



OPEN

# Effective gene expression prediction from sequence by integrating long-range interactions

Žiga Avsec<sup>1</sup>✉, Vikram Agarwal<sup>2,4</sup>, Daniel Visentin<sup>1,4</sup>, Joseph R. Ledsam<sup>1,3</sup>, Agnieszka Grabska-Barwinska<sup>1</sup>, Kyle R. Taylor<sup>1</sup>, Yannis Assael<sup>1</sup>, John Jumper<sup>1</sup>, Pushmeet Kohli<sup>1</sup>✉ and David R. Kelley<sup>1</sup>✉

**How noncoding DNA determines gene expression in different cell types is a major unsolved problem, and critical downstream applications in human genetics depend on improved solutions. Here, we report substantially improved gene expression prediction accuracy from DNA sequences through the use of a deep learning architecture, called Enformer, that is able to integrate information from long-range interactions (up to 100 kb away) in the genome. This improvement yielded more accurate variant effect predictions on gene expression for both natural genetic variants and saturation mutagenesis measured by massively parallel reporter assays. Furthermore, Enformer learned to predict enhancer-promoter interactions directly from the DNA sequence competitively with methods that take direct experimental data as input. We expect that these advances will enable more effective fine-mapping of human disease associations and provide a framework to interpret *cis*-regulatory evolution.**

Models that predict gene expression and chromatin states from DNA sequences hold the promise to better understand transcriptional regulation and how it is affected by the many noncoding genetic variants associated with human diseases and traits. These models complement population-based association studies, which are often limited to common variants and struggle to disentangle causality from association due to linkage disequilibrium (LD). Additionally, experimental validation of human genetic variants is laborious and limited to cell types or tissues that can be recapitulated in the laboratory, making it intractable to test all variants of interest in the relevant biological contexts. Although sequence-based computational models can in principle overcome these challenges, their accuracy is still limited<sup>1–4</sup>, making expression prediction from sequence a critical unsolved problem.

Deep convolutional neural networks (CNNs) achieve the current state of the art at predicting gene expression from DNA sequences for the human and mouse genomes<sup>1–4</sup>. However, to make predictions, these models are only able to consider sequence elements up to 20 kb away from the transcription start site (TSS) because the locality of convolutions limits information flow in the network between distal elements. Many well-studied regulatory elements, including enhancers, repressors, and insulators, can influence gene expression from far greater than 20 kb away<sup>5</sup>. Thus, increasing information flow between distal elements is a promising path to increase predictive accuracy.

In this work, we introduce a neural network architecture based on self-attention towards this goal. We frame the machine learning problem as predicting thousands of epigenetic and transcriptional datasets in a multitask setting across long DNA sequences. Training on most of the human and mouse genomes and testing on held out sequences, we observed improved correlation between predictions and measured data relative to previous state-of-the-art models without self-attention. We demonstrate more effective use of long-range information, as benchmarked by CRISPRi enhancer assays. The model also produces more accurate predictions of

mutation effects, as measured by direct mutagenesis assays and population eQTL studies.

## Results

**Enformer improves gene expression prediction.** We developed a new model architecture named Enformer (a portmanteau of enhancer and transformer) to predict gene expression and chromatin states in humans and mice from DNA sequences (Fig. 1a and Extended Data Fig. 1). Transformers are a class of deep learning models that have achieved substantial breakthroughs in natural language processing (NLP)<sup>6,7</sup> and were also recently applied to model short DNA sequences<sup>8</sup>. They consist of attention layers that transform each position in the input sequence by computing a weighted sum across the representations of all other positions in the sequence. Attention weight between any two positions depends on the embeddings of their current representation vectors and the distance between them. This allows the model, for example, to refine the prediction at a TSS by gathering information from all relevant regions, such as enhancers regulating the gene. Since each position directly attends to all other positions in the sequence, they allow for a much better information flow between distal elements. By contrast, convolutional layers require many successive layers to reach distal elements due to their local receptive field. Using transformer layers allowed us to substantially increase the receptive field, reaching distal regulatory elements up to 100 kb away while still being able to effectively integrate their information. By contrast, previous state-of-the-art models Basenji2 or ExPecto only reach elements up to 20 kb away (Extended Data Fig. 1). This increase in the receptive field is important because it greatly expands the number of relevant enhancers seen by the model from 47% (<20 kb) to 84% (<100 kb) as estimated from the proportions of high-confidence enhancer-gene pairs<sup>9</sup>.

Enformer substantially outperformed the previous best model, Basenji2, for predicting RNA expression as measured by Cap Analysis Gene Expression<sup>10</sup> (CAGE) at the TSS of human

<sup>1</sup>DeepMind, London, UK. <sup>2</sup>Calico Life Sciences, South San Francisco, CA, USA. <sup>3</sup>Google, Tokyo, Japan. <sup>4</sup>These authors contributed equally: Vikram Agarwal, Daniel Visentin. ✉e-mail: [avsec@google.com](mailto:avsec@google.com); [pushmeet@google.com](mailto:pushmeet@google.com); [drk@calicolabs.com](mailto:drk@calicolabs.com)

protein-coding genes, with the mean correlation increasing from 0.81 to 0.85 (Fig. 1b, left). This performance increase is twice as large as the performance increase between Basenji1 (ref. <sup>3</sup>) and Basenji2 (ref. <sup>2</sup>) and closes one-third of the gap to experimental-level accuracy, estimated at 0.94 (Extended Data Fig. 2). Gene expression predictions also better captured tissue- or cell-type specificity (Fig. 1b, right), including for closely related samples (Extended Data Fig. 3). The performance improvement was consistent across all four types of genome-wide tracks, including CAGE measuring transcriptional activity, histone modifications, TF binding, and DNA accessibility in various cell types and tissues for held-out chromosomes (Fig. 1c). The performance improvement was largest for CAGE, possibly because tissue-specific gene expression strongly depends on distal elements<sup>11</sup>. The improvement in prediction accuracy was also qualitatively evident when visualizing observed and predicted tracks of the genome (Fig. 1d). Enformer also yielded greater predictive accuracy than ExPecto<sup>1</sup>, a model trained to predict gene expression levels measured by RNA-seq, for both across-genes (0.850 versus 0.812 Spearman  $r$ ) and across-tissues (0.451 versus 0.368 Spearman  $r$ ) evaluation (Extended Data Fig. 4). These results confirm that the Enformer architecture advances prediction accuracy for both a broad range of epigenetic marks and gene expression from DNA sequence.

To pinpoint the benefit of attention layers compared with the dilated convolutions used in Basenji2, we replaced attention layers with dilated convolutions and tuned the learning rate for optimal performance. Attention layers outperformed dilated convolutions across all model sizes, numbers of layers, and numbers of training data points (Extended Data Fig. 5a). The larger receptive field was indeed crucial, because we observed a large performance drop when restricting the receptive field of Enformer to that of Basenji2 by replacing global attention layers with local ones (Extended Data Fig. 5b). We note that increasing the number of parameters improved model performance, consistent with recent advances in NLP<sup>7</sup>. Enformer uses custom relative positional basis functions in the transformer layers to more easily distinguish between proximal and distal regulatory elements, and to distinguish positions upstream and downstream of the TSS. Both properties provided a noticeable performance improvement over the typically used relative basis functions and absolute positional encodings in the NLP literature (Extended Data Fig. 6a,b). Overall, these results confirm that attention layers are better suited than dilated convolutions for gene expression prediction.

**Enformer attends to cell-type-specific enhancers.** To better understand what sequence elements Enformer is utilizing when making predictions, we computed two different gene expression contribution scores — input gradients (gradient  $\times$  input)<sup>12</sup> and attention weights (Methods and Supplementary Fig. 1) — for several genes with CRISPRi-validated enhancers<sup>9,13</sup>. Contribution scores highlight the input sequences that are most predictive for the expression of a

particular gene<sup>14,15</sup>. In silico mutagenesis and gradient  $\times$  input are tissue- or cell-type-specific, since they are computed with respect to a particular output CAGE sample (for example, K562). By contrast, attention weights are internal to the model and are shared among all tissue and cell-type predictions. We inspected the contribution scores of several genes and observed that they correlated with histone H3 acetylated at K27 (H3K27ac) and highlighted not only local promoter regions, but also distal enhancers more than 20 kb away (Fig. 2a and Supplementary Figs. 2 and 3). By contrast, the contribution scores of Basenji2 were zero for sequences beyond 20 kb from the TSS due to the limited receptive field, thereby missing several enhancers. This example suggests that Enformer is indeed looking at biologically relevant regions, such as enhancers beyond 20 kb, when making predictions, and that gene expression contribution scores could be used to prioritize relevant enhancers.

Linking candidate enhancers identified via biochemical annotations<sup>16</sup> to target genes is an important and unsolved problem<sup>5</sup>. Computational models have historically produced low accuracy owing to the combination of noisy labels and class imbalance. To systematically evaluate the ability of contribution scores to pinpoint relevant enhancers for a particular gene, we compared several contribution scores across all tested enhancer–gene pairs in two large-scale CRISPRi studies performed on the K562 cell line<sup>9,13</sup>. In these experiments, CRISPRi was used to suppress the activity of more than 10,000 candidate enhancers and measure their effect on gene expression.

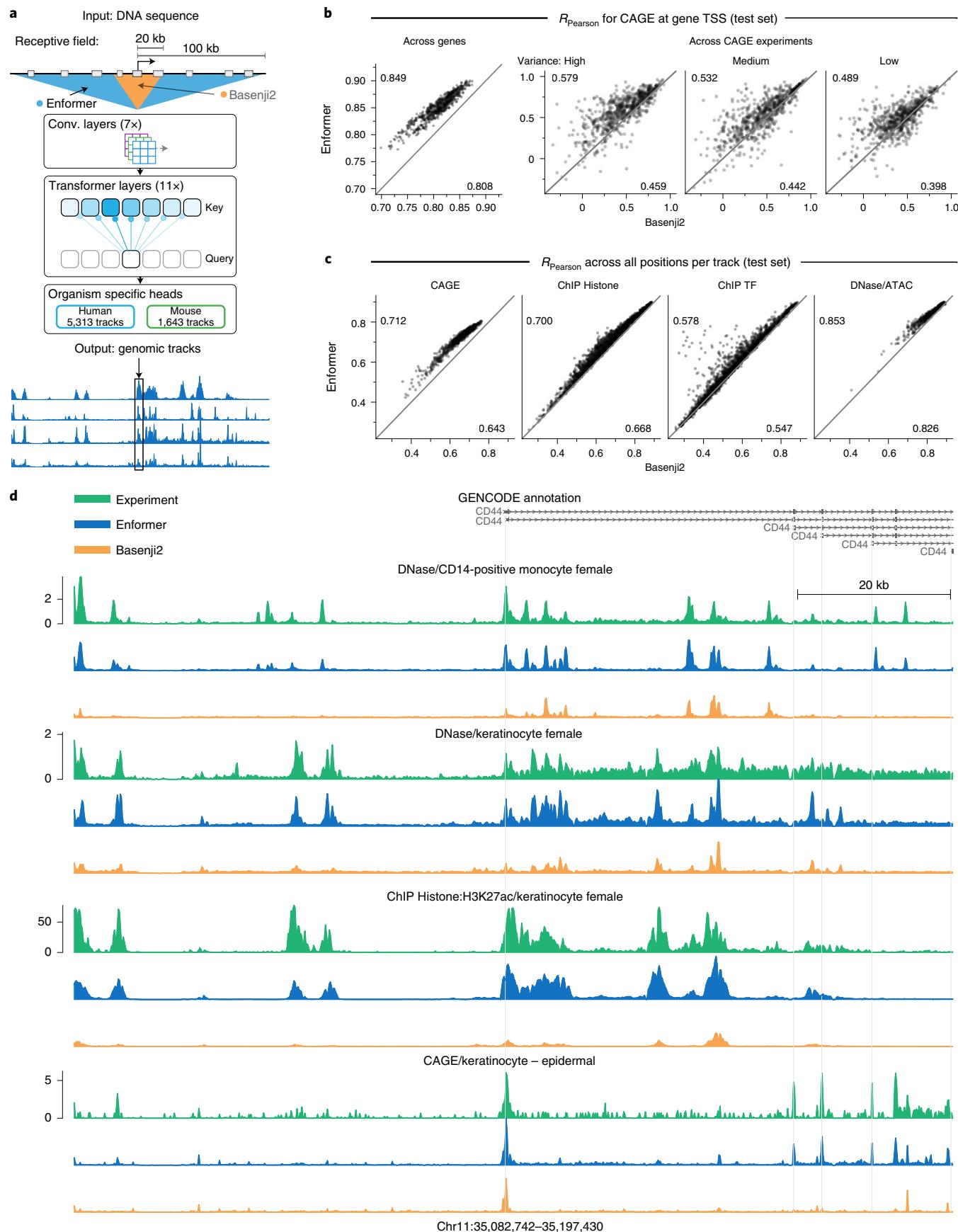
Enformer contribution scores prioritized validated enhancer–gene pairs with higher accuracy than Basenji2 contribution scores or random scores across almost all relative distances and different types of contribution scores (Fig. 2b, Enformer versus Basenji2 versus Random). The performance of Enformer was comparable to, and in some cases even better than, the ABC score<sup>13</sup>, a state-of-the-art method recently proposed specifically for enhancer prioritization. This is remarkable because the ABC score relies on experimental data, such as a HiC-based interaction frequency and H3K27ac as input (Fig. 2b, blue versus green, and Extended Data Fig. 7a), whereas Enformer uses only DNA sequence as input and was never trained to explicitly locate enhancers. This allows Enformer to also be used for arbitrary sequence variations lacking experimental data. Cell-type-specific contribution scores yielded a higher prioritization performance than cell-type-agnostic ones, suggesting that the model was using different enhancer sequences in different cell types as expected (Extended Data Fig. 7c). Thus, Enformer contribution scores are an effective strategy to prioritize candidate enhancers in cell types used for model training.

