promoter selection are in Appendix A and B, respectively.

5.2. Model architecture

We need an effective model architecture for compact promoter sequences. Our novel architecture is inspired by Avsec et al. (2021)'s model and is shown in Figure 1. It takes in a one-hot encoded promoter sequence and applies 3 length-preserving convolutional layers to learn local sequence features (e.g. TF-binding motifs). A learned [CLS] token embedding is appended to the beginning of the output of

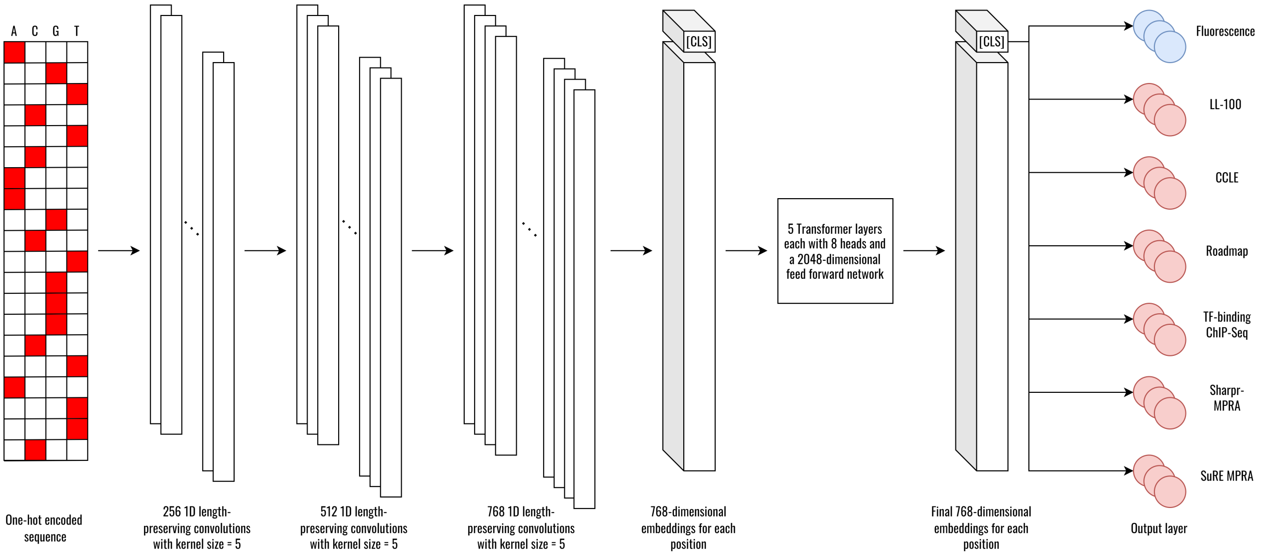
A C G T
One-hot encoded sequence
256 1D length-preserving convolutions with kernel size = 5
512 1D length-preserving convolutions with kernel size = 5
768 1D length-preserving convolutions with kernel size = 5
[CLS]
768-dimensional embeddings for each position
5 Transformer layers each with 8 heads and a 2048-dimensional feed forward network
[CLS]
Final 768-dimensional embeddings for each position
Fluorescence
LL-100
CCLE
Roadmap
TF-binding ChIP-Seq
Sharpr-MPRA
SuRE MPRA
Output layer

Figure 1: Our model architecture for evaluating various training strategies. The fluorescence outputs (in blue) are predictions of cell type-specific promoter-driven expression as measured in our experimental data and all other outputs (in red) are used either during pretraining or joint training.

Dataset	Assay	Sequence used as input	Size	Promoter-driven expression
LL-100	RNA-seq of 100 blood cancer cell lines	[TSS - 300bp, TSS - 50bp), [TSS - 100bp, TSS + 150bp), and [TSS + 100bp, TSS + 350bp)	14,969	✗
CCLE	RNA-seq of 1408 cancer cell lines	[TSS - 300bp, TSS - 50bp), [TSS - 100bp, TSS + 150bp), and [TSS + 100bp, TSS + 350bp)	13,831	✗
Roadmap	RNA-seq of 56 cell lines	[TSS - 300bp, TSS - 50bp), [TSS - 100bp, TSS + 150bp), and [TSS + 100bp, TSS + 350bp)	14,209	✗
Sharpr MPRA	MPRA in K-562 and HepG2	145bp sequences with measured expression	~950K	✓
SuRE MPRA	MPRA in K-562 and HepG2	150-500bp sequences with measured expression	~2.5M (subsampled)	✓
ENCODE TF-binding ChIP-Seq data	1363 ChIP-seq datasets from diverse cells	+ve set: 600bp sequence centered at avg position of nearby peaks -ve set: dinucleotide shuffled +ve sequences	~6M	N/A

Table 1: Summary of datasets used for pretraining or joint training.

the convolutional layers. Then, these outputs are passed through 5 transformer layers (Vaswani et al., 2017). The [CLS] token's final embedding is used as the sequence embedding from which multiple linear layers predict task-specific outputs. We use the MTL approach described in the previous section to train this model. In the Results section, we compare this architecture to two baselines and show the utility of using sequence-based transformer models over the other approaches.

5.3. Pretraining or joint training tasks

Having defined the target task of predicting cell type-specific promoter-driven expression in three experimentally-measured cell types, here we identify four large relevant datasets that can be used for pretraining or joint training. These tasks are summarized in Table 1.

5.3.1. *RNA-Seq data*

Since endogenous promoters play a crucial role in gene expression, it might be useful to pretrain our models on endogenous gene expression data measured by RNA-sequencing (RNA-Seq) in various cell types. This should enable the model to learn TF-binding motifs and their relative importances in various cell types. Thus, we pretrain on three large RNA-Seq datasets: LL-100 (Quentmeier et al., 2019), CCLE (Barretina et al., 2012), and Roadmap (Kundaje et al., 2015). LL-100, CCLE, and Roadmap contain expression values from 100, 1408, and 56 cell lines, respectively.

For each dataset, we first extract the expression values in every cell line, as measured by TPM or RPKM values. TPM values for CCLE and RPKM values from Roadmap are obtained from their respective websites. TPM values for LL-100 are obtained by processing the published raw reads using a standard pipeline (Patel et al., 2022). We filter out any genes that have mean TPM or RPKM values less than 1. Then, we extract three 250bp regions of the promoter for every gene: [TSS - 300bp, TSS - 50bp), [TSS - 100bp, TSS + 150bp), and [TSS + 100bp, TSS + 350bp), which are used to predict expression in every cell line. These regions are chosen by fitting an Xpresso model (Agarwal and Shendure, 2020) to predict median expression across all Roadmap cell lines from various 250bp windows within the TSS ± 1000bp region. We find that the highest prediction performance is obtained using windows within the TSS ± 300bp region. Thus, we choose three 250bp windows covering this region. During training, each promoter sequence window is treated as a separate example with the same associated target expression values. During testing, the predictions for the three windows are averaged to get the final prediction for the gene. We find that this approach yields better fine-tuning and joint training performance compared to using a single large input region such as TSS ± 1500bp. Genes from distinct chromosomes are used in the train, test, and validation sets, and ∼ 70%, ∼ 20% and ∼ 10% of the overall genes are assigned to the train, test, and validation sets, respectively.

5.3.2. ENCODE TF-binding ChIP-Seq data

ChIP-Seq assays are used to discover genomic regions that are bound by TFs, and pretraining on such data can help models learn TF-binding sequence motifs. We obtain peak calls (narrow peaks) from 1645 TF-binding ChIP-Seq datasets from ENCODE (ENCODE Project Consortium, 2012) that do not have any major quality issues (list of datasets is available in the code repository). Peaks that have a q-value greater than 0.05 are filtered out, and the 1363 cell types with at least 1000 peaks after q-value-based filtering are retained. Because many peaks are very close to each other, we merge peaks that occur within 100bp of each other and create a new unified peak at the mean of the individual peaks' positions. This unified peak is annotated as being a peak in all datasets from which the individual peaks originated.

We pretrain our models to predict whether a given sequence contains a peak in each of the 1363 cell types. The positive set for this classification task consists of 600bp sequences centered at every peak. In total, there are ∼ 3M peaks. The negative set is built by sampling a dinucleotide shuffled sequence for every positive sequence, similar to the approach followed by Alipanahi et al. (2015) and Zeng et al. (2016). Peaks (and their corresponding negative sequences) from distinct chromosomes are used in the train, test, and validation sets with ∼ 66.8%, ∼ 23.6%, and ∼ 9.6% of the peaks assigned to the train, test, and validation sets, respectively.

