Evaluation and optimization of sequence-based gene regulatory deep learning models

Abdul Muntakim Rafi[1], Dmitry Penzar[2,3], Daria Nogina[3], Dohoon Lee[4], Eeshit Dhaval Vaishnav[5], Danyeong Lee[4], Nayeon Kim[4], Sangyeup Kim[4], Georgy Meshcheryakov[6], Andrey Lando[7], Payman Yadollahpour[5], Arsenii Zinkevich[2,3], Dohyeon Kim[4], Yeojin Shin[4], Il-Youp Kwak[8], Byeong-Chan Kim[8], Juhyun Lee[8], Random Promoter DREAM Challenge Consortium, Sun Kim[4], Aviv Regev[5], Jake Albrecht[9], Wuming Gong[10], Ivan V. Kulakovskiy[2,6], Pablo Meyer[11], Carl de Boer[1]

[1] University of British Columbia, Vancouver, BC, Canada

[2] Vavilov Institute of General Genetics, Russian Academy of Sciences, Moscow, Russia

[3] Lomonosov Moscow State University, Moscow, Russia

[4] Seoul National University, Seoul, South Korea

[5] Broad Institute of MIT and Harvard, Massachusetts, United States

[6] Institute of Protein Research, Russian Academy of Sciences, Pushchino, Russia

[7] Yandex N. V., Moscow, Russia

[8] Chung Ang University, Seoul, Republic of Korea

[9] Sage Bionetworks

[10] University of Minnesota, Minneapolis, United States

[11] Center for Computational Health, IBM Research

*Correspondence: carl.deboer@ubc.ca, abdulmuntakim.rafi@ubc.ca

Abstract

Neural networks have proven to be an immensely powerful tool in predicting functional genomic regions, in particular with many recent successes in deciphering gene regulatory logic. However, how model architecture and training strategy choices affect model performance has not been systematically evaluated for genomics models. To address this gap, we held a DREAM Challenge where competitors trained models on a dataset of millions of random promoter DNA sequences and corresponding experimentally determined expression levels to best capture the relationship between regulatory DNA and gene expression in yeast. To robustly evaluate the models, we designed a comprehensive suite of benchmarks encompassing various sequence types. While some benchmarks produced similar results across all models, others differed substantially. For some sequence types, model performances exhibited correlation scores as high as 0.98, while for others, substantial improvement is still required. The top-performing models were all neural networks, which demonstrated substantial performance gains by customizing model architectures to the nature of the experiment and utilizing novel training

strategies tailored to genomics sequence data. Overall, our DREAM Challenge highlights the need to benchmark genomics models across different scenarios to uncover their limitations.

Introduction

In eukaryotes, transcription factors (TFs) play a crucial role in regulating gene expression and are critical components of the *cis*-regulatory mechanism (1). TFs compete with nucleosomes and each other for DNA binding and can enhance each other's binding through biochemical cooperativity and mutual competition with nucleosomes (2). *Cis*-regulatory complexity grows with the square of the number of TFs involved due to the number of potential pairwise interactions between TFs. Since yeast has about 8.5x fewer TFs than humans (~200 vs. ~1700) (1,3), the *cis*-regulatory code is theoretically about ~72x less complex in yeast, making it an ideal system to test our understanding of eukaryotic *cis*-regulation. While the field has made substantial progress in characterizing regulatory mechanisms (4–9), a quantitative understanding of *cis*-regulation remains a major challenge.

Neural Networks (NNs) have shown immense potential in deciphering gene regulation. While different network architectures, such as CNNs (4,5,7,10), RNNs(11), and transformers (8,12), have been used to create genomics models, there is limited research on how NN architectures and training strategies affect their performance. Standard datasets provide a common benchmark to evaluate and compare algorithms, leading to improved performance and continual progress in the field (13). For instance, the computer vision and natural language processing (NLP) fields have seen a continual improvement of NNs facilitated by standard datasets, such as the ImageNet data (13), MS COCO (14), etc. In contrast, genomics models are often created *ad hoc* for analyzing a specific dataset, and it remains unclear whether a model's improved performance results from improved model architecture or training data. In many cases, the models created are not directly comparable to previous models due to substantial differences in the underlying data used to train and test the models.

We organized the Random Promoter DREAM Challenge (15) to address the lack of standardized evaluation and continual improvement of genomics models. Here, we asked the participants to design sequence-to-expression models and train them on expression measurements of promoters with random DNA sequences. The models would receive regulatory DNA sequence as input and use it to predict the corresponding gene expression value. We designed a separate set of sequences to test the limits of and provide insight into model performance. Our evaluation across various benchmarks revealed that, for some sequence types, model performances are approaching the previously-estimated inter-replicate experimental reproducibility for this datatype (6), while considerable improvement remains necessary for others. The top-performing solutions in the challenge exceeded performance of all previous state-of-the-art models for similar data and demonstrated that designing NNs inspired by the state-of-the-art models from computer vision and NLP, incorporating the nature of the experiment into NN design, properly tuning hyperparameters, and novel training strategies that are more suited to genomics sequence data can result in considerable performance gains.

Summary of the challenge and the data

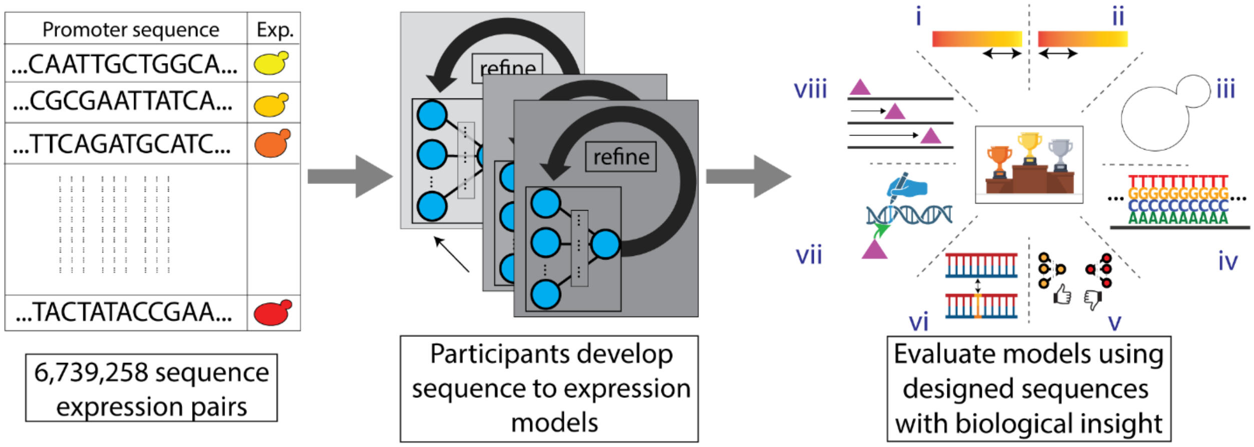


Figure 1: Overview of the challenge. Competitors received a training dataset of random promoters and corresponding expression values (left). They continually refined their models and competed for dominance in a public leaderboard (middle). At the end of the challenge, they submitted a final model for evaluation (right) using a test dataset consisting of eight sequence types: (i) high expression, (ii) low expression, (iii) native, (iv) random, (v) challenging, (vi) SNVs, (vii) motif perturbation, and (viii) motif tiling.

To generate the competition training data, we conducted a high-throughput experiment to measure the regulatory effect of millions of random DNA sequences. Prior research has shown that random DNA can display activity levels akin to genomic regulatory DNA, due to the coincidental occurrence of numerous TF binding sites (6,12,16). As a result, leveraging a vast array of random regulatory sequences enables the learning of intricate regulatory mechanisms from the ground up, without succumbing to biases associated with overrepresented genomic sequences.

Here, we cloned 80 bp random DNA sequences into a promoter-like context upstream of a yellow fluorescent protein (YFP), transformed the resulting library into yeast, grew the yeast in chardonnay grape must, and measured expression by fluorescent activated cell sorting and sequencing, as previously described (6,17,18) (**Methods**). This resulted in a training dataset of 6,739,258 random promoter sequences and their corresponding mean expression values. We provided these data to the competitors, who could use them however they like to train their model. However, competitors were not allowed to use external datasets in any form to ensure that everyone was training models on the same dataset. Ensemble predictions were also disallowed as they would almost certainly provide a boost in performance but without providing any insight into the best model types and training strategies.

We evaluated the models on a set of “test” sequences designed to probe the predictive ability of the models in different ways. The measured expressions of these sequences were quantified in the same way as the training data but in a separate experiment with more cells sorted per sequence (~100), yielding more accurate estimated expression levels compared to the training data measurements and providing higher confidence in the challenge evaluation. The test set consisted of 71,103 sequences from several promoter sequence types. We included both random sequences and sequences from the yeast genome to get an estimate of performance difference between the random sequences in the training domain and the evolved sequences. We also included sequences designed to capture known failure modes of previous models trained on similar data, namely sequences at the high/low extremes of expression and sequences designed to maximize the disagreement between a previously developed CNN and a physics-informed neural network ("biochemical model") (6,12). We previously found that predicting changes in expression between closely related sequences (i.e., nearly identical DNA sequences) is substantially more challenging, and so we included subsets where models had to predict changes that result from single nucleotide variants (SNVs), perturbations of specific TF binding sites, and tiling of TF binding sites across background sequences. Each test subset was given a different weight proportional to the number of sequences in the set and how important we considered it to be (**Table 1)**. For instance, predicting the effects of SNVs on gene expression is a critical challenge for the field due to its relevance to complex trait genetics (19), and so many such sequence pairs were included and given the highest weight in the test set. Within each sequence subset, we determined model performance using Pearson r^2 and Spearman ρ, which captured the linear correlation and monotonic relationship between the predicted and measured expression levels (or expression differences), respectively. The weighted sum of each performance metric across test subsets yielded our two final performance measurements, which we call *Pearson Score* and *Spearman Score*.

The competition ran for 12 weeks in the summer of 2022 and included two evaluation stages: the public leaderboard phase and the private evaluation phase (**Fig. 1**). The leaderboard opened six weeks into the competition and allowed teams to submit up to 20 predictions on the test data per week. At this stage, we used 13% of the test data for leaderboard evaluation and displayed only the overall Pearson r^2*,* Spearman ρ, *Pearson Score,* and *Spearman Score* to the participants*,* while keeping the performance on the promoter subsets and the specific sequences used for the evaluation hidden. The participating teams achieved increasing performance each week (**Supplementary Fig. 1**), showcasing the effectiveness of such challenges in motivating the development of better machine learning models. Over 110 teams from all over the globe competed in this stage. At the end of the challenge, 28 teams submitted their models for final evaluation. We used the rest of the test data (~87%) for the final model evaluation.

Top-performing submissions in the challenge

The top-performing submissions were all NNs. Despite recent findings suggesting the prominence of attention-based architectures (12), only one of the top five submissions in the challenge used transformers, placing 3rd. The best-performing submissions were dominated by fully convolutional NNs, with 1st, 4th, and 5th places taken by them. The best-performing solution was based on the EfficientNetV2 architecture (20,21), and the 4th and 5th solutions were based on the ResNet architecture (22). Moreover, all teams used convolutional layers as the starting point in their model design. A recurrent neural network (RNN) with bidirectional long-short-term memory (Bi-LSTM) layers (23,24) placed 2nd. Overall, the teams converged on many similar training strategies (e.g., using Adam (25) or AdamW (26) optimizers), but had substantial differences as well (**Table 1**).