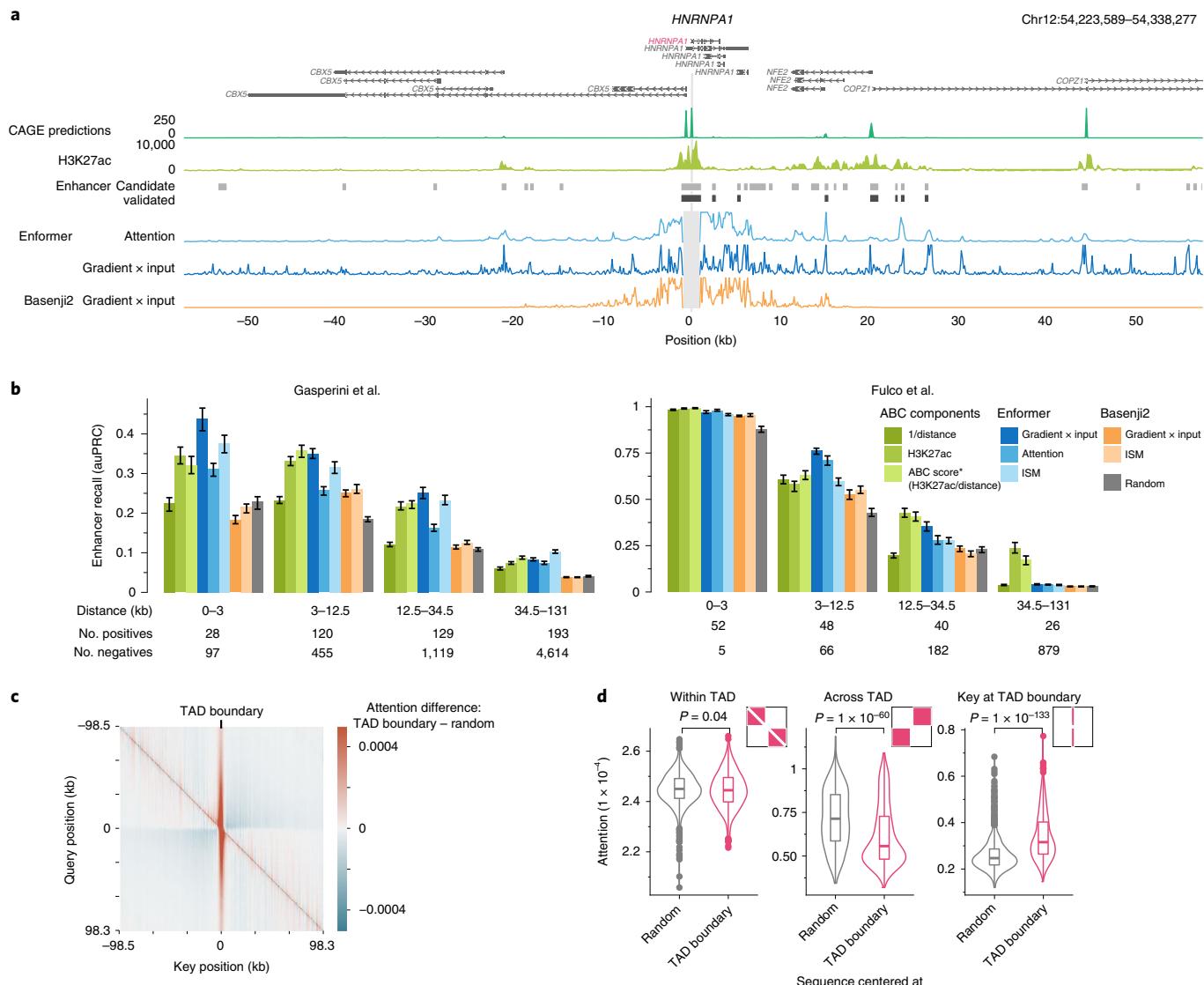
Next, we asked whether the model has learned about another important class of regulatory elements: insulator elements, which separate two topologically associating domains (TADs) and minimize enhancer–promoter crosstalk between the two. We inspected the attention matrices (which were more efficient to compute relative to input gradients owing to the many output targets) of sequences

**Fig. 1 | Enformer improves gene expression prediction in held-out genes by using a larger receptive field.** **a**, Enformer is trained to predict human and mouse genomic tracks at 128-bp resolution from 200 kb of input DNA sequence. By using transformer modules instead of dilated convolutions, it achieves a five times larger receptive field able to detect sequence elements 100 kb away, compared with 20 kb for Basenji2 (ref. <sup>2</sup>) or ExPecto<sup>1</sup> (Extended Data Fig. 1). **b**, Enformer outperforms Basenji2 in gene expression prediction from sequence both across genes and across CAGE experiments for protein-coding genes. Test set performance was measured by Pearson correlation of CAGE gene expression ( $\log(1 + x)$  transformed) computed across genes for each CAGE experiment (left) or across CAGE experiments for each test gene stratified by the observed expression variance across experiments (right). Average performance for each model is shown in the corners. Bootstrapped s.d. of these estimates is 0.004 for ‘Across genes’. Gene expression values were obtained by summing up the observed or predicted CAGE read counts at all unique TSS locations of the gene. Values for each CAGE experiment were standardized to have zero mean and variance of 1 across genes. **c**, Enformer consistently outperforms Basenji2 across all 4 assay types (columns) as measured by Pearson correlation computed across all 128-bp binned genomic positions in the human test set for 5,313 predicted tracks (points). Both models were trained and evaluated on the same dataset. Enformer performance was significantly higher across all plots in **b** and **c** (paired Wilcoxon  $P < 10^{-38}$ ). **d**, Representative example of observed and predicted genomic tracks ( $\log_{10}$  scale) at CD44 gene locus located in the test-set region with high disagreement between Enformer and Basenji2 predictions (Methods). For each experiment, all three tracks share the same y axis.

centered at TAD boundaries, and compared them with attention from sequences with no particular alignment. From the perspective of the query position, Enformer paid more attention to TAD

boundaries than to random positions (vertical red stripe, Fig. 2c) and less attention to regions on the opposite side of the boundary (off-diagonal blue blocks, Fig. 2c), consistent with the biological





**Fig. 2 | Enformer attends to cell-type-specific enhancers, enabling enhancer prioritization.** **a**, HNRNPA1 locus showing: predicted CAGE expression in K562; measured H3K27ac highlighting active enhancers; candidate (light gray) and CRISPRi-validated enhancers (dark gray) exhibiting significant HNRNPA1 expression changes from Fulco et al.<sup>13</sup>; enformer attention weight averaged across all layers and heads for a query placed at the main TSS of HNRNPA1 gene (position 0); and gradient × input<sup>12</sup> contribution scores computed with regard to the K562 CAGE track at the main TSS position for Enformer and Basenji2. **b**, Enhancer-gene pair classification performance (CRISPRi-validated versus nonvalidated candidate enhancers), stratified by relative distance, as measured by auPRC on two CRISPRi datasets<sup>9,13</sup> for different methods, models, and contribution scores (Methods). ABC score\* (H3K27ac/distance) denotes the approximate version of the ABC score<sup>13</sup> lacking Hi-C data, which exhibits similar performance (Extended Data Fig. 7a). Colored bars depict the median auPRC, and error bars show the 25th and 75th percentiles obtained by sampling 80% of enhancer-gene pairs 100 times without replacement. The auPRC metric is sensitive to class imbalance, which differs between the two datasets (1:10 for Gasperini<sup>9</sup> and 1:4 for Fulco<sup>13</sup>). **c**, Average attention matrix difference of Enformer between 1,500 sequences centered at a topologically associating domain (TAD) boundary and 1,500 sequences from the validation set without any particular centering. Attention matrices were averaged across all layers, heads, and sequences. Red stripe in the center at key = 0 means that the model is attending more to the TAD boundary than by chance. Blue regions in off-diagonal quadrants mean that the model is attending less across the TAD boundary. **d**, Attention is significantly lower across TAD boundaries (center), significantly higher at TAD boundaries (right), and shows no significant difference within them (left), as compared with 1,500 random genomic sequences. Distributions show attention across all sequences in specific attention matrix parts shown in red. *P* values were computed with the two-sided Mann-Whitney U test. The box plots mark the median, upper and lower quartiles, and 1.5× interquartile range (whiskers); outliers are shown as points ( $n = 1,500$  for each violin plot).

observation of reduced inter-TAD interactions. Both of these two patterns were statistically significant across 1,500 tested sequences (Fig. 2d, 'Across TAD' and 'Key at TAD boundary'). One of the key motifs at TAD boundaries that the model used to make DNase and CAGE predictions was CTCF, which was found to be associated with both positive and negative contribution scores (Extended Data Fig. 8). Overall, these results suggest that the model has not only

learned about the role of tissue-specific enhancers and promoters, but also about insulator elements and their role in inhibiting information flow between genomic compartments.

**Enformer improves variant effect prediction on eQTL data.** A central goal of this research is to predict the influence of genetic variants on cell-type-specific gene expression, in order to inform

fine-mapping of the many thousands of noncoding associations with phenotypes of interest from genome-wide association studies (GWAS). Computational models that predict regulatory activity from DNA sequences can process distinct alleles and compare predictions to score genetic variants<sup>3,17–19</sup>. A successful model would be able to produce the results of a gene expression quantitative trait loci (eQTL) study without having to measure hundreds to thousands of individual gene expression profiles. Thus, we studied eQTLs discovered by the GTEx project across dozens of human tissues to validate model predictions<sup>20</sup>. The primary challenge of such validation is the influence of co-occurrences between variants (that is, linkage disequilibrium) in the profiled population, which transfers the causal eQTL's effect to nearby co-occurring variants' measurements. Signed linkage disequilibrium profile (SLDP) regression is a technique developed to measure the genome-wide statistical concordance between signed variant annotations (such as our model predictions) and GWAS summary statistics (such as GTEx eQTLs) while accounting for linkage disequilibrium (Methods)<sup>21</sup>. For 379 of 648 (59.4%) CAGE datasets, the maximum SLDP Z-score across GTEx tissues (representing the most likely closest sample match) increased for Enformer predictions relative to Basenji2. Enformer maximum Z-scores increased by greater than one s.d. for 228 CAGE datasets, relative to 46 decreased by one. The maximum Z-score increased on average from 6.3 to 6.9 (Fig. 3a). Note that we do not expect increased SLDP Z-scores for CAGE samples without a relevant GTEx tissue match. We observed a qualitative improvement in the tissue similarity of the top-ranked CAGE sample for GTEx tissues, exemplified by increased SLDP Z-scores for muscle samples to GTEx skeletal muscle and adipose samples for GTEx subcutaneous adipose tissue (Fig. 3b,c). We also found that Enformer variant effect predictions for DNase hypersensitivity had greater SLDP concordance with GTEx than an alternative method called DeepSEA Beluga, used in ExPecto<sup>1</sup> (Extended Data Fig. 9). Thus, Enformer predictions for noncoding-variant activity appear to improve primarily for samples with similar cell-type composition, in line with our observations of improved tissue and cell-type specificity for held-out sequences.

