# Programmatic design and editing of *cis*-regulatory elements

Jacob Schreiber[1], Franziska Katharina Lorbeer[1], Monika Heinzl[1,2], Yang Young Lu[3], Alexander Stark[1], and William Stafford Noble[4,5]

[1]Research Institute of Molecular Pathology (IMP), Vienna BioCenter (VBC), Vienna, Austria

[2]Vienna BioCenter PhD Program, Doctoral School of the University of Vienna and Medical University of Vienna, Vienna, Austria

[3]School of Computer Science, University of Waterloo, Canada

[4]Department of Genome Sciences, University of Washington, Seattle, WA, USA

[5]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA

**Abstract**

The development of modern genome editing tools has enabled researchers to make such edits with high precision but has left unsolved the problem of designing these edits. As a solution, we propose Ledidi, a computational approach that rephrases the design of genomic edits as a continuous optimization problem where the goal is to produce the desired outcome as measured by one or more predictive models using as few edits from an initial sequence as possible. When applied across dozens of pre-trained machine learning models, we find that Ledidi can quickly design edits to precisely control transcription factor binding, chromatin accessibility, transcription, and enhancer activity across several species. By using several models simultaneously, Ledidi can programatically design edits that exhibit multiple desired

characteristics, and we demonstrate this capability by designing cell type-specific enhancers and accessible regions with controllable patterns of transcription factor binding. Finally, we introduce the concept of an affinity catalog, in which multiple sets of edits are designed that induce a spectrum of outcomes, and demonstrate the practical benefits of this approach for design tasks and scientific understanding.

# 1 Introduction

Recent advances in genome editing, driven by the discovery and development of the CRISPR-Cas9 system [1, 2, 3, 4] and subsequently base [5, 6, 7] and prime editing [8, 9, 10], have significantly reduced the cost and effort required to precisely modify genomic sequence within living cells. Consequently, genome editing is now routinely used in a variety of contexts, including engineering crops [11, 12], development of theraupeutics [13, 14], and scientific investigation [15, 16, 17, 18, 19]. To avoid unintended consequences and potential off-target effects, a desirable strategy in these applications is to make a relatively small number of edits to achieve a specific target function, e.g., altering gene expression or coding sequence.

Alongside these advances in editing techniques, a surge in the amount and quality of genomics data [20, 21, 22] has led to the development of increasingly sophisticated machine learning (ML) models [23, 24]. These models can make predictions for transcription factor (TF) binding [25, 26], histone modification, chromatin accessibility [27, 28, 29, 30], various measures of transcription [31, 32, 33], alternative splicing [34, 35], and more. Although these models differ substantially in architecture and precise output, a commonality is that they are sequence-based, in that they learn regulatory patterns encoded in DNA sequence to make predictions. Accordingly, these models have proven to be useful for identifying causal drivers among non-coding variants [36, 37, 25], and for elucidating *cis*-regulatory patterns [38, 39, 40] involving TF binding motifs and their orientation, affinity, spacing, and other higher-order syntax properties [41, 42] in leiu of running extensive experiments.

These ML models can also be used to design DNA with desired characteristics. For example, de Almeida *et al.* [43] and Taskiran *et al.* [44] designed enhancers exhibiting activity in targeted cell types in *Drosophila melangoster*. Typically, this design process involves two steps: first, the generation of candidate sequences and, second, the evaluation of these sequences using

the ML model (sometimes called an "oracle" model). In the second step, the most common evaluation is to measure the divergence between model predictions on the candidate sequences and a target outcome that is set beforehand, e.g., a precise amount of transcription. With respect to the first step, several strategies exist for generating candidate sequences. The simplest involves generating candidate sequences randomly [43]. Iterative strategies, such as *greedy substitution* and *motif implantation*, begin with random DNA and go through multiple rounds of candidate generation and evaluation, keeping the best candidate(s) from a round to use as the basis for the next round and continuing until the target outcome has been achieved [44]. More sophisticated versions of these iterative strategies, such as FastSeqProp [45], guide candidate sequences towards the target outcome using the gradient from the ML model. More recently, several types of generative ML models have been used to produce candidate sequences, including large language models (LLMs) [46], diffusion networks [47], and generative adversarial networks (GANs) [48]. We recommend De Winter, Konstantakos, and Aerts [49] as a comprehensive review of existing design methods.

Despite the algorithmic differences among these design methods, all of them focus on the creation of DNA sequences *de novo*. Although *de novo* creation provides great compositional flexibility, it is not always the most practical solution. In particular, base and prime editors enable the editing of genomes within living cells yet operate best when the number of edits needed is small. Even when one aims to design synthetic DNA constructs, starting with a region exhibiting many desirable characteristics according to available experimental measurements and minimally altering it to enhance these characteristics may be more robust than starting from scratch. In the *de novo* setting, model-based design methods can only be as good as the underlying model(s), but starting with a carefully chosen initial sequence allows designs to exhibit characteristics that the ML models do not make predictions for.

Accordingly, we propose Ledidi, a computational method for designing *edits* to an initial template sequence. Ledidi phrases this editing task as a continuous optimization problem, where the loss function being optimized is made up of two components: the deviation from the target outcome and a novel input loss, which counts the number of edits being proposed. By explicitly penalizing the number of edits, Ledidi implicitly limits unintended functional changes and off-target effects whose consequences may not be well captured by the model

being used for design.

We demonstrate that Ledidi is a powerful and flexible approach for designing genomic edits. As the generative step in the aforementioned two-step design process, Ledidi can flexibly turn any ML model into a designer of genomic edits. Across dozens of ML models and diverse forms of genomic activity, we find that Ledidi can quickly generate compact sets of edits yielding target outcomes. These edits create binding sites for TFs known to be associated with the respective processes and increase corresponding predictions from external ML models not used in the design process. Most existing design methods focus on a single characteristic, such as transcription in a single cell type. We demonstrate how Ledidi can design edits that optimize several desired characteristics simultaneously by using multiple ML models in the design process. Using this capability, we design accessible sites with controllable patterns of TF binding, cell type-specific enhancers, and show how multiple models of the same characteristic can be used together to design edits more robustly.

Finally, we leverage Ledidi's speed to show how repeating the design process across a fine-grained spectrum of target outcomes can be valuable. We call this set of designs an "affinity catalog". Affinity catalogs offer additional flexibility when the precise desired value for target outcomes are unknown beforehand, and provide a resource for understanding the full range of potential sequence edits and their relative strengths. By considering when sequence features are created or destroyed across a range of target values, one can better interpret and gain confidence in the proposed edits. Further, comparing the desired and achieved predictions across the entire catalog provides additional context as to the quality of the design that a single design cannot provide. We demonstrate that designed affinity catalogs for regulatory potential match well with experimental data, and that an affinity catalog for GATA2 binding reveals a sophisticated usage of a learned *cis*-regulatory code to achieve precise results.

# 2 Methods

## 2.1 Ledidi

Ledidi rephrases the task of designing discrete edits to an initial sequence as a continuous optimization problem (Fig 1). This problem has an objective function that is a mixture of

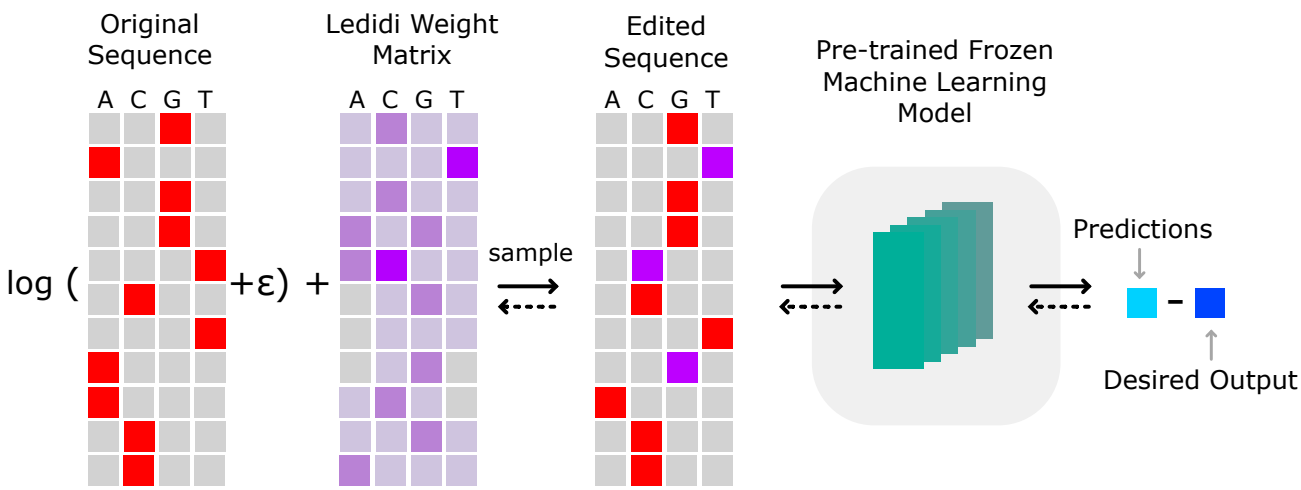


Figure 1: **A schematic of the Ledidi design process.** Ledidi turns the design of edits into a continuous optimization problem using straight-through Gumbel-softmax reparameterization. In this process, a Gumbel-softmax distribution is defined through the addition of the log of the initial sequence plus a small epsilon and the learned weight matrix (in purple). From this distribution, a discrete sequence, potentially containing edits (in purple), is generated and passed through a frozen model. A loss is then calculated between the predicted and desired outputs, and the straight-through estimator is used to pass the gradient through the discrete sequence and update the continuous weight matrix.

two terms: the first is the distance between the model output using the edited sequence $X$ and the desired output $\bar{y}$ (the output loss) and the second is the distance between the initial sequence $X_0$ and the edited sequence (the input loss). More formally, the original sequence is a one-hot encoded matrix $X_0 \in 0, 1^{n,d}$, where $n$ is the length of the sequence and $d$ is the size of the alphabet; the edited sequence is, similarly, $X \in 0, 1^{n,d}$; the model output using the edited sequence is $f(X) \in \mathbb{R}^{k,l}$, where $k$ is the length of the output sequence and $l$ is the number of output measurements per position, and the desired output is $\bar{y} \in \mathbb{R}^{k,l}$. The objective function for this optimization is

$$\min_{X} \texttt{output_loss}(f(X), \bar{y}) + \lambda \ \texttt{input_loss}(X, X_0) \tag{1}$$

and we refer to the final edited sequence from Ledidi as $\bar{X}$.