The competing teams introduced several innovative approaches to solve the expression prediction problem. Autosome.org, the best-performing team, transformed the task into a soft-classification problem by training their network to predict a vector of expression bin probabilities, which was then averaged to yield an estimated expression level, effectively recreating how the data were generated in the experiment. They also used a distinct data encoding method by adding additional channels to the traditional 4-d one-hot encoding of the DNA sequence used by most teams. The two additional channels indicated (i) whether the sequence provided as input was likely measured in only a single cell (which results in an integer expression value), and (ii) whether the input sequence is being provided in the reverse complement orientation. Furthermore, Autosome.org's model, with only two million parameters, challenged the current trend of designing deep NNs with numerous parameters, demonstrating that efficient design can considerably reduce the necessary number of parameters. Autosome.org and BHI were distinct in training their final model on the entirety of the provided training data (i.e., no sequences withheld for validation) for a prespecified number of epochs (determined previously using cross-validation using validation subsets). Unlock_DNA, the 3rd team, took a novel approach by randomly masking 5% of the input DNA sequence and having the model predict both the masked nucleotides and gene expression. This approach used the masked nucleotide predictions as a regularizer, adding a reconstruction loss to the model loss function, which stabilized the training of their large NN. BUGF, the 9th team, used a somewhat similar strategy where they randomly mutated 15% of the sequence and calculated an additional binary crossentropy loss predicting whether any bp in the sequence had been mutated. However, BUGF predicted expression using the original unchanged sequence, whereas Unlock_DNA predicted expression using the masked sequence. The 5th team, NAD, employed GloVe (27) to generate embedding vectors for each base position and used these vectors as inputs for their NN, whereas the other teams utilized traditional one-hot encoded DNA sequences. Two teams, SYSU-SAIL-2022 (11th) and Davuluri lab (16th), attempted to train DNA language models (28) on the challenge data, which involved pretraining a BERT language model (29) on the challenge data and subsequently using the BERT embeddings to train an expression predictor. SYSU-SAIL-2022's pretraining strategy differed from Davuluri lab's, as they only used the top 20% of sequences in terms of expression. However, the predictors used by these two teams were not identical, and it is not possible to definitively conclude that SYSU-SAIL-2022's pretraining strategy was superior for training language models on the random sequence space. Due to the varied network architectures and training strategies (**Table 1),** determining the exact factors that led to the success of the top submissions remains a substantial challenge.

Table 1: Breakdown of the top-performing models into key components

Participant team name	NN architecture type	DNA Encoding method	Input flanking region length	Usage of reverse strand during model training	Train validation split	Number of parameters (in millions)	Optimizer	Loss function	Learning Rate Scheduler	Metric
Autosome.org	CNN (EfficientNetV2 (21))	OHE	70	Data aug. (additional channel) + Model (additional channel)	100-0	1.9	AdamW (26)	Kullback-Leibler divergence	One Cycle LR	r, ρ*
BHI	CNN + RNN (Bi-LSTM) (23)	OHE	30	Post-hoc conjoined setting (30)	100-0	6.8	AdamW (26)	Huber	Cosine Anneal LR	r, ρ*

Participant team name	NN architecture type	DNA Encoding method	Input flanking region length	Usage of reverse strand during model training	Train validation split	Number of parameters (in millions)	Optimizer	Loss function	Learning Rate Scheduler	Metric
Unlock_DNA	Transformer	OHE	20	Input to model (concat. with forward strand)	95-5	47.4	Adam (25)	MSE + custom	One Cycle LR	r
Camformers	CNN (ResNet (22))	OHE	30	None	90-10	16.6	AdamW (26)	L1	Reduce LR On Plateau	r, ρ
NAD	CNN + Transformer	Glove (27)	0	None	90-10	15.5	AdamW (26) + GSAM (31)	smooth L1	Linear LR	r
wztr	CNN (ResNet (22))	OHE	62	Input to model (concat. with forward strand)	99-1	4.8	Adam (25)	MSE	Reduce LR On Plateau	r
High Schoolers Are All You Need (High Schoolers)	CNN + Transformer + MLP	OHE	31	Model (RC parameter sharing) (30)	98-2	4.7	Adam (25) + SWA (32)	MSE	Multi Step LR	r
BioNML	Vision Transformer (33)	OHE	30	Model (RC parameter sharing) (30)	86-14	78.7	Adamax (25) + L2 regularizer	Huber	Multi Step LR	r, CoD
BUGF	Transformer	Embedding	32	None	94-6	4.5	RAdam (34)	Multi-label focal loss (35) + custom	None	r
mt	GRU (36) +CNN	OHE	62	Model (RC parameter sharing) (30)	99.8-0.2	3.1	Adam (25)	binary cross.	None	r, CoD*

*These teams employed the metrics in a cross-validation setting to determine the optimal number of epochs for their models and ultimately saved the model weights after running for the *n* epochs, without relying on validation metric scores. In contrast, other teams utilized validation metric scores to select the best-performing model.

Top two models robustly outperform others

In order to determine the relative performance of the models, we performed a bootstrapping analysis. Here, we sampled 10% of the test data 10,000 times and, for each sample, calculated the performance of each model and the rankings of the models for both Pearson and Spearman Scores (**Methods**). We averaged the ranks from both metrics to decide their final ranks. Teams Autosome.org and BHI robustly outperformed the others, coming in 1st and 2nd place, respectively (**Fig. 2**), with Autosome.org coming second to BHI in only 0.1% and 2.25% of the time for *Pearson Score* and *Spearman Score*, respectively. In none of the bootstraps did another team outperform Autosome.org or BHI. Interestingly, Pearson Score and Spearman Score do not agree after the top two teams, indicating that they capture performance in distinct ways. For instance, Unlock_DNA, the 3rd-place team, substantially outperformed 4th-place Camformers by *Pearson Score,* but performed slightly worse than Camformers by *Spearman Score*. We consider 5th-place to be a tie between NAD and wztr as their mean ranks (5.81105 and 5.8152, respectively) were very close.

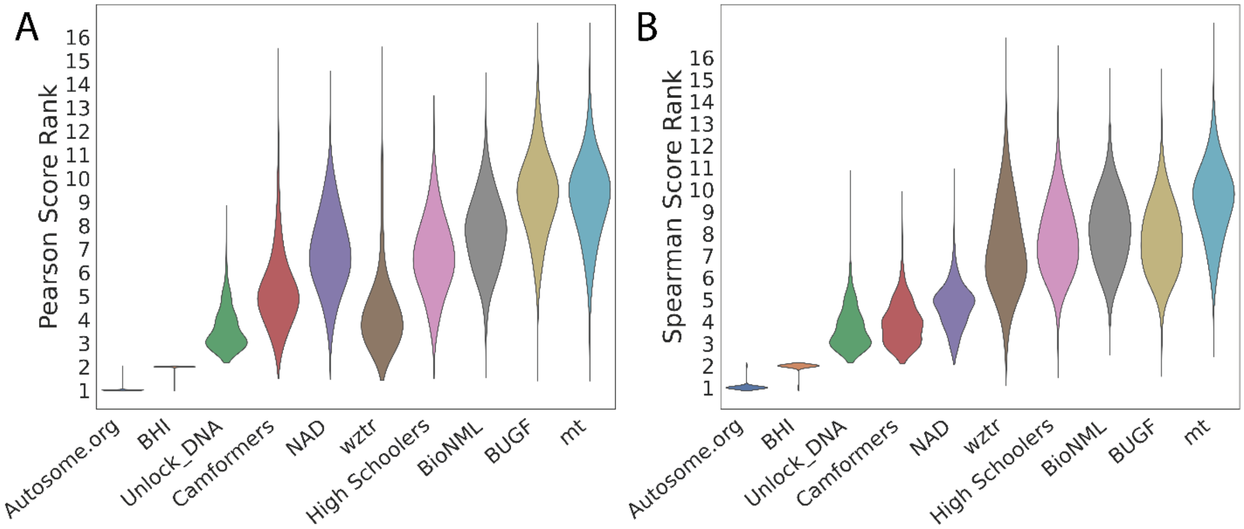


Figure 2: Bootstrapping provides a robust comparison of the model predictions. (**A, B**) Distribution of ranks in n=10,000 samples from the test dataset (*y*-axes) for the top-performing teams (*x*-axes), for (A) Pearson Score and (B) Spearman Score.

Distinct model performances on test subsets

Analysis of model performance on the different test subsets revealed distinct and shared challenges for each model. We retrained the transformer model proposed by Vaishnav et al. (12), the previous best performing model for this type of data, on the challenge data and used as a reference in the leaderboard ("reference model"). The top submissions' overall performance was substantially better than the reference model (**Fig. 3**). Interestingly, the top two models were ranked 1st and 2nd (sometimes with ties) for each test subset regardless of score metric, showcasing that their superior performance cannot be attributed to any single test subset. Further, the rankings within each test subset sometimes differed between *Pearson Score* and *Spearman Score*, reinforcing that these two measures capture performance in distinct ways.

Interestingly, models were highly variable in their ability to accurately predict variation within the extremes of gene expression. The cell sorter has a reduced signal-to-noise ratio for the highest and lowest expression levels, and the sorting bin placement can truncate the tails of the expression distribution (6,12). We included these test subsets (high and low) because we previously found that the Vaishnav et. Al model (12) tends to have limited predictive power within each expression extreme. Overall, model performance was most variable across teams in these subsets (e.g., the median difference in Pearson r^2*,* between highest and lowest performance, was 48% for high and low test subsets and 16% for the others, as shown in **Fig. 3**), suggesting that the challenge models were able to overcome this issue to varying degrees.

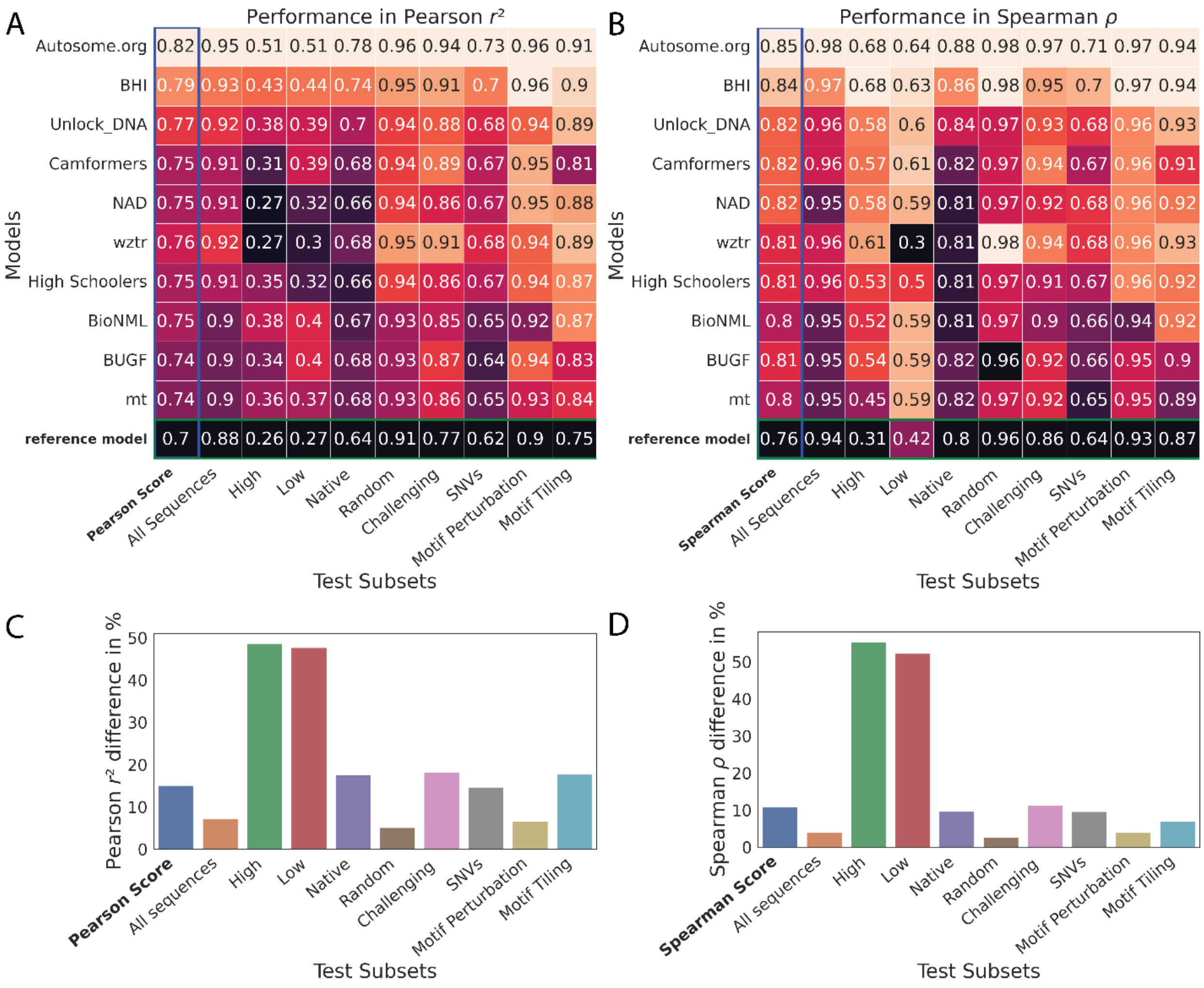


Figure 3: Performance of the top-performing teams in each test data subset. (**A,B**) Model performance (colour and numerical values) of each team (*y*-axes) in each test subset (*x*-axes), for (A) Pearson r^2 and (B) Spearman ρ. Heatmap colour palettes are min-max normalized column-wise. (**C,D**) Performance disparities observed between the best and worst models (*x*-axes) in different test subsets (*y*-axes) for (C) Pearson r^2 and (D) Spearman ρ. The calculation of the percentage difference is relative to the best model performance for each test subset.

