EPiC Series in Computing

Volume 70, 2020, Pages 163–172

Proceedings of the 12th International Conference
on Bioinformatics and Computational Biology

Convolutional neural net learns promoter sequence features driving transcription strength

Nicholas Leiby[1], Ayaan Hossain[2], and Howard M Salis[2]
[1] Two Six Labs, [2] Pennsylvania State University
nick.leiby@twosixlabs.com, auh57@psu.edu, salis@psu.edu

Abstract

Promoters drive gene expression and help regulate cellular responses to the environment. In recent research, machine learning models have been developed to predict a bacterial promoter's transcriptional initiation rate, although these models utilize expert-labeled sequence elements across a defined set of DNA building blocks. The generalizability of these methods is therefore limited by the necessary labeling of the specific components studied. As a result, current models have not been used to predict the transcriptional initiation rates of promoters with generalized nucleotide sequences. If generalizable models existed, they could greatly facilitate the design of synthetic genetic circuits with well-controlled transcription rates in bacteria.

To address these limitations, we used a convolutional neural network (CNN) to predict a promoter's transcriptional initiation rate directly from its DNA nucleotide sequence. We first evaluated the model on a published promoter component dataset. Trained using only the sequence as input, our model fits held-out test data with $R^2 = 0.90$, comparable to published models that fit expert-labeled sequence elements.

We produced a new promoter strength dataset including non-repetitive promoters with high sequence variation and not limited to combinations of discrete expert-labeled components. Our CNN trained on this more varied dataset fits held-out promoter strength with $R^2 = 0.61$. Previously-published models are intractable on a dataset like this with highly diverse inputs. The CNN outperforms classical approach baselines like LASSO on a bag of words for promoter sequence elements ($R^2 = 0.42$).

We applied recent machine learning approaches to quantify the contribution of individual nucleotides to the CNN's promoter strength prediction. Learning directly from DNA sequence, our model identified the consensus -35 and -10 hexamer regions as well as the discriminator element as key contributors to σ^{70} promoter strength. It also replicated a finding that a perfect consensus sequence match does not yield the strongest promoter.

The model's ability to independently learn biologically-relevant information directly from sequence, while performing similarly to or better than classical methods, makes it appealing for further prediction optimization and research into generalizability. This approach may be useful for synthetic promoter design, as well as for sequence feature identification.

Introduction

Promoters drive gene expression and help regulate cellular responses to the environment. Years of research have discovered many of the proteins and sequence motifs necessary for transcription. For example, in *E. coli*, transcription is mediated by the polymerase holoenzyme, which combines five subunits with one of seven sigma factors determining much of the sequence specificity.[1] The transcription of most promoters is mediated by the σ^{70} factor under standard growth conditions.[2] Distinct elements of these σ^{70} promoters cooperatively determine the magnitude of gene transcription. These elements include conserved hexamers -10 and -35 bases upstream of the transcription start site, a discriminator region downstream of the -10 hexamer, an UP element upstream of the -35 hexamer that serves as a binding site for the RNA polymerase, overall GC content, the length of a spacer segment between the -35 and -10 hexamers, and the sequence context of the promoter.[3-7]

Despite extensive research on the subject, there are limitations to our ability to identify promoters in sequence data, let alone predict the regulation of a promoter or the magnitude of the transcription it drives. Many studies have characterized individual components of a promoter in a fixed sequence context[4, 8-9], but these findings don't necessarily transfer to another promoter or context.[10] Recent research by Urtecho et al.[11] attempted to address this by measuring and modeling the transcription strength of all combinations of a set of 35 different σ^{70} promoter components. Other researchers created a biophysical model for the transcription strength of these same promoter components with high fidelity[12]. However, these methods are limited to the specific components studied and by the necessary expert labeling. Our inability to predict the strength and regulatory behavior of arbitrary novel promoters hinders the design of functional synthetic gene circuits.

To address these issues, two things are needed: 1) a dataset of transcription strengths for promoters with diversity that goes beyond permutations of discrete DNA components and 2) a model of promoter strength that does not rely on hand labeling or a circumscribed input domain.

Sequence-based deep learning models have increasingly been used with success in biology. For example, DeepBind[13] and DeepSEA[14] applied convolutional deep learning to model the sequence specificity of protein binding, outperforming the best existing conventional methods. Convolutional neural network (CNN) architectures developed for computer vision can be applied by considering a DNA sequence as an image. A genome sequence is expressed as a fixed length 1D sequence window with four channels (A, C, G, T) analogous to 2D images with three color channels (R, G, B). An advantage of convolutional neural networks in sequence analysis is their ability to detect motifs wherever they occur in a sequence in the same way an image network may learn to identify the wheel of a car or a human eye. These networks are trained directly from sequence data, and require no expert labeling.