The input loss is weighted by the mixture weight $\lambda$ to, when small, encourage a closer match to the desired output and, when large, encourage a closer match to the initial sequence when large. When $\lambda$ is set to 0, Ledidi will design sequences that most closely match the desired

output without considering the number of edits proposed, as in FastSeqProp. Although any notion of distance could be used in each of the loss terms, we choose to use the L1 distance for the input loss corresponding to the number of edits made, and the L2 distance for the output loss:

$$\min_{X} ||f(X) - \bar{y}||_2^2 + \lambda \; ||X - X_0||_1 \tag{2}$$

Unfortunately, Equation 2 is difficult to optimize because $X$ is discrete, and continuous-valued gradients cannot be directly applied to one-hot encoded sequences. We overcome this challenge by carrying out the optimization with respect to a continuous weight matrix $W \in \mathbb{R}^{n,d}$ instead of over a one-hot encoded sequence. The matrix $W$ is initialized to all zeros. At each step, we calculate $W + log(X_0 + \epsilon)$ and use these values as logits such that each position in $n$ is an independent Gumbel-softmax distribution over the alphabet $d$ and a batch of one-hot encoded sequences are sampled from these distributions [50]. This sampling process is parameterized by a temperature $\tau$, sometimes referred to as the "temperature", that controls how close to the argmax the sampling is. As $\tau$ approaches 0, all sampled sequences resemble the most likely sequence in $W + log(X_0 + \epsilon)$, and as $\tau$ approaches infinity, sampled sequences resemble those that would be drawn uniformly at random. We have found that Ledidi is robust to the precise value of $\tau$, and so use a default value of 1.

With this new variable $W$ we can solve an alternate form of Equation 1:

$$\min_{X} ||g(W, X_0) - X_0||_1 + \lambda ||f(g(W, X_0)) - \hat{y}||_2^2 \tag{3}$$

where $g$ is the sampling process and $g(W, X_0)$ yields one-hot encoded sequences. This equation turns the process of designing edits into an optimization problem that can be solved with standard gradient descent methods. In this work, we use the Adam optimizer [51] with default settings except that the initial learning rate is set to 1.0.

Having an initial sequence enables several modifications to be made to the standard edit design task. First, one can specify a *mask* where edits cannot be made by setting the values of $W$ to $-\inf$ at positions where edits are not allowed. This mask can be adapted to block specific categories of edits, e.g., not allowing edits to result in "G"s, or made more precise to not allow a specific single ""C" to become a "G" but still allowing it to become a "T". Second,

*in-painting* can be performed by removing characters or spans from the initial sequence and having Ledidi learn what character should appear at those positions. Another way of viewing in-painting is as an extension of the original design task, but with edits at certain positions able to be made for free, making their usage preferential. Finally, *priors* can be encoded in $W$ with high values indicating the higher probability of that character being generated. These modifications can be seamlessly combined with each other and also with each other idea in this work without modification or additional burden on the user.

## 2.2 Greedy Pruning

Although Ledidi's optimization uses the input loss term to reduce the number of edits made, spurious or low-influence edits can be proposed due to the sampling procedure or early termination of the optimization. Accordingly, a greedy pruning step can be subsequently performed to undo edits that do not significantly improve the quality of the sequences being edited. First, this method sequentially undoes each edit individually in $\bar{X}$, yielding $\bar{X}_1$, and records the difference in output loss $D_{1,j} = \texttt{output_loss}(\bar{X}, \bar{y}) - \texttt{output_loss}(\bar{X}_{1,j}, \bar{y})$ for each edit $j$. At the end of each round $i$, if $\min_j D_{i,j} < T$ for some user-specified threshold $T$, edit $\operatorname{argmin}_j D_{i,j}$ is undone. This procedure repeats $i$ times, yielding $\bar{X}_i$ each time, until $\min_j D_{i,j} \geq T$, at which point the procedure terminates, yielding a sequence with $i - 1$ edits reverted.

## 2.3 Training BPNet Models

Our evaluations involve many publicly available, pre-trained models, but we supplemented these with several BPNet models trained specifically for this work [26]. For each model, BAM and narrowPeak files were downloaded from the ENCODE Consortium data portal (`https://www.encodeproject.org/`). 5' mapped read ends were pooled across replicates and then split across strands into two read count bigWig files using bam2bw (`https://github.com/jmschrei/bam2bw`). For the one chromatin accessibility data set, reads were not split across strands but instead kept as a single bigWig. Finally, we used bpnet-lite (`https://github.com/jmschrei/bpnet-lite`) to train BPNet models using standard settings (64 filters for TF binding and 256 filters for chromatin accessibility). These trained BPNet models, along with their training configurations, can be downloaded from `https://zenodo.org/records/`

14604495.

## 2.4 Design of TF Binding Sites

As an initial validation of Ledidi, we designed edits that increased predictions across several models in fourteen design settings. A challenge with setting a standard design task across models is that these models were trained on data with different read depths and processing approaches, and make predictions for different outputs using different loss functions. For example, Beluga [52] predicts peaks as a classification task and only makes one prediction per output, and Enformer makes predictions for many 128bp bins for each output. These differences mean that there was not always a clear equivalent single value to set as Ledidi's desired output. Accordingly, for models that predicted tracks (Borzoi and Enformer) we summed the signal across the output window, and for BPNet/ChromBPNet/ProCapNet models we use only the count head.

To account for differences in read depth and processing, we phrased the design task using percentiles, which would be agnostic to the underlying data distribution and comparable across settings. To get these percentiles, we chunked chr12 in hg38 into non-overlapping spans of the length expected for each model, removed all chunks that had *any* unknown characters, and made predictions using the model for the relevant task. From this distribution, we chose the 100 regions closest to the 50th percentile to use as our initial sequences, and tasked Ledidi with designing edits to bring them to the 99.5th percentile. Because the $\lambda$ parameter is sensitive to the magnitude of the predictions, we set it to be

$$\lambda = \frac{1}{100} \sum_{i=0}^{99} \frac{(\bar{y} - f(X_{0,i}))^2}{100} \tag{4}$$

which is the average Euclidean distance between the initial and desired outputs divided by 100. This 100 value is unrelated to the choice of using 100 regions closest to the 50th percentile but, rather, corresponds to wanting each edit to reduce the loss by at least 1%.

This process was then repeated for the 46 design settings used subsequently to evaluate motif compositional changes.

## 2.5 Affinity Catalog Design

Affinity catalogs are collections of edits derived from running a design method repeatedly on a fine-grained range of target outcomes. These outcomes can involve increasing or decreasing model predictions, and include outcome values that fall well outside the training distribution for the ML model. Although one may naturally associate "affinity" with only TF binding, these catalogs can be designed for any characteristic that a ML model can make a prediction for. They are named such based on the intuition that the binding of TFs drives many important processes including accessibility and transcription, and, accordingly, affinity of TF binding still drives fine-grained changes in these higher-order characteristics. No special technique is needed to create an affinity catalog other than running the design step repeatedly. We designed affinity catalogs in three settings.

In Section 3.3 we designed *affinity catalogs* for GATA2 binding, chromatin accessibility, and transcription initiation at the SMYD3 promoter. These catalogs were designed using BPNet, ChromBPNet, and ProCapNet models in K562 respectively. The initial sequence was the SMYD3 promoter centered on the transcription start site. To simulate a realistic setting where one would not want to modify the gene body, and because the gene is on the negative strand, we masked out the first 1,057bp of sequence and only allowed edits in the second half of the sequence. For the GATA2 catalog our designs spanned an increase in predictions of 0.2 to 10.0 and a step size of 0.2. For the accessibility and transcription initiation catalog, our designs spanned a decrease in 3.0 to an increase of 4.0 and a step size of 0.2. We note that, for the three models used, these predicted values correspond to log fold change read/fragment counts in the region. We set $\lambda$ using the same approach as when designing TF binding sites.

Next, we designed affinity catalogs for seven 8bp-long regions in two enhancers for S2 cells in the *Drosophila melanogaster* genome using a DeepSTARR model [53, 54]. For each of the sites, we proceeded by masking out the entire sequence except for the 8bp and performing in-painting on those 8bp, tasking Ledidi with designing 8-mers to obtain a range of activities for the entire enhancer. This process keeps the entire enhancer except the 8bp the same as it was initially (masking) and attempts to achieve intended enhancer-activity levels through only designing the 8bp-long region (in-paining). We then compared the chosen 8-mer for each desired output from DeepSTARR with the recorded experimental output from that 8-mer.

Other than the mask, we used default settings for Ledidi.

Finally, we designed an affinity catalog for yeast promoter activity using LegNet [55]. Although the DREAM Yeast Promoter Challenge [56] collected a massive amount of data by surveiling over 10 million promoters, this is small compared to the $4^{80}$ space of potential promoters of 80bp. Consequently, we could not simply design affinity catalogs and look up the corresponding experimental activity for that sequence. Rather, we chose a similar strategy as the 8-mer experiment above where we began with 10 randomly chosen promoter sequences and designed 8-mers in the middle of those sequences that yielded a range of activities. To validate this, we take the average experimental signal value across all promoters that contain that 8-mer in the exact same position.

### 2.6 *In silico* marginalizations

In several evaluations we use *in silico* marginalization to quantify the influence that a motif has on model predictions. These evaluations are performed by implanting a provided motif into the center of a set of background regions and recording the difference in model predictions before and after the implantation [57]. In our evaluations, we used 1,000 background sequences that were constructed by dinucleotide shuffling the initial sequence being used for design. They are referred to as marginalizations because averaging across many background sequences marginalizes out the influence of the surrounding context and returns just the influence of implanting the motif.

## 3 Results

### 3.1 Automatic design of regulatory DNA

Our first goal was to demonstrate that Ledidi can design edits that achieve desired characteristics *in silico* across a wide range of predictive models and forms of genomic activity with minimal changes to the initial DNA sequence. Accordingly, we paired Ledidi with ten pre-trained sequence-based ML models in 14 design settings. Each task involved driving one type of genomic activity, such as TF binding, chromatin accessibility, transcription, or massively parallel reporter assay (MPRA) activity (Supplementary Table 1). To account for technical

differences in the experimental readouts and the post-processing of these readouts when used in model training, we phrased this edit design problem as taking 100 regions closest to the 50th percentile of predicted signal and increasing them to the 99.5th percentile (see Methods for details). Put another way, we took regions that were not active but also did not display abnormally low signal and edited them to strongly exhibit the activity.

We observed that Ledidi's designed edits dramatically increased predicted activity in each of the 14 settings (Fig 2A). On average, Ledidi made 16.54 edits per sequence, driving the predicted activity from the 50th to the 99.27th percentile. Puffin was an outlier, requiring 69.09 edits on average yet only yielding predictions in the 97.54th percentile, whereas Enformer (CAGE) and DeepSTARR yielding the closest designs, at the 99.48th percentiles, using averages of 13.83 and 11.08 edits respectively. Beluga required the fewest edits, with only 4.86 edits to bring CTCF binding to the 99.45th percentile and 9.07 edits to bring MAX to the 99.39th percentile. When we applied a greedy pruning algorithm to these edits (see Methods for details), allowing the predictions to be lowered by no more than one percentile, we found that we could reduce the average number of edits by 39.4% (Fig 2B). Notably, this greedy procedure allows us to remove over half of the edits proposed when using Ledidi + ChromBPNet. These findings suggest that many of Ledidi's proposed edits may have very weak effects in an attempt to precisely match the desired target. On the other hand, in some settings the number of proposed edits was smaller than expected. For instance, we found that Ledidi + Beluga designed CTCF binding sites using $\sim$4 edits on average, despite the JASPAR motif (MA0139.1) [58] being 19bp long. Further invesigation revealed that this small number was possible because Ledidi targeted regions that already contained very weak affinity CTCF sites and made key edits predicted to enable binding (Fig 2C). Together, these findings show that Ledidi can design edits broadly across differing settings, but the number of edits will be influenced by the accuracy of the model, the complexity of the targeted genomic activity, and sequence features already present in the initial sequence.

