Generating Regulatory Sequences with Cell-Type Specific Activity
CS 273B, Fall 2019

Foster Birnbaum
Department of Computer Science
Department of Biology
Stanford University
fosb@stanford.edu

Sawyer Birnbaum
Department of Computer Science
Stanford University
sawyerb@stanford.edu

Nicolai Ostberg
Stanford Center for Biomedical Research
Stanford Medicine
nostberg@stanford.edu

Josh Wolff
Department of Computer Science
Department of Bioengineering
Stanford University
jw1@stanford.edu

Abstract

Recent advances in the technology for engineering *de novo* DNA sequences raise the possibility of creating sequences that provide new or improved biological functions. One application of *de novo* sequence generation involves creating regulatory regions that impart cell-type specificity to downstream coding regions. This development could, for example, improve targeted gene therapy treatments for such diseases as diabetes and cancer and enhance cell-type specific protein function experiments. Here, we leverage the Basenji architecture, a deep-neural net architecture that provides state-of-the art performance in predicting cell-type specific regulatory activity from DNA sequences. We experiment with three techniques for sequence generation. First, we use a genetic algorithm to search for sequences that maximize activity in specific cell types, as determined by a pre-trained Basenji model. Second, we apply a GAN architecture to generate sequences that are similar to known ones with cell-type specific activation. Third, we apply gradient backpropagation on the pre-trained Basenji model and on our own Basenji-based model to construct sequences with high cell-type specific activation. Currently, these approaches do not generate sequences with any more cell-type specific activity than random sequences. We identify (1) a lack of relevant training examples and (2) the weakness of the Baseji model we used to guide our sequence generation approaches as the principal problems, and we propose improvements to the model and to our approaches. Our work thus highlights the limitations of *in silico* designing of *de novo* DNA sequences with cell-type specific activity and offers possible ways forward for the field.

1 Introduction

Improvements in the fidelity and cost of generating *de novo* DNA sequences have advanced the field of synthetic biology to the point that many seek to apply such sequences in industry and academic settings, including for drug development and basic research [1] [2]. Applying sequence synthesis techniques to the study of gene regulation is especially promising. Researchers can already use known regulatory motifs to construct regulatory networks with desired properties [3]. Ideally, though, instead of relying on known motifs, researchers could design novel regulatory sequences optimized for a desired task. For example, designing regulatory sequences with high cell-type specific activity could improve targeted gene therapy cancer treatments [4]. Current design principles rely on iteratively synthesizing, testing, and modifying sequences *in vitro*; while effective, this design-build-test cycle can be time-consuming and expensive [2]. Fortunately, while designing sequences with cell-type specific activity has proved challenging, deep learning-based techniques, such as the Basenji architecture, have made substantial progress on the related problem of predicting cell-type specific activity from DNA sequence alone [5]. Accordingly, we work to leverage the Basenji model to provide researchers with a more efficient way to generate cell-type specific regulatory sequences.

1.1 Basenji Architecture

The Basenji model provides state-of-the art predictions for cell-type specific epigenetic and transcriptional profiles based on DNA sequence alone. The multitask model operates on a one-hot encoded DNA sequence and applies a series of convolutional layers followed by a final dense layer and a softplus activation function to predict cell-type specific activity. Dr. Kelley, one of our mentors and a developer of the Basenji architecture, provided us with a pre-trained version of the architecture that operates on sequences of length 1,000 and includes 5 convolutional layers. The full model predicts on 131 kb sequences and includes 18 convolutional layers, but we used a smaller version because iterating on generation techniques for 131 kb sequences would have taken significantly more time and because Dr. Kelley advised us that smaller sequences contain enough information for the activity-prediction task. [5]

1.2 Related Work

Existing techniques for generating DNA sequences with high transcriptional activity focus on generative adversarial networks (GANs). Killoran et al. proposed a modified GAN model involving a generator that creates DNA sequences and a predictor that evaluates sequences for certain properties. They showed that their model, when using a predictor function that searches for known transcription factor binding motifs, creates at least some DNA sequences with the desired motifs. One advantage of this generator-predictor approach is its versatility: the predictor function can be defined to accomplish a variety of tasks. Indeed, Killoran et al. describe how their model could be applied to generate DNA sequences that preferentially bind to specific proteins in a transcription factor family, a related problem to designing sequences with cell-type specific activity.

However, a major flaw in Killoran et al.'s unsupervised learning approach to creating DNA sequences is that they do not demonstrate the biological relevancy of their generated sequences. This is especially concerning because no fundamental structure exists to DNA sequences (i.e., in any sequence, the four nucleotides can appear in any order any number of times), meaning unsupervised approaches may struggle to learn appropriate sequence grammars. For example, Killoran et al. noted that their generator sometimes outputs sequences with multiple copies of the same transcription factor motif in close proximity, which may not be biologically desirable. While Killoran et al. propose modifying their predictor function to incorporate grammatical structure, this approach would vastly increase the complexity of the predictor and would rely on domain-specific knowledge. Nevertheless, Killoran et al.'s work demonstrates the potential for a GAN-based approach to generate sequences with desired properties. [6]

1.3 Present Work

We apply the Basenji architecture to the problem of generating DNA sequences with cell-type specific activity in three ways. Our first approach involves using a genetic algorithm to construct sequences. We initialize these sequences around motifs we identified as associated with sequences with cell-type specific activity (i.e., "specific motifs") and used the pre-trained Basenji model to determine the fitness of each sequence when updating the population of sequences between generations. Our second approach involves applying a GAN architecture to generate new sequences similar to those with cell-type specific activity. To avoid the issues with unsupervised learning discussed above, we only provided real sequences with cell-type specific activity to enforce biologically relevant restrictions on the sequences. Our third approach involves training our own Basenji-based model (which we refer to below as the trained model) and generating sequences using backpropagation on the pre-trained model and on the trained model with a loss function designed to maximize activity in specific cell types and minimize activity in others. For these approaches, we evaluate generated sequences by measuring GC content to assess biological feasibility, by searching for the identified "specific motifs," and by using the pre-trained and trained Basenji models to assess cell-type specific activation.

2 Data

2.1 Sequences and Transcriptional Activity

Cell Type	# Sequences	% of Training Set
HeLa	532	0.052%
HepG2	944	0.092%
GM12878	1,773	0.173 %
K562	188	0.018 %

Table 1: The number of sequences with cell-type specific activity in the training set. The breakdown for the validation and testing sets are comparable. In this context, we define activity in one cell type as cell-type specific when it has a value greater than 10 and at least 10 times the max of the activity in any other cell type.