5.3.3. Sharpr-MPRA data

MPRAs measure promoter-driven expression induced by multiple promoters in parallel and thus have high throughput. We hypothesize that pretraining on MPRA data might be very beneficial for our task because of the similarity in experimental protocols - the main difference being that our data measures expression induced by stable transduction while MPRAs measure expression induced by transient transfection. The Sharpr-MPRA dataset (Ernst et al., 2016) measures expression induced by ∼ 487K 145bp promoters in K-562 and HepG2 cells. These promoters are derived from 15,720 295bp sequences centered at DNase I peaks in K-562, HepG2, HUVEC, and H1-hESC cells. Each promoter is cloned upstream of a minimal TATA or strong SV40 promoter and promoter-driven expression is measured for both conditions. Two replicates of these measurements are collected. Thus, there are 8 measurements per promoter (2 cell lines, 2 downstream promoters, 2 replicates).

This dataset was also modelled by Movva et al. (2019), who include each promoter's reverse complement as an additional training example with the same associated expression value. They also predict the average of the values from the two replicates, leading to 12 outputs per input sequence. The ∼ 20K sequences from chromosome 18 and the ∼ 30K sequences from chromosome 8 are used for testing and validation, respectively. All other sequences are used for training. We use their processed data and modelling setup for pretraining.

5.3.4. SuRE MPRA data

SuRE (van Arensbergen et al., 2017) is another MPRA that was scaled up by van Arensbergen et al. (2019) to survey the genomes of 4 individuals from 4 different populations. The genomes of these individuals are broken into 150-500bp fragments and each fragment is cloned into a reporter plasmid. These sequence fragments can drive expression and function as promoters in transfected cells if the fragment contains a valid TSS. ∼ 2.4B and ∼ 1.2B fragments were found to be expressed in K-562 and HepG2 cells, respectively. Pretraining on this large dataset allows our models to learn about the structure of promoters and the effects of single nucleotide polymorphisms (SNPs) on expression.

To the best of our knowledge, no other study has used this data for pretraining. Since pretraining on the full dataset is time-consuming due to its size, we subsample it and create a classification task. Our subsampling accounts for GC content to reduce any associated confounding. First, each tested sequence is binned into 2 expression bins, one for K-562 and one for HepG2. We define 5 bins for each cell based on the number of reads associated with each sequence: 0, (0, 10], (10, 20], (20, 30] and 30+. Most sequences have 0 reads and the number of sequences assigned to each bin decreases with higher read counts. We remove any sequences with ambiguous SNPs and compute the GC content of each sequence. For each individual, we compute a histogram of GC content over all sequences from their genome, with a bin width of 0.05. Then, for each individual and for each combination of K-562 and HepG2 expression bins (25 combinations), we subsample the individual's sequences in that bin combination while keeping the GC content distribution as close as possible to the overall GC content distribution. We aim to get 30K training sequences and 3K testing and validation sequences from each bin combination, reflecting different levels of differential expression; however, some bin combinations have fewer sequences. Ultimately, we obtain $\sim 400-600$K training sequences per individual and $\sim 50-70$K testing and validation sequences. We create datasets for each individual separately. Our models are pretrained to predict a sequence's K-562 and HepG2 expression bin in every individual.

6. Results

First, we validate our model architecture by comparing its prediction performance to the performance of two baseline architectures. Then, we evaluate the efficacy of various training strategies to find the best approach to building predictors of cell type-specific promoter-driven gene expression. Finally, we explore the TF-motifs attended to by our best model using a motif insertion-based analysis in an effort to understand the sequence determinants of expression.

In all our tables and figures, r denotes the Pearson correlation coefficient and ρ denotes the Spearman's rank correlation coefficient between the predictions and targets. The last row in each table shows the metrics obtained by comparing two experimental replicates of the fluorescence measurements in the test set, which gives a sense of the maximum prediction performance that can be obtained using these data. Replicate concordance is shown in Supplementary Figure S.1. The hyperparameters used for our experiments are detailed in Appendix C.

6.1. Model architecture validation

We evaluate two main architectural choices: **(1)** sequence-based DL models compared to simpler models with handcrafted features based on TF-binding motif occurrences, and **(2)** the inclusion of transformer layers compared to purely convolutional architectures. In particular, we compare the prediction performance of the model architecture described in Section 5.2 to two baseline architectures: **(1)** a 4-layer fully connected network (FCN) that takes as input a vector containing the number of TF-binding motif occurrences in the promoter sequence, and **(2)** a 4-layer CNN that takes the promoter sequence as input. More details about these baselines are provided in Appendix C. The performance of each of these models when used to predict the experimental fluorescence data is shown in Table 2. We find that the sequence-based model containing transformer layers as described in Section 5.2 leads to the best performance; therefore, we use this transformer-based model in all further experiments.

Model Class	Jurkat		K-562		THP-1	
	r	ρ	r	ρ	r	ρ
Motif-based FCN	0.4685 ± 0.0127	0.4539 ± 0.0115	0.4624 ± 0.0098	0.4727 ± 0.0115	0.3916 ± 0.0094	0.3726 ± 0.0128
CNN	0.5594 ± 0.0062	0.5012 ± 0.0044	0.5395 ± 0.0085	0.5077 ± 0.0060	0.5013 ± 0.0101	0.3807 ± 0.0122
CNN + Transformer	**0.6389 ± 0.0036**	**0.5996 ± 0.0093**	**0.6152 ± 0.0082**	**0.6043 ± 0.0045**	**0.5672 ± 0.0131**	**0.4742 ± 0.0136**
Test Set Replicate Concordance	0.7900 ± 0.0271	0.7348 ± 0.0116	0.7267 ± 0.0247	0.6875 ± 0.0093	0.6561 ± 0.0423	0.4987 ± 0.0133

Table 2: Prediction performance obtained using 3 model architectures (Figure 1 and two baselines). The mean and standard deviation are obtained by fitting 5 different models using 5 different train, test and validation splits of the fluorescence data.

6.2. Evaluation of training strategies

Next, we systematically evaluate various training strategies by pretraining separate models using each of the tasks described in Section 5.3 alone or in combination. We then perform either linear probing or fine-tuning of these pretrained models to predict the experimental data. We also evaluate a joint training strategy by training on the tasks in Section 5.3 together with the experimental data. We compare the prediction performance obtained using each of these methods, as evaluated on one common split of the experimental data, in Table 3. Appendix E shows the performance of our models on the pretraining/joint training tasks.

We find that the best predictor of cell type-specific promoter-driven expression is obtained by pretraining on both of the existing MPRA datasets (Sharpr-MPRA and SuRE MPRA) before fine-tuning on our smaller dataset of measured expression in the cell types of interest. The predictions from this best strategy are shown in Figure 2. We confirm these results using 5 different splits of the experimental dataset to get an estimate of the mean and standard deviation of performance (Supplementary Table S.1). Note that we do not train multiple models using joint training because of the large computational cost and relatively poor performance in Table 3. Again, we find that pretraining on all MPRA data leads to the best performance, boosting it by **$6-12\%$** compared to training from scratch on the new experimental data.

From Table 3 and Supplementary Table S.1, we also note the following observations:

(1) The ordering of the additional training tasks based on performance is similar irrespective of the transfer learning method used, suggesting that some tasks are inherently better for transfer learning than others. For our task of predicting expression from a compact promoter sequence, the MPRA datasets consistently outperform the other training tasks. These datasets are also the closest to our experimental setup, since they also capture promoter-driven expression. Thus, we find that using more relevant tasks leads to better transfer learning, akin to prior observations by Gururangan et al. (2020) among others.

