Deep exploration networks for rapid engineering of functional DNA sequences

Johannes Linder[*] **Nicholas Bogard**[†] **Alexander B. Rosenberg**[‡] **Georg Seelig**
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, WA 98195
`jlinder2@cs.washington.edu`

Abstract

Engineering gene sequences with defined functional properties is a major goal of synthetic biology. Deep learning models, combined with gradient ascent-style optimization, show promise for sequence generation. The generated sequences can however get stuck in local minima, have low diversity and their fitness depends heavily on initialization. Here, we develop deep exploration networks (DENs), a type of generative model tailor-made for exploring an input space to minimize the cost of a neural network fitness predictor. By making the network compete with itself to promote sequence diversity during training, we obtain generators capable of sampling hundreds of thousands of high-fitness sequences. We demonstrate DENs in the context of alternative polyadenylation. Using DENs, we engineered polyadenylation signals with more than 10-fold higher selection odds than gradient ascent-generated patterns, and validated their performance experimentally.

1 Introduction & Related work

Designing DNA sequences for a target cellular function is a difficult task, as the cis-regulatory information encoded in any stretch of DNA can be very complex and affect numerous mechanisms, including transcriptional and translational efficiency, splicing, 3' end processing, and more. Yet, sequence-level design of genetic components has been making rapid progress in the past few years, in part thanks to improved bioinformatics modeling. In particular deep learning has emerged as state-of-the-art in predictive modeling for many sequence-function problems (Alipanahi et. al., 2015; Zhou et. al., 2015; Quang et. al., 2019; Avsec et. al., 2019; Kelley et. al., 2016; Greenside et. al., 2018; Kelley et. al., 2018; Jaganathan et. al., 2019; Bogard et. al., 2019; Sample et. al., 2019).

Recently, gradient ascent optimization of the input sequence through a neural network fitness predictor has been proposed as an alternative to discrete search heuristics such as genetic algorithms. At its core, the method treats the input pattern as a position weight matrix (PWM) which is evaluated by the fitness predictor. The fitness score is used to compute a gradient with respect to the PWM parameters and optimized by gradient ascent. The proposed method has been used to generate TF binding motifs (Lanchantin et. al., 2016, Killoran et. al., 2017), polyadenylation signals (Bogard et. al., 2019) and protein 3D structures (Evans et. al., 2018). While showing promise, the basic method suffers from a number of limitations. First, it may easily get stuck in local minima and the converged fitness depends on initialization (Bogard et. al., 2019). Second, the method offers no means of controlling sequence diversity, which may be required for generation of large candidate sequence sets.

[*] All code available at http://www.github.com/johli/genesis

[†] Department of Electrical & Computer Engineering

[‡] Department of Electrical & Computer Engineering

Preprint. Under review.

To address these limitations, we develop a variant of generative neural network models which we call Deep Exploration Networks (DENs). In DENs, we promote sequence diversity during training by making the generator compete with itself to produce dissimilar patterns (Figure 1A). We thus force the model to explore a much larger region of the cost landscape and effectively maximize both sequence fitness and diversity. The architecture shares similarities with (Killoran et. al., 2017). However, in contrast to (Killoran et. al., 2017) where optimization is done on the input seed of a pre-trained GAN, here we optimize the weights of the generator to maximize both a fitness and diversity cost. As a result, the generator learns to sample a large and diverse set of sequences with high fitness score. The approach is conceptually similar to variational autoencoders (Kingma & Welling, 2013) and adaptive sampling methods (Brookes et. al., 2019), but rather than encoding the original pattern distribution or a conditional distribution, the model centers on the objective and maximizes pattern variation. We evaluate DENs on the task of designing 3' UTR sequences with target APA isoform abundance (Figure 1B). We find that DENs learn to generate sequences with significantly higher fitness compared to equivalent gradient ascent-generated sequences.

2 Exploration in Deep Generative Models

The predictor $\mathcal{P}$ used in a DEN is a differentiable model capable of predicting some property of an input pattern. The generator $\mathcal{G}$ is a neural network designed to produce a pattern which can be passed as input to the predictor. Here, we are interested in generating DNA sequences; these patterns are typically represented as 1-hot-coded matrices, where columns denote nucleotide position and rows denote nucleotide identity ($\{0,1\}^{N \times 4}$). The predictor output is used to define an objective (the *cost function*), and the overall goal is to generate sequences minimizing the cost. Only the generator $\mathcal{G}$ is optimized, having pre-trained $\mathcal{P}$ to accurately predict the targeted biological function.