Dr. Kelley provided us with a subset of the training, validation, and testing data used to develop the Basenji model. This dataset consists of pairs of one-hot encoded sequences of 1,000 nucleotides and real-valued CAGE activation data across the HeLa, HepG2, GM12878, and K562 cell lines. CAGE (cap analysis of gene expression) involves capturing 5' mRNA sequences, using reverse transcription enzymes to create DNA fragments from those mRNA sequences, and sequencing the fragments to assess mRNA expression [7]. By measuring 5' mRNA expression, CAGE data manifests in peaks near transcription start sites, making these data potent markers for the regulatory activity of transcriptional regions. The HeLa, HepG2, and K562 cells are immortalized cancer cell lines originally collected from the cervix, liver, and blood, respectively. The GM12878 cells are derived from B-lymophcytes that have been immortalized through infection with Epstein Barr Virus [8]. These lines thereby represent a variety of tissues and cell lineages. The data was collected as part of the FANTOM5 project and are available to download using the following identifiers: CNhs12326 (HeLa), CNhs12328 (HepG2), CNhs12333 (GM12878), and CNhs12336 (K562) [9]. Kelley at al. performed a random split of the data to form the training, validation, and testing sets [5]. The split was not stratified in any way.[1] The dataset consists of over 2 million training sequences, approximately 500,000 validation sequences, and approximately 650,00 test sequences.[2] Despite the large amount of data, as Table 1 shows, very few sequences exhibit cell-type specific activity.

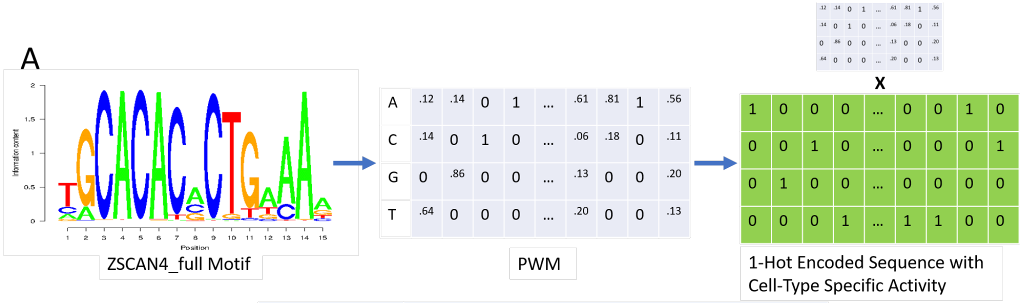
2.2 Known Motifs

We obtained a list of 843 known eukaryote transcription factor binding motifs from the MEME Suite motif database [11] [10]. The motifs range from 7 to 23 nucleotides in length, and the average length is 12.7 nucleotides. We selected this set of motifs at the recommendation of our mentors and because this set represents the majority of human transcription factors [10]. To identify motifs associated with cell-type specific activity, we selected training data sequences that satisfy the definition for cell-type specificity presented in Table 1. We also selected sequences with high specificity for the HeLa, HepG2, and GM12878 cell types combined (i.e., sequences that have an average activity across these cell types that is greater than 10 and is at least 10 times the activity in the K562 cell type). We used the the Find Individual Motif Occurrences (FIMO) tool from the MEME Suite to quantify the presence of each of the 843 known motifs in each of the selected sequences [12]. For each position in the input sequence, the FIMO tool calculates a log-likelihood ratio score for each motif representing the probability of the motif occurring at that position. This ratio is converted to a p-value, and if the p-value is less than 0.0001, the FIMO tool reports the presence of that motif at that position. We compared the lists of motifs generated by the FIMO tool for each class of sequence (i.e., sequences with specificity for each cell type individually and for the HeLa, HepG2, and GM12878 cell types combined) to identify motifs that appeared commonly in only one class. These are the "specific motifs" we classified as associated with cell-type specific activity. Figure 1

[1] Ideally, the split would have been designed to avoid leakage of training sequences into the test set by, for example, reserving whole chromosomes for the test test.

[2] To speed up training, we trained on only 1 million of the training examples and validated on only 300,000 of the validation examples.

displays this analysis pipeline and lists several of the most prevalent transcription factors in sequences with cell-type specific activity for each class.


A
Information content
0
0.5
1
1.5
2
Position
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ZSCAN4_full Motif
A .12 .14 0 1 ... .61 .81 1 .56
C .14 0 1 0 ... .06 .18 0 .11
G 0 .86 0 0 ... .13 0 0 .20
T .64 0 0 0 ... .20 0 0 .13
PWM
X
1 0 0 0 ... 0 0 1 0
0 0 1 0 ... 0 0 0 1
0 1 0 0 ... 0 0 0 0
0 0 0 1 ... 1 1 0 0
1-Hot Encoded Sequence with Cell-Type Specific Activity

B

Cell Type	Transcription Factor Binding Motifs
HeLa	ZSCAN4_full, Pou2f2_DBD_2, POU3F3_DBD_1
HepG2	TFAP2C_full_1, TFAP2C_DBD_1, TFAP2B_DBD_1
GM12878	MEF2D_DBD, RARA_full_2, EMX1_DBD_2
K562	FOXB1_DBD_2, ZNF740_DBD, MEOX2_DBD_3
HeLa, HepG2, and GM12878	KLF16_DBD, HINFP1_full_3, TFAP2A_DBD_4

Figure 1: Overview of mapping known motifs to cell-type specific activity. **A.** Statistical analysis pipeline to score how frequently each motif appears in sequences with cell-type specific activity [13]. **B.** The top three most selective motifs for each class.

3 Methods

3.1 Genetic Algorithm

For each cell type, we created a population of 100 DNA sequences of length 1,000. To do so, we randomly initialized 100 one-hot encoded length 1,000 sequences. For each sequence, we randomly selected 10 of the "specific motifs" associated with specific activity in that cell type and distributed these motifs uniformly throughout the sequence, ensuring the motifs did not overlap.