Although linkage disequilibrium generally results in GTEx eQTL associations that can be attributed only to a set of frequently co-occurring variants, the latest GTEx release includes many thousands of associations in loci with simple linkage patterns, which have been fine-mapped to a single high-probability causal variant<sup>22</sup>. To assess the utility of Enformer predictions for identifying causal variants, we defined a classification task for each tissue to discriminate likely causal variants (causal probability > 0.9, as determined by the population-based fine-mapping model SuSiE<sup>23</sup>) from likely spurious eQTLs (causal probability < 0.01), which were matched for the eGene when possible (Methods). We represented each variant by its prediction difference vector (that is, evaluating the reference minus alternative allele, summed across the sequence) for all 5,313 human datasets, and trained random forest classifiers. Enformer

predictions enabled a more accurate classifier for 47 of 48 GTEx tissues (Fig. 3d), increasing the mean area under the receiver operating characteristic curve (auROC) from 0.729 to 0.747. This improvement was consistent across all distances from the TSS (Fig. 3e), suggesting that the model not only better represents variants likely overlapping long-range enhancers (enabled by the larger receptive field), but also more effectively parses promoters and short-range enhancers. The Enformer model was also more accurate at predicting the direction of expression change of these fine-mapped eQTLs than was Basenji2 (Extended Data Fig. 10).

One example variant where the Enformer eQTL probability prediction increased relative to Basenji2 is rs11644125, which lies within an intron ~35 kb downstream of the TSS of *NLRC5*, a gene involved in viral immunity and the cytokine response (Fig. 3f). The variant has been statistically fine-mapped as likely to cause changes in monocyte and lymphocyte blood cell counts<sup>24</sup>. According to GTEx, the minor allele T decreases gene expression of *NLRC5* in whole blood relative to the major allele C. Enformer correctly predicts reduced *NLRC5* expression from the upstream TSS in many relevant CAGE samples, including PBMCs. Using *in silico* mutagenesis of the local region (Methods), we observed that the variant rs11644125 modulates the known motif of the transcription factor SP1 (ref. <sup>24</sup>). Enformer predictions suggest perturbed SP1 binding in hematopoietic cells that alters *NLRC5* expression as a mechanism for these traits.

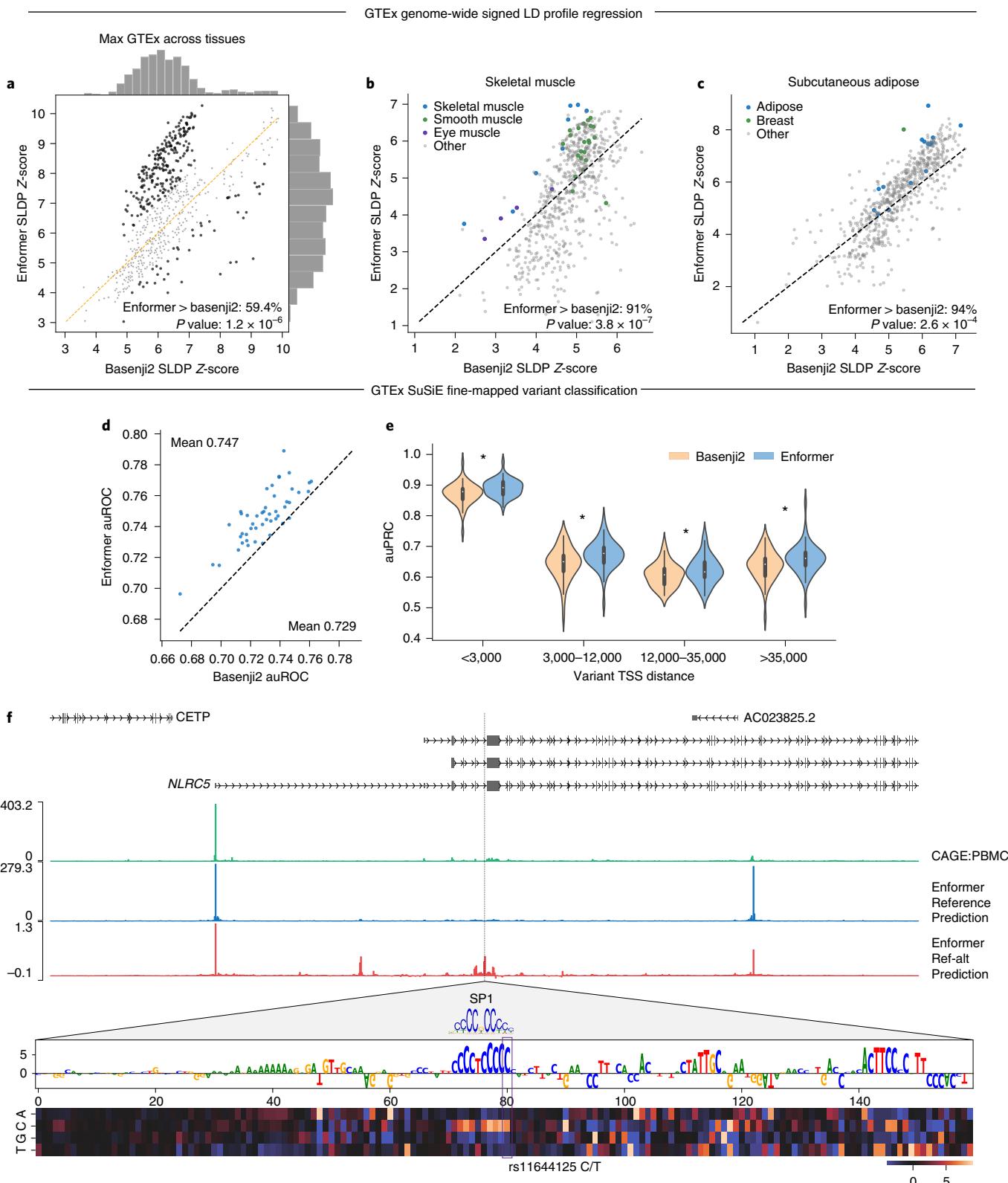
**Enformer improves MPRA mutation effect prediction.** Finally, we evaluated Enformer's performance on a second, independent variant effect prediction task using a dataset in which massively parallel reporter assays (MPRAs) directly measured the functional effect of genetic variants through saturation mutagenesis of several enhancers and promoters in a variety of cell types<sup>25</sup>. We used the same training and test sets as the CAGI5 competition<sup>26</sup>, enabling us to directly benchmark Enformer's performance relative to those of submissions from other groups. Methods derived from other groups deploy a heterogeneous set of approaches, ranging from the use of the deltaSVM strategy<sup>27</sup>, the CADD framework<sup>28</sup>, and regression models using features derived from a combination of conservation information and deep learning predictions from DeepBind<sup>29</sup> and DeepSEA<sup>18</sup> (Group 3, Group 5, and Group 7)<sup>26</sup>. For each variant, we evaluated its effect as the predicted difference between the reference and alternative allele, retrieving 5,313 features. Next, we compared two approaches: (1) we used these features to train a lasso regression model on the provided training set for each gene, and (2) we preselected a subset of features corresponding to cell-type-matched and cell-type-agnostic predictions of changes in CAGE and DNase, and generated a summary statistic of the features (that is, without additional training).

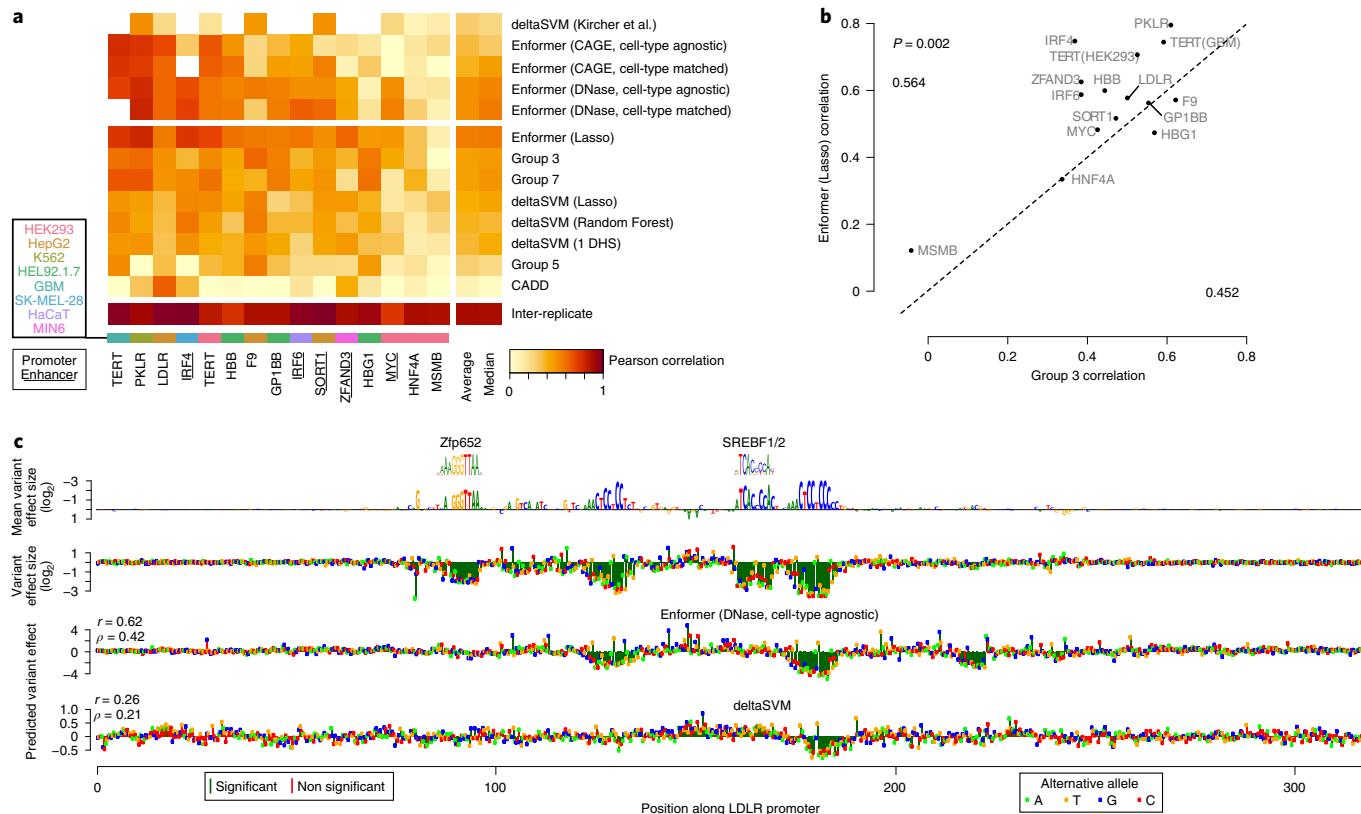
Evaluating these two approaches on each gene's test set revealed that lasso regression with Enformer predictions as features had the best average correlation across all loci, among seven alternative

**Fig. 3 | Enformer improves variant effect prediction on eQTL data as measured by SLDP regression and fine-mapped variant classification.** **a**, We computed genome-wide statistical concordance between variant effect predictions for individual CAGE datasets and GTEx eQTL summary statistics using SLDP<sup>21</sup> across all variants in the 1000 Genomes dataset. Taking the GTEx tissue with max Z-score for each sample, Enformer predictions achieved greater Z-scores for 59.4% of samples, and 228 are greater by more than one s.d. (versus 46 for Basenji2). Each point represents one of the 638 CAGE samples. We used one-sided Binomial tests to compute the *P* values in the top row panels. **b,c**, Studying SLDP in skeletal muscle (**b**) and subcutaneous adipose (**c**) GTEx tissues indicated that biologically relevant CAGE datasets (shown in blue) improve between Basenji2 and Enformer. **d**, We trained random forest classifiers to discriminate between fine-mapped GTEx eQTLs and matched negative variants in each of 48 tissues (Methods). Features derived from Enformer enabled more accurate classifiers than Basenji2 features for 47 of 48 tissues. **e**, We computed auPRC for variants in four roughly equally sized TSS distance bins. Violin plots represent measures for the  $n = 48$  tissues (white dots represent the median, thick bars the interquartile range, and thin bars the entire data range). Enformer improved accuracy at all distances (one-sided paired Wilcoxon  $P < 1 \times 10^{-4}$ ). **f**, Enformer prediction for rs11644125 improved relative to Basenji2 (data not shown) by better capturing its influence on an *NLRC5* TSS ~35 kb upstream. rs11644125 is associated with monocyte and lymphocyte counts in the UK Biobank and fine-mapped to >0.99 causal probability<sup>24</sup>. *In silico* mutagenesis of the region surrounding rs11644125 revealed an affected SP1 transcription factor motif<sup>39</sup>.

