LegNet: resetting the bar in deep learning for accurate prediction of promoter activity and variant effects from massive parallel reporter assays

Dmitry Penzar[1,2,3,*,+], Daria Nogina[4,*], Georgy Meshcheryakov[2], Andrey Lando[5], Abdul Muntakim Rafi[6], Carl de Boer[6], Arsenii Zinkevich[1,4], Ivan V. Kulakovskiy[1,2,+]

[1] Vavilov Institute of General Genetics, 119991, Moscow, Russia
[2] Institute of Protein Research, 142290, Pushchino, Russia
[3] Institute of Translational Medicine, Pirogov Russian National Research Medical University, 117997, Moscow, Russia
[4] Faculty of Bioengineering and Bioinformatics, Lomonosov Moscow State University, 119991, Moscow, Russia
[5] Yandex N. V., 119021, Moscow, Russia
[6] School of Biomedical Engineering, University of British Columbia, V6T 1Z4, Vancouver, BC Canada
[*] - equal contribution
[+] - corresponding authors: dmitrypenzar1996@gmail.com, ivan.kulakovskiy@gmail.com

Abstract

Parallel reporter assays provide rich data to decipher gene regulatory regions with deep learning. Here we introduce LegNet, a convolutional network architecture that secured the first place for our autosome.org team in the DREAM 2022 challenge of predicting gene expression from gigantic parallel reporter assays. To construct LegNet, we drew inspiration from EfficientNetV2 and reformulated the sequence-to-expression regression problem as a soft-classification task. Here, with published data, we demonstrate that LegNet outperforms existing models and accurately predicts gene expression *per se* as well as the effects of sequence alterations, such as single-nucleotide variants.

Keywords: parallel reporter assays, deep learning, gene regulation, regulatory variants, sequence variant effects, LegNet, promoter, convolution, one cycle learning rate

Introduction

The basic level of gene expression regulation in eukaryotes, the mRNA transcription, is controlled by transcription factors (TFs), which bind cis-regulatory regions, promoters, and enhancers, and affect the assembly and functioning of the mRNA transcription machinery [1]. The transcription factors can recognize particular DNA patterns, allowing them to act at particular genomic addresses and affect particular sets of target genes [2]. It is a longstanding challenge in computational biology to decipher the sequence-level regulatory code completely, from the prediction of individual TF binding sites of varying affinity to the identification of composite elements [3] and complete sequence-level annotation of promoters and enhancers.

A commonly accepted approach is bottom-up, where binding specificities of individual transcription factors are profiled with various TF-centric techniques [4], revealing TF-specific binding motifs. With individual motifs at hand, the higher-order regulatory grammar can be studied *in silico* [5]. However, genome-level analysis is hampered by numerous confounding factors, hence direct experiments are required to explicitly profile the binding preferences of TF complexes [6,7]. It remains challenging to apply the knowledge obtained *in vitro* to genomic regulatory regions and performing direct experiments for all TF combinations also remains hardly realistic.

A possible alternative approach to resolving the rules of regulatory grammar comes with massive parallel reporter assays [8], which can profile the activity of dozens of millions of synthetic or genomic regulatory sequences [9,10] in a single experiment. The resulting data are uniform and diverse enough to allow an orthogonal approach: properly trained biochemical [9] and advanced machine learning models [11,12] can provide quantitative and highly accurate predictions of regulatory activity just from the DNA sequence. In terms of machine learning, two questions remain unanswered in this setting. First, whether the current prediction errors are comparable to experimental noise or if there remains room for improvement of the computational models. Second, whether the high-level deep learning architectures such as attention transformers are truly necessary for modeling short regulatory regions, or if the task can be handled by advanced convolutional networks.

Here we introduce the LegNet convolutional network that our autosome.org team used to secure 1st place in the DREAM 2022 challenge of predicting expression yield from gigantic parallel reporter assay (GPRA) data. By using previously published GPRA data, we demonstrate that LegNet outperforms existing methods in predicting both expression and sequence variant effects.