A potential concern when performing model-based design is that the designs are based on learned statistical shortcuts [59] or spurious correlations [60] that are not biologically plausible. Accordingly, we evaluated whether seven TF binding sites designed using BPNet models elicited responses in matched outputs from external models not involved in the design process

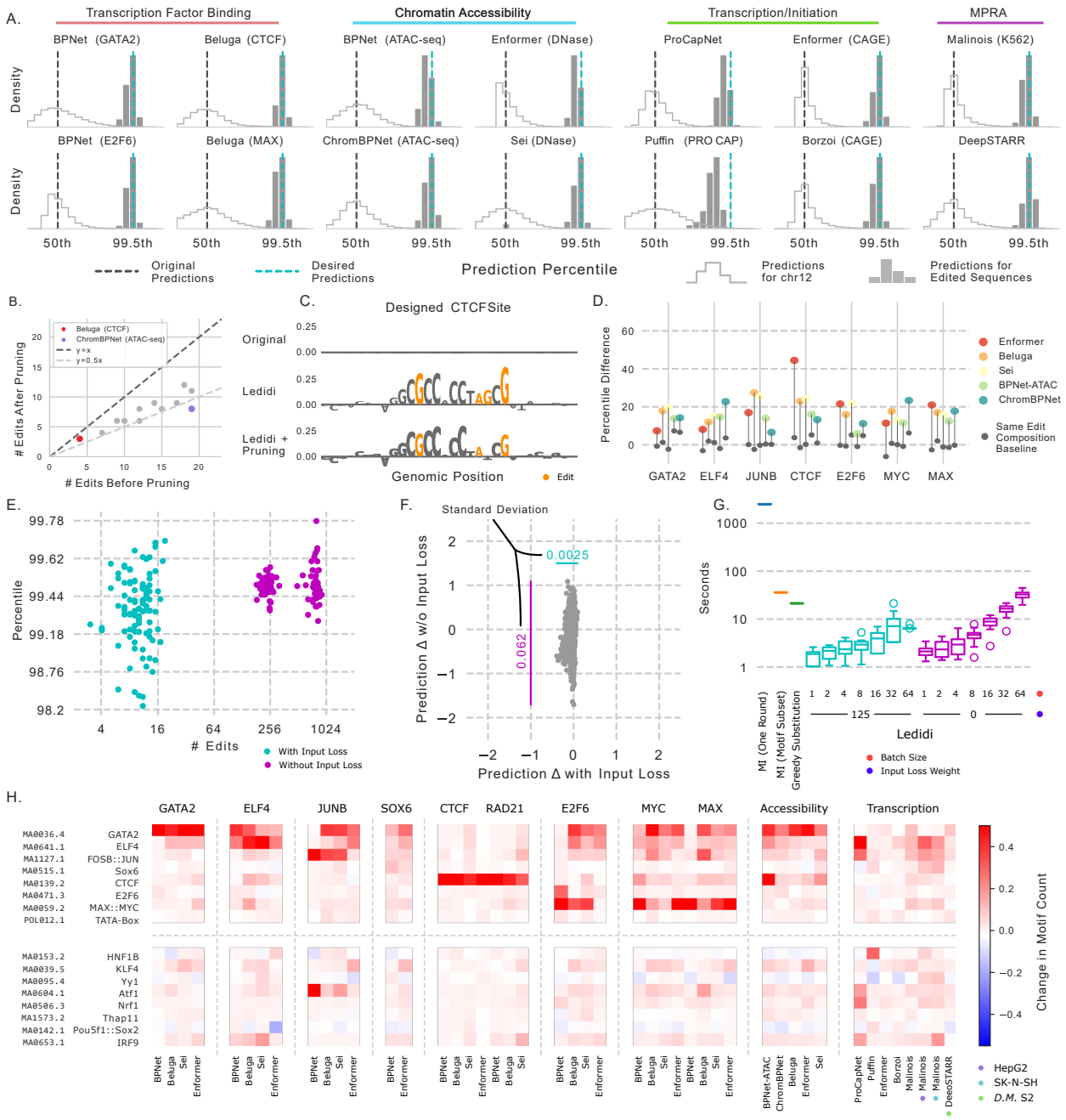


Figure 2: **Ledidi is a general-purpose designer of edits.** (A) Predicted activity of the edited sequences (filled grey histogram) in 14 design settings. (B) Average number of edits across the 100 edited sequences before and after pruning from the 99.5th percentile to the 98.5th percentile. (C) One example CTCF site with edits proposed by Ledidi shown in the colored characters, with characters sized according to their DeepLIFT/SHAP attributions. (D) Predictions from external models for sequences designed using BPNet models, with the grey dots representing a baseline of making the same composition of edits at random positions in the sequences. (E) The number of edits and prediction percentile when using an input loss (in cyan) and not using an input loss (magenta). (F) Differences in Enformer predictions across all 5,313 tasks when using or not using an input loss. (G) Timings for various design methods. (H) Change in the number of FIMO hits for select motifs after making the edits in 46 design settings.

(Fig 2D, see Methods for details). These external models included Enformer, Beluga, and Sei, which predict binding for the selected TFs directly, as well as BPNet-ATAC and ChromBPNet, which predict chromatin accessibility. In general, we expect that chromatin accessibility will increase as a consequence of the binding of these TFs. Ledidi's edits increase the predictions from these models by an average of 16.9 percentiles, with CTCF edits yielding the strongest response (+24.2 percentiles), and JunB edits yielding the weakest (+14.5 percentiles). These averages are driven by a strong response to CTCF edits by Enformer (+44.4 percentiles) and a particularly weak response from ChromBPNet to JunB edits (+6.5 percentiles). In contrast, when we apply the same *composition* of edits (i.e., number of A $\rightarrow$ C and T $\rightarrow$ G mutations, etc) randomly to the initial sequences, we find only a 0.7 percentile increase on average. In each setting, this baseline was weaker than the designed edits, demonstrating that Ledidi's edits are not based on shortcuts and do more than simply change nucleotide composition, e.g., GC content. Although the initial design task was to increase predictions from the 50th to the 99.5th percentile, we would not necessarily expect to see the same increase (49.5 percentiles) in other models due to differences in data processing and training strategy. However, this expectation and our findings suggest that subtle differences between learned regulatory rules may influence how generalizable the strongest designed sequences are.

One of our key hypotheses was that minimally editing an initial sequence would yield designs with fewer off-target effects. Accordingly, we next investigated the magnitude of these off-target effects when using or ignoring the penalty on making edits (the input loss). Here, we used a BPNet model trained to predict GATA2 binding because it is well understood and its binding site is relatively small. As expected, penalizing edits resulted in designs with significantly fewer edits ($\sim$9.8) than not penalizing edits ($\sim$360), but the predictions for the unpenalized designs more closely matched the desired 99.5th percentile (Fig 2E). Some sequences designed without a penalty made almost 1,000 edits to the 2,114bp initial template, nearly rewriting the entire sequence. This large number of edits substantially changed the GC content (40.6% to 44.0%). In contrast, designs that penalized edits only slightly changed GC content (40.6% to 40.8%). Although part of why so many edits were used is that many GATA binding sites were necessary, the primary reason is that without an input loss, there is no reason to revert neutral edits that arise due to the stochastic nature of the sampling

process. To better understand the global implications of these edits, we compared Enformer predictions in both cases for all 5,313 human outputs before and after making edits. We found that penalizing edits significantly limited the changes in prediction ($r = 0.99$ with, $r = 0.89$ without, Fig 2F). This demonstrates that designing sequences solely with a focus on one characteristic may inadvertently lead to substantial changes in other aspects, even if the design target is something conceptually simple like inducing GATA binding. However, penalizing edits as Ledidi does can mitigate the worst of these effects.

In practice, another important consideration for design methods is runtime. We compared Ledidi to several other methods, including Ledidi without the input loss, greedy substitution (sometimes called "evolutionary approaches"), and two motif implantation (MI) approaches where consensus sequences from motif databases are implanted at each possible position in the sequence. One MI approach considered the entire JASPAR2024 database, and one used only a set of 17 motifs thought to be relevant. We find that Ledidi is almost an order of magnitude faster than greedy substitution and several orders of magnitude faster than a comprehensive motif implantation method (Fig 2G). These timings match our intuition, because greedy methods involve repeatedly evaluating potential sequences individually, whereas using the gradient as Ledidi does enables multiple edits to be made simultaneously in each round.

Related to the risk of relying on statistical artficacts, another risk when using model-based design methods is that the design process pushes the models out-of-distribution. This risk is similar to how adversarial attacks on image classifiers can find individual pixels that change the image imperceptibly to the human eye but dramatically change model predictions [61]. Because many forms of genomic activity can be traced back to the influence of TF binding, we evaluated the plausibility of Ledidi's edits by quantifying the changes in the composition of these binding sites. In 46 design settings following our initial procedure, we subsequently used FIMO [62] to count the number of motif hits for a set of known relevant and manually chosen control motifs in the initial and the edited sequences (see Methods for details).

Overall, Ledidi's designed edits changed motif composition in intuitive ways. When increasing the predicted binding of GATA2, ELF4, JunB, CTCF, RAD21, MYC, and MAX, Ledidi's designs primarily relied on increasing the count of the corresponding motifs (Fig 2H). Unexpectedly, designed edits for SOX6 and E2F6 focus on adding GATA binding sites, with

E2F6 models also adding in MYC/MAX binding sites. Follow-up experiments showed that the experiments responded more strongly to the GATA consensus sequence than the E2F6 sequence, suggesting that the models have identified GATA as a pioneer factor necessary for altering accessibility prior to SOX6 and E2F6 binding. Edits increasing chromatin accessibility used several motifs but focused on GATA2 and CTCF, which are both known to be drivers of accessibility that are present in K562. Interestingly, edits increasing ELF4 binding increased motif counts for both ELF4 and GATA2, alluding to a known asymmetric relationship where GATA2 is upstream of ELF4 in hematopoietic stem cell fate decisions [63]. Encouragingly, despite the models differing in architecture, precise output, and training strategy, Ledidi's designed edits appear to change motif composition in fairly consistent ways, providing further evidence that the designed edits are generalizable. Overall, these findings indicate that Ledidi's edits achieve their desired effect by increasing the count of biologically plausible motifs and not simply by pushing the model out of distribution.

## 3.2 Designing edits driving multiple characteristics

Many applications of model-based design target only a single characteristic in a single cell type, even when a practical design goal may benefit from explicit control over multiple characteristics. Frequently, this is due to limitations in the model being used for design; a model that only predicts KLF4 binding cannot also evaluate the potential downstream transcriptional consequences of any proposed edits. Similarly, a model that makes predictions in only one cell type cannot alone be used to make cell type-specific designs. Sometimes, this is due to underspecified design objectives. Even when one characteristic is the primary focus, an unspoken preference is frequently that all other aspects of genomic activity are left unaffected, e.g., altering the binding of one TF but not disrupting the binding of others. Yet, as we saw when evaluating designs using Enformer (Fig 2F), even logically simple single-target objectives can yield edits with unintended side-effects.