We found that repeating the following method 100 times to evolve each sequence was optimal:

1. randomly make a point mutation in the sequence;
2. evaluate the sequence on the pre-trained Basenji model;
3. calculate ΔL, where ΔL is the difference between the objective function (given by Equation 1, where A_i is the activity in cell type i and t is the target cell type) evaluated on the newly mutated sequence and the objective function evaluated on the un-mutated sequence;
4. if the sequence improved, i.e., $\Delta L > 0$, the mutation is accepted; and
5. if the sequence worsened or did not improve, i.e., $\Delta L \leq 0$, decide whether to accept or reject using the Metropolis-Hastings criteria. To do so, generate a uniformly random number between 0 and 1, α, and let u be the value given by Equation 2. If $\alpha \leq u$, accept the mutation. If $\alpha > u$, reject the mutation.

$$\mathrm{L} = \max_{c \neq t}(A_c) - A_t \tag{1}$$

$$u = e^{\Delta L} \tag{2}$$

We also experimented with the following modifications to the approach presented above: not seeding the initial sequences with "specific motifs"; rejecting mutations whenever $\Delta L \leq 0$ instead of applying the Metropolis-Hastings criteria; with Equation 5 as the objective function instead of Equation 1; with 100 iterations of mutations instead of 1,000; and with replacing sequences with low objective function scores with variations of ones with high objective function scores during each iteration.

3.2 GANs

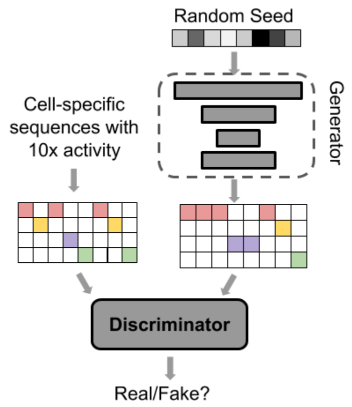


Figure 2: Overview of the GAN used to generate cell-type specific sequences. A multilayered convolutional generator network used a random seed to produce fake sequences. These sequences were then fed to a discriminator coupled with known cell-type specific sequences.

A typical GAN consists of a generator and a discriminator. The core concept of a GAN is that the data (in this case, DNA sequences) can be encoded into a real-valued, latent space. If the mapping between latent-space representation and real samples is known, new realistic samples can be generated easily. The generator and discriminator work together to learn this mapping. The generator attempts to create, from latent variables, synthetic samples that are indistinguishable from real samples. That is, it transforms a latent variable z to a sequence $G(z)$. The discriminator attempts to distinguish between the synthetic samples and real samples. For some sequence x, the discriminator assigns it a score of $D(x)$. The loss function presented in Equation 3 drives learning for the generator and the discriminator: the generator seeks to minimize the loss while the discriminator seeks to maximize it.

$$\mathrm{L} = \frac{1}{N}\sum_{i=1}^{N}\log(D(x_i)) + \frac{1}{M}\sum_{i=1}^{M}\log(1 - D(G(z_i))) \tag{3}$$

where N is the number of real samples and M is the number of generated samples.

We designed our GAN to address issues with using unsupervised learning to generate DNA sequences: as explained above, the unstructured nature of DNA makes discriminating synthetic samples from real samples challenging. Accordingly, the real data we inputted to our discriminator consisted only of DNA sequences with cell-type specific activity. In this way, the generator was incentivized not only to learn to include relevant transcription factor binding motifs but also to order the motifs in a biologically feasible way. For example, the generator should have been disincentivized from creating synthetic sequences with two copies of the same motif in close proximity because such situations occur rarely in real samples with cell-type specific activity, allowing the discriminator to separate synthetic sequences from real ones [6].

Our generator pushes randomly initialized vectors through four convolutional upsampling layers to create 1,000 base-pair-long generated sequences. The sequences are then fed into a discriminator that uses three convolutional layers with 0.3 dropout to classify the validity of the sequences. The model is trained using an Adam optimizer with a learning rate of 0.0001. The model runs for 20 epochs with a 256 batch size.

3.3 Backpropagation

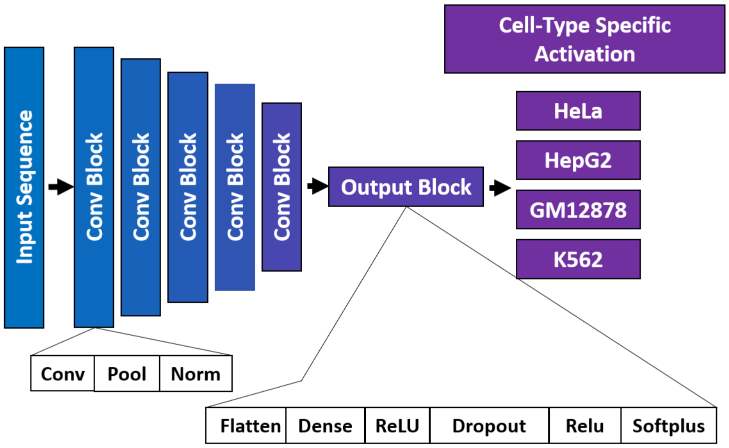


Figure 3: Our multi-task model architecture consists of five convolutional blocks with max pooling and batch-normalization followed by a flattening layer, a densely connected layer, a ReLU activation layer, a dropout layer, another densely connected layer, and a Softplus activation layer.

We applied backpropagation to the pre-trained Basenji model provided by Dr. Kelley and to our own implementation of the Basenji architecture that uses the same hyper-parameters as does Dr. Kelley's model. Figure 3 shows our model architecture. Our model differs from the provided model in several ways. First, we reduced the dimensionality of the model by flattening the activations and applying a dense layer rather than by applying a 1D cropping layer, which we thought could increase the learning capacity of the model. Second, we added dropout layers (with a dropout rate of 0.7) and L2 regularization on the dense layers (with a coefficient of 0.01) to reduce overfitting. Third, at the suggestion of Dr. Vikram, one of our mentors, we log-scaled the output data and used a root-mean-squared loss rather than a Poisson loss. Fourth, to address the disparity in frequency between sequences with cell-type specific activation and those without in our training dataset, we augmented the data by replicating sequences with high activation. Specifically, we added 50 copies of each sequence in which any cell type had an activation of more than 10 and 100 copies of each sequence with cell-type specific activation. Although these modifications provided a slight benefit for our model, it still heavily overfits.[3]

Figure 4 displays predictions on the test set for the pre-trained and trained models. Neither model does a good job predicting the activity of the high-activation sequences. (One reason we decided to try to train our own model was to hopefully achieve better performance than the pre-trained one.)