Results and Discussion

LegNet is a deep neural network designed during the DREAM 2022 challenge of predicting gene expression from millions of promoter sequences analyzed in gigantic parallel reporter assays. The design of LegNet is based on a fully-convolutional neural network architecture inspired by EfficientNetV2 [13] with some features from DenseNet [14] and additional custom blocks. In the DREAM2022 promoter expression challenge, LegNet scored first both in the overall assessment and in multiple individual subchallenges covering different categories of promoter sequences (*e.g.* high- and low-expressed promoters, synthetic promoters, *etc*). To further prove the reliability of our approach, here we applied LegNet to the previously published GPRA results [11] (**Figure 1, A**) and evaluated its performance in predicting expression *per se* as well as estimating the effect of single-nucleotide variants (**Figure 1, B**).

The dataset of Vaishnav *et al*. [11] contains more than 30 million measurements of promoter activity for yeast culture grown in a complex medium (YPD) and 20 million measurements for a defined medium (SD-Ura). For these data, the experimental setup was the same as in the DREAM 2022 challenge. For YPD and SD-Ura datasets, LegNet was trained separately with the same architecture and hyperparameters as originally in the DREAM 2022 promoter challenge except for the number of epochs, which was adjusted to account for the increased volume of the training data. The original authors' train-test split was used for model training and evaluation, see Methods for details.

Key features of LegNet

LegNet is an EfficientNetV2-based fully convolutional neural network [13] employing several domain-specific ideas and improvements to reach accurate expression modeling and prediction from a DNA sequence. The data preparation was extended over the standard one-hot encoding approach by discerning whether a target sequence was observed in the experiment only once (a singleton) or multiple times, as the singletons constitute more than half of the training data but eventually provide noisier expression estimates. Next, we included a dedicated channel denoting promoter orientation to perform training-time data augmentation with reverse complementary sequences properly. Finally, we reformulated the expression prediction as a soft-classification problem: LegNet was trained to predict not the single expression value but a vector of expression bin probabilities. At the model evaluation stage, the predicted probabilities are multiplied by bin numbers to convert the vector into a single predicted expression value, see Methods for details.

LegNet improves prediction of promoter expression

First, we evaluated LegNet in predicting native promoter expression for GRPA data from yeast grown in complex (YPD) or defined (SD-Ura) media. In both cases, LegNet demonstrated high and consistent performance, scoring significantly higher than the state-of-the-art transformer model published along with the GPRA data by Vaishnav et al. [11] (**Figure 2**). Note that the prediction "wall" encountered at around expression levels 4 (complex) and 2.5 (defined) is a known issue with the training data also learned by models of [9,11], which is likely caused by the cell sorter having limited signal-to-noise ratio in this range or inadvertently truncated distribution. We also compared LegNet against earlier deep learning approaches tested in [11] (**Figure S1**), thus highlighting the gap between LegNet (~0.96-0.98 Pearson and Spearman correlation against the ground truth test data) and conventional deep learning models such as DeepSEA and DanQ (correlations around ~0.92-0.94).

LegNet delivers accurate estimates of sequence variant effects

In the DREAM challenge, LegNet was highly successful in estimating the expression of promoters with single-nucleotide variants. To demonstrate it with independent data and further explore LegNet reliability in predicting the effects of multiple nucleotide substitutions, we utilized the GRPA data capturing expression divergence under random genetic drift. For 1,000 unique random promoter sequences, Vaishnav et al. randomly introduced single-nucleotide mutations for three generations and measured the promoter expression in each.

We evaluated the capability of LegNet to quantitatively estimate the difference between expression for original and mutated promoter sequences depending on the number of nucleotide substitutions (1,2, or 3), and compared the performance with the state-of-the-art transformer model of Vaishnav et al. Estimating the single-nucleotide variant effects was the most difficult, but in all scenarios, LegNet showed a consistent and significant increase in prediction performance (**Figure 3**).