Importantly, the models differed in their abilities to predict the expression levels of promoter sequences from the yeast genome. While the ranking of models is similar for both random and native sequences, the difference in model performance is more substantial for native sequences. We see 17.6% and 9.6% performance difference in Pearson r^2 and Spearman ρ between top vs. bottom model for native sequences, respectively, while there are only 5% and 2.7% differences for random sequences (**Fig. 3**), suggesting that top models have learned more of regulatory grammar that evolution has produced. Although there is likely more experimental noise present in the native promoter data, owing to the propensity of native DNA to include repetitive sequences, which results in lower sequence coverage and poorer quality data (**Supplementary Fig. 2**), the substantial discrepancy between performance on native and random sequences suggests that there is yet more regulatory logic to learn.

Expression differences between closely related sequences also revealed differences in model performance (**Fig. 3**, **Supplementary Fig. 3**). For instance, perturbing TF binding sites (**TFBSs**) (e.g., mutating sequences strongly matching the cognate motif for an important TF or variable numbers of binding sites added) is a comparatively large perturbation and could be predicted with simple models that capture the binding of these TFs and can count TFBS instances. However, when TFBSs are tiled across a background sequence, the same TFBS is present in every sequence, and the model must have learned how its position affects its activity, as well as capturing all the secondary TFBSs that are created or destroyed as the motif is tiled. Finally, SNVs are even harder to predict because nearly everything about the sequence is identical but for a single nucleotide that may affect the binding of multiple TFs in potentially subtle ways. Accordingly, model performance was highest for Motif Perturbation, intermediate for Motif Tiling, and lowest for SNVs. However, it is important to note that this partly results from smaller changes in expression being more dominated by experimental noise. Nonetheless, the differences in model performance suggest that the top-performing models have better captured the subtleties of *cis*-regulation since the differences in model performance are more substantial for subtler changes (% difference between best and worst compared to best in Pearson r^2 and Spearman ρ: 6.5% and 4% for Motif Perturbation, 17.7% and 7% for Motif Tiling, and 14.6% and 9.6% for SNVs; **Fig. 3)**.

Model ensemble provides marginal performance boost

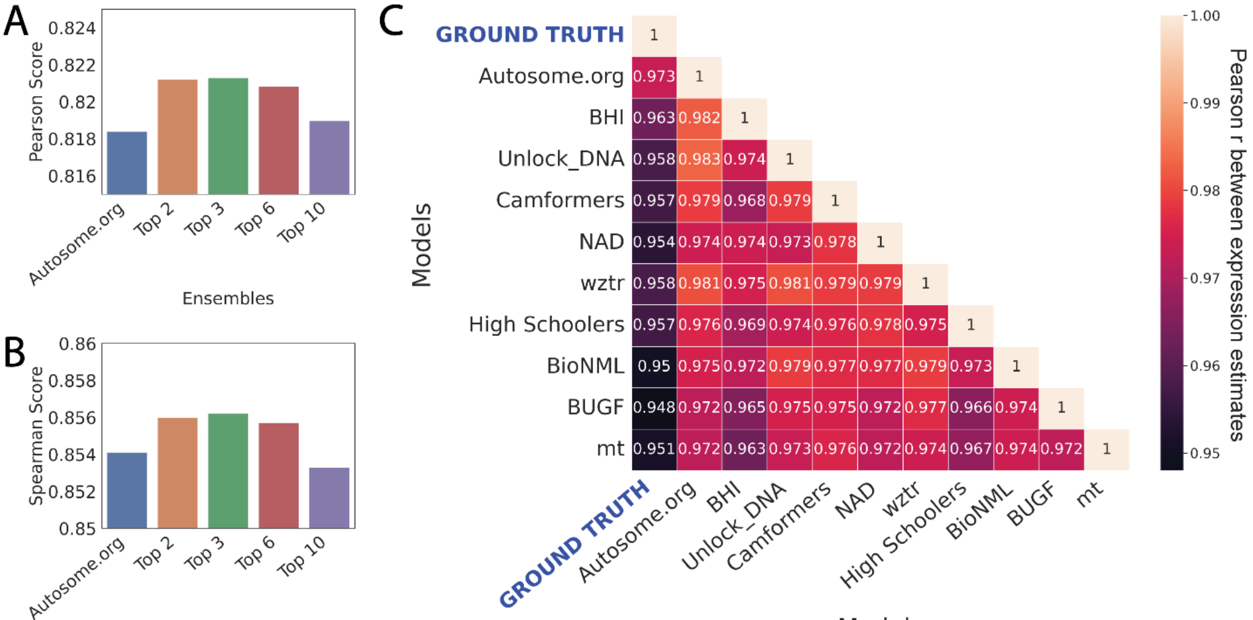


Figure 4: An ensemble of the top-performing model predictions does not improve performance substantially since the top-performing models make similar predictions. (**A,B**) Minimal improvement in *Pearson Score* and *Spearman score* (*y*-axes) achieved by linear regressors (*x*-axes) trained on top of the model predictions. (**C**)The pairwise Pearson correlation (colour) between different pairs of models and the ground truth (*x*- and *y*-axes).

We created ensembles of the top-performing models by training a linear regressor with the individual models' predictions as input. However, the improvement in performance was only slight in comparison to the individual top model performance (**Fig. 4A,B**). A pairwise comparison of the models' predictions on the test data showed a higher correlation between the models to each other than to the ground truth (**Fig. 4C, Supplementary Fig. 4**). Furthermore, we observed that the correlation remained high for individual test subsets with the exception of two sequence types (**Supplementary Fig. 5**). Interestingly, these two sequence types represented the extremes of expression space, and the networks might have learned them differently due to the low signal-to-noise ratio present in the training data at these extremes, which is evident by the differing relationships between predicted and actual expression, particularly at the high end of expression (**Supplementary Fig. 6**). Overall, the results suggest that the models have a similar understanding of the underlying biology but vary in how much of it they have captured. The models may be approaching the extent of cis-regulation that is learnable from the DREAM Challenge training data.

Discussion

The random promoter DREAM Challenge 2022 presented a unique opportunity for participants to propose novel model architectures and training strategies for modeling regulatory sequences. The top-performing models challenged the trend of using increasingly complex architectures by demonstrating that simpler NN models with fewer parameters can effectively capture much of *cis*-regulation. In particular, 3/5 of the top submissions did not use transformers, including the best-performing team (which also had the fewest parameters of the top 10), illustrating that attention may not be the most efficient way of capturing *cis*-regulatory logic in short sequences. This emphasizes the need for further exploration of the architectures and training strategies best suited for genomics sequence data.

The varied results across test subsets illustrate the complexity in evaluating cis-regulatory models effectively. For instance, performance on random sequences, which are in the same domain as the training data (also random sequences), was relatively uniform (**Fig. 3**). Whereas the domain shift to native sequences, where the relative frequencies of different regulatory mechanisms likely differ since these originated through evolution, highlights the disparities between models (**Fig. 3**). This indicates that a model that excels in modeling overall *cis*-regulation may still perform poorly for sequences involving certain regulatory mechanisms that are difficult to learn from the training data, leading to incorrect predictions of biochemical mechanisms and variant effects for sequences that use these mechanisms. This emphasizes the importance of multifaceted evaluation of genomics models (37), and designing specific datasets that test the limits of these models.

Methods

Designing the test sequences

High- and low-expression sequences were designed as described in Vaishnav et al.(12). Native test subset sequences were designed by sectioning native yeast promoters into 80 bp fragments (6). Random sequences were sampled from a previous experiment where the tested DNA was synthesized randomly (as in the training data) and quantified (6). Challenging sequences were designed using a combination of a convolutional neural network model (12) and a biochemical model (a type of physics-informed neural network) (6). Most of the SNVs represent sequence trajectories from Vaishnav et al. (12), but also include random mutations added to random, designed, and native promoter sequences. Motif Perturbation included Reb1 and Hsf1 perturbations. Sequences with perturbed Reb1 binding sites were created by inserting Reb1 consensus binding sites (strong or medium affinity; sense and reverse complement orientations), and then adding between 1 and 3 SNVs to each possible location of each motif occurrence and inserting canonical and mutated motif occurrence into 10 randomly generated sequences at position 20/80. Sequences with Hsf1 motif occurrence were designed by tiling random background sequences with between 1 and 10 Hsf1 monomeric consensus sites (ATGGAACA), added sequentially from both the right and left of the random starting sequences, or added individually within each of the possible 8 positions, or similarly tiling/inserting between 1-5 trimeric Hsf1 consensus sites (TTCTAGAANNTTCT). The Motif Tiling test subset sequences were designed by embedding a single consensus for each motif (poly-A: AAAAA, Skn7: GTCTGGCCC, Mga1: TTCT, Ume6: AGCCGCC, Mot3: GCAGGCACG, and Azf1: TAAAAGAAA) at every possible position (with the motif contained completely within the 80-bp variable region) and orientation for three background sequences as described in de Boer et al. (6).

Table 2: Summary of the test subsets

Subset	no. of sequences	Weight in evaluation metric	Description
All sequences	71103	1	All sequences in the test data.
High	968	0.3	Sequences designed to have high expression.
Low	997	0.3	Sequences designed to have low expression.
Native	997	0.3	Sequences that are present in the yeast genome.
Random	6349	0.3	Random DNA sequences.
Challenging	1953	0.5	Sequences designed to maximize the differences between a convolutional model and a biochemical model trained on the same data.
SNVs	44340 pairs	1.25	Two sequences that differ by only a single base.
Motif Perturbation (Reb1+Hsf1)	3287 pairs	0.3	Two sequences that differ due to perturbations to specific known TF binding site.
Motif tiling	2624 pairs	0.4	Two sequences that differ due to

			tiling known TF binding sites across random sequences.

Quantifying promoter expression

High complexity random DNA libraries that comprised the training data were created as described previously (6), with the following modifications. The random promoter library in E. coli theoretically contained about 74 million random promoters and was transformed into S288c (delta URA3) yeast yielding 200 million transformants, which were selected in SD-Ura media. 1 L of chardonnay grape must (filtered) was inoculated with the pool to an initiate OD600 of 0.05 and grown at room temperature without continual shaking, with the culture diluted as needed with fresh chardonnay grape must to maintain OD below 0.4, for a total growth time of 48 hours and having undergone >5 generations. Prior to each OD, the culture was gently agitated to decarbonate it, waiting for the resulting foam to die down before agitating again, and continuing until no more bubbles were released. Yeast were then sorted and sequencing libraries prepared as described previously (6), with sequencing libraries pooled and sequenced on an Illumina NextSeq.

Data processing for both N80 (training) and designed (test) libraries were done as described before (6), except that the expression levels of the designed library sequences were estimated using MAUDE (38), using the read abundance in each sorting bin as input, and estimating the initial abundance of each sequence as the average relative abundance of that sequence across all bins.

Competition rules

1. Only the provided training data could be used to train models. Models had to train from scratch without any pre-training on external datasets to avoid overfitting to sequences present in the test data (e.g. some sequences in the test data are derived from extant yeast promoters).

2. Reproducibility was a prerequisite for all submissions. The participants had to provide the code and instructions to reproduce their models. We retrained the top-performing solutions to validate their performance.

3. Augmenting the provided training data was allowed. Pseudo-labeling the provided test data was not allowed. Using the test data for any purpose during training was not allowed.