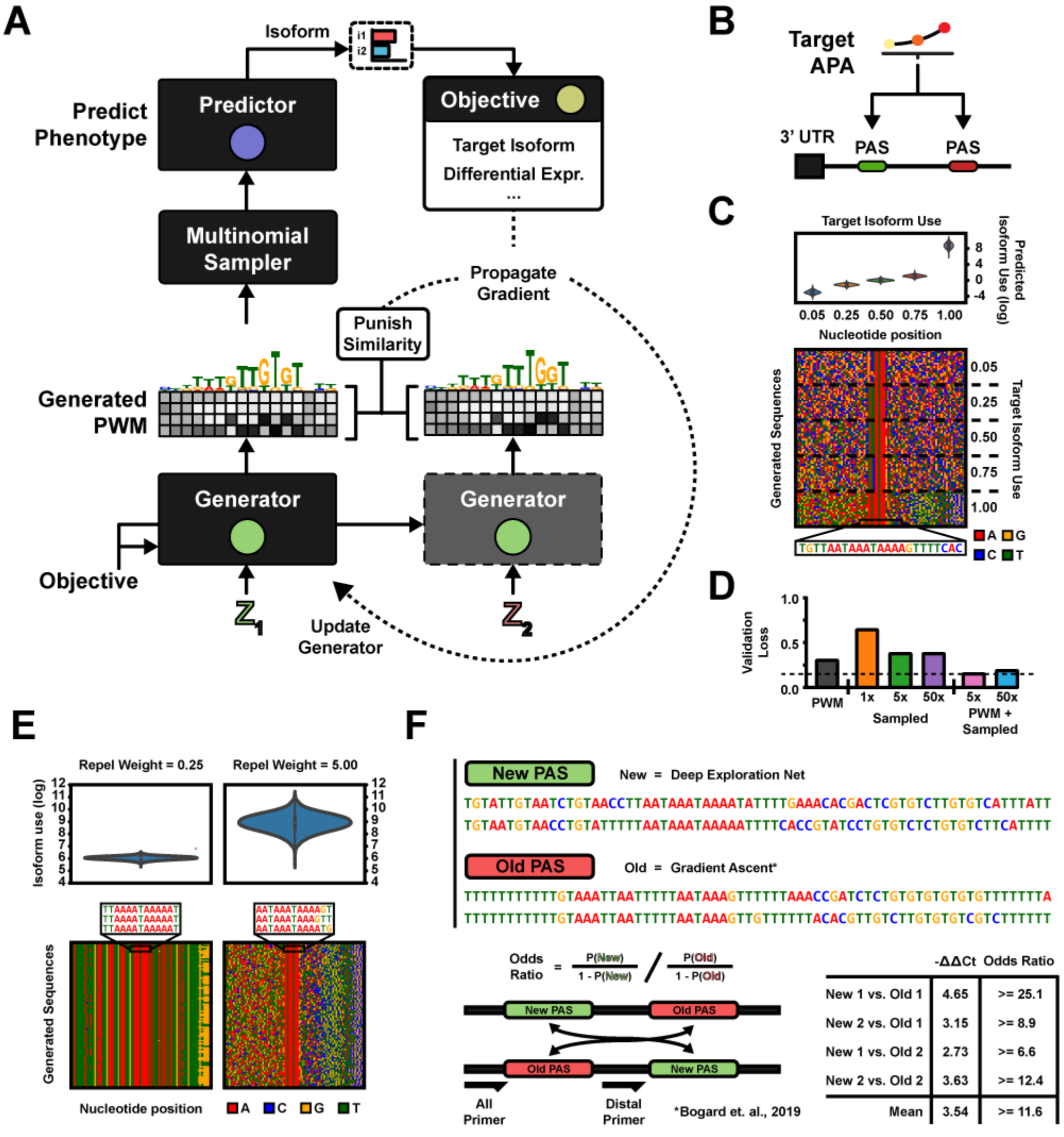
This cost layout is quite different compared to a classical GAN (Goodfellow et. al., 2014), which is typically optimized to minimize some cost $\text{C}(\mathcal{D}(\text{Data}), \mathcal{D}(\mathcal{G}(z)))$ such that an adversarial discriminator $\mathcal{D}$ can not distinguish between the real data and the distribution generated by $\mathcal{G}$. Here, we instead jointly minimize the fitness cost $\text{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z_1))$ of $\mathcal{P}$ and an adversarial diversity cost $\text{C}_{\text{diversity}}(\mathcal{G}(z_1), \mathcal{G}(z_2))$ of $\mathcal{G}$. Note that, in contrast to (Killoran et. al., 2017) where optimization is done on a single input seed z of a pre-trained GAN, $\min_z \text{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z)))$, we optimize the generator $\mathcal{G}$ itself, $\min_{\mathcal{G}} \text{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z_1)) + \text{C}_{\text{diversity}}(\mathcal{G}(z_1), \mathcal{G}(z_2))$, for all seeds $z_1, z_2 \in U(-1,1)^{100}$.