submissions from the competition (Fig. 4a). Moreover, using the Enformer predictions directly as scores, without training, performed comparably to the lasso-trained model and also outperformed the other submissions. This includes the sequence-based predictor deltaSVM<sup>27</sup>, which was trained on independent DNase and histone H3 monomethylated at K4 (H3K4me1) data derived from matched cell

types<sup>25</sup>. The lasso-trained Enformer exceeded the performance of Group 3, the winning team from CAGI5 ( $P=0.002$ , paired, one-sided Mann–Whitney U test, Fig. 4b). Visualization of the predictions that required no additional training revealed that Enformer faithfully captured the effects of two out of four transcription-factor-binding sites for the *LDLR* locus (Fig. 4c). Enformer highlighted an





**Fig. 4 | Enformer improves noncoding variant effect prediction as measured by saturation mutagenesis experiments.** **a**, Correlation of variant effect predictions with experimental values, as measured by saturation mutagenesis MPRAs<sup>25</sup>, on test sets for 15 loci curated for the CAGI5 competition<sup>26</sup>. Shown above the horizontal break is the performance of five methods that required no additional fine-tuning on each locus; shown below is that of eight methods that were additionally trained on the CAGI5 training sets. **b**, Pearson correlations of each locus for predictions derived from the Enformer versus the winning team of the CAGI5 competition. Average performance for each model is shown in the corners. Enformer shows a significant performance improvement ( $P = 0.002$ , paired, one-sided Mann-Whitney U test). **c**, Example saturation mutagenesis data from the *LDLR* promoter locus. Shown in the top row is the reference sequence scaled to the mean effect size among all alternative mutations, with measured effect sizes of individual variants in the second row. Two of the four significant elements match known motifs<sup>39</sup>, and the two unknown motifs partially resemble the SP1 binding motif. Shown in the bottom two rows are the predictions on the full dataset using methods from **a** that required no additional fine-tuning.

additional binding site that had lower effect sizes, but still showed a significant difference. By contrast, deltaSVM successfully predicted only one binding site but missed the other three, overall exhibiting 50% reduced Pearson and Spearman correlations to the measured effects relative to Enformer. For this locus, cell-type-matched predictions mirrored cell-type-agnostic predictions, indicating that the binding sites which were detected likely corresponded to general transcription factors present in most cell types.

## Discussion

A long-standing problem in regulatory genomics is that of predicting gene expression purely from DNA sequence. With a novel transformer architecture, we have made a significant improvement by greatly expanding the receptive field and increasing the information flow between distal elements. In this way, the model can better capture biological phenomena such as enhancers regulating promoters despite a large DNA-sequence distance between the two. This led to a substantial performance increase in tissue- and cell-type-specific gene expression prediction correlation from 0.81 to 0.85, one-third of the way toward the experimental-level accuracy of 0.94 estimated from replicates.

This improvement in predictive accuracy translated to improved models for two key problems of biological relevance: enhancer-promoter prediction and noncoding variant effect prediction.

We observed that the model pays attention to enhancers and considers insulators when making gene expression predictions, suggesting that it has learned canonical distal regulation patterns. Using the Enformer model, we can more accurately predict whether a natural variant or CRISPR-perturbed enhancer will cause a notable expression change than can previous approaches. By relying solely on DNA sequences as input, Enformer has several advantages over alternative variant effect prediction methods: (1) unlike most methods<sup>25</sup>, it is capable of signed prediction of activating or repressive mutations; (2) by not relying explicitly on nucleotide conservation statistics, as the majority of tools do<sup>25</sup>, its predictions are not limited to conserved enhancers, which comprise a small proportion of all enhancers<sup>30</sup>; and (3) it can make predictions for arbitrary sequences, which enables the synthetic design of enhancers that are optimized to exhibit cell-type specificity<sup>31</sup>. Altogether, these advances and advantages open exciting avenues to study the expanding catalogs of genetic variants linked to disease and enhancer biology in development and evolution.

Several paths to further improve model accuracy appear promising. Machine-learning success depends on the training data, so increasing the resolution and quality of the target tracks<sup>15</sup>, and curating data from additional organisms<sup>2</sup>, would likely boost performance. Recent work demonstrated that the highly structured 3D DNA contacts, which greatly influence long-range gene regulation,

are predictable from the underlying DNA sequence<sup>32,33</sup>. Artful combination of these models with our own could improve Enformer's modeling of insulators and distal regulation. A limitation of the current approach is that we can model and predict only for cell types and assays in the training data and cannot generalize to new cell types or assays. Parallel research has begun to address this shortcoming via representation learning of cell types and assays and could make use of the Enformer architecture in the future<sup>34,35</sup>. The sensitivity of the model to genetic variants could be further improved by training upon the growing number of functional genomic datasets, such as those derived from CRISPR perturbations and massively parallel reporter assays. Currently, the small size of these datasets has limited their usage only to model evaluation. Finally, we anticipate that recent improvements in the computational efficiency<sup>36</sup> of transformer models together with better hardware will allow us to further scale-up the models.

In the future, Enformer could be systematically applied to fine-map existing GWAS studies<sup>22</sup>, prioritize rare or de novo variants observed for rare disorders<sup>37,38</sup>, and impute regulatory activity across species to study *cis*-regulatory evolution<sup>2</sup>. To foster these downstream applications, we have made the pretrained Enformer model openly available along with code examples demonstrating its use. Furthermore, we have precomputed effect predictions for all frequent variants in the 1000 Genomes dataset and made them openly available. We hope that our model will stimulate an improved understanding of gene-regulatory architecture and facilitate the development of improved diagnostic tools for diseases of genetic origin.

## Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41592-021-01252-x>.

Received: 17 February 2021; Accepted: 27 July 2021;

Published: xx xx xxxx

## References

- Zhou, J. et al. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.* **50**, 1171–1179 (2018).
- Kelley, D. R. Cross-species regulatory sequence activity prediction. *PLoS Comput. Biol.* **16**, e1008050 (2020).
- Kelley, D. R. et al. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.* **28**, 739–750 (2018).
- Agarwal, V. & Shendure, J. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. *Cell Rep.* **31**, 107663 (2020).
- Gasperini, M., Tome, J. M. & Shendure, J. Towards a comprehensive catalogue of validated and target-linked human enhancers. *Nat. Rev. Genet.* **21**, 292–310 (2020).
- Vaswani, A. et al. Attention is all you need. in *Advances in Neural Information Processing Systems* 5998–6008 (2017).
- Brown, T. B. et al. Language models are few-shot learners. in *Advances in Neural Information Processing Systems* (2020).
- Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* <https://doi.org/10.1093/bioinformatics/btba083> (2021).
- Gasperini, M. et al. A genome-wide framework for mapping gene regulation via cellular genetic screens. *Cell* **176**, 377–390 (2019).
- FANTOM Consortium and the RIKEN PMI and CLST (DGT). A promoter-level mammalian expression atlas. *Nature* **507**, 462–470 (2014).
- Heinz, S., Romanoski, C. E., Benner, C. & Glass, C. K. The selection and function of cell type-specific enhancers. *Nat. Rev. Mol. Cell Biol.* **16**, 144–154 (2015).
- Shrikumar, A., Greenside, P. & Kundaje, A. Learning important features through propagating activation differences. in *International Conference on Machine Learning* 3145–3153 (PMLR, 2017).
- Fulco, C. P. et al. Activity-by-contact model of enhancer-promoter regulation from thousands of CRISPR perturbations. *Nat. Genet.* **51**, 1664–1669 (2019).
- Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* **20**, 389–403 (2019).
- Avsec, Ž. et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* <https://doi.org/10.1038/s41588-021-00782-6> (2021).
- ENCODE Project Consortium et al. Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature* **583**, 699–710 (2020).
- Ghandi, M. et al. gkmSVM: an R package for gapped-kmer SVM. *Bioinformatics* **32**, 2205–2207 (2016).
- Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–934 (2015).
- Kelley, D. R., Snoek, J. & Rinn, J. L. Bassett: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
- Consortium, T. G., The GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. *Science* **369**, 1318–1330 (2020).
- Reshef, Y. A. et al. Detecting genome-wide directional effects of transcription factor binding on polygenic disease risk. *Nat. Genet.* **50**, 1483–1493 (2018).
- Wang, Q. S. et al. Leveraging supervised learning for functionally informed fine-mapping of *cis*-eQTLs identifies an additional 20,913 putative causal eQTLs. *Nat. Commun.* **12**, 3394 (2021).
- Wang, G., Sarkar, A., Carbonetto, P. & Stephens, M. A simple new approach to variable selection in regression, with application to genetic fine mapping. *J. R. Stat. Soc.* **82**, 1273–1300 (2020).
- Weissbrod, O. et al. Functionally informed fine-mapping and polygenic localization of complex trait heritability. *Nat. Genet.* **52**, 1355–1363 (2020).
- Kircher, M., Xiong, C., Martin, B. & Schubach, M. Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nature* **10**, 3583 (2019).
- Shigaki, D. et al. Integration of multiple epigenomic marks improves prediction of variant impact in saturation mutagenesis reporter assay. *Hum. Mutat.* **40**, 1280–1291 (2019).
- Lee, D. et al. A method to predict the impact of regulatory variants from DNA sequence. *Nat. Genet.* **47**, 955–961 (2015).
- Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J. & Kircher, M. CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Res.* **47**, D886–D894 (2019).
- Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
- Villar, D. et al. Enhancer evolution across 20 mammalian species. *Cell* **160**, 554 (2015).
- Linder, J. et al. A generative neural network for maximizing fitness and diversity of synthetic DNA and protein sequences. *Cell Syst.* **11**, 49–62 (2020).
- Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3D genome folding from DNA sequence with Akita. *Nat. Methods* **17**, 1111–1117 (2020).
- Schweissinger, R. et al. DeepC: predicting 3D genome folding using megabase-scale transfer learning. *Nat. Methods* **17**, 1118–1124 (2020).
- Schreiber, J., Durham, T., Bilmes, J. & Noble, W. S. Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome. *Genome Biol.* **21**, 81 (2020).
- Nair, S., Kim, D. S., Perricone, J. & Kundaje, A. Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. *Bioinformatics* **35**, i108–i116 (2019).
- Tay, Y., Dehghani, M., Bahri, D. & Metzler, D. Efficient transformers: a survey. Preprint at <https://arxiv.org/abs/2009.06732> (2020).
- Richter, F. et al. Genomic analyses implicate noncoding de novo variants in congenital heart disease. *Nat. Genet.* **52**, 769–777 (2020).
- Zhou, J. et al. Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. *Nat. Genet.* **51**, 973–980 (2019).
- Gupta, S., Stamatoyannopoulos, J. A., Bailey, T. L. & Noble, W. Quantifying similarity between motifs. *Genome Biology* **8**, R24 (2007).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021

## Methods

**Model architecture.** The Enformer architecture consists of three parts: (1) 7 convolutional blocks with pooling, (2) 11 transformer blocks, and (3) a cropping layer followed by final pointwise convolutions branching into 2 organism-specific network heads (Extended Data Fig. 1). Enformer takes as input one-hot-encoded DNA sequence ( $A = [1,0,0,0]$ ,  $C = [0,1,0,0]$ ,  $G = [0,0,1,0]$ ,  $T = [0,0,0,1]$ ,  $N = [0,0,0,0]$ ) of length 196,608 bp and predicts 5,313 genomic tracks for the human genome and 1,643 tracks for the mouse genome, each of length 896 corresponding to 114,688 bp aggregated into 128-bp bins. The convolutional blocks with pooling first reduce the spatial dimension from 196,608 bp to 1,536 so that each sequence position vector represents 128 bp (although the convolutions do observe nucleotides in the adjacent pooled regions). The transformer blocks then capture long-range interactions across the sequence. The cropping layer trims 320 positions on each side to avoid computing the loss on the far ends because these regions are disadvantaged because they can observe regulatory elements only on one side (toward the sequence center) and not the other (the region beyond the sequence boundaries). Finally, the two output heads predict organism-specific tracks. The Enformer architecture is similar to the state-of-the-art model Basenji2 (ref. <sup>3</sup>). However, the following changes helped us improve and exceed its performance: Enformer uses transformer blocks instead of dilated convolutions, attention pooling instead of max pooling, twice as many channels, and 1.5 times longer input sequence (197 kb instead of 131 kb). The detailed model architecture, including the selected hyperparameters, is shown in Extended Data Fig. 1.