Methods

Experimental data overview

In this study, we used previously published results of gigantic parallel reporter assays (GPRAs) [11] that included (respectively) 30 and 20 million of promoter-driven expression measurements in the yeast *S. cerevisiae* cultured in YPD medium (complex medium: yeast extract, peptone, and dextrose) and SD-Ura medium (synthetic defined medium lacking uracil). In the GPRA experiment, yeast cells are transformed with a construction containing 80bp random promoters. These constructions contain the YFP gene regulated by such promoters and the RFP gene, which is expressed constitutively. Yeast cells are then sorted into 18 expression bins (numbered 0 to 17) with regard to their logarithmic relative protein fluorescence. The expression estimate for a particular promoter sequence is calculated as a weighted average of the numbers of expression bins where it is observed [9].

The train and test datasets were derived from the original Vaishnav *et al.* paper [11]. A total of 20,616,659 (defined medium) and 30,722,376 (complex medium) random promoter sequences were used to train LegNet in each case. The test data were collected in an independent experiment and included only the high quality measurements obtained for native (*i.e.*, present in the yeast genome) promoter sequences (N=3928 for the complex medium, N=3977 for the defined medium), see the details in [11]. A subset of the test data containing 3733 promoter sequences assessed in yeast cultures in the complex medium was used to compare the performance against conventional deep learning methods according to [11]. To evaluate how LegNet captures effects of minor alterations of promoter sequences, we used the 'genetic drift' data of [11] where 1 to 3 single-nucleotide substitutions were introduced into 1,000 random starting sequences assessed in both defined and complex media. The respective GPRA data are available in GEO (https://www.ncbi.nlm.nih.gov/geo/) under accession numbers GSE104878 and GSE163045.

Sequence-to-expression as a soft-classification problem

A straightforward formal description of a machine learning problem arising from a GPRA experiment is a regression of a single real value (the expression defined by the cell sorting bin) from a fixed-length DNA sequence. However, a direct approach cannot benefit from the nature of the experimental data. We have reformulated the sequence-to-expression regression problem as a soft classification task by transforming expression estimates into class probabilities. Given a measured expression e (the average observed bin number), we heuristically assume that the real expression is a normally distributed random variable (see Figure 2b in [9]):

$$\rho \sim N(\mu = e + 0.5,\ sd = 0.5).$$

In this approach, for each class i from 1 to 16 defined by an original measurement bin, a probability of the class is the cumulative probability to fall into $[i, i+1)$ range, with 0 and 17 classes (bins) represented by special ranges of $(-\infty, 1]$ and $[17, +\infty)$, respectively.

Thus, for the model loss, we selected the Kullback–Leibler divergence between the distribution derived from the training data and the model output vector containing 18 probabilities corresponding to each class (bin). To obtain a predicted expression value for a sequence during the expression inference step (in model validation or test scenario), the

predicted probabilities were multiplied by the corresponding bin numbers. This model layer, if joined with softmax, is called soft-argmax [3], see **Figure S2**:

$$expression = \sum_{i=0}^{17} i \cdot p_i \,.$$

Adapting GPRA data for a deep learning model

First, we padded the sequences from the 5' end with the respective constant segments of the plasmids to achieve the total fixed length of 150 base pairs. Next, sequences were encoded into 4-dimensional vectors with one-hot encoding.

We considered the integer expression estimates to belong to the *singleton* promoters observed only once across all bins. The singletons are more likely to have noisier expression estimates, compared to other promoters with non-integer expression values obtained by averaging two or more observations. To supply this information to the model, we used a binary `is_singleton` channel (1 for singletons, 0 for other training sequences). The final predictions for evaluation were made by setting `is_singleton=0`. Since the regulatory elements could be asymmetric with regard to their strand orientation and position relative to the transcription start sites, different scores are expected for the direct and reverse complementary orientation of a particular sequence. Therefore, the training data were augmented by providing each sequence both in native and reverse complementary form, explicitly specifying 0 and 1, respectively, in an additional `is_reverse` channel. We also performed the test-time augmentation by averaging the predictions made for direct (`is_reverse=0`) and reverse complementary (`is_reverse=1`) input of each promoter. A scheme of the input sequence representation is shown in **Figure S3**.