In this paper we tested the ability of CNNs to predict promoter strength directly from DNA sequence. We used a CNN to model the Urtecho et al.[11] dataset with comparable success to published models. To assess the ability of a CNN model to handle more complicated and diverse promoter sequences without manual labeling of features, we evaluated the architecture on a non-repetitive promoter library we created for synthetic biology. Promoter transcription rates were measured by combining synthesis of barcoded oligopools, DNA assembly, pooled cellular transformations, and next-generation sequencing of harvested DNA and RNA samples. The resulting analysis yields the relative differences in transcription rate for 4350 promoter sequences, which varied across a 10^6-fold range. The library was designed so that no sequence had more than 10 consecutive nucleotides in common with another. We found that a CNN is able to perform better than conventional models at predicting promoter strength on this complex promoter dataset. By using perturbation sensitivity and decomposition analysis methods developed for machine learning to interrogate the black box of the trained CNN, we determined the features of the promoter that were most impactful in driving transcription.

Results

Comparing a convolutional neural network to existing state of the art promoter strength predictions

Urtecho et al. built a dataset of promoter strengths by creating promoters from combinations of 35 distinct sequence elements (eight -10 hexamers, eight -35 hexamers, eight spacers, eight backgrounds, and three UPs) and measuring transcription with RNA/DNA-Seq.[11] This dataset contained 10898 sequences encompassing a 100-fold range of promoter strength. They fitted a simple feed-forward neural network model of promoter strength based on the presence or absence of each the 35 elements in a given sequence. [Figure 1a] This model performed well ($R^2 = 0.96$ trained on 50% of the data), but it relied on the limited input domain of 35 components and labeling of the components in each sequence. Follow-up work by Einav and Phillips[12] built a successful biophysical model for the transcription data ($R^2 \sim 0.9$ depending on modeling choices), but also relied on fitting parameters for the constrained set of promoter components.

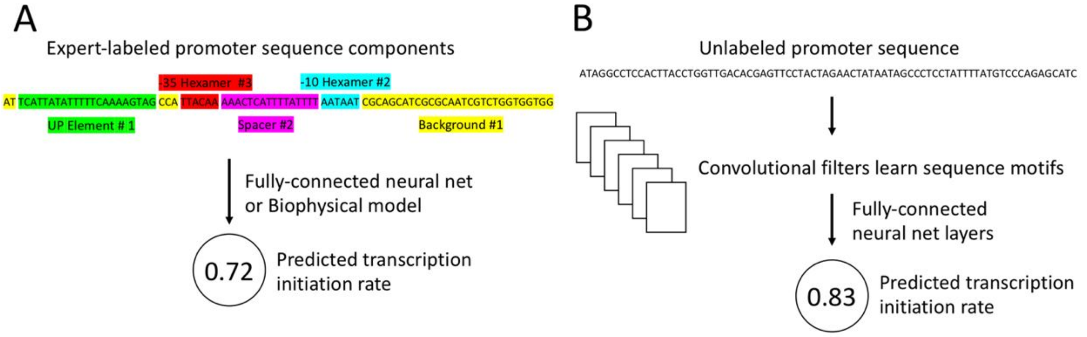
A
Expert-labeled promoter sequence components
-35 Hexamer #3
-10 Hexamer #2
AT TCATTATATTTTTCAAAAGTAG CCA TTACAA AAACTCATTTTATTTT AATAAT CGCAGCATCGCGCAATCGTCTGGTGGTGG
UP Element # 1
Spacer #2
Background #1
Fully-connected neural net or Biophysical model
0.72
Predicted transcription initiation rate
B
Unlabeled promoter sequence
ATAGGCCTCCACTTACCTGGTTGACACGAGTTCCTACTAGAACTATAATAGCCCTCCTATTTTATGTCCCAGAGCATC
Convolutional filters learn sequence motifs
Fully-connected neural net layers
0.83
Predicted transcription initiation rate

Figure 1: Comparison of promoter transcription initiation rate modeling strategies. A) Previous models have relied on expert-labeling of promoter components, and are limited to the defined inputs. B) Sequence-based models like a CNN learn rules directly from the sequence, do not rely on expert labeling, and are not limited to the defined inputs.