(2) Pretraining on certain datasets (RNA-Seq, TF-binding data) can lead to negative transfer; i.e. lower performance of fine-tuned models compared to models trained from scratch on the new experimental data. Negative transfer has also been observed in

Additional Training Tasks	Training Strategy	Jurkat		K-562		THP-1	
		r	ρ	r	ρ	r	ρ
-	Train from scratch	0.6509	0.6078	0.6370	0.6190	0.5604	0.4894
All RNA-Seq	Pretrain + Linear Probing	0.5126	0.4707	0.5188	0.5005	0.4561	0.4106
TF-binding	Pretrain + Linear Probing	0.4213	0.4514	0.4434	0.4904	0.3260	0.3644
Sharpr-MPRA	Pretrain + Linear Probing	0.6034	0.5873	0.5935	0.6038	0.5066	0.4822
SuRE MPRA	Pretrain + Linear Probing	0.6390	0.6055	0.6389	0.6432	0.5552	0.4940
All MPRA	Pretrain + Linear Probing	0.6629	0.6336	0.6565	0.6586	0.5751	0.5234
All RNA-Seq	Pretrain + Fine-tune	0.6338	0.5940	0.6269	0.6076	0.5468	0.4814
TF-binding	Pretrain + Fine-tune	0.6229	0.5826	0.6200	0.6031	0.5231	0.4579
Sharpr-MPRA	Pretrain + Fine-tune	0.6316	0.6045	0.6210	0.6320	0.5375	0.4905
SuRE MPRA	Pretrain + Fine-tune	0.6732	0.6320	0.6680	0.6611	0.5962	0.5206
All MPRA	Pretrain + Fine-tune	**0.6849**	**0.6463**	**0.6762**	**0.6710**	**0.5991**	**0.5316**
All RNA-Seq	Joint Training	0.6564	0.6025	0.6395	0.6133	0.5669	0.4970
TF-binding	Joint Training	0.6460	0.5981	0.6326	0.6029	0.5495	0.4919
Sharpr-MPRA	Joint Training	0.6519	0.6169	0.6378	0.6235	0.5547	0.5032
SuRE MPRA	Joint Training	0.6517	0.6216	0.6396	0.6394	0.5711	0.5039
All MPRA	Joint Training	0.6522	0.6121	0.6414	0.6329	0.5658	0.5006
Test Set Replicate Concordance		0.8191	0.7345	0.7371	0.6656	0.7300	0.4827

Table 3: Prediction performance obtained using various training strategies on a common split of the experimental fluorescence dataset. Linear probing, fine-tuning, and training from scratch are performed 5 times with 5 different parameter initializations and the metrics are averaged over these initializations.

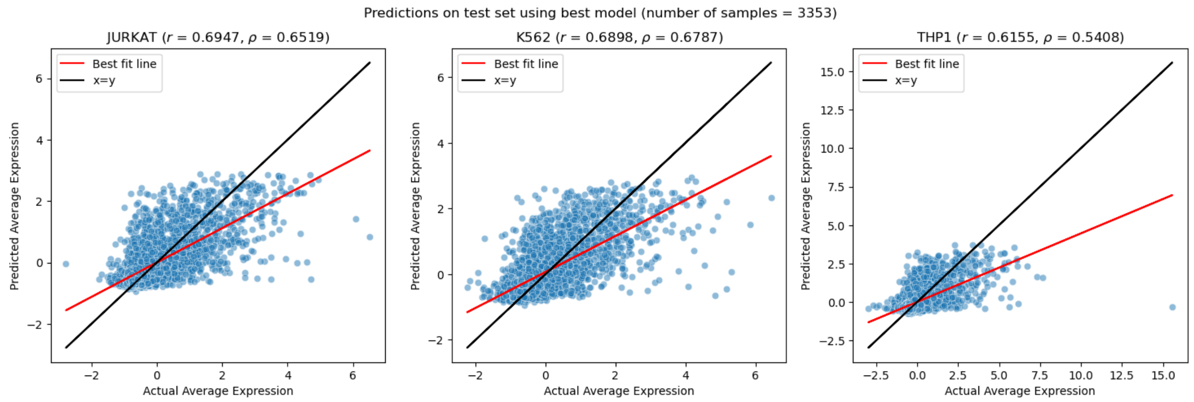

Figure 2: Scatter plots showing the predictions obtained using the best model from Table 3 for each cell type, compared to the experimental measurements. Models are trained to predict Z-scored expression values. The best model is pretrained on both of the existing MPRA datasets before fine-tuning on the fluorescence data. Fine-tuning is performed using 5 different parameter initializations and we choose the model that leads to best average ρ on the validation set as the best model.

other fields (Liu et al., 2022), and indicates that care must be taken when choosing pretraining tasks.

(3) On average, the best transfer learning method is pretraining followed by fine-tuning. Joint training produces the second-best results and pretraining followed by linear probing leads to the worst results.

6.3. Model interpretation: effect of TF-binding motifs

We finally want to use our models to understand the sequence determinants of promoter-driven expression, which is important for many biological applications where we want to understand how compact promoters drive expression, including the role of TF-binding motifs. Traditionally, the experimental data can be directly used to compare expression of tested sequences containing a given motif to those not containing the motif. However, with a small experimental dataset, such statistical tests are underpowered for many motifs that are present in relatively few tested sequences; thus, only a subset of motifs can be analyzed. In contrast, using the DL models developed above, we can test the effect of any motif of interest.

First, we obtain a list of clustered TF-binding motifs (Vierstra et al., 2020) [1]. For every motif, we generate 10 instances of that motif by sampling from its position-weight matrix (PWM). We then randomly insert (*in silico*) these 10 instances of the motif into every tested sequence from the experimental dataset that did not originally contain the motif. We use the best performing model, described above, to predict the expression of the modified and unaltered sequences. Our estimate of the influence of the motif on expression is then obtained by subtracting the predictions for the unaltered sequences from those for the modified sequences, and averaging over all sequences.

To validate our influence estimates, we check whether they match the results of the traditional approach for the subset of motifs that can be analyzed directly from the experimental data. In particular, for every motif for which we have ≥ 30 sequences in the fluorescence dataset containing that motif (determined by running FIMO (Grant et al., 2011) with default arguments and retaining detected motif occurrences with q-value < 0.01), we run a t-test to compare the measured expression of sequences in which the motif is present to those in which it is absent. We then compare significant fold-changes (q-value < 0.05) in observed expression in the presence of the motif to the predicted motif influence scores from our model. We find that they are highly correlated, with Spearman's rank correlation coefficients of 0.7673, 0.8198 and 0.8008 in Jurkat, K-562 and THP-1 cells, respectively (correlation depicted in Figure 3). This result indicates that our model can accurately predict the effects of motifs on promoter-driven gene expression.

Next, we analyze motifs with the highest and lowest predicted influence scores (note that a negative influence score means that the motif reduces predicted

[1] https://resources.altius.org/~jvierstra/projects/motif-clustering-v2.0beta/

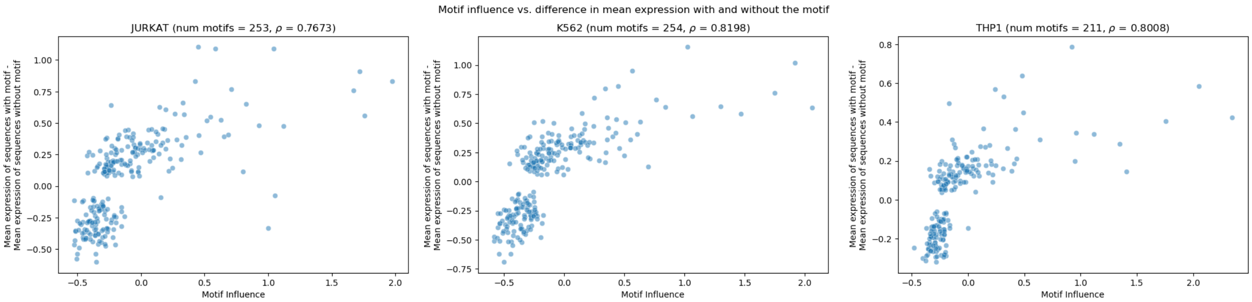

Figure 3: Scatter plots showing the correlation between our motif influence estimates in each cell type and significant changes in experimentally-measured expression in the presence of each motif (significance is computed using a t-test and q-values < 0.05 are deemed significant). Similar plots showing the correlation for all motifs that are contained in at least 30 experimental sequences are shown in Supplementary Figure S.2.