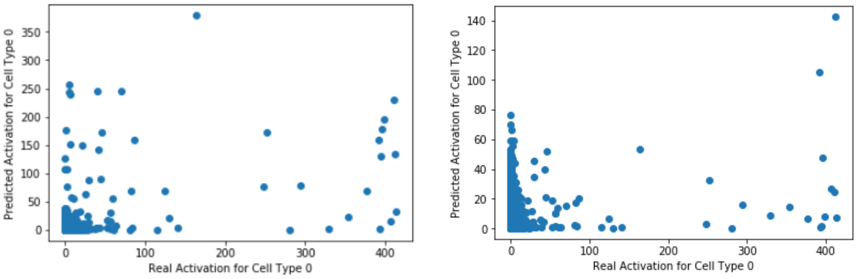


Figure 4: Correlation plots of real and predicted activations for HeLa cells (cell type 0) using the pre-trained model (left) and our trained model (right). Both models do a poor job predicting sequences with high activation. Performance is similar for the other three cell types.

$$\mathrm{L} = \left(\frac{1}{3}\sum_{c\neq t} A_c\right)^2 + \left(\frac{1}{A_t}\right)^2 - A_t \tag{4}$$

[3] We also experimented with adding TFiLM layers and residual connections but found that they did not improve performance [22].

Equation 4 presents the loss function we sought to minimize when running backpropagation on the models, where A_i is the activity in cell type i and t is the target cell type. This loss function is a slightly modified version of the following function:

$$\mathrm{L} = \big(\frac{1}{3}\sum_{c \neq t} A_c\big) - A_t \tag{5}$$

Note that minimizing the loss in Equation 5 corresponds to maximizing the activation in the target cell type while minimizing it in the other cell types. The same logic applies to Equation 4. Our modifications to Equation 5 serve to increase the penalty for predicting high activation in the non-target cell types and to increase the penalty for predicting activations less than 1 in the target cell type. Experimentally, we found that backpropagation with the simpler objective function tended to either (1) maximize the activation in all the cell types, (2) minimize the activation in all the cell types, or (3) quickly reach a broad local minimum in which activation in the K562 cell type was lower than the activation in the other cell types.

To introduce stochasticity into the sequence generation, we performed backpropagation 50 times with a different random normal initialization of the input sequence, where each input sequence is a 1,000-by-4 matrix. During each iteration, we maintained two sequences: a real-valued sequence and a one-hot encoded sequence. To generate the initial one-hot encoded sequence from the initial real-valued sequence, at each position, we selected the base corresponding to the real-valued element with the maximum value. Each iteration, we calculated the loss with respect to the one-hot encoded sequence but updated the real-valued sequence. Every 100 iterations, we checked the real-valued sequence to determine if the location of the maximum value in each row had changed and updated the one-hot encoded sequence accordingly. We chose this design because we noticed that while only maintaining a real-valued sequence would result in a sequence with the desired cell-type specific activity, translating that sequence to a one-hot encoded sequence after completing backpropagation resulted in a substantial loss of information, meaning the one-hot encoded sequence lacked the desired activity. By maintaining a one-hot encoded sequence throughout the backpropagation process and calculating the loss with respect to it, we ensured the model learned to generate a one-hot encoded sequence with the desired activity. During backpropagation, we used an Adam optimizer with a step-size of 0.0001. We updated for a minimum of 40,000 iterations.

4 Evaluation Metrics

We used three metrics to evaluate the sequences generated from each of the above approaches: we calculated the GC content to assess biological feasibility; we searched for the identified "specific motifs"; and we used the pre-trained and trained Basenji models to predict activity in each cell type.[4]

We noticed the pre-trained and trained Basenji models almost always adjusted the activity in the first three cell types (i.e., HeLa, HepG2, and GM12878) in lockstep. Indeed, according to our definition of cell-type specific activation (> 10 and > 10-times the max activation in other cell types), neither model ever predicts cell-type specific activity for any of the first three cell types. Accordingly, in addition to attempting to maximize activation in each cell type, we experimented with jointly maximizing activation in the first three types relative to the fourth (i.e., K562).

4.1 GC Content Analysis

We first verified the generated sequences' biologically feasibility. One important characteristic of DNA sequences is GC content: i.e., the percentage of the bases in the sequence that are guanine or cytosine. Although GC content varies greatly in different regions of the human genome, the average GC content across all chromosomes is 41%, and less than 10% of 100 kb fragments have a GC content in the 48-52% range [14]. We analyzed the distributions of GC content in our generated sequences to see if they matched the known distribution.[5]

[4] The ideal method for assessing the quality of the generated sequences would be to run *in vitro* experiments on the sequences; time constraints prevented us from pursuing this approach.

[5] GC content is, of course, only a crude proxy for biological feasibility. But it is also perhaps the simplest measure of sequence feasibility, and we expect that a model incapable of generating sequences with reasonable

4.2 Motif Analysis

We used the FIMO MEME Suite tool to begin to assess the cell-type specific activity of our generated sequences. For the sequences designed to have selective activity in each cell type, we searched for the appropriate "specific motifs." We quantify motif performance in two ways. First, we count the average number of times that any of the "specific motifs" appears in the set of sequences generated by each technique. Second, we compute the absolute value of the difference between the occurrence rate of individual "specific motifs" in the generated sequences compared to the training sequences with known cell-type specific activation (i.e., the sequences used to identify the "specific motifs"). The first metric provides insight into how often the "specific motifs" appear in our generated sequences, and the second provides insight into how closely the distribution of motifs in our generated sequences matches the distribution of motifs in the ones with cell-type specific activation.

4.3 Basenji Prediction

To better evaluate the cell-type specific activity of our generated sequences, we passed the sequences through the pre-trained and trained Basenji models. While the pre-trained Basenji model was used to determine sequence fitness in the genetic algorithm, this validation nevertheless provides an assessment of the effectiveness of the genetic algorithm to maximize cell-type specific activity according to a set definition of fitness. Neither Basenji model was involved in the generation of sequences using the GAN approach, making both models an unbiased predictor of cell-type specific activity for these sequences. Our trained model is sufficiently different from the pre-trained model that we expect each model to provide an unbiased estimate of the cell-type specific activity of the sequences generated through backpropagation on the other model.

We quantify our performance under the Basenji models with two metrics. The first is the average activation in the target cell type. The second is the ratio between the activation in the target cell type and the average activation in the other cell types.

5 Results

Below are the results from each of our three evaluation processes. Due to the time required to run the backpropagation process, we only generated a few sequences specific to the HeLa, HepG2, and GM12878 cell types for this approach. As expected, the results were quite poor. (This is unsurprising because both Basenji models appear incapable of predicting cell type specific activation for these cell types.) Because of our low sample size, we do not report results for these cell types for this approach.