We applied a sequence-based CNN model to the Urtecho dataset [Figure 1b]. Using 10-fold cross validation in order to evaluate the model on data not seen in training, our CNN predicted promoter transcription strength with average $R^2 = 0.90$, comparable to the above models that fit expert-labeled sequence elements (Figure 2a). This in and of itself is an interesting result, as training the model takes only 2 inputs: the DNA sequences of the promoters and the measured transcription level. The model architecture is relatively simple, with only one convolutional layer and one fully-connected dense layer, and still achieves high fidelity. Like the previous models of this dataset, the skewed distribution of transcription levels where most promoters are weak (Figure 2b) likely contributes to the good fit of our CNN model to the observations (Figure 2c).

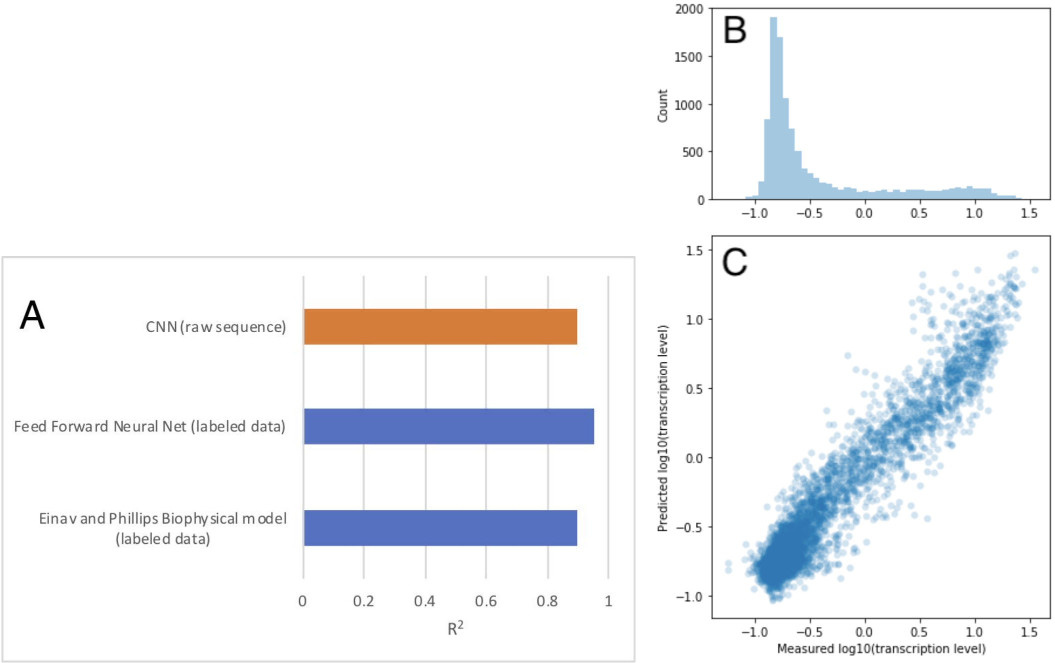

Figure 2: Model performance predicting promoter strength for a library composed of combinations of discrete promoter components.[11] A) R^2 for the sequence-based CNN model compared to published models fitting expert-labeled sequence components. B) Distribution of promoter transcription rate (log10-transformed) in this dataset. C) CNN model predictions for held-out test data versus the observed transcription rate.

Predicting promoter strength for more complex sequences

Diversity in promoter sequences, particularly avoiding sequence repetition, is an important goal in synthetic biology in order to prevent homologous recombination between engineered genes.[15] The biophysical model and simple feed-forward neural net architectures that currently represent the state of the art for promoter strength prediction only function on a limited domain of inputs- the specific set of promoter sequence building blocks contained in the dataset on which the model was trained. As the diversity of sequence components increases, these models require the addition of a new parameter for each new sequence element. Including non-repetitive promoters quickly enlarges the input domain, making published methods intractable.

A sequence-based model can handle varied promoters without explicitly fitting parameters for each element in the input sequence space. We created a non-repetitive promoter toolbox for synthetic biology encompassing 4350 E. coli σ^{70} promoter sequences such that no sequence had more than 10 consecutive nucleotides in common with another.[16] We kept the spacer length constant at 17nt and promoter length constant at 78nt, but varied GC content and sequence throughout the promoter. In particular, we included a variety of sequences deviating from the consensus -35 and -10 hexamer regions of the promoter, ranging from zero to twelve mismatches from consensus in these regions. This more complex library yielded a wide range of promoter strengths encompassing a 10^6-fold range of transcription levels (Figure 3a).