As an explicit way to design edits that control multiple characteristics simultaneously, Ledidi can use multiple outputs from a multi-task ML model and/or multiple models in the design process. This is achieved by having Ledidi's output loss be composed of several terms, where each term comes from a different model or model output. Importantly, these terms do

not have to involve matching predictions to a specific target value, as we have done so far, and can involve any loss function long as the entire output loss remains diffentiable. The design process when using multiple models works similarly to before. At each step, Ledidi samples edited sequences from the Gumbel-softmax distributions, just as before. Then, these sequences are run through *each* model being used. A loss is calculated for each model according to the user-specified term, and these losses are aggregated into a single output loss whose gradient is used to update the Ledidi weight matrix (Fig 3A).

### 3.2.1 Accessibility driven by controlled TF binding

To demonstrate this multi-characteristic design concept, we edited a site to become more accessible than anywhere else in the human genome using controllable patterns of TF binding. As our initial sequence, we choose a GATA1 enhancer with regulatory activity reported to be very specific to erythroid cell lines and that ChromBPNet predicts is only moderately accessible in K562. When we tasked Ledidi with further increasing accessibility in K562 cells, a battery of BPNet models revealed that the edits increased predicted binding of GATA2, ELF4, and SOX6. Although these are known drivers of chromatin accessibility, they are also common in other cell lines, meaning that including their binding sites in the designed sequence may also diminish the specificity of this region. We therefore set out to rephrase the design task as one that accounts for TF binding in addition to accessibility (Fig 3B).

Accordingly, we used a design objective composed of three terms: the accessibility predictions from ChromBPNet to match as before, a reward for increasing predictions from the BPNet model for a chosen TF, and a penalty for increasing predictions from BPNet models for other TFs. Mathematically, this function is $||f_1(\hat{X}) - \bar{y}|| - f_2(\hat{X}) + \frac{1}{n-2} \sum_{i=3}^{n} f_i(\hat{X})$, where the ChromBPNet model is $f_1$, the BPNet model for the chosen TF is $f_2$, and the remaining BPNet models are $f_{3...n}$. Note that the second and third terms do not involve trying to match a target value, as our previous losses have, but on maximizing and minimizing those predicted values, respectively. An intuitive way to view these terms is that, among all possible sets of edits that could increase accessibility, the terms weight more favorably those that also increase predictions for the chosen TF without increasing any other TFs.

Applying this new objective interatively to each TF results in designs that *in silico* exhibit

uniform chromatin accessibility and controllable patterns of TF binding (Figure 3C). In each setting, we observe significantly stronger binding of the target TF than when designing edits that only increase accessibility, and we observe only weak predicted binding of off-target TFs. Although the strength of the predicted binding for the chosen TF is different in each of the 12 settings, this can be explained by each TF having a different magnitude of effect on chromatin accessibility. We then compare our three-term objective with two other approaches: a two-term objective that does not contain the penalty for off-target binding, and a two-term objective where the chosen TF is given a specific value to match rather than always rewarding higher binding. We find that in all three settings the predicted chromatin accessibility remains the same but that our three-term objective yields on-target TF binding predictions as strong as the formulation without the penalty term, while also yielding significantly lower predicted off-target binding than either other formulation (Figure 3D). Taken together, these results demonstrate the benefits of being preicse about all aspects of your design goals.

### 3.2.2 Designing cell type-specific enhancers

Building on previous work that designed enhancers for multiple *Drosophila melanogaster* cell lines individually [43, 44], we next consider multi-term objectives for Ledidi that are explicitly cell type-specific, in the sense that they are designed to exhibit activity in one cell type and not the other. Gosai *et al.* [64] considered a similar design task in three human cell lines using a single model; here, we use two separate DeepSTARR models trained as part of different research projects to predict readouts of regulatory potential in Schneider-2 (S2) and ovarian somatic cells (OSC), respectively (Fig 3E). We chose 100 initial sequences from each of five pools: regions (1) that do not exhibit regulatory potential as measured by STARR-seq in either cell type, those that exhibit potential in (2) S2 or (3) OSC cells alone, in (4) both cell lines, or (5) randomly generated sequences. Then, using Ledidi paired with the two DeepSTARR models, we first turned these initial regions into S2-specific enhancers and then, separately, turned them into OSC-specific enhancers using the min-gap loss from Gosai *et al.*

In each of the ten settings, we find that Ledidi is capable of designing elements that the DeepSTARR models predict to be specific to the desired cell line (Fig 3F). Interestingly, the OSC-specific designed elements yield significantly higher predictions than the S2-specific

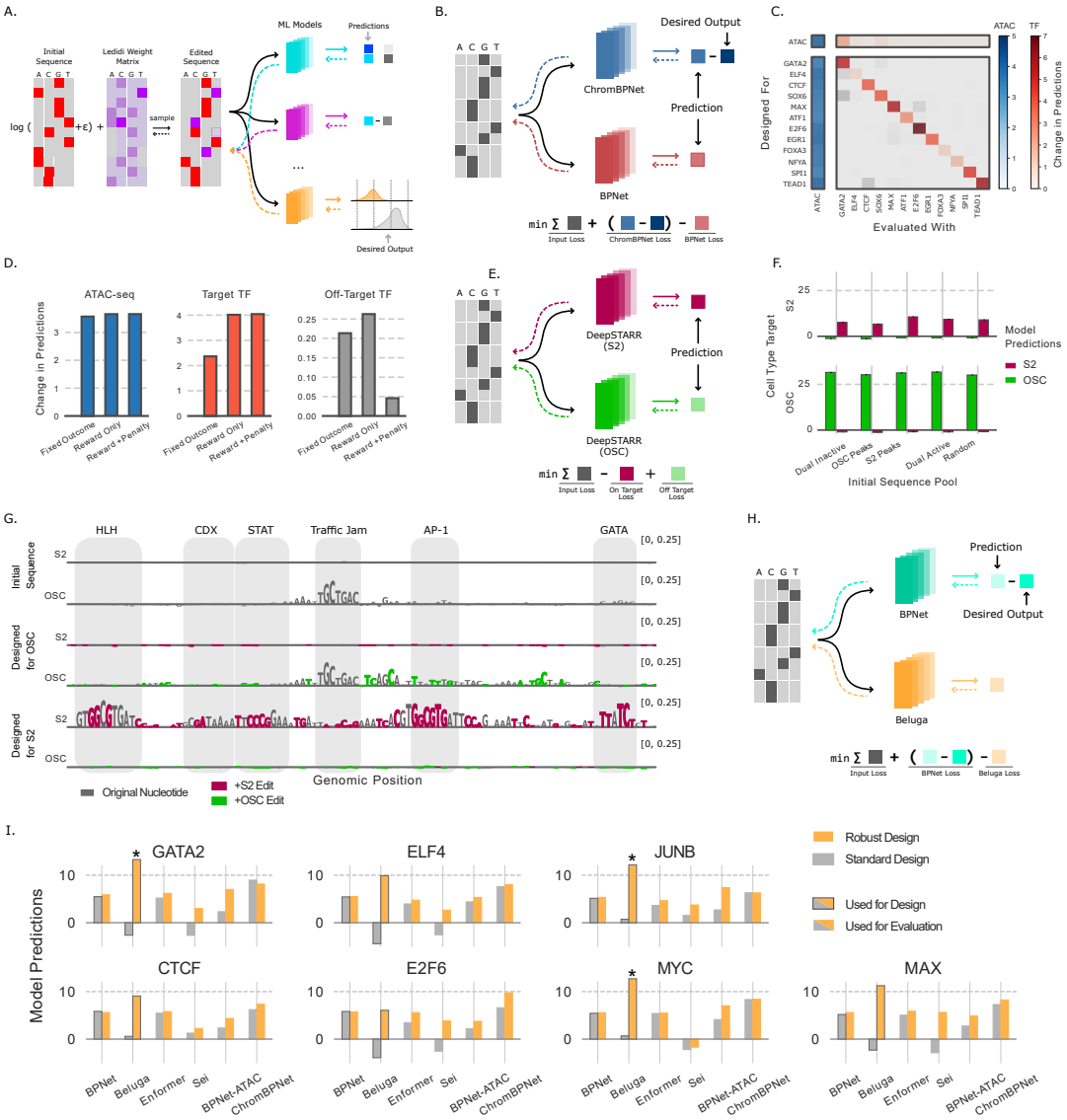


Figure 3: **Multi-term objectives enable fine-grained control of design.** (A) A schematic illustrating the use of several models and/or loss functions. (B) ChromBPNet and BPNet being used to design TF-driven accessible sites. (C) Predicted accessibility and TF binding across several design formulations. (D) Average predicted ATAC-seq (blue), on-target TF binding (red), and off-target TF binding (gray) aross the design settings in C. (E) S2- and OSC-specific DeepSTARR models being used for cell type-specific enhancer design. (F) Predictions from the models in E on the edited sequences. (G) DeepSTARR attributions for an OSC-specific peak that was subsequently edited to be a stronger OSC-specific peak or an S2-specific peak. (H) BPNet and Beluga being used together for more robust design. (I) Predictions are higher for external models when performing robust design, indicating more generalizable edits. Stars indicate that the bars continue past the visualized range.

elements, despite a similar dynamic range of predictions for these models. This difference potentially arises because the learned regulatory code in OSC has a higher impact-per-edit. However, these results suggest that, although choosing an initial sequence wisely has benefits, Ledidi can take even less suitable initial sequences and still guide them to meet the specified design criteria.

To better understand these designs, we considered feature attributions for a region that was initially an OSC-specific enhancer (Fig 3G). These attributions revealed that the initial OSC-specificity is due to a Traffic Jam (tj, sometimes referred to as MAFK) motif [65]. When tasked with making this region OSC-specific, Ledidi edits in a second tj motif next to the first motif, and edits in a weaker version of tj downstream. By focusing on this single motif, Ledidi is able to keep the region specific while also strengthening its predicted regulatory activity. In contrast, Ledidi uses a variety of motifs to make this same region S2-specific, including GATA, HLH, CDX, STAT, and AP-1. Importantly, to convert this OSC-specific enhancer into an S2-specific enhancer, Ledidi also uses a series of edits to remove the tj motif to ensure that not even a weak affinity version remains.

### 3.2.3 Achieving robust designs

Our multi-model designs so far have focused on settings where each model makes predictions for a different characteristic. Next, we investigate the inverse: can multiple models that predict the same target be used for design? We hypothesized that using multiple such models in the design process will yield more robust designs because the designs are less able to overfit to the parameters of any one model.

To test this hypothesis, we designed binding sites for seven TFs using the same settings and models as our initial cross-model validation (Figure 1F). We began by repeating the same procedure as before, where we used BPNet to design the edits and them using five other models. Then, we repeated this process but used BPNet and Beluga model together to design edits (Fig 3H). In both settings, we used the same target value for BPNet. When also using Beluga, we did not set a target value but, rather, followed the procedure we use in Sec 3.2.1 to reward edits that increase its predictions.