4. Ensembles were not allowed.

Performance evaluation

We calculated the Pearson r^2 and Spearman ρ between predictions and measurements for each test subset to capture performance on each sequence subset. We used Pearson *r,* which captures the linear correlation between predictions and measured expression levels while being robust to the scaling differences that occur between training and test sequences. As the training and the test data are not on the same scale but are linearly related by, $y = mx + b$, where m

and b are constants that are affected by sorting bin placement during the experiment, Pearson *r* was an appropriate metric for us. We also used Spearman ρ, which captures the monotonic relation between predictions and measured expression levels while being robust to outliers. The weighted sum of each performance metric across promoter types yielded our two final performance measurements, which we call *Pearson Score* and *Spearman Score*.

$$Pearson\ Score = \sum_{i=0}^{subsets} w_i \ * PearsonR^2{}_i \ / \ \sum_{i=0}^{subsets} w_i$$

$$Spearman\ Score = \sum_{i=0}^{subsets} w_i \ * Spearman_i \ / \ \sum_{i=0}^{subsets} w_i$$

Here, w_i is the weight used for the i-th test subset. $PearsonR^2{}_i$ and $Spearman_i$ are respectively the square of Pearson coefficient and the Spearman coefficient for sequences in i-th subset.

Description of the approaches used by the participants

In this section, we present an overview of the approaches employed by the participants in the challenge. For the top-performing teams, we provide a detailed description of their methodologies, while for the remaining teams, we offer a concise overview without repeating any details that have already been discussed for other teams. The performance of each approach is illustrated in **Supplementary Fig. *7*** for easier comparison.

Autosome.org: The team reformulated the initial regression task as a soft-classification problem by replacing the initial target expression value with a vector of 18 probabilities corresponding to individual bins, assuming that they can be deduced from the normal distribution with mean and variance equal to (expression + 0.5) and 0.5, respectively. To obtain a predicted expression value for a sequence during the validation step, the predicted probabilities were multiplied by their bin numbers. They used one-hot encoding for the promoter sequences, where they added a separate binary channel explicitly marking the objects with integer (and thus likely imprecise) expression measurements. They also augmented the dataset with the reverse complementary sequences and added a separate binary channel to denote the supplied strand explicitly (forward or reverse complementary). The proposed model was based on a fully-convolutional network inspired by EfficientNetV2 (21). The following architectural choices contributed to the final performance: (i) grouped convolution (39) instead of the depthwise convolution of the original EfficientNetV2, (ii) the standard residual blocks were substituted with residual channel-wise concatenations, (iii) a bilinear layer was inserted in the middle of the EfficientNetV2 SE-block.

BHI: Their approach adopts a “sandwich” architecture consisting of a one-dimensional convolutional layer, a bidirectional long-short-term memory (Bi-LSTM) layer, and another convolutional layer. Each convolutional layer used different kernel sizes. Besides the model architecture, the team found that training details specialized for DNA sequence-based deep learning models were highly important for the overall performance. Among them, the most crucial was to use a ‘post-hoc conjoined’ setting (30), which imposes a reverse-complement equivariance to the model. Test-time augmentation was also effective. Predictions were made for an original sequence, its four shifted variants (generated by -2bp, -1bp, +1bp, and +2bp shifting), and their reverse-complement sequences, then those 10 predictions were averaged to make a final prediction. While training sequences over 110bp were trimmed to the right, sequences shorter than 110bp were randomly padded with the original vector sequences on both sides. This informative padding gave a nonnegligible performance boost. Finally, to be as unbiased as possible for the distribution of the test set, predictions were quantile-transformed using the distribution of expression levels in training data as post-processing.

Unlock_DNA: The team used an end-to-end Macaron-like Transformer encoder architecture with two half-step feed-forward (FFN) layers at the beginning and end of each encoder block. A separable 1D convolution layer was inserted after the first FFN layer and in front of the multi-head attention layer. The sliding k-mers from one-hot encoded sequences were mapped onto a continuous embedding, combined with the learned positional embedding and strand embedding (forward strand vs. reverse complement strand) as the sequence input. Along with the sequence input, several positions (32 in the final model) of "pseudo" expression values were added as the input, where all input expression values were zeros. The model predicted one expression value for each "pseudo" expression position and used the mean of the prediction of all positions as the final predicted expression value.

Camformers: This team used a CNN with residual connections. The model included six convolutional layers with three residual connections allowing the model to bypass every other layer. After the penultimate convolutional layer, a max pooling operation was added to reduce the model size and improve generalization. The output of the final convolutional layer was flattened and fed into a block of two dense layers, followed by a final dense layer outputting the predicted expression level. All layers except the last used a rectified linear unit activation.

NAD: The approach has two stages: (i) generating the embedding vectors for each base position using GloVe (27) and (ii) using the embedding vectors as input of neural networks to predict the gene expression level. The proposed model combines a convolutional neural network for feature extraction and a transformer for prediction.

WZTR: The team used a fully CNN-based architecture. The model begins with two convolutional layers, and six convolution blocks follow these layers. Each convolution block is constructed of 3 convolutional layers and an average pooling at the end. Each convolutional layer consists of a hybrid convolution (40), batch normalization, ReLU activation, and residual connection. Each hybrid convolution takes in a list of dilation values [1,2,4,6], with 4 convolutions processing the input in parallel. Finally, there are three fully connected layers and an output layer. A linear combination of 256 features extracted from all the previous operations on the sequence is used to generate the predicted expression.

High Schoolers Are All You Need (High Schoolers): This team used a mix of CNN and transformer architectures, where the CNN was based on Residualbind's (41) design, with a convolutional layer (with exponential activations) followed by a residual block comprised of a series of dilated convolutional layers with increasing dilation rates. It was followed by attention pooling, a transformer layer with relative positional encodings, and a standard MLP block.

BioNML: The underlying neural network was configured to have a relatively larger set of convolutional kernels and extra dictionaries of short k-mers for spotting potential enriched DNA sequence patterns. Strand-specific streams of these patterns were normalized and consolidated with Swish activation's (42) fully learnable thresholding. The encoded patterns were fed into a ViT (33) like block but with transformer decoder type of connections and SwiGLU (43) activations for modeling any sequential interdependence. A set of suppressed signals of the encoded sequence-based patterns as queries for the transformer decoder blocks to respond to.

BUGF: A transformer model was used to predict the expression bin classes, as opposed to treating the problem as a regression problem. Random mutations were added to the sequence as an data augmentation strategy and the model was trained to predict where the mutations had been made to the input sequence. An auxiliary loss was calculated based on this prediction, which helped reduce overfitting.

mt: The approach uses GRU and CNNs to regress the strength of the targeted promoters using information encoded in the forward and reverse DNA strands.

SYSU-SAIL-2022: The team first trained a 3-layer BERT (29) using part of the train data. Then, the BERT embedding was used to train an expression predictor.

Wen Group: A deep neural network that adopted concepts of U-Net (44), Transformer, and Squeeze & Excitation blocks (45) was trained from end to end without any data augmentation.

Yuanfang Guan: A neural network that consisted of LSTM layers followed by attention layers was used to predict expression.

Metformin-121: A neural network based on bidirectional GRU was used to predict gene expression.

NGT4: A neural network based on XceptionNet (46) was used to predict gene expression. During training, the expression values were transformed evenly in the range of $i - 0.5 < x < i + 0.5$ (here, i is the integer expression) maintaining the ranking of sequences that was produced by a trained model.

Davuluri lab: The team utilized a transformer-based representation model named DNABERT (28) for predicting gene expression.

*DNABERT was pretrained on human genome, which violated the competition rules. However, we consider this to be an important benchmark that shows the limitation of DNA language models.

Wan&Barton_BBK: The team designed a model based on Temporal Convolutional Networks (47) to predict expression.

Peppa: The team designed a model based on the Enformer (8) that took 110 bp as input (compared to 200 kbp in the original) and included only 2M parameters (compared to ~200M parameters of the original).

The Dream Team: A neural network was used that incorporated convolutional, multihead attention, and LSTM layers. During training, the integer expression values were transformed by replacing them with Normal(i,0.3) distribution, where i represents the expression.

Noisy-Chardonnay: A model composed of convolutional layers followed by BiLSTM layers was trained without any data augmentation to predict expression levels.

KircherLab: The team trained a simple convolutional neural network with a GC correction step on the training data to help the model focus its decisions on motifs within the sequence rather than the general nucleotide composition.

MadLab: The model is composed of three building blocks, namely a convolutional network, a transformer and a recurrent network.

Auth: A simple hybrid architecture combining a convolutional layer and a BiLSTM layer followed by two fully connected layers was used to predict expression.

UTKbioinformatics: A neural network based on BERT was used to predict expression.

DrAshokAndFriends: An attention based ConvLSTM (48) model was used for prediction.

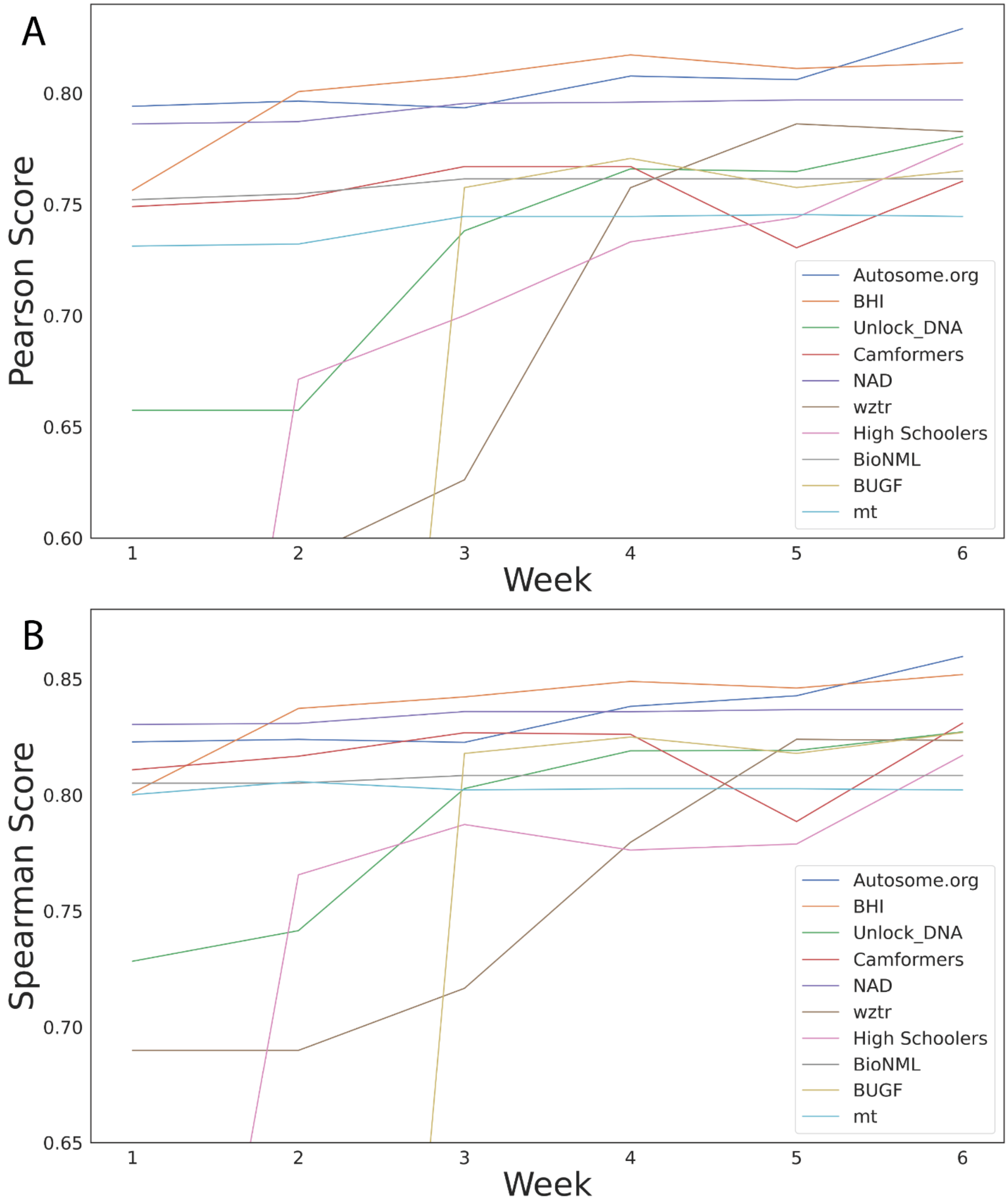
QUT_seq2exp: A sequence embedding model, dna2vec (49), was applied on the promoter sequences (in a running manner on short k-mers), which are then subsequently used as features for a transformer-based deep neural network model.