On this more complex dataset the CNN model performed better than more traditional methods at predicting promoter strength, achieving $R^2 = 0.61$ on 10-fold cross validation (Figure 3b). It's impossible to compare this directly to the simple feed forward neural network used by Urtecho et al.[11], since the domain can't be expressed as a small number of inputs.

A relevant conventional model comparison is a bag of words linear model. This determines inputs analogous to promoter component labels by sliding a window over a promoter sequence to generate 'words' of sub-sequence. It then uses LASSO regularization to reduce the number of parameters and labels included in the linear model to prevent overfitting. We used this model with a sliding window length of 6 (to capture an entire hexamer) and set the regularization constant alpha to 0.005 after optimizing with a hyperparameter sweep. The bag of words model achieved an average $R^2 = 0.42$ with 10-fold cross validation (Figure 3c). For comparison, a linear model using a single expert-labeled feature as input for each promoter that simply counts the number of mismatches from consensus in the -10 and -35 conserved hexamers, a value ranging from 0 to 12, achieved an average $R^2 = 0.38$ with 10-fold cross validation. Because the bag of words model incorporates these hexamers as possible words, this suggests that the majority of the information it uses to predict promoter strength is contained in these hexamer regions of the promoter. Withholding this information from the bag of words model by replacing the -35 and -10 hexamers in a promoter sequence with a masked sequence of non-nucleotide characters, the bag of words model performance fell to average $R^2 = 0.02$.

The CNN model given these same masked promoter sequences predicted transcription strength with average $R^2 = 0.11$. One interpretation of this is that the CNN learned something from the sequence beyond the identity of the -35 and -10 hexamer sequences and the interaction of these components with other parts of the promoter. We next examine the CNN more directly to better understand the features it identified as important for promoter strength.

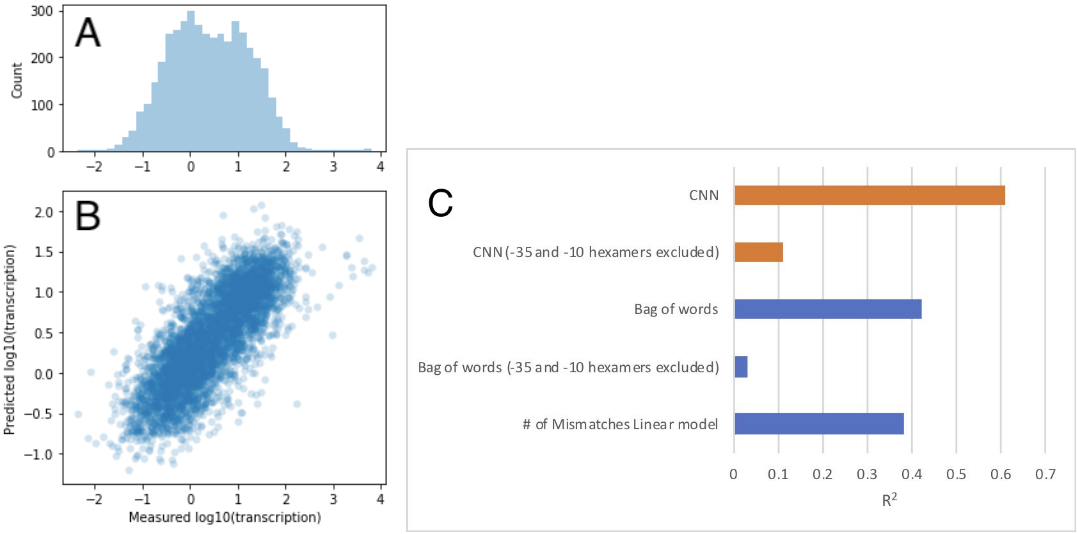

Figure 3: Model performance predicting promoter strength for a non-repetitive promoter library[16] A) Distribution of promoter transcription rate (log10-transformed) in this dataset. B) CNN model predictions for held-out test data versus the observed transcription rate. C) R^2 for the sequence-based CNN model compared to published models fitting expert-labeled sequence components.