We find that the edits designed using both models are more robust and yield stronger

responses from the held-out models than those designed using BPNet alone (Fig 3I). Across TFs, Enformer and Sei have their predictions increased by 2.54 on average and the BPNet-ATAC and ChromBPNet models have their predictions increased by 1.701 on average. The largest increase in predictions came from Sei predictions of MYC, which increased by 8.50 from a negative value to a strong positive value. The only two settings where using Beluga led to worse outcomes was ChromBPNet with GATA and JunB, where the predictions were already very strong. Unsurprisingly, we find that the Beluga predictions are significantly increased when using Beluga in the design process. In each setting, we see that BPNet's predictions are similar whether Beluga is used or not. This finding reinforces the point that there are many paths to achieve a desired design objective and that multi-term objectives can push the design towards a path that has other desirable properties; thus, using multiple terms does not require sacrificing on any of them.

## 3.3 Affinity catalogs reveal how Ledidi uses cis-regulatory logic

Having shown that Ledidi is capable of quickly designing edits that achieve a desired objective, we next sought to answer three questions: (1) how precisely can Ledidi match a specified design target value, (2) can designs be made that exhibit activity more extreme than observed in the genome, and (3) what sequence features are being used to do so? We answer these questions by introducing the concept of an *affinity catalog*, where designs are generated for a fine-grained range of target values — potentially, both increasing and decreasing predictions — rather than focusing on a single value, as in our designs thus far. These catalogs provide useful context for a given design by illuminating where models are most likely to be successful, demonstrating how subtle aspects of the *cis*-regulatory code are being used by Ledidi to achieve precise values, and providing flexibility to researchers who may not know what specific value to design toward. Although any design method can be used to create an affinity catalog in theory, the efficiency of Ledidi makes it feasible to generate the large number of designs required for these catalogs.

We used Ledidi to design affinity catalogs for GATA2 binding, chromatin accessibility, and transcription initiation at the promoter of SMYD3 (see Methods for details) using a BPNet, ChromBPNet, and ProCapNet model respectively. These represent three very different genomic processes driven by increasingly sophisticated *cis*-regulatory features. In each case,

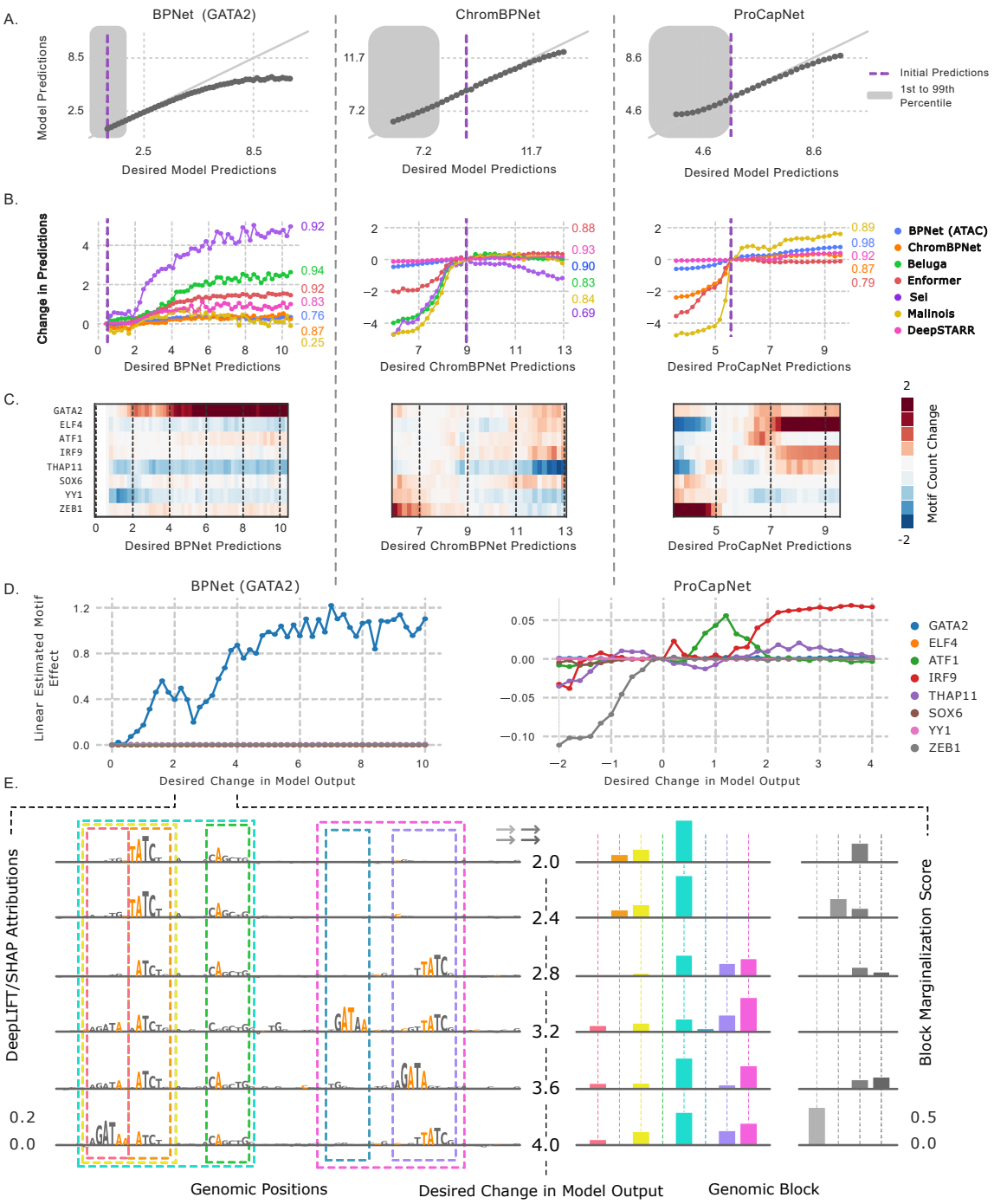


Figure 4: **Affinity catalogs reveal subtle principles of *cis*-regulation.** (A) Affinity catalogs for GATA2 binding, chromatin accessibility, and transcription initiation. At each step, design was performed 25 times and the results for A-D are averaged across those. (B) External validation of these affinity catalogs using a battery of models, showing the change in their predictions across the catalog. (C) Changes in motif counts across the catalogs. (D) Expected change in model predictions, obtained by multiplying the marginalized effect of each motif by the count of the motif. ELF4 has been excluded from the ProCapNet panel due to its overwhelming influence. (E) Proposed edits at one heavily-edited region for a slice of the affinity catalog. Bars on the right indicate the prediction delta when marginalizing out the respective colored box on the left panel. The four gray arrows indicate four downstream binding sites that are measured by marginalizations but whose attributions are not visualized.

Ledidi successful produced predictions that closely matched the desired fine-grained continuum of activity (Fig 4A). Perhaps not surprisingly, we observed that this agreement diminished on the edges of the catalogs, with the most notable disagreements occurring at very high desired levels of GATA2 binding. We initially hypothesized that this divergence arises because Ledidi cannot make designs more extreme than those observed in the genome; however, when we consider the distribution of genomic predictions, we see that designs do not diverge far past the 99th percentile in each of the three settings (grey shaded regions in Figure 4A). Upon further investigation, we found that the divergence is an optimization artifact caused by the balance between the input and output loss. Because each edit imposes a cost, Ledidi will begin by making edits that individually yield the strongest effects. Once such edits have been exhausted, each step toward more extreme GATA2 binding requires, on average, more edits to achieve. Eventually, the number of required edits to increase predictions comprises a larger portion of the loss than is made up for by more closely matching the desired output. Consequently, when we change the mixture weight (see Methods for details), we observe that down-weighting the input loss yields better agreement in the extremes, whereas up-weighting the input loss causes divergence to happen sooner. This finding indicates that affinity catalogs may also be useful in practice for setting the mixture weight correctly, to ensure that this divergence does not happen near where the target value is.

As discussed in Section 3.2.3, a potential risk when designing edits inducing activity more extreme than seen in the genome is that the edits may overfit to artifacts in the model being used for design and not actually yield activity that strong *in vivo*. Accordingly, we extended our previous cross-model validation to the three affinity catalogs using models that should both directly (through a prediction matched on the exact target) and indirectly (through prediction of a related form of activity) respond to the designed edits. When validating our catalogs using Beluga, Sei, and Enformer, we use the output most closely matching that of the designer model.

First, we validated the GATA2 catalog using Beluga, Enformer, and Sei (Pearson correlations of 0.94, 0.92, and 0.92 respectively), which all directly predict GATA2 binding. Then, we used ChromBPNet and BPNet-ATAC (correlations of 0.87 and 0.76, respectively), which predict chromatin accessibility, and Malinois and DeepSTARR (correlations of 0.25 and 0.83,

respectively), which predict enhancer activity. We found generally high correlations in the GATA2 catalog except for the Malinois model, with the strongest agreement coming from the three models that directly predict GATA2 binding (Fig 4B). We observe a slightly different trend for the chromatin accessibility and transcription initiation catalogs. In these catalogs, most models show strong monotonic alignment in the portion of the catalog where activity is decreased relative to the initial sequence, but then plateau or even decrease in the portion of the catalog corresponding to increasing targets. Despite this general trend, the models whose targets are most related to the model being used for design (BPNet-ATAC for chromatin accessibility, Malinois for transcription initiation, and to a lesser extent DeepSTARR in both settings) show strong agreement in both portions of the catalog. Taken together, these results indicate that Ledidi is finding generalizable edits even when designing out-of-distribution elements.

Next, we considered how motif composition changes across the catalogs. Unsurprisingly, we found that the GATA2 catalog primarily involves increasing the number of GATA2 sites (Fig 4C). In contrast, the accessibility catalog involved multiple trends, with ZEB1 binding sites being added to decrease accessibility and THAP11 being removed to increase accessibility. Additionally, in this catalog, we found that GATA2, ELF4, ATF1 and IRF9 sites are used interchangeably because each can increase accessibility and hence on average exhibit weak usage. Potentially, the most nuanced findings are in the transcription initiation catalog. Here, ELF4 binding sites are clearly the primary driver, and ZEB1 sites can reduce transcription initiation, likely through their influence on accessibility. However, several other weaker patterns also emerge, such as the temporary use of AF1 sites to induce small increases in activity before transitioning back to ELF4, the use of IRF9 to slightly boost signal, and the cyclic patterns of GATA2 and THAP11 usage, potentially to fine-tune the specific levels of initiation. These patterns indicate not only that the affinity catalogs are based on the activity of known motifs, but suggest that Ledidi is leveraging subtle learned *cis*-regulatory patterns to match the precise desired output.

Although considering changes in motif composition is useful for validating the edits independently of a predictive model, these changes in motif composition do not necessarily influence model predictions. Consequently, we supplement this analysis by multiplying the change in

motif count by the marginalized effect of that motif on model predictions (see Methods for details), yielding the linear estimated motif effect (LEME, Fig 4D). LEME scores for the GATA2 catalog indicate that the GATA2 motif is the exclusive driver of model predictions, despite slight changes in motif composition observed in Fig 4C, potentially indicating that edits are being made to convert binding sites for irrelevant TFs into GATA2 binding sites. Indeed, THAP11 motifs (GGACTACA, MA1573.1 in JASPAR) can be converted into GATA motifs (GGATAACA, MA0036.2 in JASPAR), including strong flanking context, in only two edits. When considering LEME scores for transcription initiation, many key patterns are preserved, including the temporary use of ATF1, repression from ZEB1, and the cyclic patterns of THAP1, but the actual influence of GATA seems to be minimal.