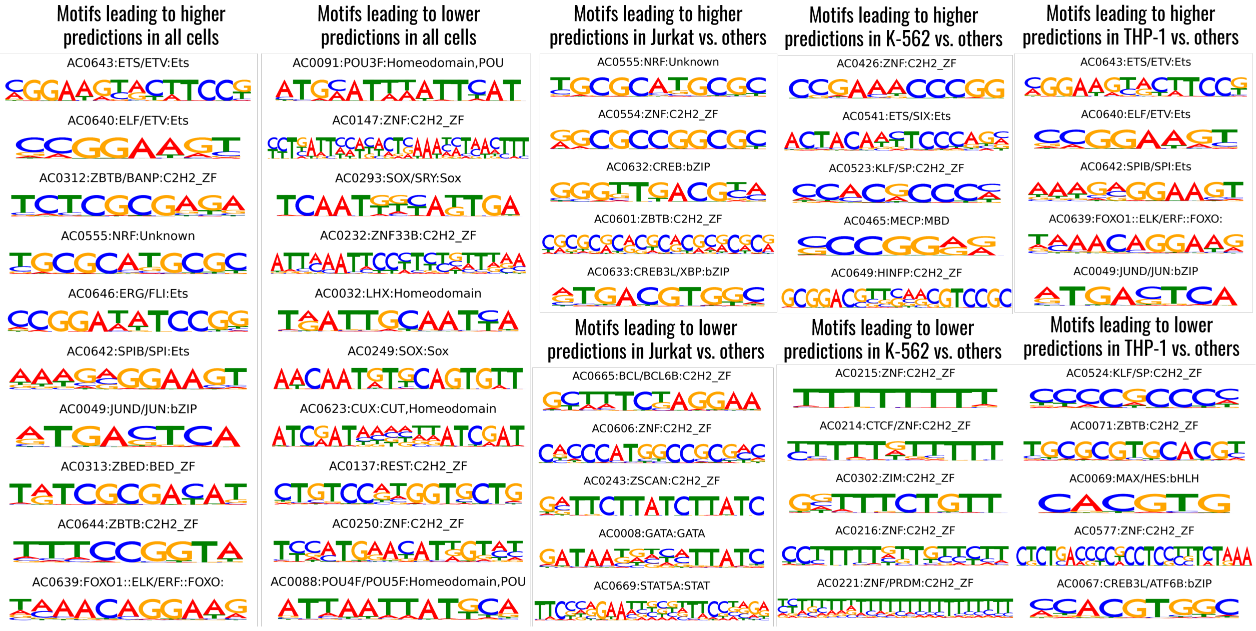

Figure 4: Top TF-binding motifs that influence model predictions.

expression). We first Z-score the cell type-specific influence scores across all motifs in each cell type; let the resulting scores be $m_{Jurkat}, m_{K-562}, m_{THP-1}$ for a motif m. Then, to identify top upregulating motifs in a given cell type (e.g. Jurkat), we consider all motifs with positive influence scores in that cell type and choose those with the highest differential influence score (e.g. $m_{Jurkat} - \max(m_{K-562}, m_{THP-1})$). To identify top downregulating motifs, we similarly consider motifs with negative influence scores and choose those with the smallest differential score (e.g. $m_{Jurkat} - \min(m_{K-562}, m_{THP-1})$).

Figure 4 shows the PWMs of the top motifs that are predicted to uniformly drive up or down expression in all three cell types or to differentially drive expression in each of the three cell types. Several of these predictions are consistent with prior biological knowledge. For example, the motif for NRF1, an activator that plays a role in homeostasis and the activation of housekeeping genes (Zhang and Xiang, 2016), leads to higher overall predicted expression. The motif for REST, a repressor (Chong et al., 1995), leads to lower predicted expression. ZBTB proteins are responsible for regulating various T cell processes (Cheng et al., 2021), and we find that the ZBTB motif leads to higher predicted differential expression in Jurkats, which are T cells. These results suggest that our model can identify biologically meaningful motifs that play a role in promoter-driven expression.

7. Conclusion

We propose a novel pretraining-based approach that leverages existing large datasets to train predictors of cell type-specific promoter-driven expression using small datasets. After a thorough analysis of various pretraining tasks and transfer methods, we find that the best predictors can be obtained by first pretraining on existing promoter-driven expression data from MPRAs and then fine-tuning on small datasets measuring expression in specific cell types. We see a $6-12\%$ improvement in performance using this approach compared to training models from scratch. Finally, we use our best models to explore the effects of various TF-binding motifs on expression and find meaningful effects that concur with previous results. Our approach can be easily adopted to model any other promoter-driven expression dataset in a data-efficient manner and can be used for designing promoters for gene therapy.

Acknowledgments

This work was partially supported by an H2H8 Graduate Research Grant to AJR, a Schmidt Futures grant to SL, and the U.S. National Institutes of Health grant R00HG009677 to NMI.

This research used the Savio computational cluster resource provided by the Berkeley Research Computing program at the University of California, Berkeley (supported by the UC Berkeley Chancellor, Vice Chancellor for Research, and Chief Information Officer).

References

Vikram Agarwal and Jay Shendure. Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks. *Cell reports*, 31(7):107663, 2020.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33 (8):831–838, 2015.

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. *Nature methods*, 18 (10):1196–1203, 2021.

Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful zero-shot predictors of non-coding variant effects. *bioRxiv*, pages 2022–08, 2022.

Christian Bock. torchmtl: A lightweight module for multi-task learning in pytorch, 2020. URL https://github.com/chrisby/torchMTL.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pages 773–782. PMLR, 2019.

Alan R Brooks, Richard N Harkins, Peiyin Wang, Hu Sheng Qian, Pengxuan Liu, and Gabor M Rubanyi. Transcriptional silencing is associated with extensive methylation of the cmv promoter following adenoviral gene delivery to muscle. *The Journal of Gene Medicine: A cross-disciplinary journal for research on the science of gene transfer and its clinical applications*, 6(4):395–404, 2004.

Shifu Chen, Yanqing Zhou, Yaru Chen, and Jia Gu. fastp: an ultra-fast all-in-one fastq preprocessor. *Bioinformatics*, 34(17):i884–i890, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Zhong-Yan Cheng, Ting-Ting He, Xiao-Ming Gao, Ying Zhao, and Jun Wang. Zbtb transcription factors: key regulators of the development, differentiation and effector function of t cells. *Frontiers in Immunology*, 12, 2021.

Jayhong A Chong, José Tapia-Ramirez, Sandra Kim, Juan J Toledo-Aral, Yingcong Zheng, Michael C Boutros, Yelena M Altshuller, Michael A Frohman, Susan D Kraner, and Gail Mandel. Rest: a mammalian silencer protein that restricts sodium channel gene expression to neurons. *Cell*, 80(6):949–957, 1995.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012.

Jason Ernst, Alexandre Melnikov, Xiaolan Zhang, Li Wang, Peter Rogov, Tarjei S Mikkelsen, and Manolis Kellis. Genome-scale high-resolution mapping of activating and repressive nucleotides in regulatory regions. *Nature biotechnology*, 34 (11):1180–1190, 2016.

Charles E Grant, Timothy L Bailey, and William Stafford Noble. Fimo: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C Lin, Peter Laslo, Jason X Cheng, Cornelis Murre, Harinder Singh, and Christopher K Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular cell*, 38(4):576–589, 2010.

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, 2021.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Anshul Kundaje, Wouter Meuleman, Jason Ernst, Misha Bilenky, Angela Yen, Alireza Heravi-Moussavi, Pouya Kheradpour, Zhizhuo Zhang, Jianrong Wang, Michael J Ziller, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

Johannes Linder, Nicholas Bogard, Alexander B Rosenberg, and Georg Seelig. A generative neural network for maximizing fitness and diversity of synthetic dna and protein sequences. *Cell systems*, 11(1):49–62, 2020.

Zhili Liu, Jianhua Han, Kai Chen, Lanqing Hong, Hang Xu, Chunjing Xu, and Zhenguo Li. Task-customized self-supervised pre-training with scalable dynamic routing. *Transfer*, 55:65, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1–21, 2014.

Carol H Miao, Kazuo Ohashi, Gijsbert A Patijn, Leonard Meuse, Xin Ye, Arthur R Thompson, and Mark A Kay. Inclusion of the hepatic locus control region, an intron, and untranslated region increases and stabilizes hepatic factor ix gene expression in vivo but not in vitro. *Molecular Therapy*, 1(6):522–532, 2000.

Shentong Mo, Xi Fu, Chenyang Hong, Yizhen Chen, Yuxuan Zheng, Xiangru Tang, Zhiqiang Shen, Eric P Xing, and Yanyan Lan. Multi-modal self-supervised pre-training for regulatory genome across cell types. *arXiv preprint arXiv:2110.05231*, 2021.

Mariela Montaño-Samaniego, Diana M Bravo-Estupiñan, Oscar Méndez-Guerrero, Ernesto Alarcón-Hernández, and Miguel Ibáñez-Hernández. Strategies for targeting gene therapy in cancer cells with tumor-specific promoters. *Frontiers in oncology*, 10:605380, 2020.