Investigating the black box of a neural network: perturbation and decomposition analysis

Because of the internal complexity of neural networks, they are often thought of as black boxes and difficult to interpret. However, much research has focused on explaining and visualizing their decision making. We applied two general techniques to interrogate the CNN model and learn what patterns drive its predictions. First, we used perturbation sensitivity analysis to systematically alter a promoter sequence and look at the change in model predictions for these modified sequences.[17-18] Second, we used layer-wise relevance propagation as a means of decomposing the impact of particular sequence bases on the model's prediction.[19]

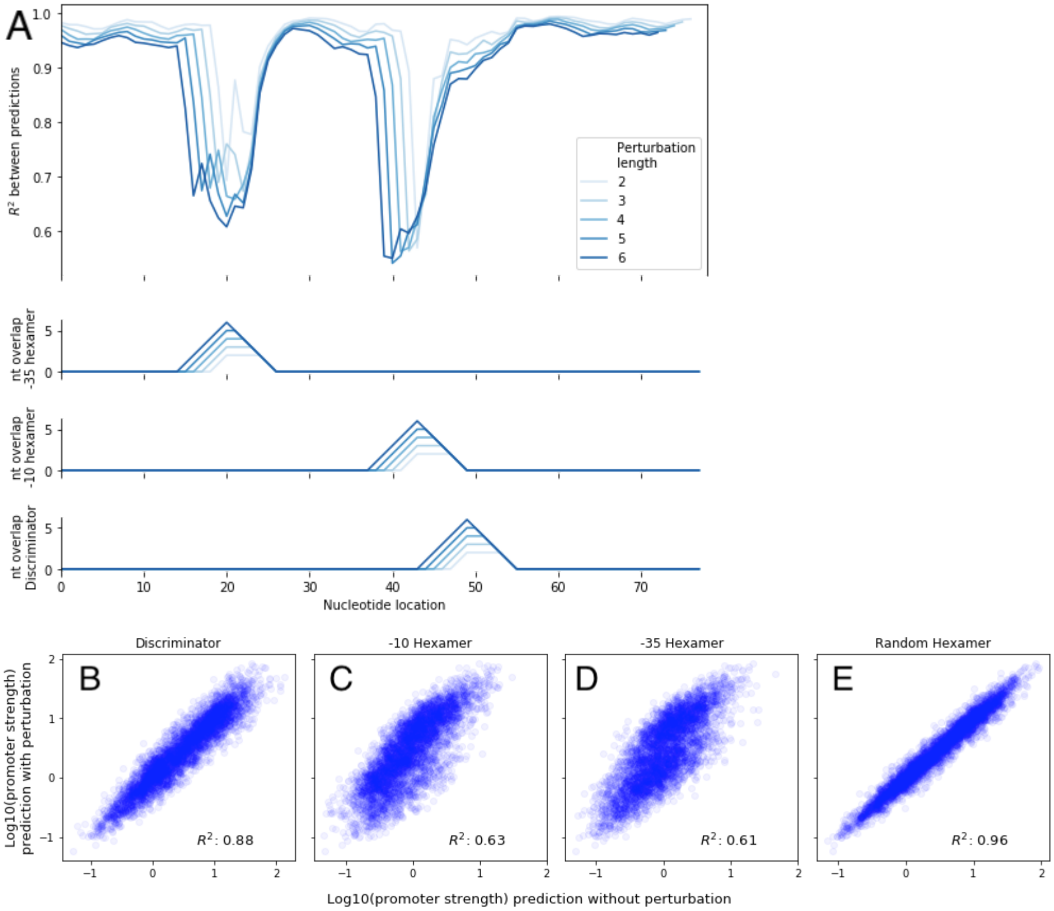

Figure 4: The effects of perturbations in promoter sequence on model predictions of promoter strength. A) Randomizing the sequence in a sliding window of varying length and comparing the CNN-predicted promoter strength for the base sequence versus the perturbed sequence reveals the parts of the promoter that are most important to the CNN. These include the -35 and -10 hexamer and the discriminator. Focusing the perturbation on a portion of the promoter shows the distribution of the effect on individual promoter sequences. Randomizing the hexamer sequence corresponding to the discriminator (B), the -10 hexamer (C), and the -35 hexamer (D) have much larger impacts than randomizing a random hexamer not included in the above (E).

For perturbation analysis, we trained the CNN model on promoter sequences. However, because we are interested in how the model learns from the sequences on which it has been trained, we do not hold out sequences for testing. We compared the CNN-predicted promoter strength of a sequence to the prediction for the same sequence with an added perturbation- a randomized segment of nucleotides. By sliding a window of perturbation across a sequence, and testing many sequences in this way, we gain a picture of which regions of the promoter are most important to the CNN model based on the magnitude of the prediction difference caused by the perturbations.