Attention pooling summarizes a contiguous chunk of the input sequence  $\mathbf{x}_{k:k+L_p}^{\text{full}} = \mathbf{x} \in R^{L_p \times C}$  across  $L_p$  positions for each of the  $C$  channels and returns the output value  $\mathbf{h} \in R^C$  as follows:

$$h_j = \frac{\sum_i \exp(\mathbf{x}_i \cdot \mathbf{w}_j) x_{ij}}{\sum_i \exp(\mathbf{x}_i \cdot \mathbf{w}_j)},$$

where  $i$  indexes sequence position in the pooling window, which is weighted by the exponentiated dot product  $\mathbf{x}_i \cdot \mathbf{w}_j$ , and  $\mathbf{w} \in R^{C \times K}$  is a matrix of learned weights. We apply attention pooling to contiguous chunks of the original input sequence using window size  $L_p = 2$  and stride of 2. We initialize  $\mathbf{w}$  to  $2 \times 1$ , where  $\mathbf{1}$  is the identity matrix to prioritize the larger value, making the operation similar to max pooling. This initialization gave slightly better performance than did random initialization or initialization with zeros, representing average pooling.

We use multi-head attention (MHA) layers to share information across the sequence and model long-range interactions, such as those between promoters and enhancers. Each head has a separate set of weights  $\mathbf{w}^q \in R^{C \times K}$ ,  $\mathbf{w}^k \in R^{C \times K}$ , and  $\mathbf{w}^v \in R^{C \times V}$  which transform the input sequence  $\mathbf{x} \in R^{C \times L}$  into queries  $\mathbf{q} = \mathbf{x} \cdot \mathbf{w}^q$ , keys  $\mathbf{k} = \mathbf{x} \cdot \mathbf{w}^k$ , and values  $\mathbf{v} = \mathbf{x} \cdot \mathbf{w}^v$ . Queries represent the current information at each position and dots product represent the information each position will be looking for to attend to. Their dot product plus the relative positional encodings  $\mathbf{R}_{ij}$  forms the attention matrix, which is computed as  $\mathbf{a}_{ij} = \text{softmax}(\mathbf{q}_i \mathbf{k}_j^T / \sqrt{K} + \mathbf{R}_{ij})$ , where the entry  $a_{ij}$  represents the amount of weight query at position  $i$  puts on the key at position  $j$ . Values represent the information that each position will propagate forward to positions that attend to it. Each single attention head computes its output as a weighted sum across all input positions:  $\mathbf{av}$ . This allows each query position to use information across the whole sequence. The multiple heads compute with independent parameters, and we concatenate the outputs from each head to form the final layer output followed by a linear layer to combine them. Our layers used 8 heads, value size of 192, and key/query size of 64.

MHA applications in NLP typically operate directly on the input sequence, tokenized into words and embedded in a richer embedding space. The convolution tower preceding MHA in the Enformer model serves to perform an analogous operation of embedding nucleotide segments and contributes a compelling inductive bias for adjacent nucleotides to function together in motifs. We chose to compute at 128-bp resolution because it roughly represents a well-studied length of regulatory elements that contain several motifs and is an appropriate bin size at which to aggregate the experimental data to be predicted. Finer resolution has potential benefits when the data support it<sup>15</sup>, but would extend the sequence length entering the quadratic complexity MHA and make the model engineering intractable on currently available hardware.

To inject positional information, we add relative positional encodings<sup>40</sup>  $\mathbf{R}_{ij}$  to the  $\mathbf{q}_i \mathbf{k}_j^T$  attention term as formulated in the Transformer-XL paper<sup>41</sup>. Relative positional encodings provide a parameterized baseline for how actively two positions in the sequence should influence each other during the layer's transformation as a function of their pairwise distance. Specifically, we use  $\mathbf{R}_{ij} = \mathbf{q}_i^T \mathbf{r}_{i-j} + \mathbf{u} \mathbf{k}_j^T + \mathbf{v} \mathbf{r}_{i-j}^T$ , where  $\mathbf{r}_{i-j} = \mathbf{w}^r \mathbf{f}(i-j)$  is a linear function of different relative basis functions  $\mathbf{f}(i-j)$ , and  $\mathbf{u}$  and  $\mathbf{v}$  are the position-agnostic embeddings used to evaluate the preference for specific keys ( $\mathbf{u}$ ) or relative distances ( $\mathbf{v}$ ). We use three different basis function classes for  $\mathbf{f}(i-j)$ , as visualized in Extended Data Fig. 5b:

1.  $f_i^{\text{exponential}}(r) = e^{-\log(2) \frac{r}{r_{1/2,i}}}$ , where  $r_{1/2,i}$  is placed linearly in the log-space between 3 and sequence length.
2.  $f_i^{\text{central mask}}(r) = \begin{cases} 1, & \text{if } r \leq 2^i \\ 0, & \text{otherwise} \end{cases}$

3.  $f_i^{\text{gamma}}(r) = \text{Gamma}\left(r | \alpha = \frac{\mu_i}{\sigma^2}, \beta = \frac{\mu_i^2}{\sigma^2}\right)$ , where  $\text{Gamma}(r | \alpha, \beta)$  is the gamma probability distribution function.  $\mu_i$  is placed linearly from (sequence length / number of features) to sequence length and  $\sigma = \text{sequence length} / (2 \times \text{number of features})$ .

For each basis function, we use a symmetric  $f(|x|)$  and asymmetric  $\text{sign}(x) \times f(|x|)$  version to introduce directionality. We use the same number of relative positional basis functions as the value size of MHA (192). The 192 basis functions are equally divided among the basis function classes and the symmetric versus asymmetric versions thereof. With 3 basis function classes, each basis function class provides 64 positional features (32 symmetric and 32 asymmetric).

Dropout rates of 0.01 and 0.05 were used for positional encoding features and the final attention matrix respectively in MHA. All other dropout rates are annotated in Extended Data Fig. 1a.

**Model training and evaluation.** The model was trained, evaluated, and tested on the same targets, genomic intervals, and using the same Poisson negative log-likelihood loss function as Basenji2 (ref. <sup>3</sup>). Briefly, the cross-species training/validation/test sets were constructed using the following procedure to partition homologous sequences into the same set. First, we divided both the human and mouse genomes into 1 Mb regions. We constructed a bipartite graph, in which the vertices represent these regions. Next, we placed edges between 2 regions if they have >100 kb of aligning sequence in the hg38-mm10 syntenic net format alignment downloaded from the UCSC Genome Browser<sup>42</sup>. Finally, we partitioned connected components in the bipartite graph randomly into training, validation, and test sets.

The dataset contains 34,021 training, 2,213 validation, and 1,937 test sequences for the human genome, and 29,295 training, 2,209 validation, and 2,017 test sequences for the mouse genome. For the human genome, each example contains 2,131 transcription factor (TF) chromatin immunoprecipitation and sequencing (ChIP-seq), 1,860 histone modification ChIP-seq, 684 DNase-seq or ATAC-seq, and 638 CAGE tracks (total 5,313, Supplementary Table 2). For the mouse genome, each example contains 308 TF ChIP-seq, 750 histone modification ChIP-seq, 228 DNase-seq or ATAC-seq, and 357 CAGE tracks (total 1,643, Supplementary Table 3). We modified the Basenji2 dataset by extending the input sequence to 196,608 bp from the original 131,072 bp using the hg38 reference genome.

To train a model simultaneously on human and mouse genomes, we alternated between a batch containing data from the human genome and the mouse genome. The main Enformer model with 1,536 channels was implemented in Sonnet v2, TensorFlow (v2.4.0), and was trained on 64 TPU v3 cores with batch size of 64 (1 per core) for 150,000 steps (approximately 3 days) using all-reduce gradient aggregation across the cores at every step. Batch normalization statistics were also aggregated across multiple replicas using 0.9 momentum. We used the Adam optimizer from Sonnet v2 (ref. <sup>43</sup>) with a learning rate of 0.00005 and default settings for other hyperparameters:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1 \times 10^{-8}$ . The optimal learning rate was discovered by grid search yielding the highest performance on the validation set. We linearly increased the learning rate from 0 to target value in the first 5,000 steps of training. We clipped gradients to a maximum global norm of 0.2. We used the same data augmentation as Basenji2 (ref. <sup>3</sup>) during training by randomly shifting the input sequence by up to 3 bp and reverse-complementing the input sequence while reversing the targets. Finally, we fine-tuned the Enformer model on human data for 30,000 steps using a lower learning rate of 0.0001.

We used the **pretrained Basenji2 model** for all main model comparisons and retrained an equivalent model for ablation and hyperparameter sweeps shown in Extended Data Fig. 5. In these comparative analyses, we used 768 channels (1/2 of the original Enformer model obtained by using a value size of 96 in MHA), 131 kb input sequence, and batch size 32 trained on 32 TPU v3 cores. We did not fine-tune these models on the human data. For models using dilated convolutions instead of transformer blocks, we used a higher learning rate of 0.02 without ramp up of the learning rate. As for Enformer, the optimal learning rate was discovered by grid search yielding the highest performance on the validation set. All models were trained for 500,000 steps while only storing the model with the highest Spearman correlation of CAGE TSS gene expression across genes averaged across experiments computed on the validation set every 1,000 steps.

We used the validation set for hyperparameter selection and the test set for Basenji2 comparison. We considered two evaluation metrics: (1) Pearson correlation computed across all 128-bp binned genomic positions in the validation/test set for each output track; and (2) Pearson correlation of CAGE gene expression values ( $\log(1 + x)$ -transformed and standardized across genes for each experiment) of all protein-coding genes in the validation/test set computed either for each CAGE experiment across genes (main metric) or across CAGE experiments for each gene (shown in Fig. 1b). Observed and predicted gene expression values were obtained by summing up the observed/predicted CAGE read counts at all unique TSS locations of the gene. For each TSS location, we used the 128-bp bin overlapping the TSS as well as the two neighboring bins (3 bins in total). We used test-time augmentation during model evaluation: we averaged the predictions from 8 sequences randomly augmented the same way as during training ( $\leq 3$  bp shifts and reverse-complementation). We only evaluated the performance of our

model on the test set once to generate Fig. 1 and did not use the test set during model development.

To select a representative example, we visualized the top 10 transcripts with highest discrepancy between Enformer and Basenji2 performance on the ‘Across CAGE experiments’ metric measuring tissue specificity for 33% of the most tissue-specific genes. We picked the sixth transcript in the list (ENST00000524922) because it cleanly showed differences across all three categories of genomic tracks (DNA accessibility, histone modifications, and gene expression).

**Enhancer prioritization.** We obtained a set of enhancer–gene pairs tested using a CRISPRi assay perturbing the enhancer of interest while measuring the expression change of the gene in K562 cells from two studies: Gasperini et al.<sup>9</sup> using scRNA-seq to measure expression changes, and Fulco et al.<sup>13</sup> using Flow-FISH. We transformed the enhancer and gene coordinates from hg19 to hg38 using the UCSC liftOver web tool<sup>42</sup>. Each enhancer–gene pair contains a label denoting whether a significant expression change was induced after CRISPRi treatment. We denoted the set of all enhancers as ‘candidate’ enhancers and those that showed a change in expression as ‘validated’ enhancers. We evaluated different methods on their ability to classify or prioritize enhancer–gene pairs that exhibited a significant expression change using area under precision–recall curve (auPRC)<sup>13</sup>.