Rajiv Movva, Peyton Greenside, Georgi K Marinov, Surag Nair, Avanti Shrikumar, and Anshul Kundaje. Deciphering regulatory dna sequences and noncoding genetic variants using neural network models of massively parallel reporter assays. *PLoS One*, 14(6):e0218073, 2019.

Lior Nissim, Ming-Ru Wu, Erez Pery, Adina Binder-Nissim, Hiroshi I Suzuki, Doron Stupp, Claudia Wehrspaun, Yuval Tabach, Phillip A Sharp, and Timothy K Lu. Synthetic rna-based immunomodulatory gene circuits for cancer immunotherapy. *Cell*, 171(5):1138–1150, 2017.

Alexi Nott, Inge R Holtman, Nicole G Coufal, Johannes CM Schlachetzki, Miao Yu, Rong Hu, Claudia Z Han, Monique Pena, Jiayang Xiao, Yin Wu, et al. Brain cell type–specific enhancer–promoter interactome maps and disease-risk association. *Science*, 366(6469):1134–1139, 2019.

Harshil Patel, Phil Ewels, Alexander Peltzer, Rickard Hammarén, Olga Botvinnik, Gregor Sturm, Denis Moreno, Pranathi Vemuri, silviamorins, Lorena Pantano, Mahesh Binzer-Panchal, Gavin Kelly, FriederikeHanssen, Maxime U. Garcia, nf-core bot, Chris Cheshire, rfenouil, Jose Espinosa-Carrasco, marchoeppner, Peng Zhou, Gisela Gabernet, Christian Mertes, Daniel Straub, Matthias Hörtenhuber, Paolo Di Tommaso, Sven F., George Hall, Senthilkumar Panneerselvam, Denis OMeally, and jun wan. nf-core/rnaseq: nf-core/rnaseq v3.6 - Platinum Platypus, March 2022. URL https://doi.org/10.5281/zenodo.6327553.

Hilmar Quentmeier, Claudia Pommerenke, Wilhelm G Dirks, Sonja Eberth, Max Koeppel, Roderick AF MacLeod, Stefan Nagel, Klaus Steube, Cord C Uphoff, and Hans G Drexler. The ll-100 panel: 100 cell lines for blood cancer studies. *Scientific reports*, 9(1):1–14, 2019.

Nilofer Sayed, Prince Allawadhi, Amit Khurana, Vishakha Singh, Umashanker Navik, Sravan Kumar Pasumarthi, Isha Khurana, Anil Kumar Banothu, Ralf Weiskirchen, and Kala Kumar Bharani. Gene therapy: Comprehensive overview and therapeutic applications. *Life sciences*, page 120375, 2022.

Jamie L Shirley, Ype P de Jong, Cox Terhorst, and Roland W Herzog. Immune responses to viral gene therapy vectors. *Molecular Therapy*, 28(3): 709–722, 2020.

Tabula Sapiens Consortium. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022.

Andrea Telatin, Piero Fariselli, and Giovanni Birolo. Seqfu: a suite of utilities for the robust and reproducible manipulation of sequence files. *Bioengineering*, 8(5):59, 2021.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pages 10358–10368. PMLR, 2021.

Joris van Arensbergen, Vincent D FitzPatrick, Marcel de Haas, Ludo Pagie, Jasper Sluimer, Harmen J Bussemaker, and Bas van Steensel. Genome-wide mapping of autonomous promoter activity in human cells. *Nature biotechnology*, 35(2):145–153, 2017.

Joris van Arensbergen, Ludo Pagie, Vincent D FitzPatrick, Marcel de Haas, Marijke P Baltissen, Federico Comoglio, Robin H van der Weide, Hans Teunissen, Urmo Võsa, Lude Franke, et al. High-throughput identification of human snps affecting regulatory element activity. *Nature genetics*, 51(7): 1160–1169, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jeff Vierstra, John Lazar, Richard Sandstrom, Jessica Halow, Kristen Lee, Daniel Bates, Morgan Diegel, Douglas Dunn, Fidencio Neri, Eric Haugen, et al. Global reference mapping of human transcription factor footprints. *Nature*, 583(7818):729–736, 2020.

Ming-Ru Wu, Lior Nissim, Doron Stupp, Erez Pery, Adina Binder-Nissim, Karen Weisinger, Casper Enghuus, Sebastian R Palacios, Melissa Humphrey, Zhizhuo Zhang, et al. A high-throughput screening and computation platform for identifying synthetic promoters with enhanced cell-state specificity (specs). *Nature communications*, 10(1):1–10, 2019.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*, 2017.

Haoyang Zeng, Matthew D Edwards, Ge Liu, and David K Gifford. Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

Wenhuan Zeng, Anupam Gautam, and Daniel H Huson. Mulan-methyl-multiple transformer-based language models for accurate dna methylation prediction. *bioRxiv*, pages 2023–01, 2023.

Yiguo Zhang and Yuancai Xiang. Molecular and cellular basis for the unique functioning of nrf1, an indispensable transcription factor for maintaining cell homoeostasis and organ integrity. *Biochemical Journal*, 473(8):961–1000, 2016.

Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10): 931–934, 2015.

Jan Zrimec, Filip Buric, Mariia Kokina, Victor Garcia, and Aleksej Zelezniak. Learning the regulatory code of gene expression. *Frontiers in Molecular Biosciences*, 8:673363, 2021.

A. Experimental methods

A.1. Library cloning

The promoter library was synthesized by Twist Biosciences in a pooled fashion using microarray sythethesis. 25bp overhangs were added to each 250bp promoter sequence to allow for PCR amplification and Golden Gate assembly (5'-TAGTCGGCTAGATGCGTCTCCTACG(Nx250)GGTACGAGACGACTGTCTTTCCCCT-3'). 20ng of the oligopool was PCR amplified in a 50µL reaction using 1.5µL of of each 10µM primer (TAGTCGGCTAGATGCGTCTCC and AGGGGAAAGACAGTCGTCTCG), and 25µL KAPA HiFi HotStart ReadyMix (Roche KK2602). The thermocycling protocol was 98°C for 3 minutes followed by 12 cycles of 98°C for 20s, 69°C for 15s, 72°C for 15s with a final extension at 72°C for 1 minute. 1µL of the reaction was analyzed by gel electrophoresis, and a single band was visualized at 300bp. The remainder of the reaction was purified using DNA Clean & Concentrator-5 (Zymo D4004) and eluted in 12µL of nuclease-free H_2O. The amplified oligopool was then cloned into a 3rd generation lentiviral vector immediately upstream of a minimal CMV promoter driving the expression of enhanced green fluorescent protein (EGFP) using a 25µL Golden Gate reaction containing 250ng backbone plasmid, 2X molar of the purified oligopool, 1µL Esp3I (Thermo Fisher FD0454), 1µL T4 DNA ligase (NEB M0202L, 400U/µL) and 2.5µL T4 ligase buffer (NEB B0202S). After an initial 5 minute digestion at 37°C, 30 cycles of 37°C digestion and 16°C ligation were followed by 20 minutes of ligation at 16°C, 30 minutes of digestion at 37°C and 20 minutes of heat-inactivation at 80°C. The reaction was purified using DNA Clean & Concentrator-5 (Zymo D4004) and eluted in 6µL of nuclease-free H_2O. 2µL were transformed into Endura electrocompetent cells (Biosearch Technologies 60242-2) following the manufacturer's protocol. After recovery, the cells were plated on a single large 245mm x 245mm LB plate with carbenicillin, and serial dilutions were plated on standard sized plates up to $1{:}1\times10^6$ to assess library coverage. After overnight incubation at 30°C, colonies were counted on the dilution plates to assure a library coverage of at least 30X. Colonies from the large plate were scraped into liquid suspension and collected into a 50mL conical tube before the plasmid pool was prepared using NucleoBond Xtra Midi EF (Macherey-Nagel 740420). Subsequent analysis of the plasmid pool using gel electrophoresis confirmed a homogenously sized plasmid species that was not digestible with Esp3I (Thermo Fisher FD0454).