Figure 4a shows the average effect on predictions of sliding windows of different length perturbations across all of the training sequences in the non-repetitive promoter dataset. The correlation between promoter strength predictions with and without perturbation is lowest when the perturbation overlaps 3 distinct regions of the promoter: the -35 hexamer, the -10 hexamer, and the discriminator region of the promoter just downstream of the -10 hexamer. This suggests these parts of the promoter sequence are most important to the CNN predictions. By focusing perturbations in these regions, we evaluated the distribution of this impact. We randomized the 6nt segment of each promoter corresponding to these regions (Figure 4c-e) as well as a random 6nt segment outside of these regions as a control (Figure 4f). This highlights the importance of these hexamer regions to the CNN prediction relative to the rest of the promoter.

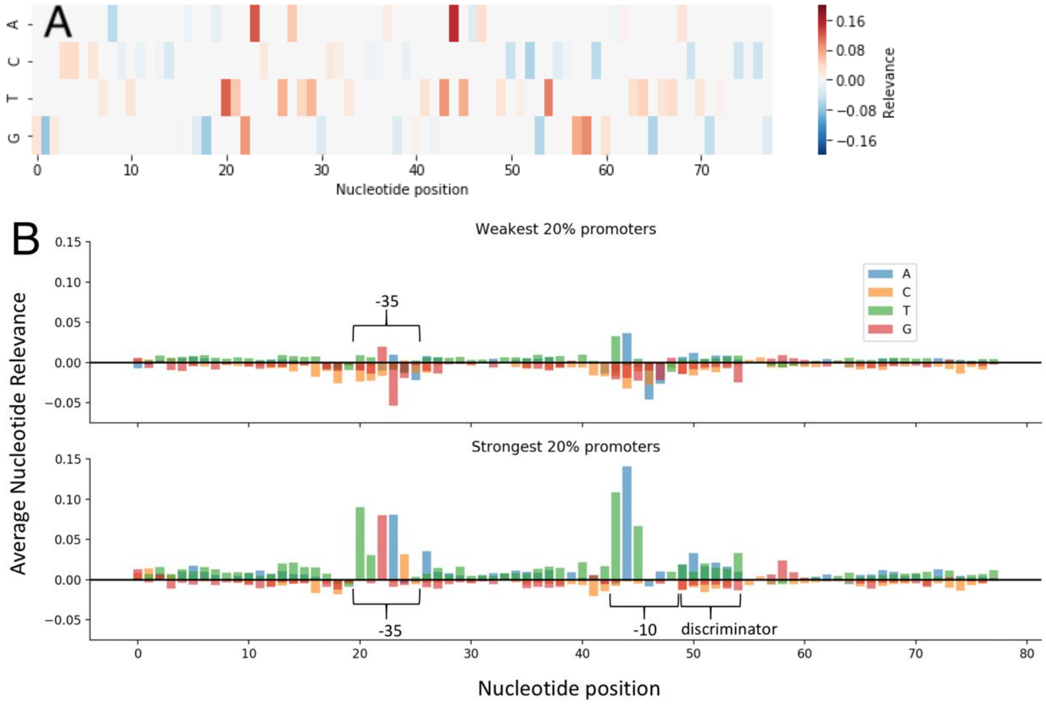

Figure 5: Decomposition analysis of CNN model weights determining prediction logic. A) a heatmap showing the relative relevance of nucleotides and positions for a single promoter sequence to the CNN prediction. B) An aggregation of binned nucleotide relevance scores for the bottom and top 20% bins of promoter strengths. This shows the importance of particular sequence segments and motifs the CNN predictions.

While perturbation analysis is informative about regions of the input important to the model on aggregate, it has a couple of weaknesses. Perturbation by changing a sequence is not an efficient way of determining the contributions of individual bases to a specific promoter strength prediction. Random sequence perturbations also represent artifacts- sequence patterns that may not be like anything the model was originally trained on.

Another tool for neural network explanation is Layer-wise Relevance Propagation (LRP)- a gradient-based method that identifies important bases in a sequence by running a backward pass on the network. During the backwards pass, neurons that contribute the most to higher layers (and therefore the model's output) receive most relevance from them.[19] This process determines for each base a unitless relevance score reflecting the sign and magnitude of the impact of that base on the model's prediction, often represented as a heatmap.