LegNet architecture

Our model (**Figure S1, A**) is based upon a fully-convolutional neural network architecture inspired by EfficientNetV2 [13] with selected features from DenseNet [14] and additional custom blocks.

The first LegNet block (Stem block) is a standard convolution with `kernel_size=7`, followed by BatchNorm and SiLU activation (**Figure S1, B**). The output of the first block is passed to the sequence of six convolution blocks of EfficientNet-like structure (**Figure S1, C**) but using the grouped convolution instead of the depthwise of the original EfficientNetV2. The standard residual connections were replaced with residual channel-wise concatenation (**Figure S1, C**). All convolutions are used with padding set to the mode `'same'`. Convolutions followed by batch normalization were trained with no bias. The resize block is of the same structure as the stem block used at the start of the network (**Figure S1, B**).

The Squeeze and Excitation (SE) block used as a part of EfficientNet-like block is a modification of that of the original EfficientNetV2 (**Figure S1, E**). The number of parameters in the bilinear block inside of SE block is reduced with low-rank representation of the parameterized tensor via canonical polyadic decomposition implementation provided by the TensorLy [15] library.

The final block consists of a single convolutional layer with `kernel_size=1` followed by channel-wise Global Average Pooling and SoftMax activation (**Figure S1, D**). We used 256 channels for the first block and [128, 128, 64, 64, 64, 64] channels for six EfficientNetV2-like blocks, respectively. The total number of parameters in our model is 1,852,846.

Model training procedure

To train our neural network, we used One Cycle Policy [16] with FastAI [17] modifications: (1) two phases (instead of the original three), (2) the cosine annealing strategy instead of the linear one, (3) the AdamW optimizer (`weight_decay=0.01`) instead of the SGD with momentum. The parameters of the One Cycle Policy were selected using 1/10 of the training data of the DREAM2022 promoter expression challenge. To select the max learning rate (0.005) for the One Cycle Learning Rate Policy, we used the LR-range test as suggested in [18].

Each epoch consisted of 1000 batches of size 1024. The model was trained for 150 epochs (defined medium) and 300 epochs (complex medium) achieving a reasonable trade-off between training time and validation variance. For the final model, we used the hyperparameters based on the validation on the last k-fold (10th) of the training data, but the final model was trained from scratch on the whole training dataset.

We used the same weight initialization as in EfficientNetV2 [13]. The training of the final model took about 12 hours for the defined medium model and 24 hours for the complex medium model using the NVIDIA RTX A5000 GPU and PyTorch version 1.11.0+cu113.

Conclusions

In this study, we presented LegNet, a new deep-learning approach for predicting promoter expression from DNA sequence. With the data from gigantic parallel reporter assays, we have demonstrated LegNet efficiency in predicting expression *per se* as well as quantitatively estimating the effects of sequence variants, and have shown that LegNet significantly outperforms conventional models and the previous state-of-the-art transformer model. Thus, while today the researchers' preference is biased toward complex architectures, we conclude that the fully convolutional networks should be considered as a solid method of choice for the computational modeling of short gene regulatory regions and predicting sequence alteration effects.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

Project name: LegNet
Project home page: https://github.com/autosome-ru/LegNet. The GitHub repository includes the Python code to reproduce the results presented in this study and a Jupyter Notebook tutorial.
Archived version: https://github.com/autosome-ru/LegNet/releases/tag/0.0.1.
Operating system(s): Linux

Programming language: Python>=3.9
Other requirements: NumPy >= 1.2.3, PyTorch >=1.12.0, Pytorch-Ignite>=0.4.9
License: MIT

Train and test GPRA data were taken from Vaishnav et al. [11] Zenodo record: https://zenodo.org/record/4436477. The variant effect analysis was perform with the GPRA data available in GEO (https://www.ncbi.nlm.nih.gov/geo/) under accession numbers GSE104878 and GSE163045. The performances of DeepSEA, DeepAtt, DanQ and attention-based model were given according to the following GitHub repository that accompanied Vaishnav et al. study: https://github.com/1edv/evolution.