A.2. Cell lines and culture conditions

Jurkat, K-562, and THP-1 cells were obtained from American Type Culture Collection (TIB-152, CCL-243, and TIB-202) and grown in RPMI + GlutaMAX (Gibco 61870036) supplemented with 10% FBS (Gibco 26140079), 1x penicillin/streptomycin (Gibco 15140122), 1mM sodium pyruvate (Gibco 11360070) and 10mM HEPES (Gibco 15630080). Jurkat and K-562 cells were generally maintained between 1×10^5-1×10^6 cells/mL, and THP-1 wells were maintained between 2×10^5-1×10^6 cells/mL. All suspension cell lines were split every 2-4 days by counting cell density and diluting cells into a new flask with fresh medium warmed to 37°C. Lenti-X 293T cells were attained from Takara Bio (632180) and grown in DMEM, high glucose, pyruvate (Gibco 11995065) supplemented with 10% FBS (Gibco 26140079) and 1x penicillin/streptomycin (Gibco 15140122). Lenti-X cells were split every 2-4 days by aspirating medium, treating with TrypLE Express (Gibco 12604021), and reseeding cells into a new flask with fresh medium warmed to 37°C. Incubator conditions were kept at 37°C, 5% CO_2 and >90% RH. All cell lines were routinely tested for mycoplasma contamination every 2-4 months with MycoStrip mycoplasma detection kit (InvivoGen rep-mysnc-100).

A.3. Lentiviral production and titration

Large scale lentiviral production was performed in Lenti-X cells by polyethylenimine (PEI, Polysciences 23966) transfection into confluent T225 flasks containing DMEM, high glucose, pyruvate (Gibco 11995065) supplemented with 10% FBS (Gibco 26140079) and 10mM HEPES (Gibco 15630080). 40µg of DNA were transfected into each flask using 2nd generation packaging plasmids pMD2.G (Addgene #12259) and psPAX2 (Addgene #12260) along with the lentiviral plasmid pool at a mass ratio of 1:2:4. After 72 hours of incubation, lentiviral particles were concentrated 10X using Lenti-X Concentrator (Takara Bio 631232) per the manufacturer's instructions, and single use aliquots were frozen at -80°C. Functional titration of each batch of lentivirus was performed in Jurkat, K-562, and THP-1 cells by transducing 4×10^4 cells via 90 minute spinfection at 1000g and 32°C in 96 well plates with 8μg/mL polybrene (Millipore TR-1003-G). At least five serial dilutions of lentivirus were used, and transductions were performed in quadruplicate. After overnight incubation, media containing lentivirus was removed and replaced with fresh media with and without 2μg/mL puromycin (Gibco A1113803). After five days of selection, cell survival in each well was quantified on a Tecan Spark plate reader using CellTiter-Glo 2.0 Cell Viability Assay (Promega G9242), and percent survival was calculated as the ratio of luminescence in the presence versus absence of puromycin for each lentiviral dilution. Finally, functional lentiviral titer was calculated for all dilutions with 5-30% survival and averaged for each cell line.

A.4. High-throughput measurements of promoter activity

8×10^7 Jurkat, K-562, or THP-1 cells were transduced in duplicate via 90 minute spinfection at 1000g and 32°C in 50mL conical tubes with 8×10^6 infection units (IUs) of virus and 8μg/mL polybrene (Millipore TR-1003-G) for a multiplicity of infection (MOI) of 0.1 and a library coverage of 400X. After transfer to T225 flasks and overnight incubation, media containing lentivirus was removed and replaced with fresh media containing 2μg/mL puromycin (Gibco A1113803). After five days of selection, cells were expanded a further 2-10 days in the absence of puromycin to dilute dead cells and attain at least 4×10^7 cells (2000X coverage) for sorting. Selected cells were sorted into four 25% bins of EGFP fluorescence using a BD FACSAria Fusion Special Order Research Product. At least 2×10^7 total cells were sorted for a library coverage of 1000X. Cells from each bin were pelleted, and the supernatant was removed for short-term storage at -20°C.

A.5. Library preparation and sequencing

Genomic DNA was extracted from sorted cell pellets using Quick-DNA Midiprep Plus Kit (Zymo D4075) using the manufacterer's instructions. Next generation sequencing (NGS) libraries were prepared using two consecutive PCR steps. In PCR1, the promoters contained in each sorted bin were amplified from the total amount of corresponding genomic DNA using 4μL of each 100μM primer and 400μL NEBNext Ultra II Q5 Master Mix (NEB M0544X). Each 800μL reaction was divided into 8×100μL reactions in a 96 well PCR plate before thermocycling at 98°C for 30s, followed by 20 cycles of 98°C for 10s, 63°C for 30s and 65°C for 45s, with a final extension at 65°C for 5 minutes. All eight completed reactions for each bin were combined into a single tube and vortexed thoroughly before 50μL were purified using a 0.7X AMPure XP bead cleanup (Beckman Coulter A63881). Sequencing adapters and barcodes were then added to the promoter amplicons in PCR2 by combining 2μL of purified PCR1, 2μL of index primers at 10μM each and 25μL NEBNext Ultra II Q5 Master Mix (NEB M0544X). The 50μL reaction was thermocycled at 98°C for 30s, followed by 7 cycles of 98°C for 10s and 65°C for 75s, with a final extension at 65°C for 5 minutes. The PCR2 products were run on a 2% agarose gel, and each produced a single 428bp band, which was extracted using Monarch DNA Gel Extraction Kit (NEB T1020L). Gel-extracted PCR2 products from each bin were then quantified by Qubit 1X dsDNA HS Assay (Thermo Fisher Q33231) and pooled at equimolar ratios before requantification with Qubit and fragment analysis with Agilent 2100 Bioanalyzer using the High Sensitivity DNA Kit (Agilent Technologies 50674626). Prepared libraries were loaded onto the Illumina NextSeq 2000 at 750-850pM and sequenced using 300 cycle v3 kits with P1 or P2 flow cells (Illumina 20050264 and 20046813) to attain at least 1000X sequencing coverage for each replicate.

A.6. Sequencing analysis

Raw BCL files were converted to fastq files and demultiplexed with bcl-convert v4.0.3 (Illumina). Paired-end reads were trimmed, merged and filtered using fastp (Chen et al., 2018) followed by dereplication and counting with seqfu (Telatin et al., 2021). Only reads with zero mismatches to a promoter in our library were counted, and only promoters with at least five reads in each replicate across all cell lines were considered in downstream analyses.

A.7. Quantifying expression strength of promoters

The expression strength of each promoter was calculated as the log (base 2) ratio of reads in the highest quartile EGFP bin to the lowest quartile EGFP bin after adding one read to each bin, and the average expression strength (across the two replicates) was calculated for each promoter in each cell line.

B. Generation of promoter sequences for the experimental dataset

We constructed a promoter library for the experiments described above, which was then used to train and fine-tune our models, containing the following types of sequences.

B.1. Class I (9991 promoters)

These promoters were extracted from the promoters of endogenous differentially expressed (DE) genes. Gene expression data from LL-100 (Quentmeier et al., 2019) and CCLE (Barretina et al., 2012) were used to identify DE genes. Although we measure expression in Jurkat, K-562 and THP-1 cells, the cell types used for this DE analysis were Jurkat, THP-1 and *NK-92*. We later switched from NK-92s to K-562s due to experimental difficulties. DE genes were identified by DESeq2 (Love et al., 2014). Briefly, for each of the three cell lines, we identified a set of “globally" up/down-regulated genes that were up/down-regulated in that cell line and related cell lines (other immune cells of the same type), when compared to all other cell lines. For each of the three cell lines, we also identified a set of “locally" up/down-regulated genes that were up/down-regulated in that cell line and related cell lines when compared to the other two chosen cell lines and cell lines related to them. For each cell line, we took the intersection of its globally and locally up/down-regulated genes and considered the 1111 top DE genes per cell line (711 up-regulated and 400 down-regulated). Following the rationale from section 5.3.1, we extracted three 250bp promoter sequences for every gene − [TSS - 300bp, TSS - 50bp), [TSS - 100bp, TSS + 150bp), and [TSS + 100bp, TSS + 350bp) − to get a total of 3333 promoters per cell line and 9991 promoters overall (after removing duplicates) to test in our experiments.