We define $\text{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z))$ in terms of the predictor's output. For example, in the case of models that predict isoform abundances, we may minimize the (symmetric) KL-divergence between predicted and target proportions: $\text{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z)) = \text{KL}(\mathcal{P}(\mathcal{G}(z))||t) + \text{KL}(t||\mathcal{P}(\mathcal{G}(z)))$, where $\mathcal{P}(\mathcal{G}(z), t \in [0,1]$.

The distinguishing feature of DENs is to enforce exploration during training by controlling the degree of sequence diversity generated by the network. We do so by making the generator compete with itself; we penalize any two generated sequence patterns based on similarity. This mechanism is implemented by running the generator twice at each step of the optimization with two random seeds $z_1, z_2 \in \text{U}(-1,1)$. Here, we penalize sequence patterns using a multi-offset cosine similarity loss. We found empirically that minimizing a slack-bound cosine similarity gives the best results, where a fraction of the sequences can be identical up to a margin ϵ without incurring any loss. Given two sequence patterns $S^{(1)}$ and $S^{(2)}$ generated by $\mathcal{G}$, we define $\text{C}_{\text{diversity}}(S^{(1)}, S^{(2)})$ as:

$$C_{\text{diversity}}(S^{(1)}, S^{(2)}) = \max\left(\left(\frac{1}{N} \max_{\sigma} \sum_{i=1}^{N-\sigma} \sum_{j=1}^{M} s^{(1)}_{i+\max(\sigma,0),j} \cdot s^{(2)}_{i+\max(-\sigma,0),j}\right) - \epsilon, 0\right)$$

G generates patterns $X = \mathcal{G}(z)$, representing nucleotide log probabilities ($X \in \mathcal{R}^{N \times 4}$). By applying Softmax $s_{ij} = \frac{e^{x_{ij}}}{\sum_{k=1}^{M} e^{x_{ik}}}$, we turn the logits into a PWM $S \in [0,1]^{N \times 4}$. We can directly pass S to the predictor $\mathcal{P}$, however, the gradient propagated backward through S approaches zero as the nucleotide logits push the Softmax probabilities toward their extremes. We can alternatively sample K independent 1-hot-coded patterns from S and backpropagate the gradient using straight-through estimation (Bengio, Léonard & Courville, 2013; Courbariaux et. al., 2016; Bogard et. al., 2019). As our results indicate below, using both representations for $\text{C}_{\text{fitness}}(S)$ and walking down the average gradient can enhance convergence even further. For the diversity loss $C_{\text{diversity}}(S^{(1)}, S^{(2)})$, however, we only use the sampled (straight-through) representation.



TGTATTGTAATCTGTAACCTTAATAAATAAAATATTTTGAAACACGACTCGTGTCTTGTGTCATTTATT
TGTAATGTAACCTGTATTTTTAATAAATAAAAATTTTCACCGTATCCTGTGTCTCTGTGTCTTCATTTT

TTTTTTTTTTTGTAAATTAATTTTTAATAAAGTTTTTTAAACCGATCTCTGTGTGTGTGTTTTTTTA
TTTTTTTTTTTGTAAATTAATTTTTAATAAAGTTGTTTTTTACACGTTGTCTTGTGTGTCGTCTTTTTT

	-ΔΔCt	Odds Ratio
New 1 vs. Old 1	4.65	>= 25.1
New 2 vs. Old 1	3.15	>= 8.9
New 1 vs. Old 2	2.73	>= 6.6
New 2 vs. Old 2	3.63	>= 12.4
Mean	3.54	>= 11.6