Competing interests

The authors declare that they have no competing interests.

Funding

The study was primarily supported by RSF grant 20-74-10075 to IVK. AZ was supported by a personal fellowship from Non-commercial Foundation for Support of Science and Education "INTELLECT". The funding body had no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Authors' contributions

DP, DN, GM, AL, and AZ developed and tested LegNet. AMR and CB performed GPRA data preprocessing and interpretation of results. IVK supervised the study. All authors participated in the paper preparation.

Acknowledgments

We thank Google TRC for providing free access to computational resources used in the model development and assessment. The LegNet logo was generated with Midjourney AI and used under CC BY-NC license.

References

1. Wasserman WW, Sandelin A. Applied bioinformatics for the identification of regulatory elements. Nat Rev Genet. 2004;5:276–87.
2. Lovering RC, Gaudet P, Acencio ML, Ignatchenko A, Jolma A, Fornes O, et al. A GO catalogue of human DNA-binding transcription factors. Biochim Biophys Acta Gene Regul Mech. 2021;1864:194765.
3. Kel-Margoulis OV, Kel AE, Reuter I, Deineko IV, Wingender E. TRANSCompel: a database on composite regulatory elements in eukaryotic genes. Nucleic Acids Res. 2002;30:332–4.
4. Jolma A, Yan J, Whitington T, Toivonen J, Nitta KR, Rastas P, et al. DNA-Binding Specificities of Human Transcription Factors. Cell. Elsevier; 2013;152:327–39.
5. Boeva V. Analysis of Genomic Sequence Motifs for Deciphering Transcription Factor Binding and Transcriptional Regulation in Eukaryotic Cells. Front Genet. 2016;7:24.
6. Isakova A, Groux R, Imbeault M, Rainer P, Alpern D, Dainese R, et al. SMiLE-seq identifies binding motifs of single and dimeric transcription factors. Nat Methods. 2017;14:316–22.
7. Jolma A, Kivioja T, Toivonen J, Cheng L, Wei G, Enge M, et al. Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. Genome Res. 2010;20:861–73.
8. Klein JC, Agarwal V, Inoue F, Keith A, Martin B, Kircher M, et al. A systematic evaluation of the design and context dependencies of massively parallel reporter assays. Nat Methods. Nature Publishing Group; 2020;17:1083–91.
9. de Boer CG, Vaishnav ED, Sadeh R, Abeyta EL, Friedman N, Regev A. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. Nat Biotechnol. 2020;38:56–65.
10. Sahu B, Hartonen T, Pihlajamaa P, Wei B, Dave K, Zhu F, et al. Sequence determinants of human gene regulatory elements. Nat Genet. Nature Publishing Group; 2022;54:283–94.
11. Dhaval Vaishnav E, de Boer CG, Molinet J, Yassour M, Fan L, Adiconis X, et al. The evolution, evolvability, and engineering of gene regulatory DNA. Nature. 2022;603:455–63.
12. Almeida BP de, Reiter F, Pagani M, Stark A. DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of enhancers [Internet]. bioRxiv; 2021 [cited 2022 Nov 15]. p. 2021.10.05.463203. Available from: https://www.biorxiv.org/content/10.1101/2021.10.05.463203v1
13. Tan M, Le QV. EfficientNetV2: Smaller Models and Faster Training [Internet]. arXiv; 2021 [cited 2022 Nov 15]. Available from: http://arxiv.org/abs/2104.00298
14. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely Connected Convolutional Networks [Internet]. arXiv; 2018 [cited 2022 Nov 15]. Available from: http://arxiv.org/abs/1608.06993
15. Kossaifi J, Panagakis Y, Anandkumar A, Pantic M. TensorLy: Tensor Learning in Python [Internet]. arXiv; 2018 [cited 2022 Dec 10]. Available from: http://arxiv.org/abs/1610.09555
16. Smith LN, Topin N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates [Internet]. arXiv; 2018 [cited 2022 Nov 15]. Available from: http://arxiv.org/abs/1708.07120
17. fast.ai - fast.ai—Making neural nets uncool again [Internet]. [cited 2022 Nov 15]. Available from: https://www.fast.ai/
18. Silver NC, Hittner JB, May K. Testing Dependent Correlations With Nonoverlapping Variables: A Monte Carlo Simulation. J Exp Educ. Routledge; 2004;73:53–69.