B.2. Class II (7998 promoters)

Promoters in this class were constructed using HOMER (Heinz et al., 2010), a motif detection tool. We supplied the DE genes identified above for the Class I promoters to HOMER, analyzing the [TSS - 300bp, TSS + 50bp] regions of these genes to identify enriched motifs. HOMER identifies two types of enriched motifs, known motifs (which we obtained from Vierstra et al. (2020)) and de-novo motifs. We identified known motifs that were enriched with q-values less than 0.05 and de-novo motifs that were enriched with p-values less than 1e-10. For each cell type, we then generated 2666 promoters, 1500 using motifs enriched in that cell type’s upregulated genes and 1166 using a mix of motifs enriched in that cell type’s upregulated genes and motifs enriched in the other two cell types’ downregulated genes. To generate the promoters, we inserted up to 18 randomly sampled motifs from the above set into an endogenous promoter segment, ([TSS - 100bp, TSS + 150bp)) from an upregulated gene in NK-92s. The exact inserted sequence for each motif was obtained by sampling from its PWM. This process resulted in inserting more than 100bp of motifs into the original 250bp endogenous promoter segment for $\sim 77\%$ of the Class II promoters.

B.3. Class III (2011 promoters)

Finally, we extracted sequences from the promoters of endogenous highly expressed genes, which were chosen as follows:

1. 1004 genes with the lowest coefficient of variation in their TPM values across all cell lines in the CCLE dataset (restricted to those with a TPM of at least 1).
2. 1007 genes that were up-regulated in all three of the selected cell lines (and related cell lines) vs. all other cell lines in the CCLE dataset, identified using DESeq.

We used the [TSS - 100bp, TSS + 150bp) regions of these genes as 250bp promoter sequences to test in our experiments.

C. Modelling details

C.1. Architectures of baseline models

The motif occurrences-based FCN used as a baseline has 4 layers. We use FIMO (Grant et al., 2011) to extract the number of occurrences of clustered TF-binding motifs (Vierstra et al., 2020) in the sequences in the fluorescence dataset (FIMO is run with default arguments and we retain detected motif occurrences with q-value < 0.01). Vectors containing these occurrence counts for all motifs are input to the FCN. Then, 4 fully connected layers with 2048, 1024, 1024 and 512 neurons are applied to get embeddings for each input (each layer except the last uses ReLU activations). These embeddings are then used by a linear output layer to make the fluorescence predictions.

The baseline CNN has 4 convolutional layers followed by 2 fully connected layers. One-hot encoded sequences are fed as inputs to the network. Then 4 1D length preserving convolutional layers with 512, 768, 768 and 1024 filters of size 5 are applied. Two 1D max pooling layers of size 5 are applied between the second and third layer, and after the last layer. The outputs of the CNN are flattened and passed through 2 fully connected layers with 2048 neurons (and with ReLU activation) and 1024 neurons. The final outputs of this network are then used by a linear output layer to make the fluorescence predictions.

Note that all convolutional layers in both the baseline CNN and our main transformer-based model use ReLU activation and are followed by batch norm and dropout (0.1 dropout probability) layers.

C.2. Hyperparameters

1. We use a cluster consisting of GPU-enabled nodes to train our models. The nodes either use Nvidia A40s or V100s.
2. All models are trained using the AdawW optimizer (Loshchilov and Hutter, 2017).
3. For regression tasks, we Z-score all target values before fitting models.
4. All models trained from scratch to predict fluorescence use a 1e-5 learning rate, 1e-4 weight decay, and 96 batch size. They are trained for a maximum of 50 epochs but if the average Spearman's rank correlation coefficient of the fluorescence data's validation set does not improve for 5 epochs, we stop training.
5. We use a 1e-5 learning rate and 1e-4 weight decay during pretraining and joint training. If the SuRE MPRA dataset is used for pretraining/joint training, we use a batch size of 12. Otherwise, we use a batch size of 96.
6. We pretrain for a maximum of 50, 10, 20, 10 and 8 epochs on RNA-Seq, TF-binding, Sharpr-MPRA, SuRE MPRA and all MPRA data respectively. Again, we stop training if the validation loss does not decrease for 5 epochs. In every epoch, if using more than one dataset for pretraining, we cycle through each dataset to sample batches and an epoch is done when we have run through the largest dataset fully. This leads to smaller datasets being run through more than once in an epoch but we tend to use similar sized datasets making this less of an issue.
7. During joint training, as the fluorescence dataset is small compared to some of the other datasets, in every epoch, we run through the full fluorescence dataset but only run through an equal number of batches for the other datasets. This is done to avoid overfitting on the fluorescence data's training set - if we use the same scheme as pretraining, we would run through the fluorescence data multiple times in an epoch which might cause overfitting. We jointly train for a maximum of 50 epochs but if the average Spearman's rank correlation coefficient of the fluorescence data's validation set does not improve for 5 epochs, we stop training.
8. Fine-tuning is performed for a maximum of 50 epochs but if the average Spearman's rank correlation coefficient of the fluorescence data's validation set does not improve for 5 epochs, we stop training. We again use a 1e-5 learning rate, 1e-4 weight decay, and 96 batch size.
9. Linear probing is also performed for a maximum of 50 epochs but if the average Spearman's rank correlation coefficient of the fluorescence data's validation set does not improve for 5 epochs, we stop training. We use a higher 1e-3 learning rate, 1e-4 weight decay, and 96 batch size.

D. Performance of training strategies on multiple splits of the fluorescence dataset

Table S.1 summarizes the prediction performances obtained using various training strategies when 5 different train, test, and validation splits are used.

Pretraining Tasks	Transfer Method	Jurkat		K-562		THP-1	
		r	ρ	r	ρ	r	ρ
-	Train from scratch	0.6389 ± 0.0036	0.5996 ± 0.0093	0.6152 ± 0.0082	0.6043 ± 0.0045	0.5672 ± 0.0131	0.4742 ± 0.0136
All RNA-Seq	Linear Probing	0.4974 ± 0.0180	0.4528 ± 0.0153	0.4949 ± 0.0184	0.4803 ± 0.0137	0.4497 ± 0.0206	0.3937 ± 0.0078
TF-binding	Linear Probing	0.4141 ± 0.0109	0.4491 ± 0.0155	0.4164 ± 0.0106	0.4607 ± 0.0161	0.3294 ± 0.0107	0.3532 ± 0.0101
Sharpr-MPRA	Linear Probing	0.5850 ± 0.0110	0.5596 ± 0.0101	0.5813 ± 0.0198	0.5887 ± 0.0093	0.5097 ± 0.0253	0.4597 ± 0.0100
SuRE MPRA	Linear Probing	0.6355 ± 0.0123	0.6053 ± 0.0112	0.6329 ± 0.0124	0.6302 ± 0.0072	0.5574 ± 0.0229	0.4901 ± 0.0154
All MPRA	Linear Probing	0.6454 ± 0.0097	0.6176 ± 0.0087	0.6415 ± 0.0118	0.642 ± 0.0085	0.5732 ± 0.0169	0.5033 ± 0.0083
All Tasks	Linear Probing	0.6247 ± 0.0165	0.5774 ± 0.0170	0.6235 ± 0.0136	0.595 ± 0.0150	0.5591 ± 0.0209	0.4692 ± 0.0197
All RNA-Seq	Fine-tune	0.6279 ± 0.0091	0.5885 ± 0.0106	0.6052 ± 0.0148	0.5916 ± 0.0074	0.5530 ± 0.0183	0.4701 ± 0.0133
TF-binding	Fine-tune	0.6138 ± 0.0148	0.5784 ± 0.0133	0.5940 ± 0.0134	0.5916 ± 0.0084	0.5327 ± 0.0181	0.4523 ± 0.0212
Sharpr-MPRA	Fine-tune	0.6358 ± 0.0127	0.6056 ± 0.0173	0.6158 ± 0.0195	0.6211 ± 0.0108	0.5610 ± 0.0228	0.4841 ± 0.0145
SuRE MPRA	Fine-tune	0.6635 ± 0.0122	0.6340 ± 0.0099	0.6510 ± 0.0116	0.6497 ± 0.0075	0.5931 ± 0.0221	0.5154 ± 0.0138
All MPRA	Fine-tune	**0.6814 ± 0.0106**	**0.6402 ± 0.0077**	**0.6684 ± 0.0108**	**0.6591 ± 0.0053**	**0.6179 ± 0.0193**	**0.5310 ± 0.0120**
All Tasks	Fine-tune	0.6610 ± 0.0153	0.6172 ± 0.0104	0.6542 ± 0.0172	0.6348 ± 0.0080	0.5988 ± 0.0219	0.5039 ± 0.0163
Test Set Replicate Concordance		0.7900 ± 0.0271	0.7348 ± 0.0116	0.7267 ± 0.0247	0.6875 ± 0.0093	0.6561 ± 0.0423	0.4987 ± 0.0133

Table S.1: Summary of prediction performances obtained using various training strategies. The averages and standard deviations are computed over 5 different models that are fit using 5 different train, test and validation splits of the fluorescence data.