To prioritize enhancer–gene pairs with sequence-based models, we computed three different scores: gradient  $\times$  input, attention, and in silico mutagenesis (ISM). For each enhancer–gene pair, we determined the major TSS of the gene by taking the highest predicted CAGE value in K562 using Enformer. We extracted the DNA sequence centered at the main TSS and computed the following different enhancer–gene scores:

1. Gradient  $\times$  input: We computed the absolute value of the gradient of the CAGE targets (either using the K562-specific CAGE targets or all CAGE targets, Extended Data Fig. 7c) at the TSS with regard to the input reference sequence nucleotide. Note that since our input sequence is one-hot encoded, taking the input gradient of the nonzero channel (the reference nucleotide), is equivalent to computing gradient  $\times$  input attributions<sup>12</sup>. We note that ‘CAGE at TSS’ always means summing the absolute gradient values from three adjacent bins, as is also done in gene-focused model evaluation. The three bins include the bin overlapping the TSS and one flanking bin on each side. The enhancer–gene score was obtained by summing the absolute gradient  $\times$  input scores in the 2-kb window centered at the enhancer.
2. Attention: We first averaged transformer attention matrices across all heads and layers. We extracted the row corresponding to the query index positioned at the TSS, so that keys correspond to different spatial positions and the attention values specify how much the model attended to these positions when making predictions for the TSS. We only computed this contribution score for Enformer. The enhancer–gene score was obtained by summing the attention scores in the 2-kb window centered at the enhancer.
3. ISM: The in silico mutagenesis enhancer–gene score was computed by comparing K562 CAGE predictions at the TSS from the reference sequence with predictions from modified sequence where the 2-kb enhancer sequence was replaced by a random sequence:  $|f(\text{modified}) - f(\text{reference})|$ .

To reproduce the ABC score introduced in Fulco et al.<sup>13</sup>, we obtained the BigWig of H3K27ac ChIP-seq data in K562 from ENCODE with file accession ENCF779QTH and DNase with file accessions ENCF413AHU and ENCF936BDN. We summed the normalized reads from replicates. For each track and enhancer, we summed up the signal at the enhancer in a fixed window of 2 kb centered at the enhancer. This fixed and broader window yielded better performance compared to the variable window size of ~500 bp as used in the original ABC score (Extended Data Fig. 4a).

**GTEX SLDP.** We predicted the effect of a genetic variant on various annotations by computing a forward pass through the model using the reference and alternative alleles, subtracting their difference, and summing outputs across the sequence to obtain a signed score for each training dataset. We averaged scores computed using the forward and reverse complement sequence and small sequence shifts to the left and right. We computed scores for all 1000 Genomes SNPs.

We used SLDP<sup>30</sup> to estimate the functional correlation between these scores and GTEX v7a summary statistics for 48 tissues while accounting for population linkage disequilibrium structure (Supplementary Information).

**Fine-mapped GTEX classification.** To study specific eQTLs without needing to consider LD, we studied statistical fine-mapping of GTEX v8 using the SuSiE method<sup>20,23</sup>. We focused on variants with posterior inclusion probability (PIP) in a credible causal set  $>0.9$ , which ranged from a minimum of 166 variants for substantia nigra to 2,740 for tibial nerve. We arranged a classification task to discriminate between these positive causal variants and a matched set of negative variants. When available, we chose a negative variant matched to each causal variant from the set with PIP  $< 0.01$  but  $|Z\text{-score}| > 4$  tested for the same gene. When unavailable for the same gene, we chose from the set with PIP  $< 0.01$  and  $|Z\text{-score}| > 6$  genome-wide.

To determine how informative different variant annotations are, we trained separate random forest classifiers for each tissue to distinguish causal from noncausal variants using eight-fold crossvalidation. We selected the default hyperparameters of the scikit-learn 0.22 implementation after finding negligible accuracy gains from modifying them<sup>44</sup>. However, owing to the large number of features derived from the training datasets, setting the maximum features considered per decision tree split to  $\log_2$  of the total number of features greatly improved the computational efficiency. We fit 100 iterations of stochastic crossvalidation shuffling and random forest fitting to delineate a low-variance estimate of model accuracy. We performed statistical tests comparing two different model feature sets by comparing the  $8 \times 100$  distinct test set auROCs.

For signed GTEX analysis, we benchmarked model predictions on the basis of their ability to discriminate causal variants that increase versus decrease gene expression. In this analysis, we removed variants that affect gene expression in opposite directions for different cis-genes. We manually matched FANTOM5 CAGE sample descriptions to the GTEX tissues. We skipped cases with more than three possible matches. In cases with two or three possible matches, we chose the CAGE sample with the best average concordance between the Basenji2 and Enformer predictions. We computed auROC statistics by ranking causal variants by their signed prediction for that sample.

**Benchmarking variant effect predictions on saturation mutagenesis data.** We acquired training and test sets as well as the predictive accuracies of individual competition participants from the CAGI5 competition<sup>26</sup> (M. Kircher, personal communication, <https://genomeinterpretation.org/content/expression-variants>). For each variant and locus, we evaluated its effect as the predicted difference between the reference and alternative allele summed in four flanking bins representing 512 bp, producing 5,313 features based on the human datasets. All CAGE features were log-transformed after adding a pseudocount of 1 prior to computing this difference. For each allele, we averaged predictions for the forward and reverse-complemented sequence. We scaled the features from the test set with scaling factors computed on the features from the training set, such that the training features had a mean of 0 and s.d. of 1. Following our previous work<sup>45</sup>, we then trained a lasso regression model for each locus using these features and the corresponding training set. The strength of the regularization was controlled by a single  $\lambda$  parameter, which was optimized using tenfold crossvalidation for each training set using the cv.glmnet function of the glmnet library in R.

For our training-free comparisons, we selected the subset of features corresponding to cell-type-matched and cell-type-agnostic predictions of changes in CAGE and DNase. For the cell-type-agnostic models, we used the subset of all 638 CAGE or 674 DNase features (Supplementary Table 2). For the cell-type-matched models, we additionally required the CAGE/DNase features to contain the following substrings: (1) ‘HepG2’ for *F9*, *LDLR*, and *SORT1*, (2) ‘K562’ for *GP1BB*, *HBB*, *HBG1*, and *PKLR*, and (3) ‘HEK293’ for *HNF4A*, *MSMB*, *TERT* (performed in HEK293T cells), and MYCr6983267. For several loci, a perfectly matched DNase or CAGE sample did not exist. We therefore selected the most closely matched feature based on the following substrings: (1) ‘pancreas’ for *ZFAND3*, (2) ‘glioblastoma’ for *TERT* (performed in GBM cells), (3) ‘keratinocyte’ for *IRF6*, and (4) ‘SK-MEL’ for *IRF4*. For each locus, we extracted the features matching the aforementioned substrings, and used the first principal component (PC) of the indicated features as our summary statistic, inverting the sign of the PC if it was negatively correlated to the mean of the features.

**Reporting Summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

## Data availability

Gene annotation was obtained from <https://www.gencodegenes.org/> (v32). Basenji2 training, validation, and test data was obtained from [https://console.cloud.google.com/storage/browser/basenji\\_barnyard/data](https://console.cloud.google.com/storage/browser/basenji_barnyard/data). Processed CRISPRi data for Fulco et al 2019<sup>13</sup> was obtained from supplementary material and for Gasperini et al 2019<sup>9</sup> from GEO accession GSE120861. H3K27ac ChIP-seq data in K562 used for analysis in Fig. 2 was obtained from <https://www.encodeproject.org/> with file accession ENCF779QTH and DNase with file accessions ENCF413AHU and ENCF936BDN. TAD boundaries processed by Fudenberg et al 2020<sup>32</sup> were obtained from [https://console.cloud.google.com/storage/browser/basenji\\_hic/insulation](https://console.cloud.google.com/storage/browser/basenji_hic/insulation). Fine-mapped eQTLs are available from the supplementary material of Wang et al 2021<sup>33</sup> and the negative set from [https://console.cloud.google.com/storage/browser/dm-enformer/data/gtex\\_fine](https://console.cloud.google.com/storage/browser/dm-enformer/data/gtex_fine). We acquired training and test sets as well as the predictive accuracies of individual competition participants from the CAGI5 competition<sup>26</sup> (M. Kircher, personal communication, <https://genomeinterpretation.org/content/expression-variants>). For comparison to ExPecto, we used the provided data from <https://github.com/FunctionLab/ExPecto/tree/master/resources>.

## Code availability

All components of our core algorithm, including the full model architecture and example code to train and evaluate the model are available under the open

source Apache 2.0 license at the following URL: <https://github.com/deepmind/deepmind-research/tree/master/enformer>. The code is also archived at Zenodo <https://doi.org/10.5281/zenodo.5098375><sup>46</sup>. In addition, layer components of the model are now available in the existing Basenji repository for biological sequence deep learning at <https://github.com/calico/basenji> also under the open source Apache 2.0 license.

Pre-trained Enformer model is available on TF-Hub so that users can easily run it on new data: <https://tfhub.dev/deepmind/enformer/1>. We also plan to release it in the Kipoi model repository<sup>47</sup>. We provide code examples ([enformer-usage.ipynb](#)) on how to use that model to score genetic variants. Finally, we provide variant effect predictions for all frequent variants in the 1000 genomes cohort (MAF>0.5% in any population) [here](#), with an open creative-commons CC-BY 4.0 license. To make these predictions more accessible, we distilled the 5,313 features into 20 highly informative variant scores using PCA (Methods) to keep the released file sizes manageable (<1 GB in total for 10 M variants, instead of 100 GB) while retaining high predictive accuracy (GTEx fine-mapping classification auROC of 0.743 compared to 0.747 using all features).

## References

40. Shaw, P., Uszkoreit, J. & Vaswani, A. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* 464–468 (2018).
41. Dai, Z. *et al.* Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* 2978–2988 (2019).
42. Kent, W. J. The Human Genome Browser at UCSC. *Genome Research* **12**, 996–1006 (2002).
43. Reynolds, M. *et al.* Open sourcing Sonnet — a new library for constructing neural networks. <https://deepmind.com/blog/open-sourcing-sonnet> (2017).
44. Pedregosa, F. *et al.* Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
45. Klein, J. C. *et al.* A systematic evaluation of the design and context dependencies of massively parallel reporter assays. *Nat. Methods* **17**, 1083–1091 (2020).
46. Avsec, Žiga *et al.* Enformer (Version 3.0) (Zenodo, 2021); <https://doi.org/10.5281/zenodo.5098375>
47. Avsec, Ž. *et al.* The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nat. Biotechnol.* **37**, 592–600 (2019).

## Acknowledgements

We thank M. Kircher (Berlin Institute of Health) for sharing saturation mutagenesis MPRA datasets and variant effect predictions with us as well as A. Pritzel, A. W. R. Nelson, A. Patterson, A. Obika, C. Meyer, D. Hassabis, M. Dunlop, N. Latysheva, N. Alic, S.-J. Dunn, S. Petersen, T. Niccoli, T. Sargeant, and T. Back for their contributions and support.

## Author contributions

Ž.A., J.R.L., D.R.K., and P.K. initiated the project. Ž.A., V.A., and D.R.K. conceived of the study and designed the analyses. Ž.A. designed the model with help from D.V., J.J., and P.K.. Ž.A., D.V., K.R.T., and Y.A. implemented the model. Ž.A. performed model performance analyses, A.G.-B. and Ž.A. performed enhancer prioritization analysis, D.R.K. performed variant effect analysis on population genetic data, and V.A. performed variant effect analyses for MPRA data. J.R.L., J.J., P.K., and D.R.K. supervised the study. Ž.A., V.A., and D.R.K. prepared the manuscript with input from all authors. The authors received no specific external funding for this work.

## Competing interests

Ž.A., A.G.-B., K.R.T., Y.A., J.J., and P.K. are employed by DeepMind. V.A., and D.R.K. are employed by Calico Life Sciences. J.R.L. is employed by Google. The remaining authors declare no competing interests.