Zeta: A transformer model was used for predicting expression.

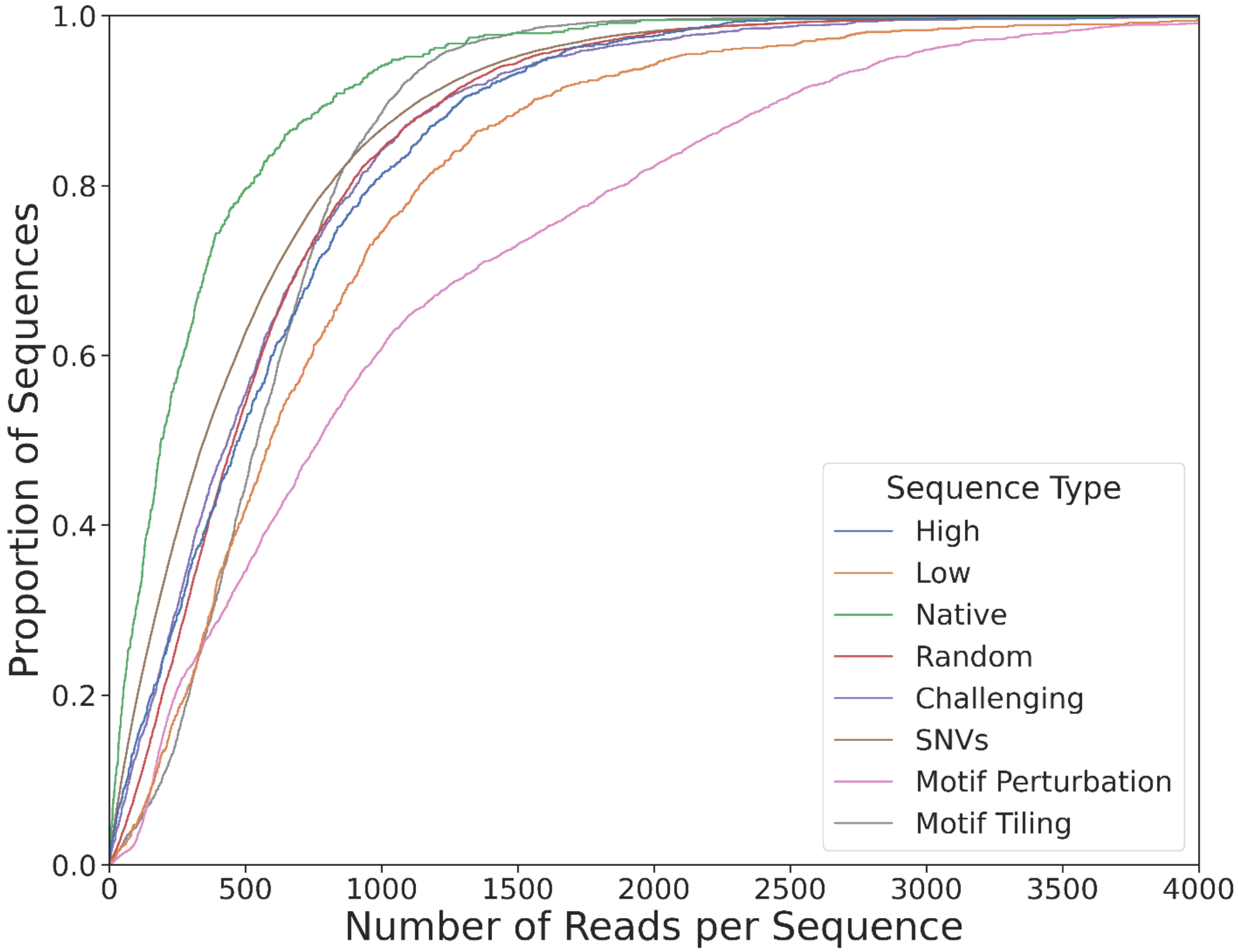
Data availability

The processed training and test data used in this study are available at Zenodo database under DOI 10.5281/zenodo.7395397.

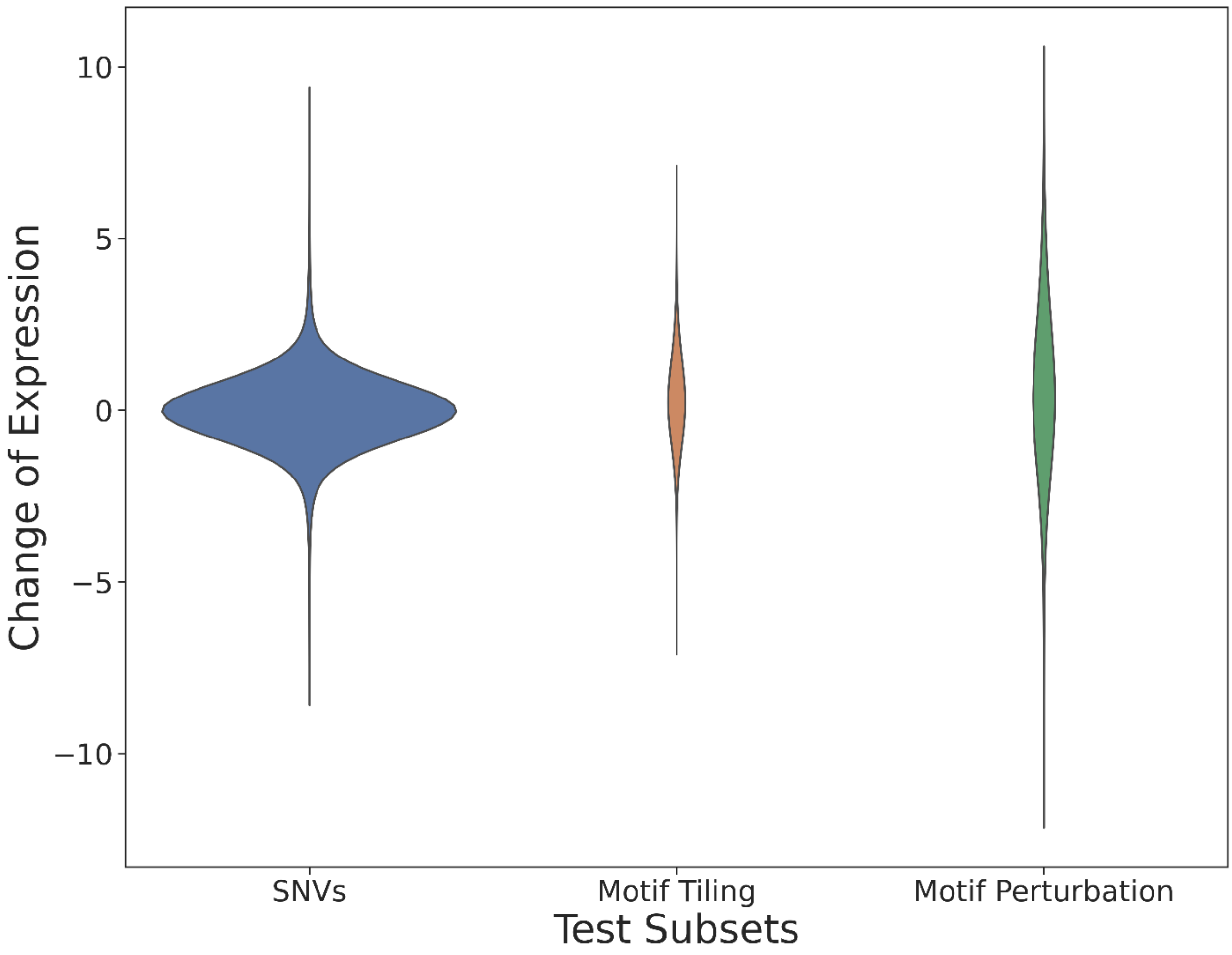
Supplementary Figures



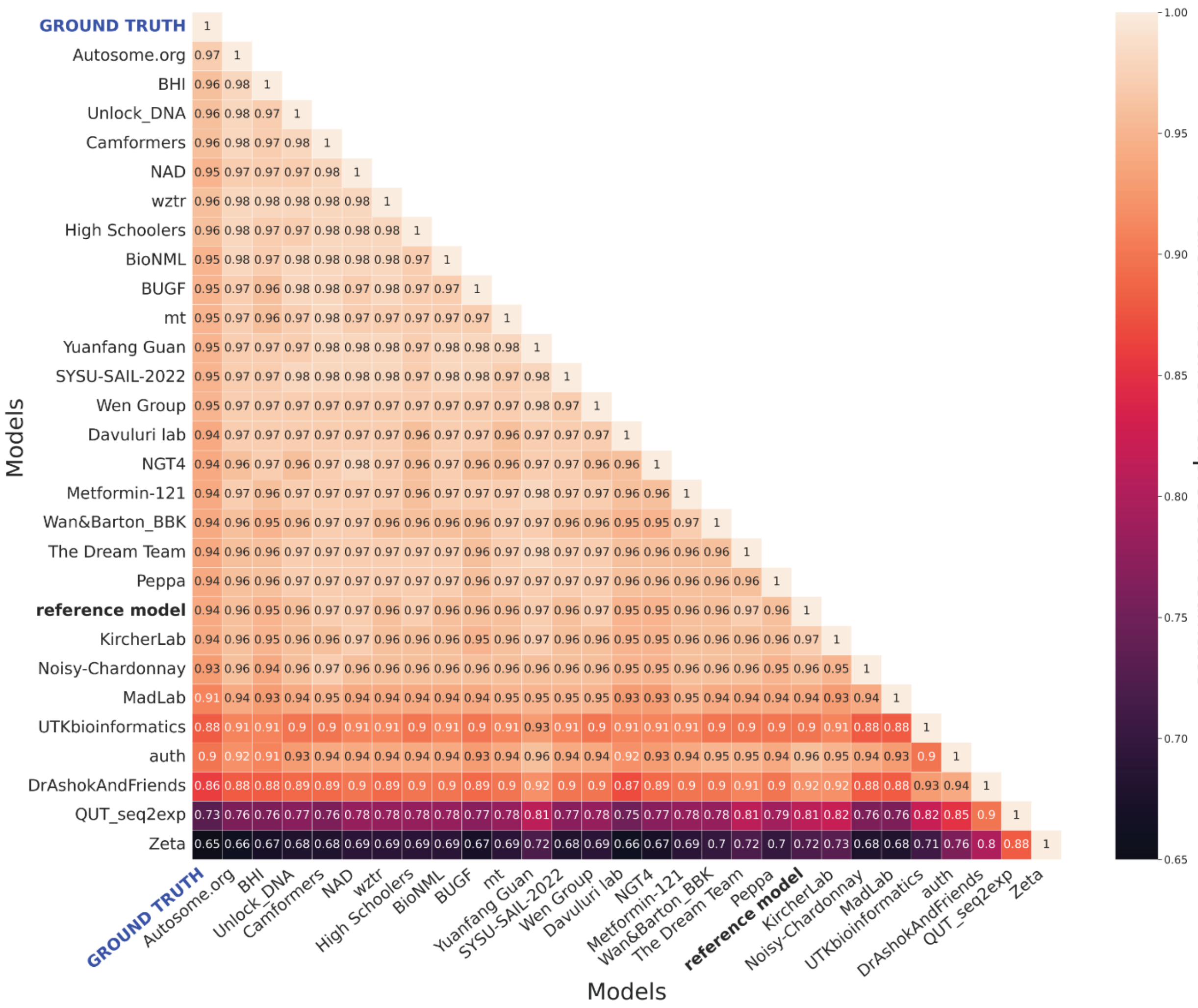
Supplementary Figure 1: Progression of the top-performing teams' performance in the DREAM Challenge public leaderboard. (A,B) Performance (*y*-axes) in (**A**) Pearson Score and (**B**) Spearman Score achieved by the participating teams (colours) each week (*x*-axes), showcasing the effectiveness of such challenges in motivating the development of better machine learning models.