E. Performance of models on pretraining or joint-training tasks

Table S.2 shows the performance of our pretrained and jointly trained models on all additional tasks used for pretraining or joint training (note that joint training is performed together with the experimental fluorescence dataset). r denotes the Pearson correlation coefficient and ρ denotes the Spearman's rank correlation coefficient between the predictions and targets, Acc stands for accuracy, and F1 stands for F1-score. For the RNA-Seq datasets (LL-100, CCLE and Roadmp), the metrics shown in the table are obtained by averaging the metrics over all cell types in the datasets. Similarly, for the TF-binding datasets, the metrics shown in the table are obtained by averaging the metrics over all ChIP-seq datasets. For the Shapr-MPRA dataset, note that we have 12 outputs per input sequence, 4 of them being predictions for the average promoter-driven expression (average over the two replicates) in K-562 and HepG2 cells when the input sequence is bundled with either a minimal TATA promoter (minP) or strong SV40 promoter (SV40P). The table shows metrics obtained using these 4 outputs. Our results are very similar to those obtained by Movva et al. (2019). For the SuRE MPRA dataset, the table shows the average accuracy in predicting the K-562 and HepG2 expression bins with the average being computed over the 4 individuals from whose genomes the sequences were extracted.

Tasks	Training Strategy	LL-100		CCLE		Roadmap		TF-binding		Sharpr-MPRA				SuRE MPRA	
		r	ρ	r	ρ	r	ρ	Accuracy	F1	K-562 minP ρ	HepG2 minP ρ	K-562 SV40P ρ	HepG2 SV40P ρ	K-562 Acc	HepG2 Acc
All RNA-Seq	Pretraining	0.4493	0.4390	0.4673	0.4711	0.4379	0.4413	-	-	-	-	-	-	-	-
TF-binding	Pretraining	-	-	-	-	-	-	0.9747	0.9746	-	-	-	-	-	-
Sharpr-MPRA	Pretraining	-	-	-	-	-	-	-	-	0.3369	0.2261	0.1350	0.2090	-	-
SuRE MPRA	Pretraining	-	-	-	-	-	-	-	-	-	-	-	-	0.4018	0.3435
All MPRA	Pretraining	-	-	-	-	-	-	-	-	0.2803	0.2100	0.1493	0.2223	0.4011	0.3429
All RNA-Seq	Joint Training	0.4612	0.4495	0.4754	0.4786	0.4385	0.4435	-	-	-	-	-	-	-	-
TF-binding	Joint Training	-	-	-	-	-	-	0.5154	0.4989	-	-	-	-	-	-
Sharpr-MPRA	Joint Training	-	-	-	-	-	-	-	-	0.1350	0.0903	0.0565	0.0868	-	-
SuRE MPRA	Joint Training	-	-	-	-	-	-	-	-	-	-	-	-	0.4018	0.3435
All MPRA	Joint Training	-	-	-	-	-	-	-	-	0.1259	0.1359	0.0775	0.1658	0.3649	0.3129

Table S.2: Prediction performance of models on the pretraining or joint-training tasks.

F. Additional Figures

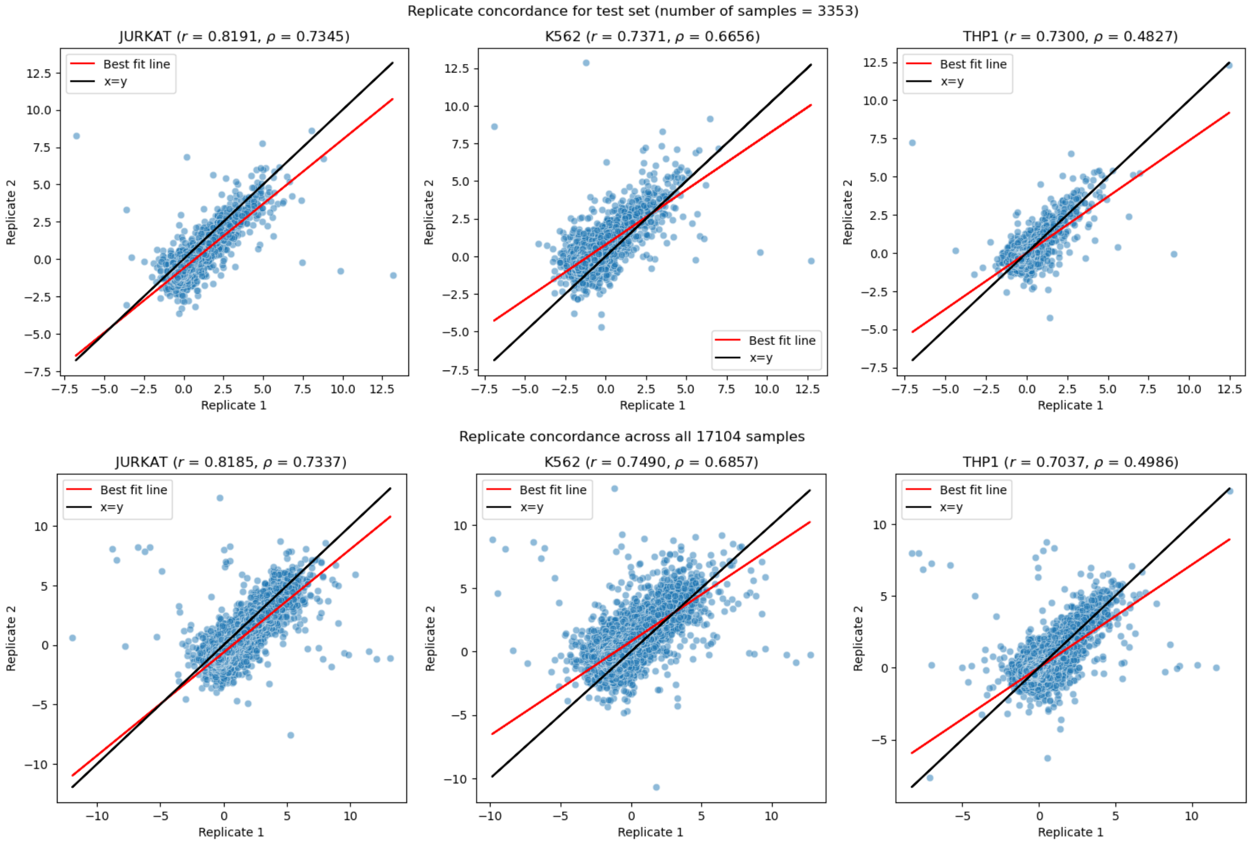

Figure S.1: We collect two replicate measurements of fluorescence and the scatter plots above show the correlation between these two measurements in each cell type (columns). The first row of plots shows the correlation for the test set used in Table 3 and the second row shows the correlation across all samples.

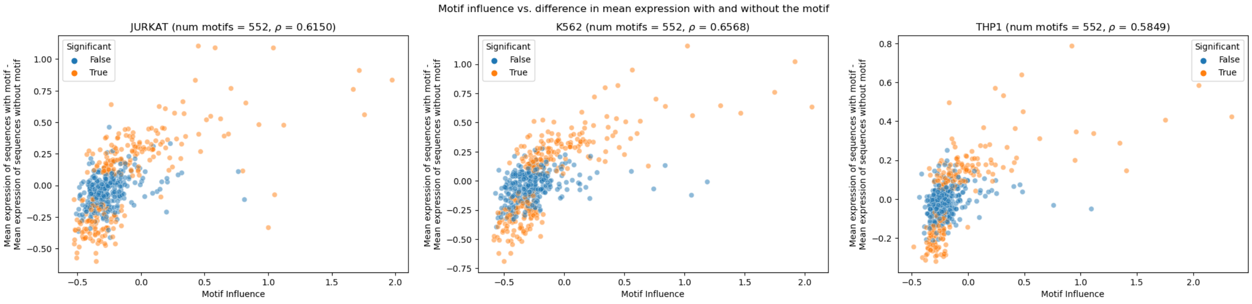

Figure S.2: Scatter plots showing the correlation between our motif influence estimates in each cell type and the changes in experimentally-measured expression in the presence of that motif. Significance of expression changes is computed using a t-test, and q-values < 0.05 are deemed significant (orange).