5.1 GC Content Analysis

	Mean					Standard Deviation				
Cell Type	HeLa	HepG2	GM12878	Joint	K562	HeLa	HepG2	GM12878	Joint	K562
Random Sequence	0.5	0.5	0.5	0.5	0.5	0.02	0.02	0.02	0.02	0.02
Cell-Type Specific Activity	0.4	0.4	0.4	0.4	0.4	0.7	0.8	0.7	0.7	0.07
Genetic Algorithm	0.34	0.86	0.63	0.61	0.27	0.12	0.08	0.07	0.23	0.12
GAN	0.24	0.28	0.37	0.3	0.38	0.02	0.01	0.02	0.06	0.01
Backpropagation (pre-trained model)	N/A	N/A	N/A	0.52	0.52	N/A	N/A	N/A	0.02	0.02
Backpropagation (trained model)	N/A	N/A	N/A	0.50	0.50	N/A	N/A	N/A	0.02	0.02

Table 2: Mean and standard deviation of normal distributions fit to GC content results. We include as baselines results from random sequences and from the sequences with known cell-type specific activity.

Figure 5 presents the results of the GC content analysis; for reference, it also shows the GC content distributions in the sequences with cell-type specific activity. For a more quantitative analysis of our GC content results, we fit normal distributions for each of the methods and cell types. (As Figure 5 shows, GC content is distributed roughly normally in the real sequences.) Table 2 displays the mean and standard deviation of these distributions.

As these results show, all of our approaches fail to generate sequences that are similar in GC content to the provided training sequences. Of our approaches, the GAN appears to achieve the best results,

GC content will also be unable to satisfy more complex requirements. As our approaches generally struggled to generate sequences with the expected GC content, we decided to focus on improving performance under our existing metrics rather than on developing a more sophisticated scheme for measuring feasibility.

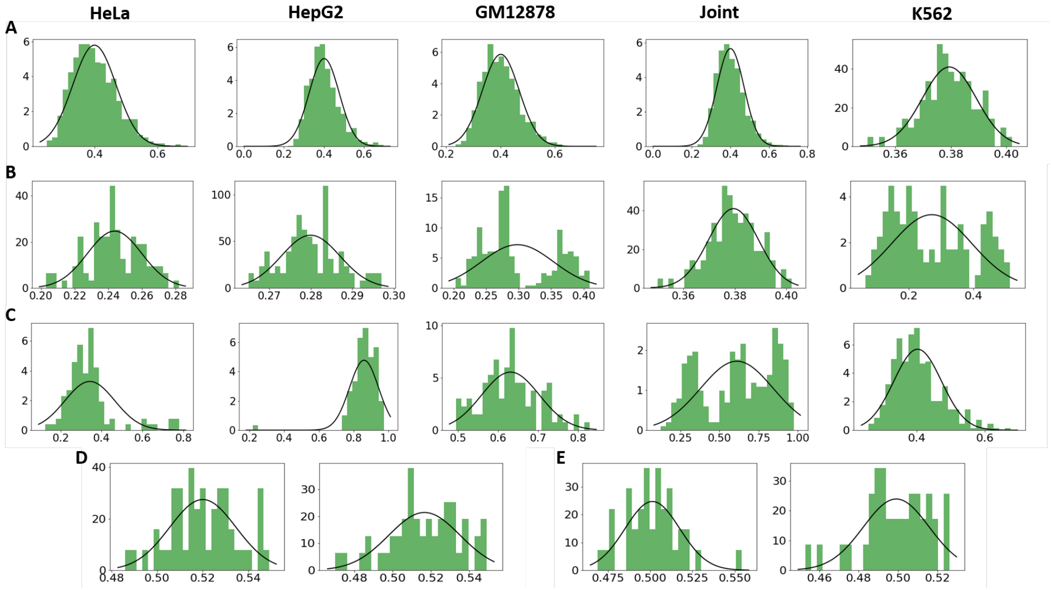


Figure 5: The results of the GC content analysis. **A.** The distribution of GC content in sequences with high cell-type specific activation. **B.** The distribution in sequences produced by the genetic algorithm; **C.** produced by the GAN; **D.** produced by backpropagation on the pre-trained model; and **E.** produced by backpropagation on our trained model. For **D.** and **E.**, the left graph shows the distribution in sequences designed to have joint activity in the first three cell types, and the right graph shows the distribution in sequences designed to have activity only in the K562 cell type.

with mean GC content between 24% and 38%, which is at least somewhat close to the expected mean of 40%. The genetic algorithm generates sequences with widely varying GC content, perhaps indicating that the provided Basenji model does not use GC content as a factor when predicting activation. This would make sense given that the high-activation sequences have the same GC content as the human genome generally (Figure 5). This inference is supported by the results from the backpropagation approach. These results are similar to the expected GC content of random sequences and thus demonstrate that during backpropagation the model did not learn to modify the GC content (which, given our random initialization approach, started at about 50%).

5.2 Motif Analysis

	Occurrence Frequency					Occurrence Delta				
Cell Type	HeLa	HepG2	GM12878	Joint	K562	HeLa	HepG2	GM12878	Joint	K562
Random Sequence	9.4	9.4	9.4	9.4	9.4	0.21	0.21	0.21	0.21	0.21
Cell-Type Specific Activity	19.0	19.4	15.9	18.1	19.5	N/A	N/A	N/A	N/A	N/A
Genetic Algorithm	48.9	32.3	23.0	34.7	42.6	0.70	0.60	0.43	0.57	0.67
GAN	27.8	19.4	15.9	14.8	27.3	0.60	0.36	0.35	0.44	0.80
Backpropagation (pre-trained model)	N/A	N/A	N/A	6.3	14.9	N/A	N/A	N/A	0.22	0.22
Backpropagation (trained model)	N/A	N/A	N/A	8.78	16.2	N/A	N/A	N/A	0.21	0.32

Table 3: "Specific motif" frequency and delta for each generation technique. We include as baselines results from 10,000 random sequences and from the sequences with known cell-type specific activity. The delta is in comparison with the frequencies for the sequences with known cell-type specific activity; this is why the delta for those sequences is N/A.