Supplementary Figure 2: Library coverage differs between sequence supsets and is lowest for native sequences. Cumulative proportion (*y*-axis) of the number of reads per sequence (*x*-axis) for different sequence types (colours).

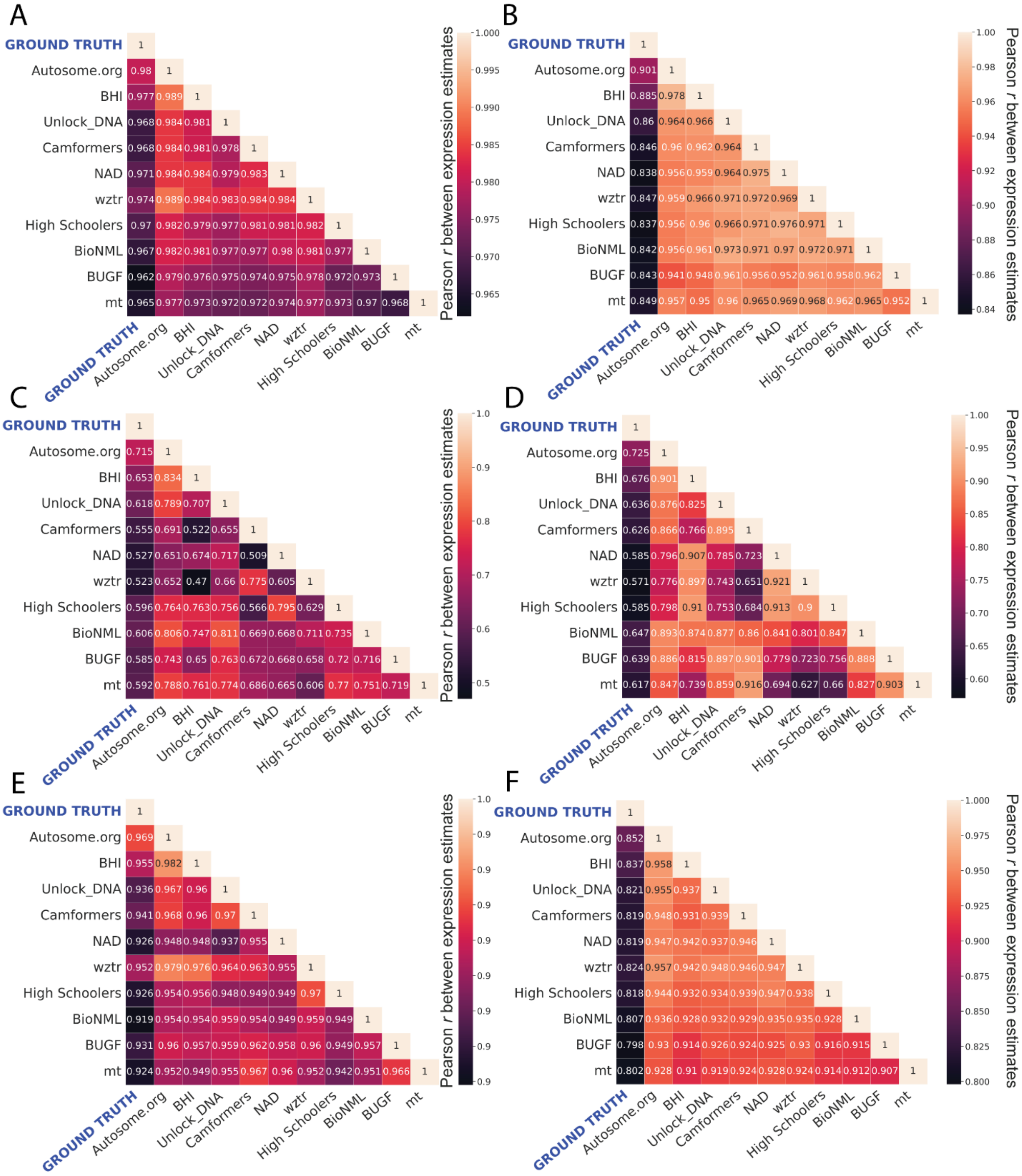


Supplementary Figure 3: Expression changes (*y*-axis) are biggest for motif perturbation, smallest for SNVs, and intermediate for motif tiling.

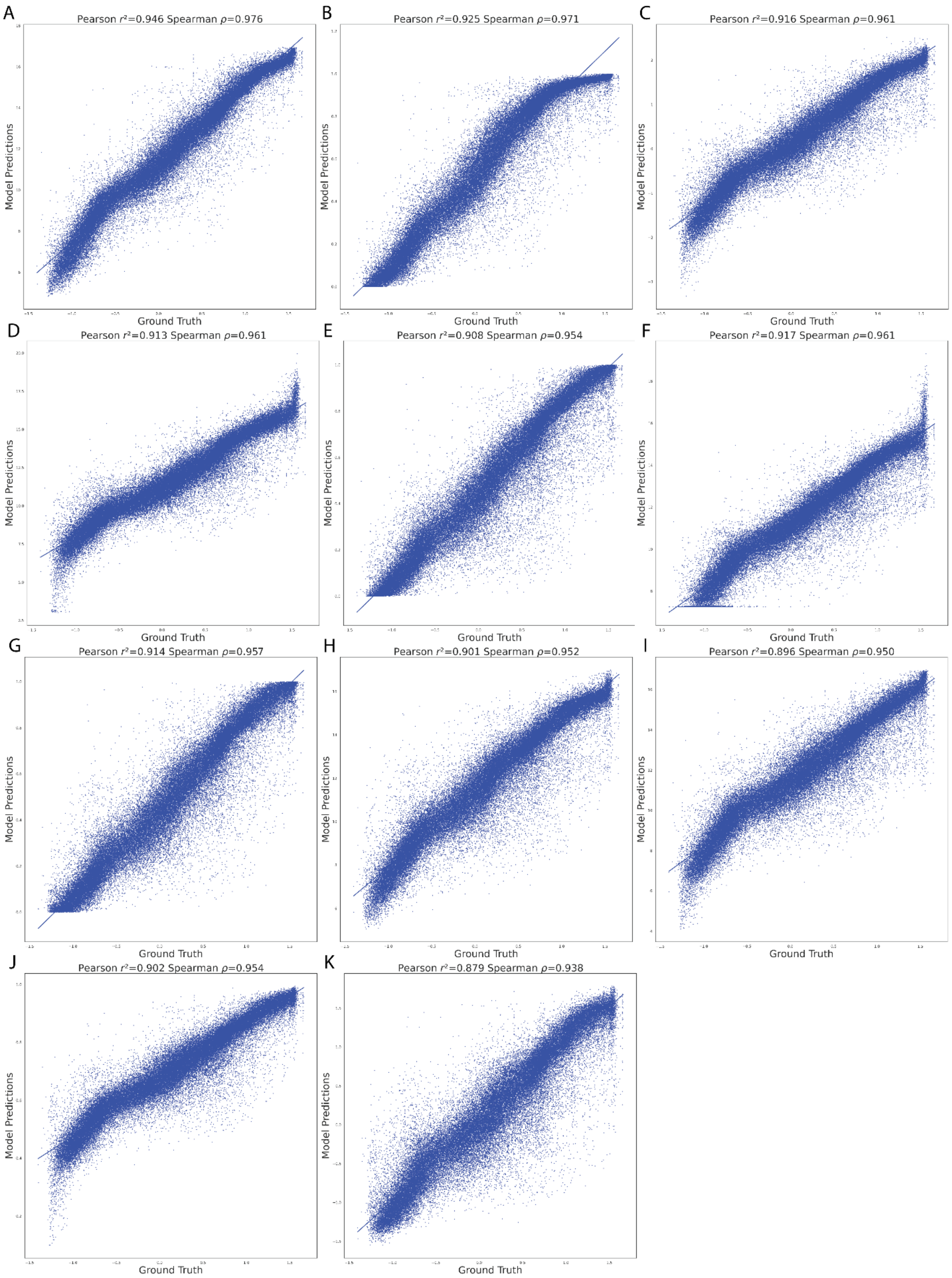


	GROUND TRUTH	Autosome.org	BHI	Unlock_DNA	Camformers	NAD	wztr	High Schoolers	BioNML	BUGF	mt	Yuanfang Guan	SYSU-SAIL-2022	Wen Group	Davuluri lab	NGT4	Metformin-121	Wan&Barton_BBK	The Dream Team	Peppa	**reference model**	KircherLab	Noisy-Chardonnay	MadLab	UTKbioinformatics	auth	DrAshokAndFriends	QUT_seq2exp	Zeta
GROUND TRUTH	1																												
Autosome.org	0.97	1																											
BHI	0.96	0.98	1																										
Unlock_DNA	0.96	0.98	0.97	1																									
Camformers	0.96	0.98	0.97	0.98	1																								
NAD	0.95	0.97	0.97	0.97	0.98	1																							
wztr	0.96	0.98	0.98	0.98	0.98	0.98	1																						
High Schoolers	0.96	0.98	0.97	0.97	0.98	0.98	0.98	1																					
BioNML	0.95	0.98	0.97	0.98	0.98	0.98	0.98	0.97	1																				
BUGF	0.95	0.97	0.96	0.98	0.98	0.97	0.98	0.97	0.97	1																			
mt	0.95	0.97	0.96	0.97	0.98	0.97	0.97	0.97	0.97	0.97	1																		
Yuanfang Guan	0.95	0.97	0.97	0.97	0.98	0.98	0.98	0.97	0.98	0.98	0.98	1																	
SYSU-SAIL-2022	0.95	0.97	0.97	0.98	0.98	0.98	0.98	0.97	0.98	0.98	0.97	0.98	1																
Wen Group	0.95	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.98	0.97	1															
Davuluri lab	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.96	0.97	0.97	0.97	1														
NGT4	0.94	0.96	0.97	0.96	0.97	0.98	0.97	0.96	0.97	0.96	0.96	0.97	0.97	0.96	0.96	1													
Metformin-121	0.94	0.97	0.96	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.97	0.98	0.97	0.97	0.96	0.96	1												
Wan&Barton_BBK	0.94	0.96	0.95	0.96	0.97	0.97	0.96	0.96	0.96	0.96	0.97	0.97	0.96	0.96	0.95	0.95	0.97	1											
The Dream Team	0.94	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.97	0.98	0.97	0.97	0.96	0.96	0.96	0.96	1										
Peppa	0.94	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.96	0.96	0.96	0.96	0.96	1									
reference model	0.94	0.96	0.95	0.96	0.97	0.97	0.96	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.95	0.95	0.96	0.96	0.97	0.96	1								
KircherLab	0.94	0.96	0.95	0.96	0.96	0.97	0.96	0.96	0.96	0.95	0.96	0.97	0.96	0.96	0.95	0.95	0.96	0.96	0.96	0.96	0.97	1							
Noisy-Chardonnay	0.93	0.96	0.94	0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95	0.96	0.96	0.96	0.95	0.96	0.95	1						
MadLab	0.91	0.94	0.93	0.94	0.95	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.93	0.93	0.95	0.94	0.94	0.94	0.94	0.93	0.94	1					
UTKbioinformatics	0.88	0.91	0.91	0.9	0.9	0.91	0.91	0.9	0.91	0.9	0.91	0.93	0.91	0.9	0.91	0.91	0.91	0.9	0.9	0.9	0.9	0.91	0.88	0.88	1				
auth	0.9	0.92	0.91	0.93	0.94	0.94	0.94	0.94	0.94	0.93	0.94	0.96	0.94	0.94	0.92	0.93	0.94	0.95	0.95	0.94	0.96	0.95	0.94	0.93	0.9	1			
DrAshokAndFriends	0.86	0.88	0.88	0.89	0.89	0.9	0.89	0.9	0.9	0.89	0.9	0.92	0.9	0.9	0.87	0.89	0.9	0.9	0.91	0.9	0.92	0.92	0.88	0.88	0.93	0.94	1		
QUT_seq2exp	0.73	0.76	0.76	0.77	0.76	0.78	0.78	0.78	0.78	0.77	0.78	0.81	0.77	0.78	0.75	0.77	0.78	0.78	0.81	0.79	0.81	0.82	0.76	0.76	0.82	0.85	0.9	1	
Zeta	0.65	0.66	0.67	0.68	0.68	0.69	0.69	0.69	0.69	0.67	0.69	0.72	0.68	0.69	0.66	0.67	0.69	0.7	0.72	0.7	0.72	0.73	0.68	0.68	0.71	0.76	0.8	0.88	1