Figure 5a shows a representative relevance heatmap for a single promoter sequence. Nucleotides that increased the prediction of promoter strength are shown in red, and those that decreased the prediction are shown in blue. By binning promoter sequences by strength and calculating the average LRP relevance score for each nucleotide in the grouped promoters, we visualized the positions and bases most responsible for changing the CNN model predictions of transcription strength (Figure 5b). This method showed that the first 5 bases of the -35 conserved hexamer led to the prediction of a strong promoter, as well as the first 3 bases of the -10 conserved hexamer. The fact that perfect consensus with the conserved hexamers was not associated with the strongest promoters is consistent with previous research, with the postulated explanation that perfect consensus leads to over-tight polymerase binding and less transcription.[11] LRP also revealed that AT bases in the discriminator region, but not GC, led to predictions of strong promoters. In contrast, the presence of a G in the 4th position of the -35 conserved hexamer (consensus TTGACA) is the single variant that most drives predictions of weak promoters. While some of these observations are not novel, each is a testable hypothesis that was generated from sequence data input without expert labels or knowledge. The ability to efficiently identify these points of focus is another advantage making sequence-based models an interesting direction for further research in promoter modeling.

Conclusions

A convolutional neural network model learned to predict promoter strength from sequence alone. Without expert knowledge or labeling, it identified the consensus conserved -35 and -10 hexamer regions, as well as the discriminator element, as key contributors to σ^{70} promoter strength. A CNN was as successful as published models in predicting transcription rate on a promoter strength dataset that measured transcription for promoters composed of permutations of 35 distinct components. It was more successful than conventional models in predicting transcription for varied, non-repetitive promoters for which published methods are not tractable.

The CNN's ability to independently learn biologically-relevant information directly from sequence makes it appealing for further prediction optimization and research into generalizability. In the future, this approach may be useful for synthetic promoter design, as well as for sequence feature identification.

Methods

Promoter strength datasets

The promoter strength dataset from Urtecho et al.[11] was obtained from NCBI: GSE108535. RNA_exp_average for barcodes were joined with promoter sequences from this dataset. 10898 promoters with non-zero RNA_exp_average and not labeled as negative controls were included.

The non-repetitive σ^{70} promoter library was designed and transcription strength was measured in previous work.[16] Briefly, promoter oligos were spliced into vectors and transformed into *E. coli.* Total RNA and DNA were extracted from 2 biological replicates and sequenced on an Illumina HiSeq. Transcription rate for a promoter was measured by counting RNA reads for the promoter barcode and normalizing with respect to the total RNA reads and the DNA reads for the promoter barcode. This transcription rate was further normalized against the J23100 promoter.

CNN Model Architecture and Evaluation

The convolutional neural network was implemented in Keras using a 2D convolutional layer, a fully connected dense layer, and a single node output layer with linear activation representing the promoter strength prediction. Convolutional kernels were 7x6: the 7 nucleotide channels represent the 4 DNA nucleotides plus J for start, O for end, and X for padding, and a filter width of 6 was chosen to fully capture a sequence hexamer motif within a kernel. The promoter sequence was therefore modeled as a 7-channel matrix with width N. The N dimension is the promoter length-78 for [16] and 150 for [11] - plus one full convolutional filter-width of padding and a start/end nucleotide marker on both sides of the sequence. The log10 transform of the transcription rate was used as the independent variable.

K-fold cross validation was used with K = 10 and non-overlapping test sets in the fold groups to ensure that the model was tested on all of the data in one of the iterations, but not tested on any data included in the training for that iteration. Average R^2 was calculated as the mean of these 10 iterations.

Three methods were used to prevent overfitting. Small minibatches (n = 4) were used. Neuron dropout was included in training after the convolutional layer and fully connected layer with P = 0.3. Early stopping terminated model training and used the weights from the best epoch if performance on a validation dataset, consisting of 10% of the training data, did not improve for 6 consecutive epochs. Otherwise training continued for a max of 50 epochs.

For decomposition analysis, the Innvestigate toolkit for Keras[19] was used to calculate individual nucleotide relevance to the model prediction for a sequence. The LRPEpsilon method was used with epsilon = 0.01.

References

[1] Feklístov A, Sharon BD, Darst SA, and Gross CA (2014) Bacterial Sigma Factors: A Historical, Structural, and Genomic Perspective. Annu. Rev. Microbiol 68, 357–376. DOI: 10.1146/annurev-micro-092412-155737

[2] Gruber TM, and Gross CA (2003) Multiple sigma subunits and the partitioning of bacterial transcription space. Annu. Rev. Microbiol 57, 441–466. DOI: 10.1146/annurev.micro.57.030502.090913