Figures

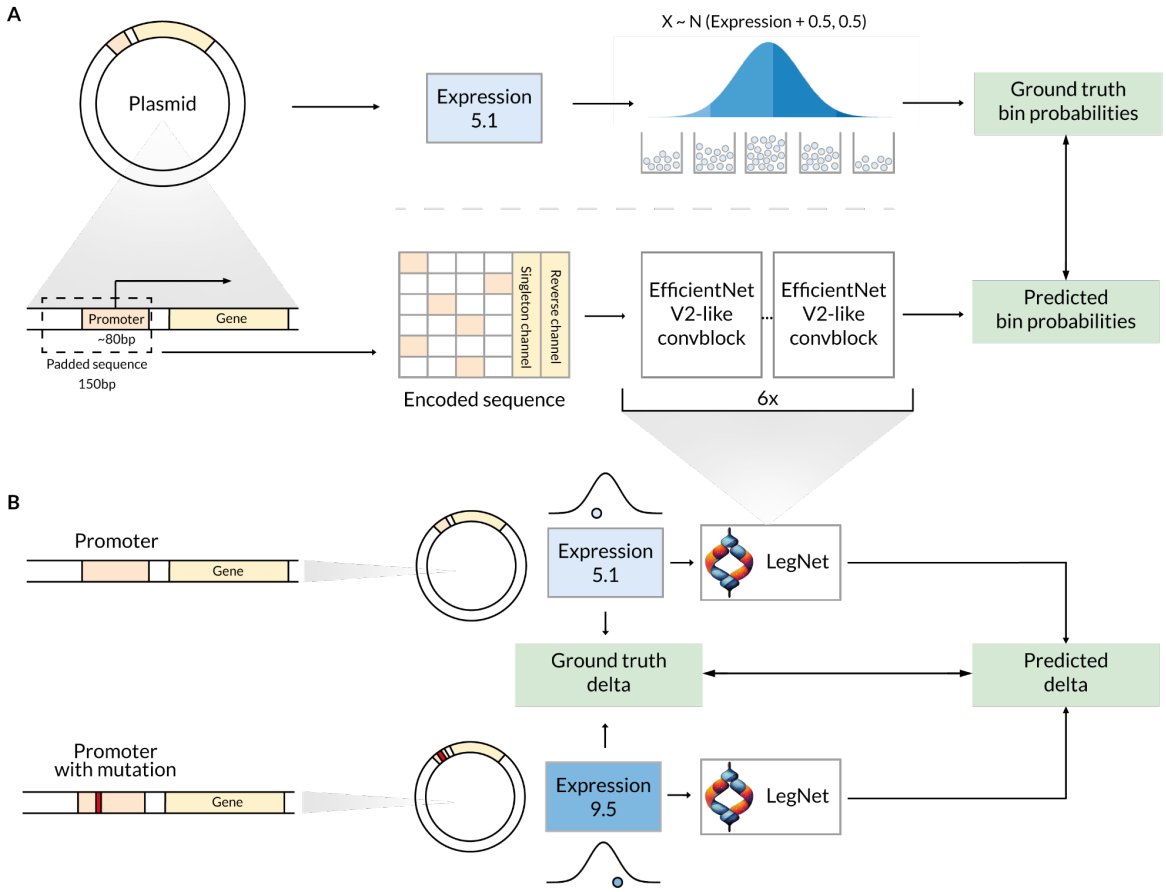

Figure 1. Learning and predicting promoter expression and effects of single-nucleotide variants from massive parallel reporter assays with LegNet.
A. An overall pipeline. The regression task is reformulated as the soft-classification problem mirroring the original experimental setup where cells were sorted into different bins depending on reporter protein fluorescence. Bottom: sequence encoding and prediction of the expression bin probabilities with LegNet.
B. Variant effect estimation with LegNet. Both original and mutated promoter sequences are passed separately to the trained neural network. The variant effect is estimated as a difference between corresponding predictions and compared against the ground truth experimental data.