Table 3 presents the results of our motif analysis. The genetic algorithm and GAN approaches consistently include the "specific motifs" much more frequently than would be expected from random sequences, indicating that they are are learning to identify and value the presence of these motifs. The genetic model especially uses "specific motifs," which makes sense given that we seeded the initial sequences with these motifs. If anything, these approaches use too many of the "specific motifs": such motifs appear more frequently in the generated sequences than in the actual sequences with cell-type specific activity, and the deltas for the sequences generated by the GAN and the genetic algorithm are greater than the deltas for random sequences, indicating that both approaches do a poor job matching the distribution of motifs. The backpropagation approaches achieve smaller deltas, but at the cost of reduced use of the "specific motifs," especially for the sequences trained to have specific activity in the HeLa, HepG2, and GM12878 cell types.

5.3 Basenji Prediction

	Mean Activation and Ratio using Pre-trained Model				
Cell Type	HeLa	HepG2	GM12878	Joint	K562
Random Sequence	μ: 47.0, Δ: 1.15	μ: 45.5, Δ: 1.14	μ: 50.9, Δ: 1.47	μ: 48.1, Δ: 2.46	μ: 29.7, Δ: 0.48
Cell-Type Specific Activity	μ: 0.36, Δ: 1.19	μ: 0.41, Δ: 1.16	μ: 0.49, Δ: 1.16	μ: 0.50, Δ: 2.67	μ: 0.53, Δ: 0.39
Genetic Algorithm	μ: 4.91 , Δ: 1.39	μ: 17.9, Δ: 1.32	μ: 265.7, Δ: 1.22	μ: 90.8, Δ: 2.17	μ: 0.061, Δ 0.54:
GAN	μ: 4.52 , Δ: 1.25	μ: 0.49, Δ: 1.16	μ: 8.24, Δ: 1.42	μ: 4.24, Δ: 2.42	μ: 8.40, Δ: 0.51
Backpropagation (pre-trained model)	N/A:	N/A	N/A	μ: 12.9, Δ: 11.3	μ: 0.79, Δ: 0.57
Pre-trained Model (raw)	N/A	N/A	N/A	μ: 182.4, Δ: 152.0	μ: 1.28, Δ: 1.47
Backpropagation (trained model)	N/A	N/A	N/A	μ: 27.0, Δ: 2.8	μ: 27.4, Δ:0.45
Trained Model (raw)	N/A	N/A	N/A	N/A	N/A

Table 4: Average pre-trained model predicted activation for each cell type for each sequence generation technique and average ratio between activation in the target cell type and the other cell types. We include as baselines results from a random sequence and from the sequences with known cell-type specific activity. We also include results using the "raw," real-valued sequences produced by backpropagation.

	Mean Activation and Ratio using Trained Model				
Cell Type	HeLa	HepG2	GM12878	Joint	K562
Random Sequence	μ: 19.7, Δ: 1.12	μ: 20.8, Δ: 1.31	μ: 20.5, Δ: 1.18	μ: 20.3, Δ: 2.07	μ: 11.6, Δ: 0.50
Cell-Type Specific Activity	μ: 2.47, Δ: 1.03	μ: 3.00, Δ: 1.34	μ: 2.51, Δ: 1.25	μ: 2.43 , Δ: 2.16	μ: 3.00, Δ: 0.50
Genetic Algorithm	μ: 5.29, Δ: 1.07	μ: 16.5, Δ: 1.31	μ: 28.0, Δ:1.21	μ: 31.1, Δ: 2.06	μ: 1.75, Δ: 0.63
GAN	μ: 2.85, Δ: 1.05	μ: 1.86, Δ: 1.23	μ: 20.0, Δ: 1.14	μ: 8.24, Δ: 1.74	μ: 23.5, Δ: 0.57
Backpropagation (pre-trained model)	N/A	N/A	N/A	μ: 4.36, Δ: 2.06	μ: 6.93, Δ: 0.53
Pre-trained Model (raw)	N/A	N/A	N/A	N/A	N/A
Backpropagation (trained model)	N/A	N/A	N/A	μ: 5.34, Δ: 4.3	μ: 2.92, Δ: 5.9
Trained Model (raw)	N/A	N/A	N/A	μ: 281.0, Δ: 200.7	μ: 50.0, Δ: 41.7

Table 5: Basenji prediction results using the model we trained. See Table 4 for more details.

Tables 4 and 5 present the results from the Basenji prediction evaluation. Critically, for both the pre-trained and trained Basenji models, the predicted activation in the target cell type for the sequences with known cell-type specific activity is very low. (This is consistent with the results displayed in Figure 4.) This is concerning because it indicates that the models are unable to recognize sequences with cell-type specific activity and therefore suggests that neither model is a good proxy for the actual activity of our generated sequences.

As expected, the results are uniformly poor for the first thee cell types, with deltas not better than those achieved by the random sequences. For the genetic algorithm and GAN approaches, the results are similar for the joint analysis and for the fourth cell type. However, the sequences generated by backpropagation do demonstrate substantial cell-type specific activity, at least when evaluated on the same model under which the backpropagation was preformed. In particular, the "raw" sequences have massive deltas of up to 200.7. This indicates that the backpropagation process works insofar as it successfully adjusts the values in the input to achieve high and specific activation for the target cell type. But despite the steps described in Section 3.4, the process of continually converting from a real-valued "raw" sequence to a one-hot encoded sequence still undoes much of the gains achieved through backpropagation, with the delta falling to as little as 4.3 for the joint sequences and 1.47 for the fourth cell-type sequences. However, these deltas are still far higher than those achieved by the random sequences or by the other generative approaches. More concerning, though, is that when these one-hot sequences are evaluated under the model that was not used to generate them, the deltas decline to levels consistent with the deltas from the other approaches. This might reflect the weakness of both of the Basenji models and signal that one or both model is not learning generalizable features for cell-type specific activity.

Regarding the results for the genetic algorithm and the GAN, we observed the following:

1. The mean target cell-type activations vary widely for the genetic algorithm sequences when evaluated on the pre-trained model, even for the first three cell types. We believe this to be a result of variation in the pre-trained model's ability to predict that the activity in one cell type is higher than in the others. For example, on the one hand, the model almost never predicts that activity in the K562 cell type is higher than in the others. Accordingly, because the genetic algorithm could not make the K562 activation higher than the others, it learned to make the predicted activations as small as possible so that the K562 activation was as close

as possible to the activations in the other cell types. As a result, the mean K562 activation for sequences designed to maximize specificity activity in the K562 cell type is 0.061. On the other hand, the pre-trained model more often predicts that activity in the GM12878 cell type is higher than in the others. Accordingly, the algorithm learned to maximize activations in all cell types in order to maximize the magnitude of the difference between the GM12878 activation and the next largest activation. As a result, the mean GM12878 activation for sequences deigned to maximize specific activity in the GM12878 cell type is 265.7.