Because LEME captures only motif count and marginalized influence on model predictions, any deviation in trend from the model predictions must arise from other aspects of *cis*-regulatory logic such as affinity, spacing, presence of co-factors, and flanking context. Accordingly, we examined more closely the dip in LEME between the desired output of 2 and 4 in the GATA2 affinity catalog (Fig 4D). For each of the steps in this window, we calculated attribution scores for the designed sequences and considered a set of partially overlapping *in silico* marginalizations meant to measure non-linear interactions between motifs in the design (Fig 4E).

When focusing on the most densely edited area of the sequence, we find that subtle aspects of the *cis*-regulatory code have been used by Ledidi to precisely match targeted values (Figure 4E). To achieve an increase in predictions of 2.0, a GATA motif (yellow) has been edited in with a TAL co-factor motif (green) where neither one alone drives predictions significantly but, together (cyan), act non-linearly to drive predictions alongside a downstream motif not visualized (gray). To increase predictions by only 0.4 to 2.4, the downstream motif is weakened and a second downstream motif is induced. This weakening is necessary because only adding in the second downstream motif would yield predictions greater than the target of 2.4. When increasing further by 0.4 to 2.8, a second GATA site is added in (purple) but the original GATA motif (yellow) must be weakened for a similar reason. By considering the attributions and marginalizations within this range, we can see that aspects of spacing and affinity are carefully leveraged to precisely match desired values.

## 3.4 Experimental validation of an affinity catalog

A challenge in experimentally verifying affinity catalogs using publicly available experimental data is that the potential design space grows exponentially with length. This means that even when a large number of sequences have been assayed, any particular design is unlikely to have been observed. Consequently, we considered two settings where a constrained version of the design task could be validated.

Reiter *et al.* [53] used STARR-seq to investigate the influence of known TF binding sites on overall enhancer activity by considering all possible 8-mers at seven locations across enhancers (ced6 and ZnT63C) in *Drosophila melanogaster* S2 cells. Alongside this experimental data, they provide a DeepSTARR [54] model that predicts STARR-seq activity (the same model as used in Section 3.2.2 for S2 cells). As a constrained version of the design task, we paired Ledidi with this DeepSTARR model to design affinity catalogs for each of these seven locations, masking out all positions except for the 8-mer being considered. Because their experimental data contains all possible 8-mers and their respective strengths, we can simply look up the experimental signal for each designed 8-mer and compare that to the desired output from Ledidi.

At each of the seven sites, we observe strong correlations between the affinity catalog design targets and the experimental signal, ranging between 0.893 and 0.925. The ced6 enhancer contained four sites: two GATA sites known to drive enhancer activity and identified by DeepSTARR as being relevant, and two background sites (CACA and GAGA) are included as negatives (Fig 5A). As expected, the minimal experimental signal at these negative sites was higher than at the two GATA Sites (Fig 5B) because even irrelevant 8-mers do not disrupt GATA's activity. Across all four sites, there is a flattening of the experimental data on the sides of the affinity catalogs. Similarly to the affinity catalogs of Sec 3.3, this flattening is because the desired outputs are more extreme than the experimental readouts. When we consider the portion of the catalog between the 1st and 99th percentile of DeepSTARR predictions across all 8-mers, we observe significantly higher correlations across all four sites. We then repeated this process for the three binding sites in the ZnT63C enhancer, where all three motifs are active drivers (Fig 5C). We observe the same shape and trends, with a flattening on both sides outside the dynamic range of the predictions (Fig 5D). Taken together, these results indicate

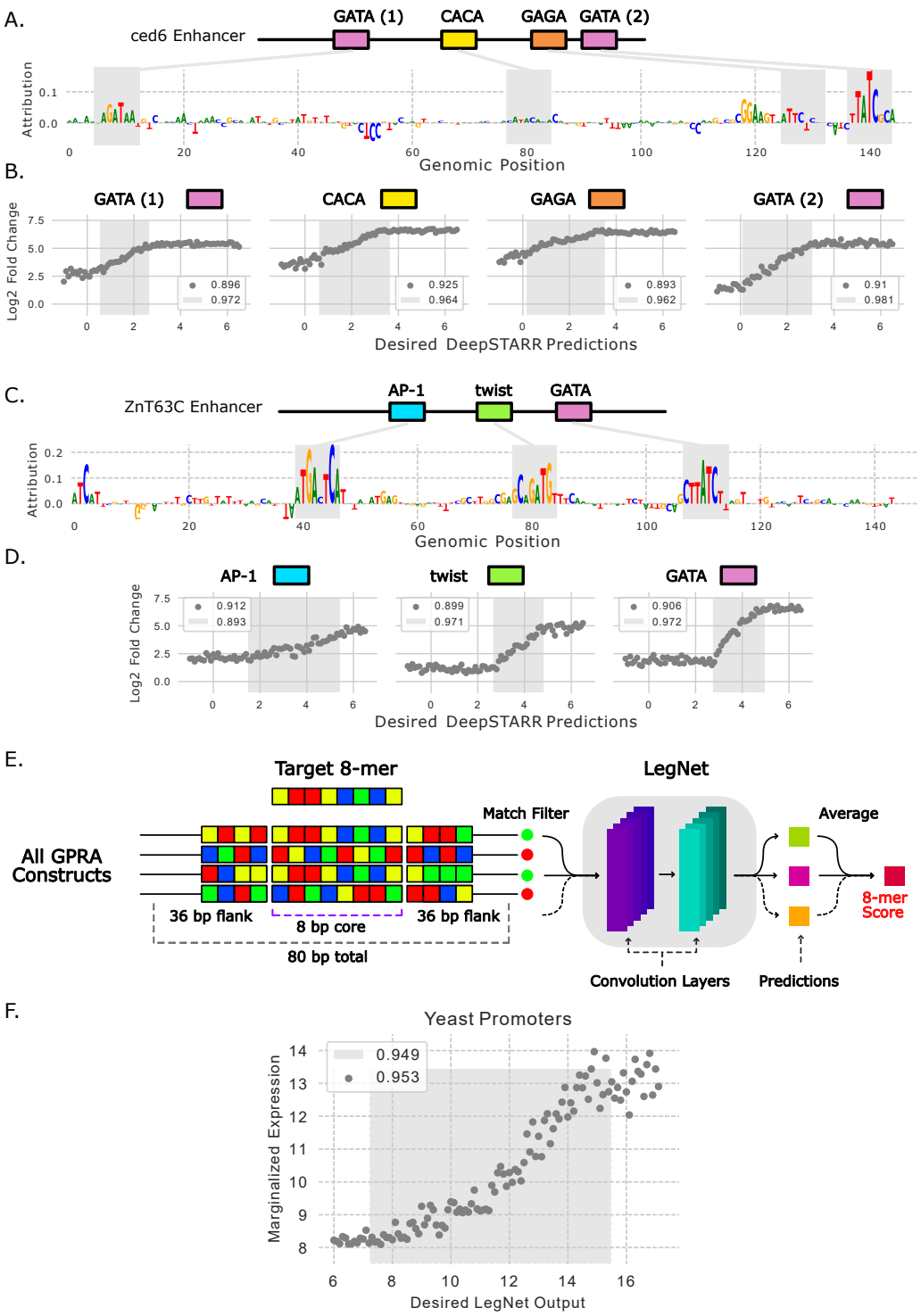


Figure 5: **Experimental validation of affinity catalogs.** (A) Four sites chosen to be comprehensively evaluated at the ced6 enhancer and attribution scores derived from a DeepSTARR model applied to the endogenous sequence. (B) Affinity catalogs designed for each of the four sites with the 1st and 99th percentile of DeepSTARR predictions on endogenous sequence outlined in gray. (C/D) The same as A/B but for the ZnT63C enhancer. (E) A schematic of the procedure to calculate marginalize 8-mer scores on the GPRA data from the DREAM Yeast Promoter Challenge using the LegNet model. (F) An affinity catalog of designed 8-mers comparing the desired LegNet predictions versus the marginalized 8-mer scores.

that Ledidi is capable of designing affinity catalogs whose subtle increases are confirmed by experimental signal but that affinity catalogs are most accurate within the dynamic range of the training data.

As a second experimental evaluation of affinity catalogs, we considered the DREAM Yeast Promoter Challenge [56]. This challenge was a competition aimed at developing ML models to predict gene expression in *Saccharomyces cerevisiae* using promoter sequence alone, with LegNet [55] being the winning submission. Data for this challenge were collected by a gigantic parallel reporter assay (GPRA) that screened more than 10 million randomly generated 80 bp promoters. However, despite this massive amount of data, we still could not evaluate our affinity catalogs without constraints because 10 million is only a small fraction of the $4^{80}$ potential promoters. To circumvent this sparsity, we adopted a similar strategy to our first setting and used Ledidi + LegNet to design affinity catalogs of 8-mers positioned within the center of the 80 bp promoters (see Methods for details). We evaluated these catalogs by comparing the desired output from LegNet with the average experimental signal across sequences that contained the same 8-mer at the same positions (Fig 5E) and observed a strong correlation of 0.953 (Fig 5F). Although there is flattening on either end of the affinity catalog that is likely driven by the same factors as the other catalogs, this effect cannot be fully explored due to hard limits on the range of predictions from LegNet and restricting our window between the 1st and 99th percentile only slightly reduces correlation to 0.949.

# 4 Discussion

In this work we introduced Ledidi, a computational method that turns any sequence-based ML model into a designer of edits. We demonstrated Ledidi's capabilities and flexibility by applying it out-of-the-box to dozens of models to design edits that increase TF binding, chromatin accessibility, transcription, and enhancer activity, across several species. Ledidi is significantly faster than inference-based design methods because each gradient step conceptually answers the question "What edits should be made from this point?" rather than systematically asking "Does this edit help?" A key conceptual difference between Ledidi and other design approaches is that Ledidi begins with an initial sequence and minimally edits it to induce desired properties, rather than performing *de novo* design. Consequently, a component of the most successful

designs will be choosing a good initial sequence, especially when the desired properties are complex and not fully captured by a single ML model. When the optimal choice is not immediately clear, we recommend using collections of public genomics datasets to find regions that already exhibit many of the desired characteristics [66]. When there truly is no reasonable initial sequence, one can adopt the strategy from the *de novo* settings and use randomly generated or randomly selected sequences.

A frequent challenge when designing regulatory elements is that one wants to precisely control multiple characteristics simultaneously. In these settings, Ledidi naturally accommodates multi-term design objectives, where each term can come from a different output from the same model or even from separate models that have been trained in different research settings. This point is important for several reasons. First, because models are usually developed with a specific goal in mind, the state-of-the-art model for one characteristic, e.g., MPRA activity, may not also make state-of-the-art predictions for other characteristics, e.g., TF binding, that you care about. Second, when a new assay is developed or an old assay applied in a new setting, it is likely infeasible to retrain and comprehensively evaluate massive models, such as Enformer, with this relatively small amount of additional data included. However, training a light-weight specialist model for these new experiments would be easy and can be done without affecting the performance of the massive model. Finally, even when a multi-output model makes predictions for all the characteristics one cares about, some outputs may be less accurate than others, even to the point where they cannot be trusted for design. In these settings, replacing the defective outputs with a light-weight specialized model may be necessary.