[3] Campbell EA, Muzzin O, Chlenov M, Sun JL, Olson CA, Weinman O, Trester-Zedlitz ML, and Darst SA (2002) Structure of the Bacterial RNA Polymerase Promoter Specificity σ Subunit. Mol. Cell 9, 527–539. DOI: 10.1016/s1097-2765(02)00470-7

[4] Liu M, Tolstorukov M, Zhurkin V, Garges S, and Adhya S (2004) A mutant spacer sequence between -35 and -10 elements makes the Plac promoter hyperactive and cAMP receptor protein-independent. PNAS 101, 6911–6916. DOI: 10.1073/pnas.0401929101

[5] Estrem ST, Gaal T, Ross W, and Gourse RL (1998) Identification of an UP element consensus sequence for bacterial promoters. PNAS 95, 9761–9766. DOI: 10.1073/pnas.95.17.9761

[6] Carr SB, Beal J, and Densmore DM (2017) Reducing DNA context dependence in bacterial promoters. PLoS One 12, e0176013. DOI: 10.1371/journal.pone.0176013

[7] Winkelman JT, Chandrangsu P, Ross W, and Gourse RL. (2016) Open complex scrunching before nucleotide addition accounts for the unusual transcription start site of E. coli ribosomal RNA promoters. PNAS 113 (13) 1787-1795. DOI: 10.1073/pnas.1522159113

[8] Ross W, Aiyar SE, Salomon J, and Gourse RL (1998) Escherichia coli promoters with UP elements of different strengths: modular structure of bacterial promoters. J. Bacteriol 180, 5375–5383. [PubMed: 9765569]

[9] Meng H, Wang J, Xiong Z, Xu F, Zhao G, & Wang Y (2013) Quantitative design of regulatory elements based on high-precision strength prediction using artificial neural network. PLoS One, 8 e60288. DOI: 10.1371/journal.pone.0060288

[10] Mutalik VK, Nonaka G, Ades SE, Rhodius VA, and Gross CA (2009) Promoter Strength Properties of the Complete Sigma E Regulon of Escherichia coli and Salmonella enterica. J. Bacteriol 191, 7279–7287. DOI: 10.1128/JB.01047-09

[11] Urtecho G, Tripp AD, Insigne KD, Kim H, and Kosuri S (2019) Systematic Dissection of Sequence Elements Controlling σ^{70} Promoters Using a Genomically Encoded Multiplexed Reporter Assay in Escherichia coli. Biochemistry 58 (11), 1539-1551 DOI: 10.1021/acs.biochem.7b01069

[12] Einav T, Phillips R (2019) How the avidity of polymerase binding to the –35/–10 promoter sites affects gene expression. PNAS, 116 (27) 13340-13345; DOI: 10.1073/pnas.1905615116

[13] Alipanahi B, Delong A, Weirauch MT, and Frey, BJ (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology, 33, 831. DOI: 10.1038/nbt.3300

[14] Zhou J and Troyanskaya PG (2015) Predicting effects of noncoding variants with deep learning–based sequence model. Nature Methods 12, 931–934. DOI: 10.1038/nmeth.3547

[15] Jack BR, Leonard SP, Mishler DM, Renda BA, Leon D, Suárez GA, and Barrick JE (2015) Predicting the Genetic Stability of Engineered DNA Sequences with the EFM Calculator. ACS Synthetic Biology 2015 4 (8), 939-943. DOI: 10.1021/acssynbio.5b00068

[16] Hossain A, Halper SM, Cetnar DP, Reis AC and Salis HM. Automated Design of Thousands of Highly Non-Repetitive Genetic Parts for Engineering Evolutionary Robust Genetic Systems (In review, Nature Biotechnology)

[17] Samek W, Binder A, Montavon G, Lapuschkin S, and Müller K (2017) Evaluating the Visualization of What a Deep Neural Network Has Learned. IEEE Transactions on Neural Networks and Learning Systems, 28 (11), 2660-2673. DOI: 10.1109/TNNLS.2016.2599820

[18] Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLoS ONE 10 (7): e0130140. DOI: 10.1371/journal.pone.0130140

[19] Alber, Maximilian & Lapuschkin, Sebastian & Seegerer, Philipp & Hägele, Miriam & Schütt, Kristof & Montavon, Grégoire & Samek, Wojciech & Müller, Klaus-Robert & Dähne, Sven & Kindermans, Pieter-Jan. (2018). iNNvestigate neural networks! http://arxiv.org/abs/1808.04260

Acknowledgements

This work was supported by the DARPA Synergistic Discovery and Design (SD2) program. We thank Jedediah Singer and Scott Novotney at Two Six Labs for helpful conversations.