2. The GAN model preforms consistently worse than the genetic algorithm. This perhaps indicates problems with our approach of learning to generate sequences that are difficult to distinguish from ones with cell-type specific activation. Biologically viable sequences that are very similar to ones with cell-type specific activation may not themselves have these cell-type specific properties: the GAN approach will not distinguish these sequences from those that do have cell-type specific behavior.

6 Discussion

We used the Basenji architecture, the state-of-the-art deep-learning approach for predicting cell-type specific regulatory activity from DNA sequence alone, to generate *de novo* regulatory sequences with cell-type specific activity.

Of our three approaches, each shows promise in different areas. In terms of biological feasibility, which we assessed by analyzing the GC content distribution, the GAN generated sequences with GC contents closest to the accepted mean of 40% [14]. In our first assessment of cell-type specific activity, the genetic algorithm generated sequences with the highest average number of "specific motifs," suggesting it at least learned to maintain the motifs seeded during the initialization step. The relatively high number of these motifs present in randomly generated sequences most likely indicates that the motifs are short enough that almost any sequence of length 1,000 contains some of them. This suggests the key factor linking the presence of "specific motifs" and cell-type specific activity is the distribution of the motifs, both in terms of the presence of certain motifs (which, according to our analysis, none of our approaches succeed at emulating any better than generating random sequences does) and in terms of spatial organization along the sequence (which we do not evaluate). In the final assessment of cell-type specific activity, none of the approaches perform significantly better than do randomly generated sequences when evaluated using the model (or models, when applicable) that did not directly contribute to the sequence generation. We think the primary causes of this are the lack of training examples with high cell-type specific activity (Table 1) and the inability of the models to accurately predict cell-type specific activity (Tables 4 and 5). Importantly, we confirmed the validity of the backpropagation implementation by confirming that performing backpropagation on each model results in sequences that that model predicts have the desired activation property. Below, we discuss several strategies to improve each approach.

The ability to quickly generate *de novo* sequences with cell-type specific activity would provide synthetic biologists with a valuable tool in their efforts to apply DNA synthesis technology to disease treatment and basic research. Two potential applications of designing regulatory sequences with cell-type specific activity are to enable more targeted gene therapies and to enable additional ways of testing the activity of proteins in certain tissues. Regarding gene therapy, we could generate a virus with a vector sequence that is only promoted in the target cells of interest. For example, we could create a sequence containing the human insulin gene under the control of a regulatory region designed for activity specifically in pancreatic cells to improve treatment for diabetic patients. This technique could also improve non-metastatic cancer gene therapy treatments wherein we desire to make cancer cells produce a cytotoxic small molecule or protein. Indeed, tissue-specific promoters are already recognized as critical to the effectiveness of gene therapy treatments and to the alleviation of toxic off-target effects [16] [17]. Regarding tissue-specific basic research, we consider the development of cell-type specific regulatory sequences as an improvement to Cre-Lox recombination. In Cre-Lox recombination, a specific gene is knocked out in select tissues to help identify the function of the associated gene product in those tissues [18]. The development of an *in silico* method of generating regulatory sequences designed to have different activities in different cell types would enable researchers to experiment with a continuum of expression levels in specific tissues, rather than only with a binary knockout mutant. Accordingly, this work has broad implications to disease treatment and basic research.

7 Future Work

7.1 More Data and Better Models

All of our approaches are hindered by a lack of data. While we do have millions of labeled sequences, we have only a few thousand examples of sequences with cell-type specific activation. As a result, our GAN model has few examples of "real" sequences, and the pre-trained and trained models have few examples with cell-type specific activation on which to learn how to predict such activation. Indeed, perhaps as a result of the lack of relevant training examples, even on sequences experimentally measured to have cell-type specific activation, the pre-trained and trained models fail to predict cell-type specific activation.

We could address the data and model limitations in several ways. First, we could train and evaluate on a larger subset of the data used by Dr. Kelley to train the full Basenji model, potentially providing us with thousands of additional sequences with cell-type specific behavior. Second, we could use the full Basenji model that operates over 131 kb sequences, which is presumably a significantly better model than the models we employed that operate over only 1 kb sequences because it can better incorporate distal regulatory information [5]. Third, we could switch the task to a classification one and train a model to predict only if a sequence has cell-type specific activation in the desired cell type instead of predicting the precise activation in each cell type: this less-detailed task may be easier to accomplish. Fourth, we could move away from the Basenji architecture and, for example, train a model that takes in the cell type as a feature and predicts activity in that cell type.

In addition, our analysis could be expanded to include more cell types and more transcription and epigenetic data. For example, we could attempt to predict ChIP-seq data, which should exhibit larger, more cell-type specific peaks than CAGE data does and therefore should be easier for the model to predict with cell-type specificity.

7.2 Improvements to Our Approaches

The genetic algorithm could be revised in several ways. Future approaches might include running the algorithm until convergence, such that ΔL as outlined in Section 3.2 is not positive for some number of sequential attempted mutations. Also, the algorithm could apply other mutation types besides point mutations, such as recombination among sequences.

For the GAN approach, we could experiment with the modified GAN architecture described by Killoran et al., which uses a discriminator paired with a predictor [6]. We would continue to task the predictor with distinguishing between generated sequences and real sequences with cell-type specific activity. But using the predictor may allow us to emphasize generating sequences that actually have cell-type specific activity rather than ones that are just similar in structure to the real sequences.

For the backpropagation approach, we could initialize our sequences with physiologically relevant GC contents. We could also experiment with more complex architectures for our trained model, potentially using recurrent layers or a transformer structure. Increasing the complexity of the model may allow the model to better learn the fine-grained differences between sequences with and without cell-type specific activity. (Although, to be sure, we would also need to think about how to reduce overfitting, which is already present.) Another option involves adopting additional methods for data augmentation, although how we would modify the existing sequences without jeopardizing the quality of the associated activation data remains unclear.

Furthermore, we could evaluate the potential of using a variational autoencoder to generate sequences. The variational autoencoder would be simpler and perhaps easier to train than the GAN but would allow us the same ability to sample from a latent space and generate sequences with, at least in theory, cell-type specific activity.