## Additional information

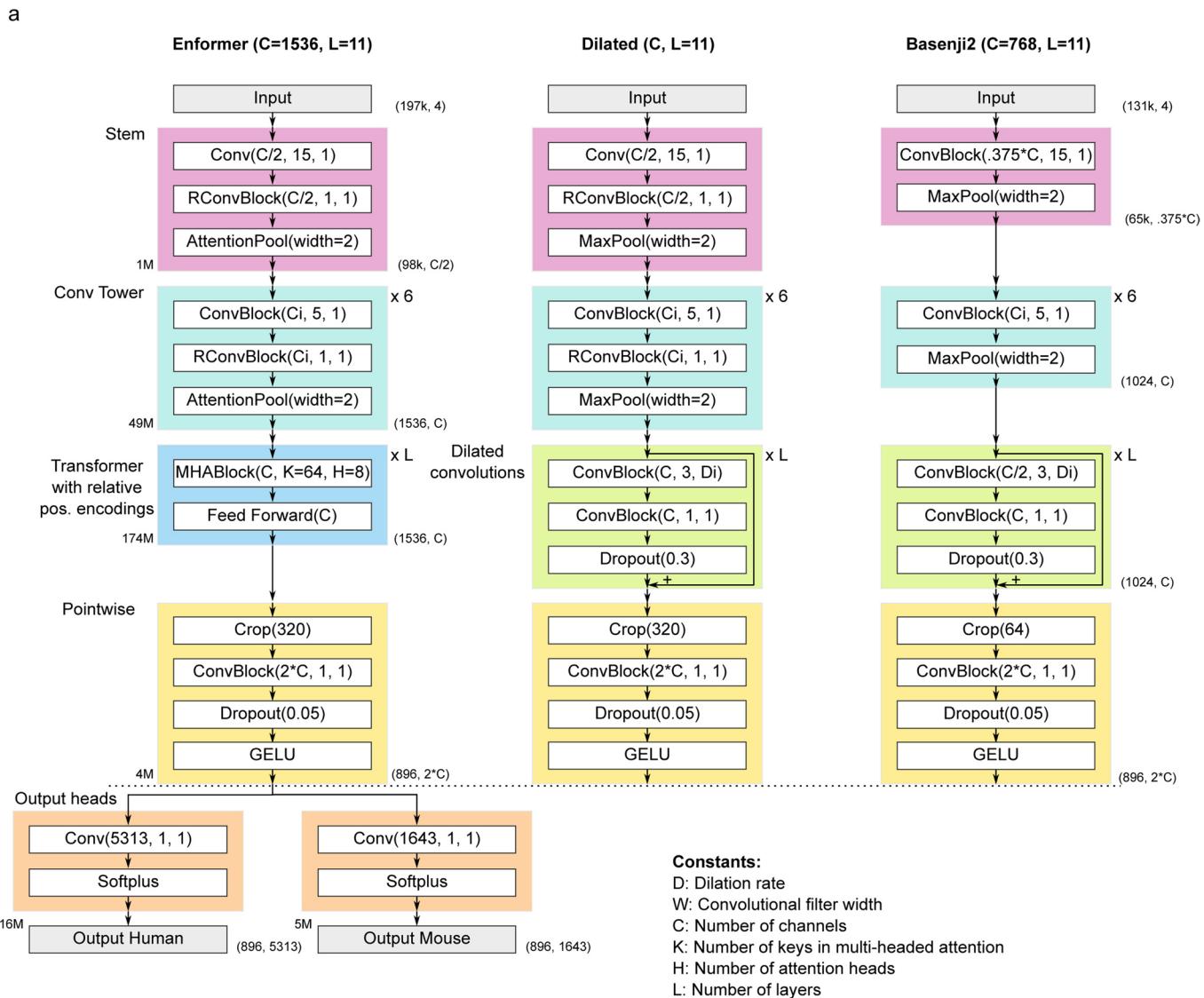
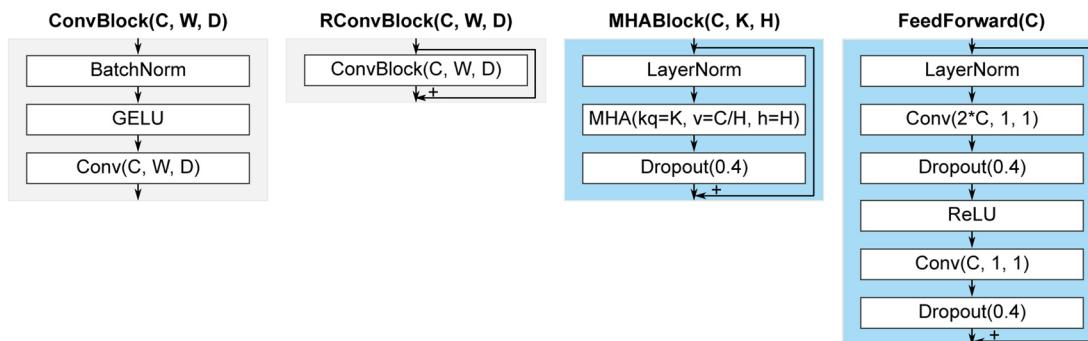
**Extended data** is available for this paper at <https://doi.org/10.1038/s41592-021-01252-x>.

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41592-021-01252-x>.

**Correspondence and requests for materials** should be addressed to Žiga Avsec, Pushmeet Kohli or David R. Kelley.

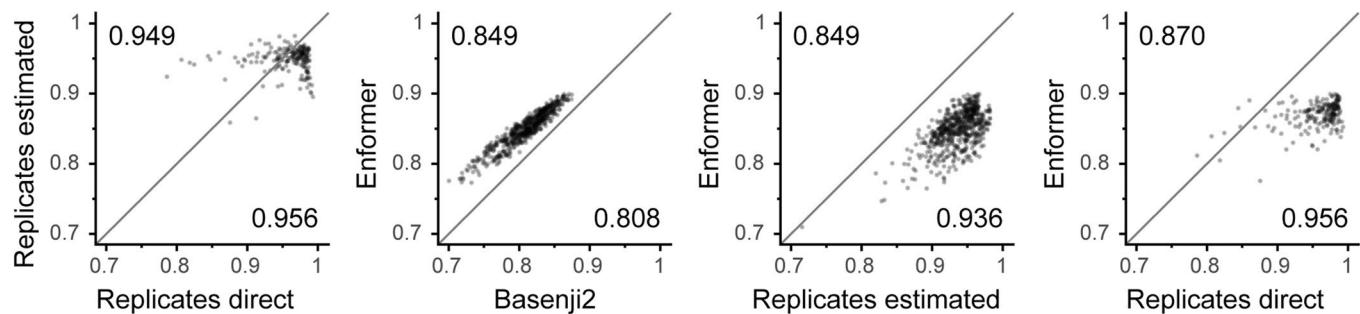
**Peer review information** *Nature Methods* thanks Qiao Liu, Wing Hung Wong and the other, anonymous, reviewers for their contribution to the peer review of this work. Primary Handling editor: Lin Tang, in collaboration with the *Nature Methods* team.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**b**

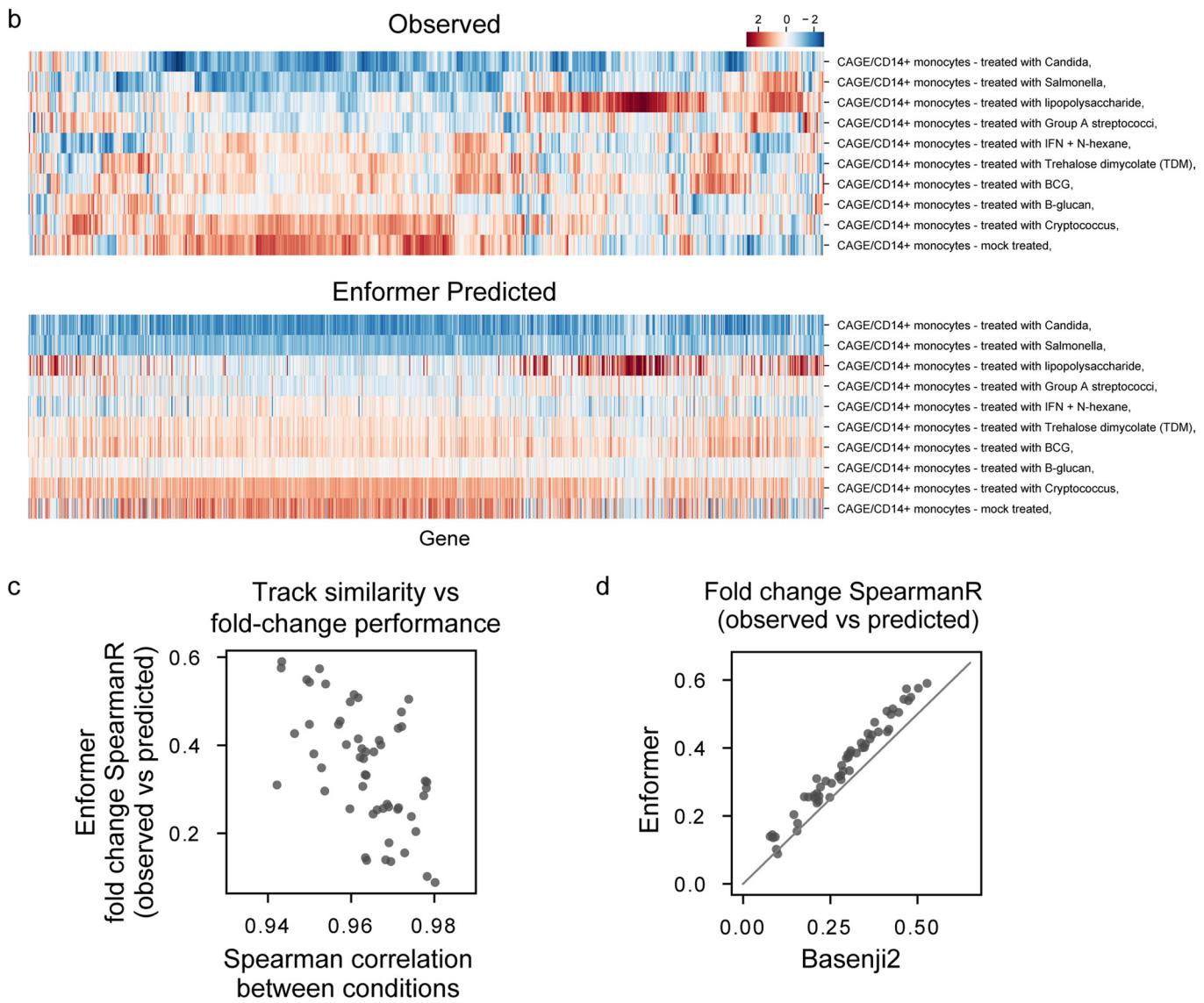
**Extended Data Fig. 1 | Enformer model architecture and comparison to Basenji2.** **a)** From left to right: Enformer model architecture, ‘dilated’ architecture used in ablation studies obtained by replacing the transformer part of the model with dilated convolutions, and Basenji2. Output shapes (without batch dimensions) are shown as tuples on the right side of the blocks. The number of trainable parameters for different parts of Enformer are shown on the left side of the blocks. The two main hyperparameters of the model are the number of transformer/dilated layers, L, and the number of channels, C. All models have the same two output heads as shown on the Enformer at the bottom. The number of channels in the convolutional tower Ci was increased by a constant multiplication factor to reach C channels starting from C/2 (or 0.375\*C for Basenji2) in 6 layers. For dilated layers, we increased the dilation rate Di by a factor of 1.5 at every layer (rounded to the nearest integer). **b)** Definition of different network blocks in terms of basic neural network layers. MHA denotes multi-headed attention using relative positional encodings with kq representing the number of key/query size, v representing the value size and h the number of heads. Number of relative positional basis functions is equal to value size v.