We anticipate that designing affinity catalogs will be broadly valuable in practice, even when one only needs a single design with a specific target outcome. Sometimes, the exact value to design against is not obvious *a priori* and having a catalog of potential designs is useful to contextualize the choice. By considering at which target values edits are introduced, one can evaluate the reliability of the edit or the induced motif. Conversely, these catalogs can reveal which edits are induced solely to meet a specific value, e.g., by switching to suboptimal affinities or spacings. Finally, by considering the full curve comparing desired values to predicted values, one can easily determine whether a model is being asked to do something too extreme in a way that considering a single design cannot.

Although increasing model complexity can improve predictive performance, this complexity can also significantly slow down methods like Ledidi. For example, we found that BPNet using an A100 SXM4 40G GPU usually takes less than five seconds to design edits increasing TF binding in one initial sequence, but can take several minutes to do the same with Enformer. These findings hold even when considering predictions for the same experiment and limiting the receptive field of Enformer from 197 kb to only 2 kbp. This slowdown can come from two sources: using a model that considers sequences much longer than needed for your design task, and using a model with slow internal units such as transformers. Accordingly, we suggest that designers consider inference cost when choosing which model is best for their setting, and that model developers consider the cost of inference as a benchmark when proposing new models.

This work has focused on the use of supervised ML models, but there are two ways one can use genomic language models (GLMs) with Ledidi. First, because design can be viewed as an "attack" on the oracle model, design methods can sometimes be prone to creating very unnatural sequences. Ledidi's input loss implicitly mitigates this problem by limiting the number of proposed edits, but does not completely correct it. A more explicit way to address this problem is by using a GLM as one term in a multi-term objective, where the GLM evaluates how "genome-like" each edited sequence is. Second, one can use a two-step design process where a GLM is used to design batches of initial sequences that are then subsequently refined using Ledidi. This process may be most valuable when an initial sequence is not readily available from the genome or when working with a species that does not yet have extensive experimental validation that can be used to guide selection.

An important note is that model-based design methods such as Ledidi are only as good as the underlying oracles. We strongly recommend that the first step of design, whether using Ledidi or some other method, is to rigorously validate the ML models used, even when they have already been published. This validation should begin with checking predictive performance, but should then continue to more sophisticated analyses such as considering *in silico* marginalizations and attributions to ensure that the model responds appropriately to biologically relevant sequence features. When the ML model relies on statistical artifacts, or the training data quality is so poor that overall performance measures are uninformative, then design is impossible. After verifying the quality of the model and using it to design sequences,

we recommend validating that the designs exhibit the desired characteristics using external models not used in the design process, checking motif composition, and applying any other available domain knowledge.

**Acknowledgements** We thank Vincent Loubiere, Nikolaus Mandlburger, and Jay Lou for useful discussions and feedback. Research at the Institute of Molecular Pathology (IMP) is supported by Boehringer Ingelheim GmbH and the Austrian Research Promotion Agency (FFG, FO999902549). For the purpose of Open Access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

### Author contributions

| | JS | FKL | MH | YL | AS | WN |
|---|---|---|---|---|---|---|
| Conceptualization | • | | | • | | |
| Data curation | • | • | | | | |
| Formal analysis | • | | | | | |
| Funding acquisition | • | • | • | | • | • |
| Investigation | • | | | | | |
| Methodology | • | | | • | | |
| Project administration | • | | | | • | • |
| Software | • | | • | | | |
| Resources | • | • | • | | | |
| Supervision | • | | | | • | • |
| Validation | • | • | • | | | |
| Visualization | • | | | | | |
| Writing—original draft | • | | | | | |
| Writing—review & editing | • | | | | | • |

# References

[1] Andrew V Anzalone, Luke W Koblan, and David R Liu. Genome editing with CRISPR-cas nucleases, base editors, transposases and prime editors. *Nat. Biotechnol.*, 38(7):824–844, July 2020.

[2] F Ann Ran, Patrick D Hsu, Jason Wright, Vineeta Agarwala, David A Scott, and Feng Zhang. Genome engineering using the CRISPR-Cas9 system. *Nat. Protoc.*, 8(11):2281–2308, November 2013.

[3] Tianxiang Li, Yanyan Yang, Hongzhao Qi, Weigang Cui, Lin Zhang, Xiuxiu Fu, Xiangqin He, Meixin Liu, Pei-Feng Li, and Tao Yu. CRISPR/Cas9 therapeutics: progress and prospects. *Signal Transduct. Target. Ther.*, 8(1):36, January 2023.

[4] Ayman Eid, Sahar Alshareef, and Magdy M Mahfouz. CRISPR base editors: genome editing without double-stranded breaks. *Biochem. J.*, 475(11):1955–1964, June 2018.

[5] Elizabeth M Porto, Alexis C Komor, Ian M Slaymaker, and Gene W Yeo. Base editing: advances and therapeutic opportunities. *Nat. Rev. Drug Discov.*, 19(12):839–859, December 2020.

[6] A light-inducible RNA base editor for precise gene expression. *Nat. Biotechnol.*, pages 1–2, March 2025.

[7] Nicole M Gaudelli, Alexis C Komor, Holly A Rees, Michael S Packer, Ahmed H Badran, David I Bryson, and David R Liu. Programmable base editing of A•T to G•C in genomic DNA without DNA cleavage. *Nature*, 551(7681):464–471, November 2017.

[8] Zhihan Zhao, Peng Shang, Prarthana Mohanraju, and Niels Geijsen. Prime editing: advances and therapeutic applications. *Trends Biotechnol.*, 41(8):1000–1012, August 2023.

[9] Joss B Murray, Patrick T Harrison, and Janine Scholefield. Prime editing: therapeutic advances and mechanistic insights. *Gene Ther.*, 32(2):83–92, March 2025.

[10] Peter J Chen and David R Liu. Prime editing for precise and highly versatile genome manipulation. *Nat. Rev. Genet.*, 24(3):161–177, March 2023.

[11] Kutubuddin A Molla, Simon Sretenovic, Kailash C Bansal, and Yiping Qi. Precise plant genome editing using base editors and prime editors. *Nat. Plants*, 7(9):1166–1187, September 2021.

[12] Sanskriti Vats, Jitesh Kumar, Humira Sonah, Feng Zhang, and Rupesh Deshmukh. Prime editing in plants: prospects and challenges. *J. Exp. Bot.*, 75(17):5344–5356, September 2024.

[13] Lauren C Testa and Kiran Musunuru. Base editing and prime editing: Potential therapeutic options for rare and common diseases. *BioDrugs*, 37(4):453–462, July 2023.

[14] Jessie R Davis, Samagya Banskota, Jonathan M Levy, Gregory A Newby, Xiao Wang, Andrew V Anzalone, Andrew T Nelson, Peter J Chen, Andrew D Hennes, Meirui An, Heejin Roh, Peyton B Randolph, Kiran Musunuru, and David R Liu. Efficient prime editing in mouse brain, liver and heart with dual AAVs. *Nat. Biotechnol.*, 42(2):253–264, February 2024.

[15] Gabriella E Martyn, Michael T Montgomery, Hank Jones, Katherine Guo, Benjamin R Doughty, Johannes Linder, Ziwei Chen, Kelly Cochran, Kathryn A Lawrence, Glen Munson, Anusri Pampari, Charles P Fulco, David R Kelley, Eric S Lander, Anshul Kundaje, and Jesse M Engreitz. Rewriting regulatory DNA to dissect and reprogram gene expression. *bioRxivorg*, page 2023.12.20.572268, December 2023.

[16] Colin K W Lim, Tristan X McCallister, Christian Saporito-Magriña, Garrett D McPheron, Ramya Krishnan, M Alejandra Zeballos C, Jackson E Powell, Lindsay V Clark, Pablo Perez-Pinera, and Thomas Gaj. CRISPR base editing of cis-regulatory elements enables the perturbation of neurodegeneration-linked genes. *Mol. Ther.*, 30(12):3619–3631, December 2022.

[17] Päivi Pihlajamaa, Otto Kauko, Biswajyoti Sahu, Teemu Kivioja, and Jussi Taipale. A competitive precision CRISPR method to identify the fitness effects of transcription factor binding sites. *Nat. Biotechnol.*, 41(2):197–203, February 2023.

[18] Alina Vaitsiankova, Tanay Thakar, and Alberto Ciccia. Base-editing screens illuminate variant effects in human hematopoiesis. *Cell Rep. Methods*, 3(7):100541, July 2023.

[19] Matthew A Coelho, Magdalena E Strauss, Alex Watterson, Sarah Cooper, Shriram Bhosle, Giuditta Illuzzi, Emre Karakoc, Cansu Dinçer, Sara F Vieira, Mamta Sharma, Marie Moullet, Daniela Conticelli, Jonas Koeppel, Katrina McCarten, Chiara M Cattaneo, Vivien Veninga, Gabriele Picco, Leopold Parts, Josep V Forment, Emile E Voest, John C Marioni, Andrew Bassett, and Mathew J Garnett. Base editing screens define the genetic landscape of cancer drug resistance mechanisms. *Nat. Genet.*, 56(11):2479–2492, November 2024.

[20] ENCODE Project Consortium, Michael P Snyder, Thomas R Gingeras, Jill E Moore, Zhiping Weng, Mark B Gerstein, Bing Ren, Ross C Hardison, John A Stamatoyannopoulos, Brenton R Graveley, Elise A Feingold, Michael J Pazin, Michael Pagan, Daniel A Gilchrist, Benjamin C Hitz, J Michael Cherry, Bradley E Bernstein, Eric M Mendenhall, Daniel R Zerbino, Adam Frankish, Paul Flicek, and Richard M Myers. Perspectives on ENCODE. *Nature*, 583(7818):693–698, July 2020.

[21] Jennifer E Rood, Samantha Wynne, Lucia Robson, Anna Hupalowska, John Randell, Sarah A Teichmann, and Aviv Regev. The human cell atlas from a cell census to a unified foundation model. *Nature*, 637(8048):1065–1071, January 2025.

[22] IGVF Consortium. Deciphering the impact of genomic variation on function. *Nature*, 633(8028):47–57, September 2024.

[23] Gökcen Eraslan, Žiga Avsec, Julien Gagneur, and Fabian J Theis. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.*, 20(7):389–403, July 2019.

[24] Charlotte Capitanchik, Oscar G Wilkins, Nils Wagner, Julien Gagneur, and Jernej Ule. From computational models of the splicing code to regulatory mechanisms and therapeutic implications. *Nat. Rev. Genet.*, 26(3):171–190, March 2025.

[25] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods*, 12(10):931–934, October 2015.

[26] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.*, 53(3):354–366, March 2021.

[27] Anusri Pampari, Anna Shcherbina, Evgeny Z Kvon, Michael Kosicki, Surag Nair, Soumya Kundu, Arwa S Kathiria, Viviana I Risca, Kristiina Kuningas, Kaur Alasoo, William James Greenleaf, Len A Pennacchio, and Anshul Kundaje. ChromBPNet: bias factorized, base-resolution deep learning models of chromatin accessibility reveal cis-regulatory sequence syntax, transcription factor footprints and regulatory variants. *bioRxivorg*, page 2024.12.25.630221, January 2025.