Figure 1: (A) The generator is run twice on two seeds, producing two sequence patterns. One pattern is evaluated by the fitness predictor, resulting in an objective gradient. The patterns are also penalized by similarity, resulting in an exploration gradient, and the generator is updated by both gradients. (B) The generative task is to design proximal APA signals which are polyadenylated at target proportions. (C) Evaluation of five separate DENs trained to generate sequences according to APA isoform targets 5%, 25%, 50%, 75% and 100% ('Max'). (Top) Predicted isoform proportions of 1,000 sampled sequences per target. Mean proportions are within 1% of target proportions. (Bottom) Generator sequence diversity, illustrated by 20 randomly sampled sequences per objective on a pixel grid where rows denote sequence samples. 0% duplication rate at 500,000 sampled sequences. (D) DEN validation loss after 1,500 training iterations, using three modes of sequence representation: 1. PWM, 2. ST-sampled One-hots and 3. Both. '1x', '5x' etc. refers to the number of samples drawn. (E) The diversity loss was evaluated by re-training the Max-APA DEN with a small (left) and large (right) loss coefficient respectively. (Top) Predicted isoform proportion for 1,000 generated sequences. (Bottom) Sequence diversity illustrated 100 sampled sequences. Small coefficient (left): Mean log odds = 6.06 +- 0.12, 99.5% duplication rate at 100,000 samples. Large coefficient (right): Mean log odds = 8.91 +- 0.72, 0% duplication rate at 100,000 samples. (F) Two Max-APA sequences generated by the DEN were synthesized on minigene reporters in competition with baseline gradient ascent-generated sequences using the same fitness predictor. Isoform fold changes were assayed using qPCR. The newly generated sequences have on average 9.4-fold increased preference.

3 Experiments

We evaluate DENs in the context of Alternative Polyadenylation (APA). APA is a 3' end processing event where competing polyA signals (PAS) within a 3' UTR give rise to multiple mRNA isoforms (Figure 1B) (Di Giammartino et al., 2011; Tian and Manley, 2017). A typical PAS consists of a core sequence element (CSE), as well as diverse upstream and downstream sequence elements (USE, DSE). We have previously developed a neural network for predicting APA isoform abundance (APARENT; Bogard et. al., 2019), which we use here as the predictor. The DEN was tasked with generating APA signals with precisely defined target isoform abundances as well as maximally strong signals. The generator network follows a DC-GAN architecture (Radford et. al., 2015). We built the DEN in Keras (Chollet et. al., 2015) and optimized the generator with Adam (Kingma et. al., 2014).

We trained 5 generators, each optimized to the following target isoform proportions: 0%, 25%, 50%, 75% and maximal use ('Max'). The objectives were encoded in the cost function by minimizing the KL-divergence between the predicted APA isoform proportion $\mathcal{P}(\mathcal{G}(z))$ and the target $t \in [0, 1]$:

$$\mathrm{C}_{\text{fitness}}(\mathcal{P}(\mathcal{G}(z)) = \mathrm{KL}(\mathcal{P}(\mathcal{G}(z))||t) + \mathrm{KL}(t||\mathcal{P}(\mathcal{G}(z)))$$

After training, each generator could produce accurate sequence samples (Figure 1C, Top): Each generated isoform distribution mean was within 1% from the target proportion. The generated sequences for the Max-objective were predicted to be extremely efficient signals (on average 99.98% predicted use). All five generators exhibited high diversity (Figure 1C, Bottom), with 0% duplication rate. We re-trained the Max-isoform generator when using one-hot samples (straight-through), the continuous PWM, or a combination of both as input to the predictor (Figure 1D). Using both representations, the loss decreased to less than 50% the magnitude of the PWM loss after 30 epochs.

To evaluate the importance of exploration, we re-trained the Max isoform-generator under two different parameter settings; in one instance, we lowered the diversity loss coefficient to a small value, and in another instance we increased it (Figure 1E). With a low coefficient, the generator only learns to sample few, low-diversity sequences (mean log odds = 6.06, 99.5% duplication rate). With an increased coefficient, generated sequences become much more diverse and the mean isoform odds increase almost 20-fold (mean log odds = 8.91, 0% duplication rate). These results suggest that exploration during training may drastically improve the final fitness of the generator.