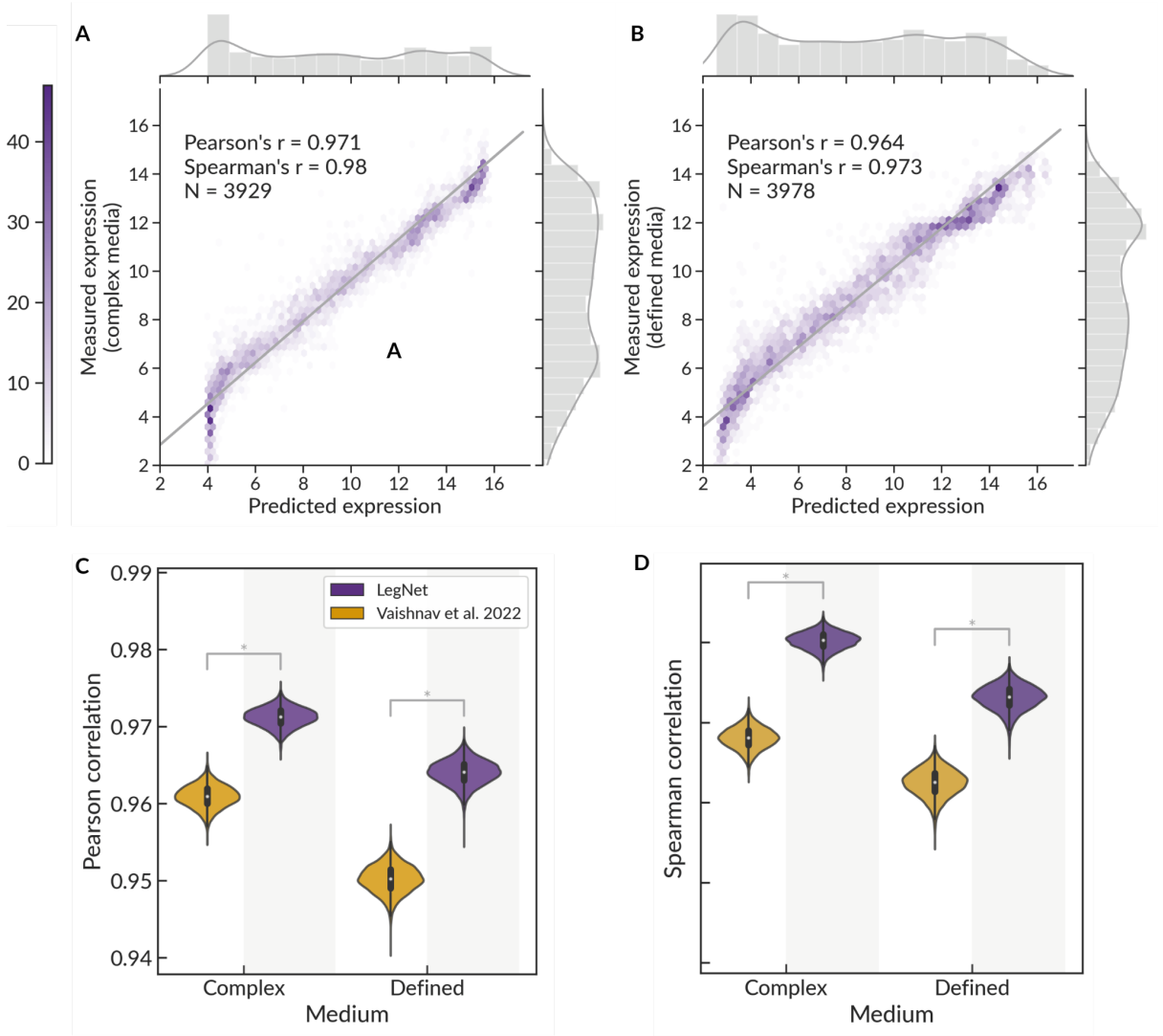

Figure 2. LegNet accurately predicts promoter expression.
A-B. Prediction of native promoter expression for yeast grown in complex medium (YPD, A) and defined medium (SD-Ura, B), hexagonal binning plots.
C-D. Comparison of LegNet prediction performance for native yeast promoter sequences compared to the transformer model of Vaishnav et al. C: Pearson correlation between predictions and ground truth, D: Spearman correlation. Violin plots show bootstrap with n=10,000. *p < 0.001, Silver dependent correlations test [18] for the total data.