[28] Kaelan J Brennan, Melanie Weilert, Sabrina Krueger, Anusri Pampari, Hsiao-Yun Liu, Ally W H Yang, Jason A Morrison, Timothy R Hughes, Christine A Rushlow, Anshul Kundaje, and Julia Zeitlinger. Chromatin accessibility in the drosophila embryo is determined by transcription factor pioneering and enhancer activation. *Developmental cell*, 58(19):1898, October 2023.

[29] Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nat. Genet.*, 54(7):940–949, July 2022.

[30] Yan Hu, Max A Horlbeck, Ruochi Zhang, Sai Ma, Rojesh Shrestha, Vinay K Kartha, Fabiana M Duarte, Conrad Hock, Rachel E Savage, Ajay Labade, Heidi Kletzien, Alia Meliki, Andrew Castillo, Neva C Durand, Eugenio Mattei, Lauren J Anderson, Tristan Tay, Andrew S Earl, Noam Shoresh, Charles B Epstein, Amy J Wagers, and Jason D Buenrostro. Multiscale footprints reveal the organization of cis-regulatory elements. *Nature*, 638(8051):779–786, February 2025.

[31] Kelly Cochran, Melody Yin, Anika Mantripragada, Jacob Schreiber, Georgi K Marinov, and Anshul Kundaje. Dissecting the cis-regulatory syntax of transcription initiation with deep learning. *bioRxivorg*, page 2024.05.28.596138, May 2024.

[32] 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.

[33] Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R Kelley. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *Genomics*, (biorxiv;2023.08.30.555582v1):2023.08.30.555582, September 2023.

[34] Tony Zeng and Yang I Li. Predicting RNA splicing from DNA sequence using pangolin. *Genome Biol.*, 23(1):103, April 2022.

[35] Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F McRae, Siavash Fazel Darbandi, David Knowles, Yang I Li, Jack A Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B Schwartz, Eric D Chow, Efstathios Kanterakis, Hong Gao, Amirali

Kia, Serafim Batzoglou, Stephan J Sanders, and Kyle Kai-How Farh. Predicting splicing from primary sequence with deep learning. *Cell*, 176(3):535–548.e24, January 2019.

[36] Gonzalo Benegas, Gokcen Eraslan, and Yun S Song. Benchmarking DNA sequence models for causal regulatory variant prediction in human genetics. *bioRxiv*, page 2025.02.11.637758, February 2025.

[37] Xiaoyu Wang, Fuyi Li, Yiwen Zhang, Seiya Imoto, Hsin-Hui Shen, Shanshan Li, Yuming Guo, Jian Yang, and Jiangning Song. Deep learning approaches for non-coding genetic variant effect prediction: current progress and future prospects. *Brief. Bioinform.*, 25(5):bbae446, July 2024.

[38] Ferran Muiños, Francisco Martínez-Jiménez, Oriol Pich, Abel Gonzalez-Perez, and Nuria Lopez-Bigas. In silico saturation mutagenesis of cancer genes. *Nature*, 596(7872):428–432, August 2021.

[39] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv [cs.AI]*, May 2017.

[40] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv [cs.CV]*, April 2017.

[41] Avanti Shrikumar, Katherine Tian, Žiga Avsec, Anna Shcherbina, Abhimanyu Banerjee, Mahfuza Sharmin, Surag Nair, and Anshul Kundaje. Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5.6.5. *arXiv [cs.LG]*, October 2018.

[42] Shushan Toneyan and Peter K Koo. Interpreting cis-regulatory interactions from large-scale deep neural networks. *Nat Genet*, pages 1–11, September 2024.

[43] Bernardo P de Almeida, Christoph Schaub, Michaela Pagani, Stefano Secchia, Eileen E M Furlong, and Alexander Stark. Targeted design of synthetic enhancers for selected tissues in the drosophila embryo. *Nature*, 626(7997):207–211, February 2024.

[44] Ibrahim I Taskiran, Katina I Spanier, Hannah Dickmänken, Niklas Kempynck, Alexandra Pančíková, Eren Can Ekşi, Gert Hulselmans, Joy N Ismail, Koen Theunis, Roel Vandepoel, Valerie Christiaens, David Mauduit, and Stein Aerts. Cell-type-directed design of synthetic enhancers. *Nature*, 626(7997):212–220, February 2024.

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

[46] Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. Designing realistic regulatory DNA with autoregressive language models. *Genome Res.*, 34(9):1411–1420, October 2024.

[47] Lucas Ferreira DaSilva, Simon Senan, Zain Munir Patel, Aniketh Janardhan Reddy, Sameer Gabbita, Zach Nussbaum, César Miguel Valdez Córdova, Aaron Wenteler, Noah Weber, Tin M Tunjic, Talha Ahmad Khan, Zelun Li, Cameron Smith, Matei Bejan, Lithin Karmel Louis, Paola Cornejo, Will Connell, Emily S Wong, Wouter Meuleman, and Luca Pinello. DNA-diffusion: Leveraging generative models for controlling chromatin accessibility and gene expression via synthetic regulatory elements. *bioRxivorg*, February 2024.

[48] Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K Speicher, Christoph S Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, Verena Siewers, Florian David, Jens Nielsen, and Aleksej Zelezniak. Controlling gene expression with deep generative design of regulatory DNA. *Nat. Commun.*, 13(1):5099, August 2022.

[49] Seppe De Winter, Vasileios Konstantakos, and Stein Aerts. Modelling and design of transcriptional enhancers. *Nat Rev Bioeng*, pages 1–16, February 2025.

[50] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv [stat.ML]*, November 2016.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv [cs.LG]*, December 2014.

[52] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.*, 50(8):1171–1179, August 2018.

[53] Franziska Reiter, Bernardo P de Almeida, and Alexander Stark. Enhancers display constrained sequence flexibility and context-specific modulation of motif function. *Genome Res.*, 33(3):346–358, March 2023.

[54] Bernardo P de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark. DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. *Nat. Genet.*, 54(5):613–624, May 2022.

[55] Dmitry Penzar, Daria Nogina, Elizaveta Noskova, Arsenii Zinkevich, Georgy Meshcheryakov, Andrey Lando, Abdul Muntakim Rafi, Carl de Boer, and Ivan V Kulakovskiy. LegNet: a best-in-class deep learning model for short DNA regulatory regions. *Bioinformatics*, 39(8):btad457, August 2023.

[56] Abdul Muntakim Rafi, Daria Nogina, Dmitry Penzar, Dohoon Lee, Danyeong Lee, Nayeon Kim, Sangyeup Kim, Dohyeon Kim, Yeojin Shin, Il-Youp Kwak, Georgy Meshcheryakov, Andrey Lando, Arsenii Zinkevich, Byeong-Chan Kim, Juhyun Lee, Taein Kang, Eeshit Dhaval Vaishnav, Payman Yadollahpour, Random Promoter DREAM Challenge Consortium, Sun Kim, Jake Albrecht, Aviv Regev, Wuming Gong, Ivan V Kulakovskiy, Pablo Meyer, and Carl G de Boer. A community effort to optimize sequence-based deep learning models of gene regulation. *Nat. Biotechnol.*, pages 1–11, October 2024.

[57] Amr M Alexandari, Connor A Horton, Avanti Shrikumar, Nilay Shah, Eileen Li, Melanie Weilert, Miles A Pufall, Julia Zeitlinger, Polly M Fordyce, and Anshul Kundaje. De novo distillation of thermodynamic affinity from deep learning regulatory sequence models of in vivo protein-DNA binding. *bioRxivorg*, May 2023.

[58] Ieva Rauluseviciute, Rafael Riudavets-Puig, Romain Blanc-Mathieu, Jaime A Castro-Mondragon, Katalin Ferenc, Vipin Kumar, Roza Berhanu Lemma, Jérémy Lucas, Jeanne Chèneby, Damir Baranasic, Aziz Khan, Oriol Fornes, Sveinung Gundersen, Morten Johansen, Eivind Hovig, Boris Lenhard, Albin Sandelin, Wyeth W Wasserman, François Parcy, and Anthony Mathelier. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, 52(D1):D174–D182, January 2024.

[59] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, November 2020.

[60] Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, and Aidong Zhang. Spurious correlations in machine learning: A survey. *arXiv [cs.LG]*, February 2024.

[61] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 23(5):828–841, October 2019.

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

[63] Alex J Tipping, Cristina Pina, Anders Castor, Dengli Hong, Neil P Rodrigues, Lorenza Lazzari, Gillian E May, Sten Eirik W Jacobsen, and Tariq Enver. High GATA-2 expression inhibits human hematopoietic stem and progenitor cell function by effects on cell cycle. *Blood*, 113(12):2661–2672, March 2009.

[64] Sager J Gosai, Rodrigo I Castro, Natalia Fuentes, John C Butts, Kousuke Mouri, Michael Alasoadura, Susan Kales, Thanh Thanh L Nguyen, Ramil R Noche, Arya S Rao, Mary T Joy, Pardis C Sabeti, Steven K Reilly, and Ryan Tewhey. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, 634(8036):1211–1220, October 2024.

[65] Michelle A Li, Jeffrey D Alls, Rita M Avancini, Karen Koo, and Dorothea Godt. The large maf factor traffic jam controls gonad morphogenesis in drosophila. *Nat. Cell Biol.*, 5(11):994–1000, November 2003.

[66] Nikoletta Psatha, Pavel Sova, Grigorios Georgolopoulos, Kiriaki Paschoudi, Mineo Iwata, Jordan Bloom, Tatyana Ulyanova, Hao Wang, Alexandra Kirtsou, Ninos-Ioannis Vasiloudis, Matthew S Wilken, John A Stamatoyannopoulos, Evangelia Yannaki, Thalia Papayanopoulou, George Stamatoyannopoulos, and Jeff Vierstra. Large-scale discovery of potent, compact and lineage specific enhancers for gene therapy vectors. *bioRxiv*, page 2023.10.04.559165, October 2023.

# Supplementary Materials

## Supplementary Tables

| Model | Assay Type | Target | Target Index | Input Length |
|---|---|---|---|---|
| BPNet | TF ChIP-seq | GATA2 | | 2114 |
| BPNet | TF ChIP-seq | E2F3 | | 2114 |
| Beluga | TF ChIP-seq | CTCF | 338 | 2000 |
| Beluga | TF ChIP-seq | MAX | 636 | 2000 |
| BPNet | ATAC-seq | | | 2114 |
| ChromBPNet | ATAC-seq | | | 2114 |
| Enformer | DNase-seq | | 625 | 2000 |
| Sei | DNase-seq | | 1490 | 4096 |
| ProCapNet | ProCap | | | 2114 |
| Puffin | ProCap | | 5 | 1000 |
| Enformer | CAGE | | 5111 | 2000 |
| Borzoi | CAGE | | 306 | 2048 |
| Malinois | MPRA | | 0 | 600 |
| DeepSTARR | STARR-seq | | 0 | 249 |

Supplementary Table 1: **Metadata of models used for initial design.** The type of machine learning model, assay the model was trained on, target of the assay (if applicable), output target used when the models are multi-task, and length of the input sequences used for each of the fourteen design settings. All design settings are in human K562 except for DeepSTARR which is in *Drosophila melanogaster* S2 cells.

# Supplementary Figures