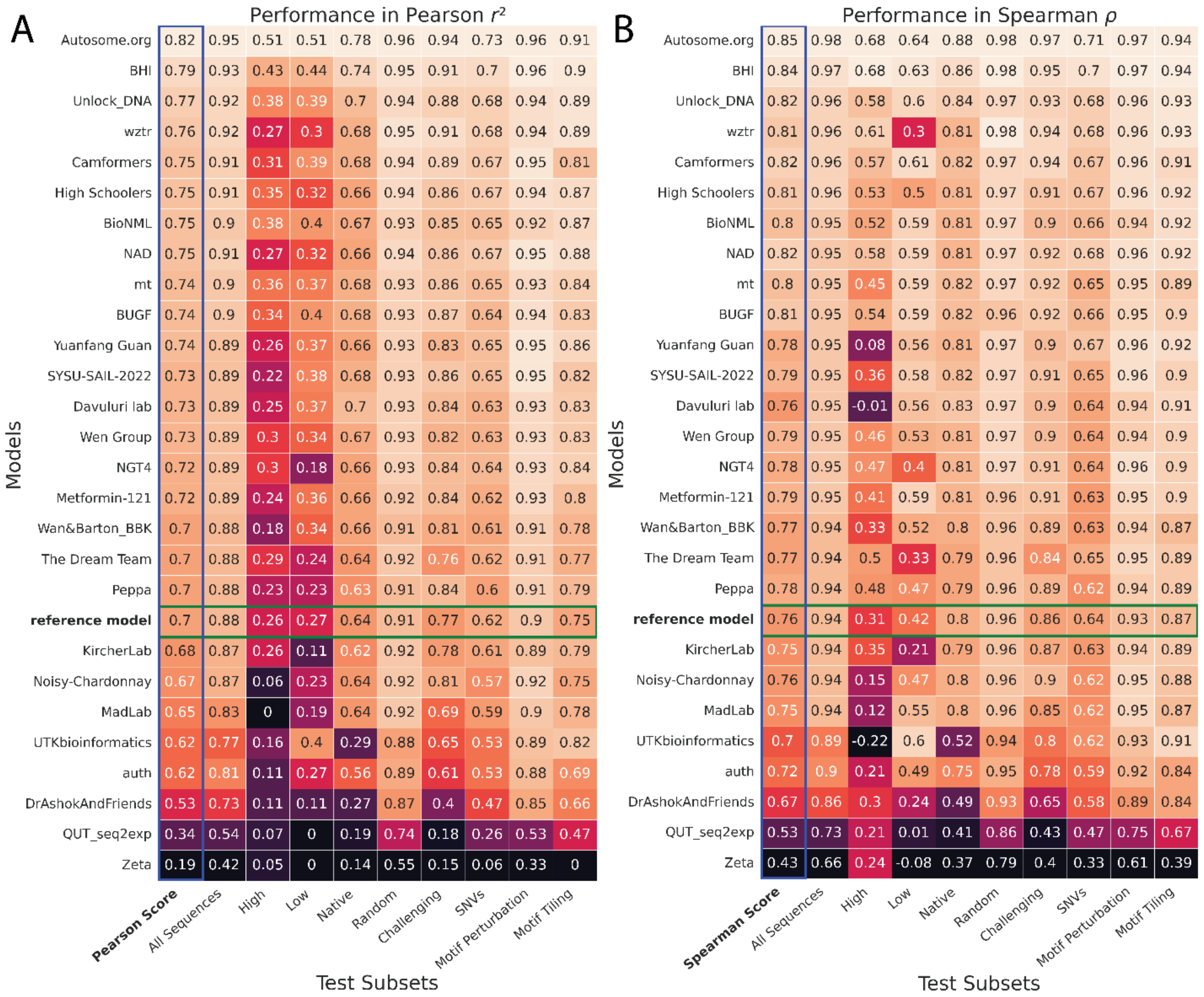
Supplementary Figure 4: The pairwise Pearson correlation between the ground truth and the predictions made by all models.



Supplementary Figure 5: The pairwise Pearson correlation between predictions made by different models for different test subsets, (A) random, (B) native, (C) high, (D) low, (E) challenging, and (F) SNVs.


A
Pearson r^2=0.946 Spearman ρ=0.976
Model Predictions
Ground Truth
B
Pearson r^2=0.925 Spearman ρ=0.971
Model Predictions
Ground Truth
C
Pearson r^2=0.916 Spearman ρ=0.961
Model Predictions
Ground Truth
D
Pearson r^2=0.913 Spearman ρ=0.961
Model Predictions
Ground Truth
E
Pearson r^2=0.908 Spearman ρ=0.954
Model Predictions
Ground Truth
F
Pearson r^2=0.917 Spearman ρ=0.961
Model Predictions
Ground Truth
G
Pearson r^2=0.914 Spearman ρ=0.957
Model Predictions
Ground Truth
H
Pearson r^2=0.901 Spearman ρ=0.952
Model Predictions
Ground Truth
I
Pearson r^2=0.896 Spearman ρ=0.950
Model Predictions
Ground Truth
J
Pearson r^2=0.902 Spearman ρ=0.954
Model Predictions
Ground Truth
K
Pearson r^2=0.879 Spearman ρ=0.938
Model Predictions
Ground Truth

Supplementary Figure 6: Models capture expression at high levels in different ways. Scatter plots displaying the relationship between model predictions (*y*-axes) and ground truth (*x*-axes) for the top-performing teams in the challenge, for: (**A**) Autosome.org, (**B**) BHI, (**C**) Unlock_DNA, (**D**) Camformers, (**E**) NAD, (**F**) wztr, (**G**) High Schoolers, (**H**) BioNML, (**I**) BUGF, (**J**) mt, and (**K**) reference model.



Models	Pearson Score	All Sequences	High	Low	Native	Random	Challenging	SNVs	Motif Perturbation	Motif Tiling
Autosome.org	0.82	0.95	0.51	0.51	0.78	0.96	0.94	0.73	0.96	0.91
BHI	0.79	0.93	0.43	0.44	0.74	0.95	0.91	0.7	0.96	0.9
Unlock_DNA	0.77	0.92	0.38	0.39	0.7	0.94	0.88	0.68	0.94	0.89
wztr	0.76	0.92	0.27	0.3	0.68	0.95	0.91	0.68	0.94	0.89
Camformers	0.75	0.91	0.31	0.39	0.68	0.94	0.89	0.67	0.95	0.81
High Schoolers	0.75	0.91	0.35	0.32	0.66	0.94	0.86	0.67	0.94	0.87
BioNML	0.75	0.9	0.38	0.4	0.67	0.93	0.85	0.65	0.92	0.87
NAD	0.75	0.91	0.27	0.32	0.66	0.94	0.86	0.67	0.95	0.88
mt	0.74	0.9	0.36	0.37	0.68	0.93	0.86	0.65	0.93	0.84
BUGF	0.74	0.9	0.34	0.4	0.68	0.93	0.87	0.64	0.94	0.83
Yuanfang Guan	0.74	0.89	0.26	0.37	0.66	0.93	0.83	0.65	0.95	0.86
SYSU-SAIL-2022	0.73	0.89	0.22	0.38	0.68	0.93	0.86	0.65	0.95	0.82
Davuluri lab	0.73	0.89	0.25	0.37	0.7	0.93	0.84	0.63	0.93	0.83
Wen Group	0.73	0.89	0.3	0.34	0.67	0.93	0.82	0.63	0.93	0.83
NGT4	0.72	0.89	0.3	0.18	0.66	0.93	0.84	0.64	0.93	0.84
Metformin-121	0.72	0.89	0.24	0.36	0.66	0.92	0.84	0.62	0.93	0.8
Wan&Barton_BBK	0.7	0.88	0.18	0.34	0.66	0.91	0.81	0.61	0.91	0.78
The Dream Team	0.7	0.88	0.29	0.24	0.64	0.92	0.76	0.62	0.91	0.77
Peppa	0.7	0.88	0.23	0.23	0.63	0.91	0.84	0.6	0.91	0.79
reference model	0.7	0.88	0.26	0.27	0.64	0.91	0.77	0.62	0.9	0.75
KircherLab	0.68	0.87	0.26	0.11	0.62	0.92	0.78	0.61	0.89	0.79
Noisy-Chardonnay	0.67	0.87	0.06	0.23	0.64	0.92	0.81	0.57	0.92	0.75
MadLab	0.65	0.83	0	0.19	0.64	0.92	0.69	0.59	0.9	0.78
UTKbioinformatics	0.62	0.77	0.16	0.4	0.29	0.88	0.65	0.53	0.89	0.82
auth	0.62	0.81	0.11	0.27	0.56	0.89	0.61	0.53	0.88	0.69
DrAshokAndFriends	0.53	0.73	0.11	0.11	0.27	0.87	0.4	0.47	0.85	0.66
QUT_seq2exp	0.34	0.54	0.07	0	0.19	0.74	0.18	0.26	0.53	0.47
Zeta	0.19	0.42	0.05	0	0.14	0.55	0.15	0.06	0.33	0

Models	Spearman Score	All Sequences	High	Low	Native	Random	Challenging	SNVs	Motif Perturbation	Motif Tiling
Autosome.org	0.85	0.98	0.68	0.64	0.88	0.98	0.97	0.71	0.97	0.94
BHI	0.84	0.97	0.68	0.63	0.86	0.98	0.95	0.7	0.97	0.94
Unlock_DNA	0.82	0.96	0.58	0.6	0.84	0.97	0.93	0.68	0.96	0.93
wztr	0.81	0.96	0.61	0.3	0.81	0.98	0.94	0.68	0.96	0.93
Camformers	0.82	0.96	0.57	0.61	0.82	0.97	0.94	0.67	0.96	0.91
High Schoolers	0.81	0.96	0.53	0.5	0.81	0.97	0.91	0.67	0.96	0.92
BioNML	0.8	0.95	0.52	0.59	0.81	0.97	0.9	0.66	0.94	0.92
NAD	0.82	0.95	0.58	0.59	0.81	0.97	0.92	0.68	0.96	0.92
mt	0.8	0.95	0.45	0.59	0.82	0.97	0.92	0.65	0.95	0.89
BUGF	0.81	0.95	0.54	0.59	0.82	0.96	0.92	0.66	0.95	0.9
Yuanfang Guan	0.78	0.95	0.08	0.56	0.81	0.97	0.9	0.67	0.96	0.92
SYSU-SAIL-2022	0.79	0.95	0.36	0.58	0.82	0.97	0.91	0.65	0.96	0.9
Davuluri lab	0.76	0.95	-0.01	0.56	0.83	0.97	0.9	0.64	0.94	0.91
Wen Group	0.79	0.95	0.46	0.53	0.81	0.97	0.9	0.64	0.94	0.9
NGT4	0.78	0.95	0.47	0.4	0.81	0.97	0.91	0.64	0.96	0.9
Metformin-121	0.79	0.95	0.41	0.59	0.81	0.96	0.91	0.63	0.95	0.9
Wan&Barton_BBK	0.77	0.94	0.33	0.52	0.8	0.96	0.89	0.63	0.94	0.87
The Dream Team	0.77	0.94	0.5	0.33	0.79	0.96	0.84	0.65	0.95	0.89
Peppa	0.78	0.94	0.48	0.47	0.79	0.96	0.89	0.62	0.94	0.89
reference model	0.76	0.94	0.31	0.42	0.8	0.96	0.86	0.64	0.93	0.87
KircherLab	0.75	0.94	0.35	0.21	0.79	0.96	0.87	0.63	0.94	0.89
Noisy-Chardonnay	0.76	0.94	0.15	0.47	0.8	0.96	0.9	0.62	0.95	0.88
MadLab	0.75	0.94	0.12	0.55	0.8	0.96	0.85	0.62	0.95	0.87
UTKbioinformatics	0.7	0.89	-0.22	0.6	0.52	0.94	0.8	0.62	0.93	0.91
auth	0.72	0.9	0.21	0.49	0.75	0.95	0.78	0.59	0.92	0.84
DrAshokAndFriends	0.67	0.86	0.3	0.24	0.49	0.93	0.65	0.58	0.89	0.84
QUT_seq2exp	0.53	0.73	0.21	0.01	0.41	0.86	0.43	0.47	0.75	0.67
Zeta	0.43	0.66	0.24	-0.08	0.37	0.79	0.4	0.33	0.61	0.39

Supplementary Figure 7: Performance of the teams in each test data subset. (A,B) Model performance (colour and numerical values) of each team (y-axes) in each test subset (x-axes), for (A) Pearson r^2 and (B) Spearman ρ. Heatmap colour palettes are min-max normalized column-wise.