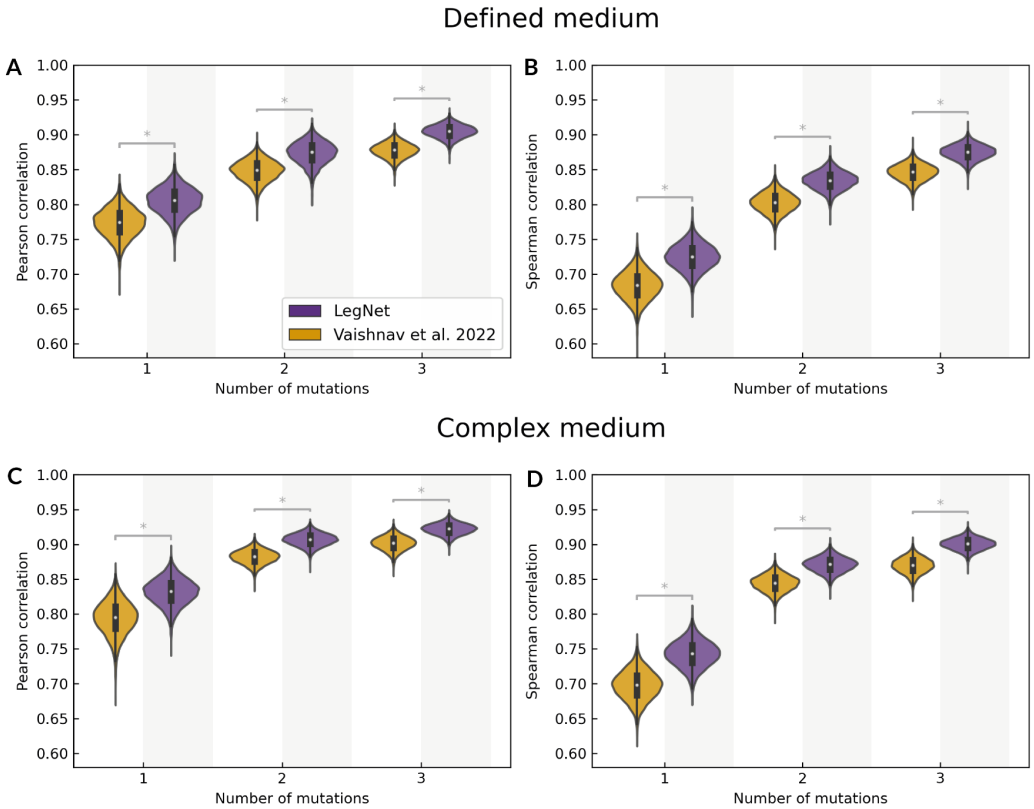

Figure 3. LegNet demonstrates better prediction of variant effects for yeast grown in complex (**A-B**) and defined (**C-D**) medium compared to the transformer model of Vaishnav et al. A, C: Pearson correlation between predictions and ground truth; B, D: Spearman correlation. Violin plots show bootstrap with n=10,000, *p < 0.0001, Silver dependent correlations test [18] for the total data.