8 Conclusions

As biologists become authors, the potential to pen genetic sequences with cell-type specific properties is vast. Unfortunately, at present, our results primarily demonstrate the challenges in achieving that potential. High-activation sequences are rare, and cell-type specific high-activation sequences are even rarer, severely limiting the amount of data available for deep learning-based approaches to sequence generation. Additionally, given the limitations of even the state-of-the-art models for predicting sequence activity *in silico*, the analysis of generated sequences depends on either undertaking time-consuming and expensive lab work or relying on imperfect proxies for sequence activity. Nonetheless, our work outlines several approaches for sequence generation that we maintain hold promise. Collectively, the three techniques we experimented with proved capable of generating sequences with roughly normal GC content, with a high occurrence of motifs associated with cell-type specific activity, and with cell-type specific activity according to the weak models we used to evaluate performance. With more data and with better models for predicting cell-type specific information, we expect that some combination of the approaches we considered will be able to generate sequences with desired cell-type specific properties.

9 Peer Review Revisions

We thank our reviewers for providing thoughtful feedback regarding our methods and results. Most of our reviewers' comments concerned clarifications to aspects of our methodology: we addressed all of these comments.

We checked the sequences generated by the GAN approach and verified that the model had not suffered mode collapse and had not simply memorized and reproduced high-activation sequences. None of the sequences generated for any cell type exactly matched sequences in our dataset.

We expanded our GC content evaluation by providing a quantitative comparison of the distributions. Although the reviewers suggested running a statistical test to determine the significance of the differences between the distributions, we decided that doing so was unnecessary given the clear differences between the distributions for the generated sequences and the real ones. Additionally, while one reviewer suggested running an ablation analysis, we decided that doing so was unnecessary because none of our approaches produced particularly strong results.

10 Code Source

Code is available at https://github.com/Sawyerb/CS273B.git.

11 Acknowledgments

Our mentors Dr. David Kelley and Dr. Vikram Agarwal have been enormously helpful, providing us with advice on possible approaches for sequence generation, with data, and with ideas for improving the performance of our trained Basenji model. We are very grateful for their support.

References

[1] Kosuri, S. and Church, G.M., 2014. Large-scale de novo DNA synthesis: technologies and applications. Nature methods, 11(5), p.499.

[2] Hughes, R.A. and Ellington, A.D., 2017. Synthetic DNA synthesis and assembly: putting the synthetic in synthetic biology. Cold Spring Harbor perspectives in biology, 9(1), p.a023812.

[3] Benner, S.A. and Sismour, A.M., 2005. Synthetic biology. Nature Reviews Genetics, 6(7), p.533.

[4] Cross, D. and Burmester, J.K., 2006. Gene therapy for cancer treatment: past, present and future. Clinical medicine & research, 4(3), pp.218-227.

[5] Kelley, David R., et al. “Sequential Regulatory Activity Prediction across Chromosomes with Convolutional Neural Networks.” Oct. 2017, doi:10.1101/161851.

[6] Killoran, N., Lee, L.J., Delong, A., Duvenaud, D. and Frey, B.J., 2017. Generating and designing DNA with deep generative models. arXiv preprint arXiv:1712.06148.

[7] Kanamori-Katayama, M., Itoh, M., Kawaji, H., Lassmann, T., Katayama, S., Kojima, M., Bertin, N., Kaiho, A., Ninomiya, N., Daub, C.O. and Carninci, P., 2011. Unamplified cap analysis of gene expression on a single-molecule sequencer. Genome research, 21(7), pp.1150-1159.

[8] Hussain, T. and Mulherkar, R., 2012. Lymphoblastoid cell lines: a continuous in vitro source of cells to study carcinogen sensitivity and DNA repair. International journal of molecular and cellular medicine, 1(2), p.75.

[9] Forrest, A.R., Kawaji, H., Rehli, M., Baillie, J.K., De Hoon, M.J., Haberle, V., Lassmann, T., Kulakovskiy, I.V., Lizio, M., Itoh, M. and Andersson, R., 2014. A promoter-level mammalian expression atlas. Nature, 507(7493), p.462.

[10] Jolma, A., Yan, J., Whitington, T., Toivonen, J., Nitta, K.R., Rastas, P., Morgunova, E., Enge, M., Taipale, M., Wei, G. and Palin, K., 2013. DNA-binding specificities of human transcription factors. Cell, 152(1-2), pp.327-339.

[11] Timothy L. Bailey, Mikael Bodén, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, William S. Noble, "MEME SUITE: tools for motif discovery and searching", Nucleic Acids Research, 37:W202-W208, 2009.

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

[13] Ccg.epfl.ch. PWMTools. [online] Available at: https://ccg.epfl.ch//cgi-bin/pwmtools/pwmviewer.cgi?ID=ZSCAN4_full&lib=jolma2013.

[14] Bernardi, G., 2000. Isochores and the evolutionary genomics of vertebrates. Gene, 241(1), pp.3-17.

[15] Ghandi, Mahmoud, et al. “Enhanced Regulatory Sequence Prediction Using Gapped k-Mer Features.” PLoS Computational Biology, Public Library of Science, 17 July 2014, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4102394/. /

[16] Saukkonen, Kirsi, and Akseli Hemminki. “Tissue-Specific Promoters for Cancer Gene Therapy.” Expert Opinion on Biological Therapy, U.S. National Library of Medicine, May 2004, https://www.ncbi.nlm.nih.gov/pubmed/15155160.

[17] Zheng, Changyu, and Bruce J Baum. “Evaluation of Promoters for Use in Tissue-Specific Gene Delivery.” Methods in Molecular Biology (Clifton, N.J.), U.S. National Library of Medicine, 2008, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2685069/.

[18] Sauer, B. “Functional Expression of the Cre-Lox Site-Specific Recombination System in the Yeast Saccharomyces Cerevisiae.” Molecular and Cellular Biology, U.S. National Library of Medicine, June 1987, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC365329/.

[19] “A Genetic Algorithm Tutorial.” CiteSeerX, https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.184.3999.

[20] Ancona, Marco, et al. "Towards better understanding of gradient-based attribution methods for deep neural networks." arXiv preprint arXiv:1711.06104 (2017).

[21] Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

[22] Birnbaum, Sawyer, et al. "Temporal FiLM: Capturing Long-Range Sequence Dependencies with Feature-Wise Modulations." Advances in Neural Information Processing Systems. 2019.