Acknowledgement

We extend our sincere gratitude to Jonathan Caton from Google Brain for his invaluable technical assistance in setting up the cloud resources for the challenge participants. Without his expert guidance and support, the successful organization of the challenge would not have been possible. We are also deeply grateful to TPU Research Cloud for providing the necessary cloud resources, which allowed us to ensure a level playing field for all challenge participants. We also thank Deep Genomics for providing travel grants to the top-performing teams in the challenge.

Consortia:

Random Promoter Dream Challenge Consortium members:

Susanne Bornelöv[1], Fredrik Svensson[2], Maria-Anna Trapotsi[1], Duc Tran[3], Tin Nguyen[3], Xinming Tu[4], Wuwei Zhang[4], Wei Qiu[4], Rohan Ghotra[5,6], Yiyang Yu[5,6], Ethan Labelson[6,7], Aayush Prakash[8], Ashwin Narayanan[9], Peter Koo[6], Xiaoting Chen[10], David T. Jones[2], Michele Tinti[11], Yuanfang Guan[12], Ding Maolin, Chen Ken, Ke Ding[13], Gunjan Dixit[13], Jiayu Wen[13], Zhihan Zhou[14], Pratik Dutta[15], Rekha Sathian[15], Pallavi Surana[15], Yanrong Ji[14], Han Liu[14], Ramana V Davuluri[15], Yu Hiratsuka[16], Mao Takatsu[16], Tsai-Min Chen[17,18], Chih-Han Huang[19], Hsuan-Kai Wang, Edward S.C. Shih[18], Sz-Hau Chen[20], Chih-Hsun Wu[21], Jhih-Yu Chen[17], Kuei-Lin Huang[22], Ibrahim Alsaggaf[23], Patrick Greaves[23], Carl Barton[23], Cen Wan[23], Nicholas Abad[24], Cindy Körner[24], Lars Feuerbach[24], Yichao Li[25], Sebastian Röner[26], Pyaree Mohan Dash[26], Max Schubach[26], Onuralp Soylemez[27], Andreas Møller[28], Gabija Kavaliauskaite[28], Jesper Madsen[28], Zhixiu Lu[29], Owen Queen[29], Ashley Babjac[29], Scott Emrich[29], Konstantinos Kardamiliotis[30], Konstantinos Kyriakidis[30], Andigoni Malousi[30], Ashok Palaniappan[31], Krishnakant Gupta[31], Prasanna Kumar S[31], Jake Bradford[32], Dimitri Perrin[32], Robert Salomone[32], Carl Schmitz[32], Chen JiaXing[33], Wang JingZhe[33], Yang AiWei[33]

Affiliations:

[1] University of Cambridge

[2] University College London

[3] University of Nevada Reno

[4] University of Washington

[5] Syosset High School

[6] Cold Spring Harbor Laboratory

[7] Friends Academy

[8] Half Hollow Hills High School

[9] Jericho High Schoo

[10] Cincinnati Children's Hospital Medical Center

[11] The Wellcome Centre for Anti-Infectives Research, Dundee University

[12] University of Michigan

[13] Australia National University

[14] Northwestern University

[15] Stony Brook University

[16] Niigata University School of Medicine

[17] National Taiwan University

[18] Academia Sinica

[19] ANIWARE

[20] Development Center for Biotechnology, Taipei

[21] National Chengchi University

[22] School of Medicine, China Medical University

[23] Univerity of London

[24] German Cancer Research Center, Heidelberg

[25] St. Jude Children's Research Hospital

[26] Berlin Institute of Health at Charité – Universitätsmedizin

[27] Global Blood Therapeutics

[28] University of Southern Denmark

[29] University of Tennessee at Knoxville

[30] Aristotle University of Thessaloniki

[31] SASTRA University

[32] Queensland University of Technology

[33] Beijing Normal University-Hong Kong Baptist University United International College

References:

1. Phillips, T. Regulation of transcription and gene expression in eukaryotes. *Nature Education* **2008**, *1*(1), 199.
2. Zeitlinger, J. Seven myths of how transcription factors read the cis-regulatory code. *Current opinion in systems biology* **2020**, *23*, 22–31. doi:10.1016/j.coisb.2020.08.002.
3. Lambert, S. A.; Jolma, A.; Campitelli, L. F.; Das, P. K.; Yin, Y.; Albu, M.; et al. The Human Transcription Factors. *Cell* **2018**, *172*(4), 650–665. doi:10.1016/j.cell.2018.01.029.
4. Alipanahi, B.; Delong, A.; Weirauch, M. T.; Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **2015**, *33*(8), 831–838. doi:10.1038/nbt.3300.

5. Zhou, J.; Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods* **2015**, *12*(10), 931–934. doi:10.1038/nmeth.3547.

6. de Boer, C. G.; Vaishnav, E. D.; Sadeh, R.; Abeyta, E. L.; Friedman, N.; Regev, A. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature Biotechnology* **2020**, *38*(1), 56–65. doi:10.1038/s41587-019-0315-8.

7. Agarwal, V.; Shendure, J. Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks. *Cell Reports* **2020**, *31*(7), 107663. doi:10.1016/j.celrep.2020.107663.

8. Avsec, Ž.; Agarwal, V.; Visentin, D.; Ledsam, J. R.; Grabska-Barwinska, A.; Taylor, K. R.; et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods* **2021**, *18*(10), 1196–1203. doi:10.1038/s41592-021-01252-x.

9. Avsec, Ž.; Weilert, M.; Shrikumar, A.; Krueger, S.; Alexandari, A.; Dalal, K.; et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics* **2021**, *53*(3), 354–366. doi:10.1038/s41588-021-00782-6.

10. Kelley, D. R.; Snoek, J.; Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research* **2016**, *26*(7), 990–999. doi:10.1101/gr.200535.115.

11. Quang, D.; Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research* **2016**, *44*(11), e107. doi:10.1093/nar/gkw226.

12. Vaishnav, E. D.; de Boer, C. G.; Molinet, J.; Yassour, M.; Fan, L.; Adiconis, X.; et al. The evolution, evolvability and engineering of gene regulatory DNA. *Nature* **2022**, *603*(7901), 455–463. doi:10.1038/s41586-022-04506-6.

13. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **2015**, *115*(3), 211–252. doi:10.1007/s11263-015-0816-y.

14. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; et al. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*. Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2014; pp 740–755. doi:10.1007/978-3-319-10602-1_48.

15. Meyer, P.; Saez-Rodriguez, J. Advances in systems biology modeling: 10 years of crowdsourcing DREAM challenges. *Cell Systems* **2021**, *12*(6), 636–653. doi:10.1016/j.cels.2021.05.015.

16. Sahu, B.; Hartonen, T.; Pihlajamaa, P.; Wei, B.; Dave, K.; Zhu, F.; et al. Sequence determinants of human gene regulatory elements. *Nature Genetics* **2022**, *54*(3), 283–294. doi:10.1038/s41588-021-01009-4.

17. Kinney, J. B.; Murugan, A.; Callan, C. G.; Cox, E. C. Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proceedings of the National Academy of Sciences* **2010**, *107*(20), 9158–9163. doi:10.1073/pnas.1004290107.

18. Sharon, E.; Kalma, Y.; Sharp, A.; Raveh-Sadka, T.; Levo, M.; Zeevi, D.; et al. Inferring gene regulatory logic from high-throughput measurements of thousands of

systematically designed promoters. *Nature Biotechnology* **2012**, *30*(6), 521–530. doi:10.1038/nbt.2205.

19. Shastry, B. S. SNPs in disease gene mapping, medicinal drug development and evolution. *Journal of Human Genetics* **2007**, *52*(11), 871–880. doi:10.1007/s10038-007-0200-z.

20. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*; PMLR, 2019; pp 6105–6114.

21. Tan, M.; Le, Q. EfficientNetV2: Smaller Models and Faster Training. In *Proceedings of the 38th International Conference on Machine Learning*; PMLR, 2021; pp 10096–10106.

22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Las Vegas, NV, USA, 2016; pp 770–778. doi:10.1109/CVPR.2016.90.

23. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

24. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv August 9, 2015. doi:10.48550/arXiv.1508.01991.

25. Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv January 29, 2017. doi:10.48550/arXiv.1412.6980.

26. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. arXiv January 4, 2019. doi:10.48550/arXiv.1711.05101.

27. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp 1532–1543. doi:10.3115/v1/D14-1162.

28. Ji, Y.; Zhou, Z.; Liu, H.; Davuluri, R. V. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* **2021**, *37*(15), 2112–2120. doi:10.1093/bioinformatics/btab083.

29. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv May 24, 2019. doi:10.48550/arXiv.1810.04805.

30. Zhou, H.; Shrikumar, A.; Kundaje, A. Towards a Better Understanding of Reverse-Complement Equivariance for Deep Learning Models in Genomics. In *Proceedings of the 16th Machine Learning in Computational Biology meeting*; PMLR, 2022; pp 1–33.

31. Zhuang, J.; Gong, B.; Yuan, L.; Cui, Y.; Adam, H.; Dvornek, N.; et al. Surrogate Gap Minimization Improves Sharpness-Aware Training. arXiv March 19, 2022. doi:10.48550/arXiv.2203.08065.

32. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A. G. Averaging Weights Leads to Wider Optima and Better Generalization. arXiv February 25, 2019. doi:10.48550/arXiv.1803.05407.

33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv June 3, 2021. doi:10.48550/arXiv.2010.11929.

34. Liu, L.; Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; et al. On the Variance of the Adaptive Learning Rate and Beyond. arXiv October 25, 2021. doi:10.48550/arXiv.1908.03265.

35. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. arXiv February 7, 2018. doi:10.48550/arXiv.1708.02002.

36. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv December 11, 2014. doi:10.48550/arXiv.1412.3555.

37. Karollus, A.; Mauermeier, T.; Gagneur, J. Current sequence-based models capture gene expression determinants in promoters but mostly ignore distal enhancers. bioRxiv September 17, 2022, p 2022.09.15.508087. doi:10.1101/2022.09.15.508087.

38. de Boer, C. G.; Ray, J. P.; Hacohen, N.; Regev, A. MAUDE: inferring expression changes in sorting-based CRISPR screens. *Genome Biology* **2020**, *21*(1), 134. doi:10.1186/s13059-020-02046-8.

39. Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* **2017**, *60*(6), 84–90. doi:10.1145/3065386.

40. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. arXiv April 30, 2016. doi:10.48550/arXiv.1511.07122.

41. Koo, P. K.; Majdandzic, A.; Ploenzke, M.; Anand, P.; Paul, S. B. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLOS Computational Biology* **2021**, *17*(5), e1008925. doi:10.1371/journal.pcbi.1008925.

42. Ramachandran, P.; Zoph, B.; Le, Q. V. Searching for Activation Functions. arXiv October 27, 2017. doi:10.48550/arXiv.1710.05941.

43. Shazeer, N. GLU Variants Improve Transformer. arXiv February 12, 2020. doi:10.48550/arXiv.2002.05202.

44. Ronneberger, O.; Fischer, P.; Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.org. <https://arxiv.org/abs/1505.04597v1> Accessed 23.04.19.

45. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. arXiv May 16, 2019. doi:10.48550/arXiv.1709.01507.

46. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. arXiv April 4, 2017. doi:10.48550/arXiv.1610.02357.

47. Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; Hager, G. D. Temporal Convolutional Networks for Action Segmentation and Detection. arXiv November 16, 2016. doi:10.48550/arXiv.1611.05267.

48. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. arXiv September 19, 2015. doi:10.48550/arXiv.1506.04214.

49. Ng, P. dna2vec: Consistent vector representations of variable-length k-mers. arXiv January 23, 2017. doi:10.48550/arXiv.1701.06279.