Finally, we characterized experimentally whether DEN-generated polyA signals truly are more optimal than sequences generated by the baseline gradient ascent method. To that end, we synthesized APA reporters with two adjacent polyA signals (Figure 1F): Each reporter contained one of the newly generated Max-target signals, as well as one of the strongest gradient ascent-optimized signals from (Bogard et. al., 2019). In order to discount first-come-first-serve bias, we assayed both orientations for each reporter. The reporters were cloned onto plasmids and delivered to HEK293 cells. We quantified RNA isoform levels using a qPCR assay, measuring the Ct values of total and distal RNA. Using Ct values to estimate odds ratios, we found that the DEN-generated sequences were on average 11.6-fold more preferred than the gradient ascent-sequences. To put this in perspective, the strongest gradient ascent-sequence had usage odds of 127:1 (99.22%) relative to a distal bGH PAS separated by 200 nt. The DEN-sequences would have usage odds of 1481:1 (99.93%) relative to the same signal.

4 Conclusion

We developed an end-to-end differentiable generative network model, Deep Exploration Networks (DENs), capable of synthesizing large, diverse sets of sequences with high fitness. The model could generate polyadenylation signals with precisely defined target isoform ratios, and it could generate polyadenylation signals that were far stronger than any previously designed sequence.

DENs incorporate many techniques to improve its generative capabilities, but the single most important contribution is the control of exploration during training. By having the generator sample two sequences given different seeds, we developed a hinge-style loss which penalized sequence pairs based on similarity. Our analysis showed that the magnitude by which we punish sequence similarity almost entirely determines final generator diversity and, importantly, also largely determines the final fitness of the generated patterns. During training, the optimizer trades off exploring (repelling similar patterns) with exploiting (maximizing pattern fitness) until convergence is reached. In the end, this scheme produces generative models with (1) high fitness, and (2) controllable diversity.

References

Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning." Nature biotechnology 33.8 (2015): 831.

Avsec, Žiga, et al. "Deep learning at base-resolution reveals motif syntax of the cis-regulatory code." bioRxiv (2019): 737981.

Avsec, Žiga, et al. "The Kipoi repository accelerates community exchange and reuse of predictive models for genomics." Nature biotechnology (2019): 1.

Bengio, Yoshua, Nicholas Léonard, and Aaron Courville. "Estimating or propagating gradients through stochastic neurons for conditional computation." arXiv preprint arXiv:1308.3432 (2013).

Bogard, Nicholas, et al. "A Deep Neural Network for Predicting and Engineering Alternative Polyadenylation." Cell (2019).

Brookes, David H., Hahnbeom Park, and Jennifer Listgarten. "Conditioning by adaptive sampling for robust design." arXiv preprint arXiv:1901.10060 (2019).

Chollet, François. "Keras (2015)." (2017).

Courbariaux, Matthieu, et al. "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830 (2016).

Di Giammartino, Dafne Campigli, Kensei Nishida, and James L. Manley. "Mechanisms and consequences of alternative polyadenylation." Molecular cell 43.6 (2011): 853-866.

Evans, R., et al. "De novo structure prediction with deep learning based scoring." Annu Rev Biochem 77 (2018): 363-382.

Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

Greenside, Peyton, et al. "Discovering epistatic feature interactions from neural network models of regulatory DNA sequences." Bioinformatics 34.17 (2018): i629-i637.

Jaganathan, Kishore, et al. "Predicting splicing from primary sequence with deep learning." Cell 176.3 (2019): 535-548.

Kelley, David R., Jasper Snoek, and John L. Rinn. "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks." Genome research 26.7 (2016): 990-999.

Kelley, David R., et al. "Sequential regulatory activity prediction across chromosomes with convolutional neural networks." Genome research 28.5 (2018): 739-750.

Killoran, Nathan, et al. "Generating and designing DNA with deep generative models." arXiv preprint arXiv:1712.06148(2017).

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

Lanchantin, Jack, et al. "Deep motif: Visualizing genomic sequence classifications." arXiv preprint arXiv:1605.01133(2016).

Quang, Daniel, and Xiaohui Xie. "FactorNet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data." Methods (2019).

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

Sample, Paul J., et al. "Human 5ÚTR design and variant effect prediction from a massively parallel translation assay." Nature biotechnology 37.7 (2019): 803.

Tian, Bin, and James L. Manley. "Alternative polyadenylation of mRNA precursors." Nature reviews Molecular cell biology 18.1 (2017): 18.

Zhou, Jian, and Olga G. Troyanskaya. "Predicting effects of noncoding variants with deep learning–based sequence model." Nature methods 12.10 (2015): 931.