a



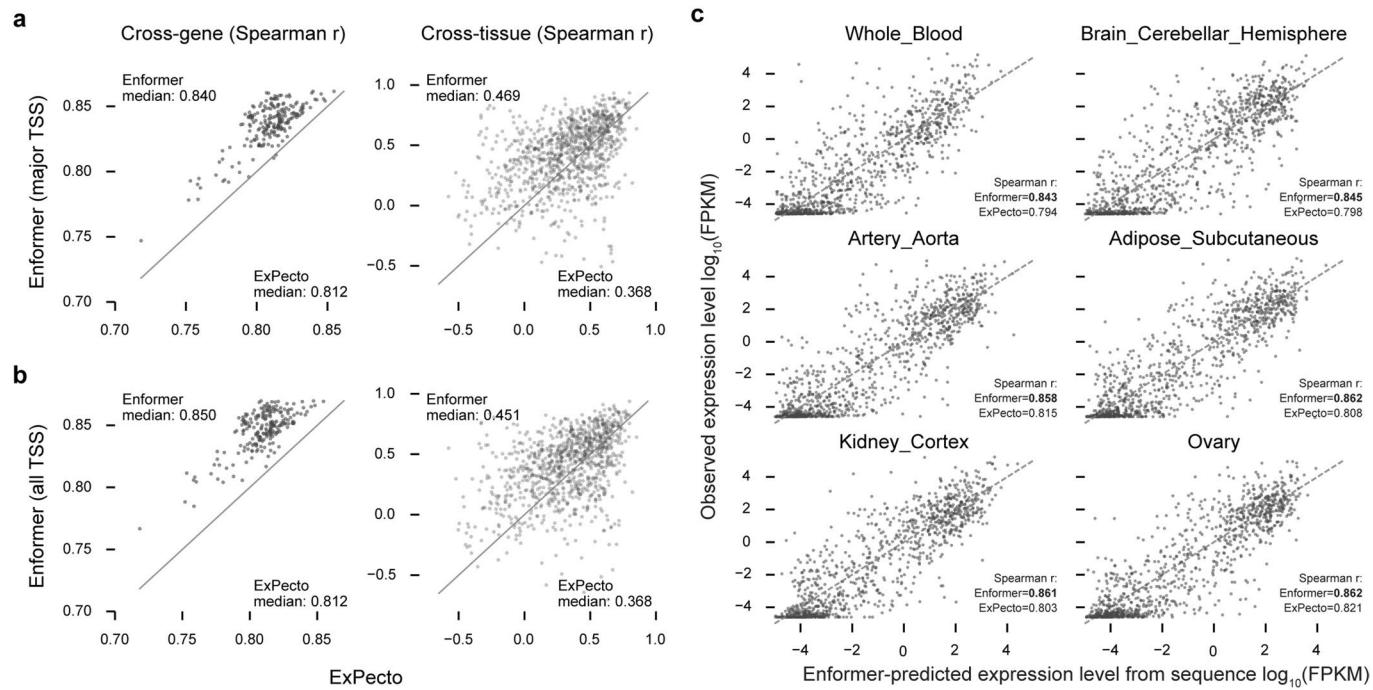
**Extended Data Fig. 2 | Replicate level accuracy. a)** Gene expression correlation ( $\log(1+x)$  `pearsonR`) for each CAGE track across protein-coding genes comparing experimental-level accuracy computed in two ways (estimated and direct) to Enformer. For ‘Replicates direct’, CAGE replicate experiments were partitioned into two groups and compared against each other. For ‘Replicates estimated’, a predictive model was used to impute CAGE values of a particular track from all other tracks.

<b>a</b>	Group 1: CAGE:CD14+ monocytes - treated with BCG, CAGE:CD14+ monocytes - treated with IFN + N-hexane, CAGE:CD14+ monocytes - treated with Trehalose dimycolate (TDM), CAGE:CD14+ monocytes - mock treated, CAGE:CD14+ monocytes - treated with Group A streptococci, CAGE:CD14+ monocytes - treated with lipopolysaccharide, CAGE:CD14+ monocytes - treated with Salmonella, CAGE:CD14+ monocytes - treated with Cryptococcus, CAGE:CD14+ monocytes - treated with Candida, CAGE:CD14+ monocytes - treated with B-glucan,	Group 4: CAGE:embryonic kidney cell line: HEK293/SLAM untreated CAGE:embryonic kidney cell line: HEK293/SLAM infection, 24hr
	Group 2: CAGE:Mast cell, expanded, CAGE:Mast cell, expanded and stimulated,	Group 5: CAGE:cord blood derived cell line:COBL-a untreated CAGE:cord blood derived cell line:COBL-a 24h infection(-C) CAGE:cord blood derived cell line:COBL-a 24h infection
	Group 3: CAGE:Smooth muscle cells - airway, asthmatic, CAGE:Smooth muscle cells - airway, control	Group 6: CAGE:Skeletal Muscle Cells, CAGE:Skeletal muscle cells differentiated into Myotubes - multinucleated,
		Group 7: CAGE:Hep-2 cells treated with Streptococci strain 5448, biol_ CAGE:Hep-2 cells treated with Streptococci strain JRS4, biol_ CAGE:Hep-2 cells mock treated, biol_

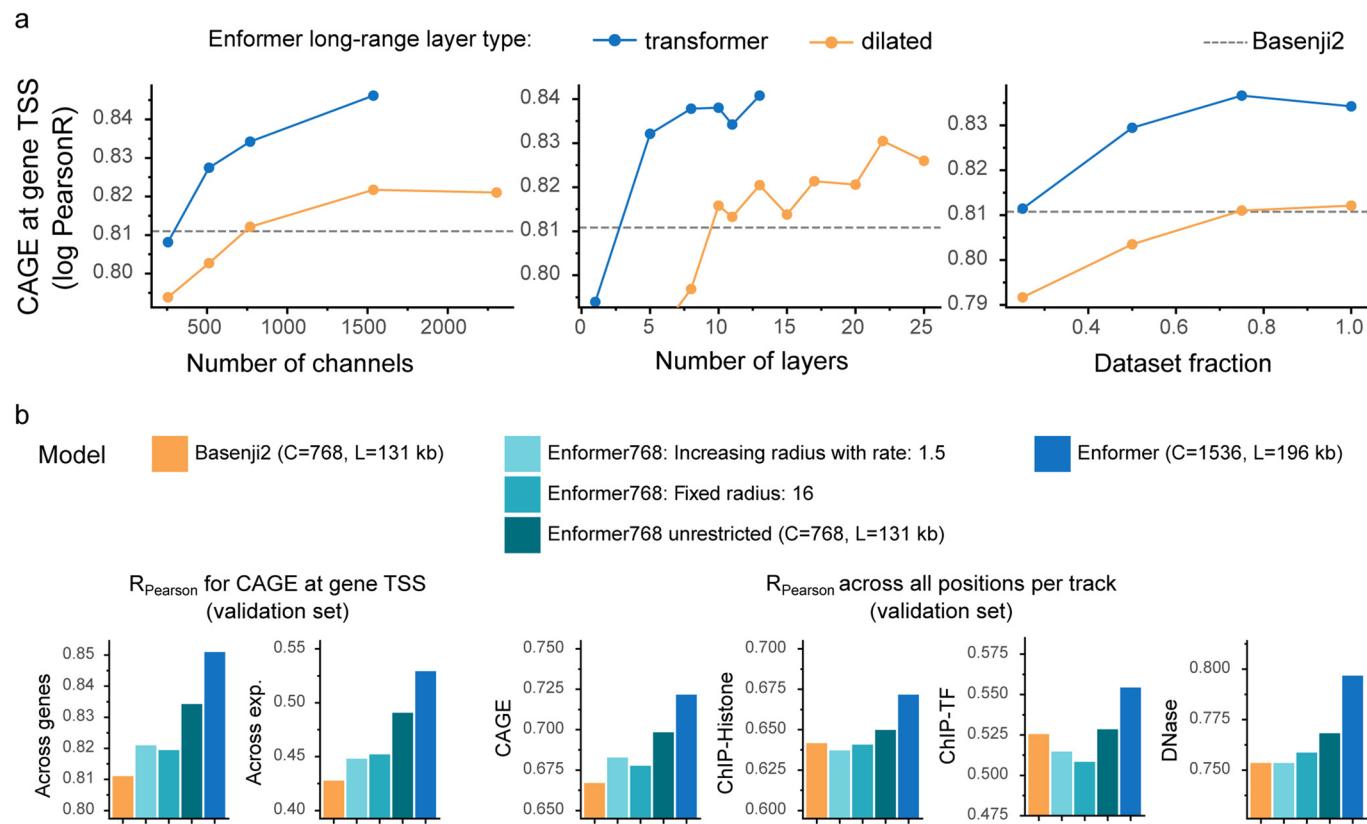


Extended Data Fig. 3 | See next page for caption.

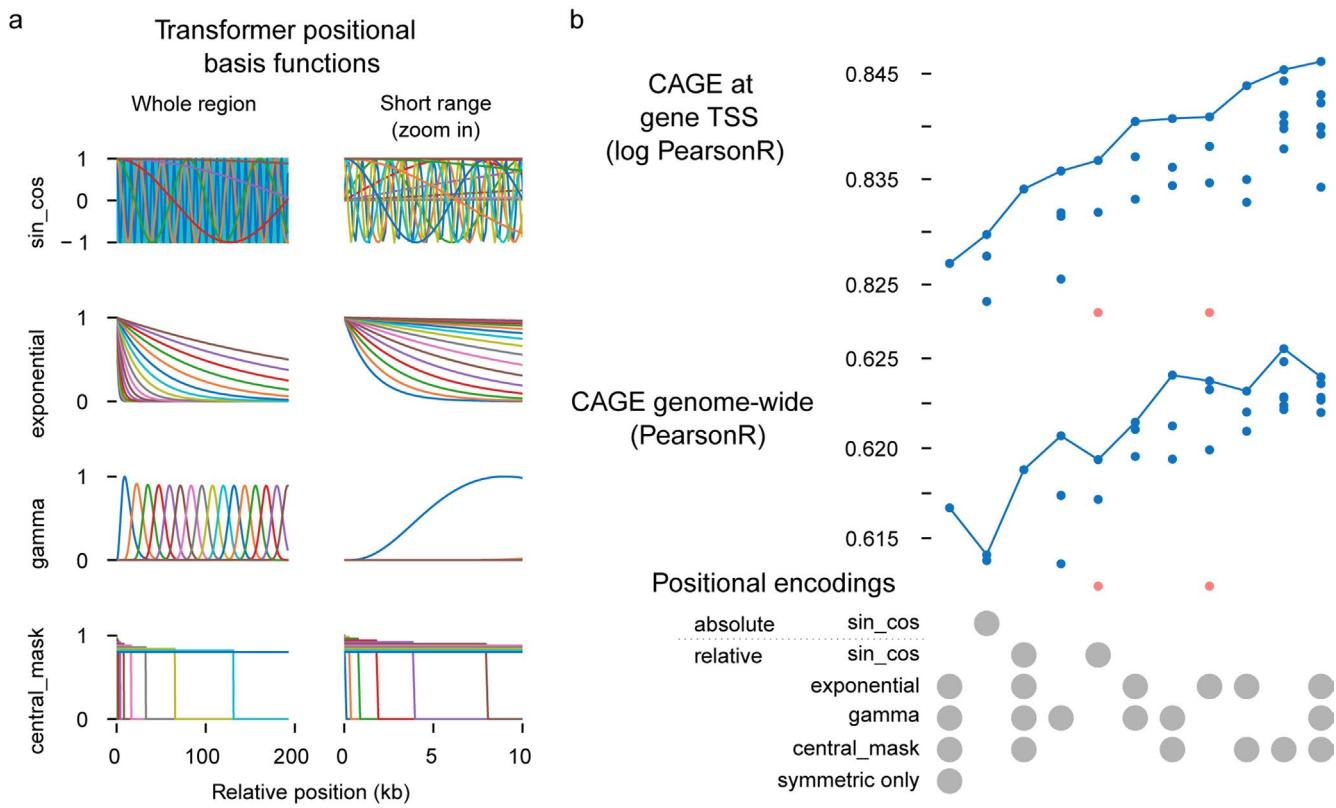
**Extended Data Fig. 3 | Predictive performance for treated samples.** **a)** Groups of CAGE experiments where the biological samples were perturbed in different ways. **b)** Observed and predicted gene expression matrices ( $\log(1+x)$  transformed) for CD14+ monocytes and genes in the held-out test set. The most prominent change in gene expression due to the lipopolysaccharide treatment was also captured by the Enformer model. Observed matrix was hierarchically clustered for both rows and columns. Enformer predicted heatmap follows the same row and column ordering as the observed matrix. **c)** Predictive performance in the test set for CAGE gene expression fold change for all within-group pairs from a (y-axis) compared to the observed gene expression correlation between two pairs (x-axis). Fold change of highly correlated CAGE samples is more difficult to predict. **d)** Enformer shows higher fold-change predictive performance compared to Basenji2.



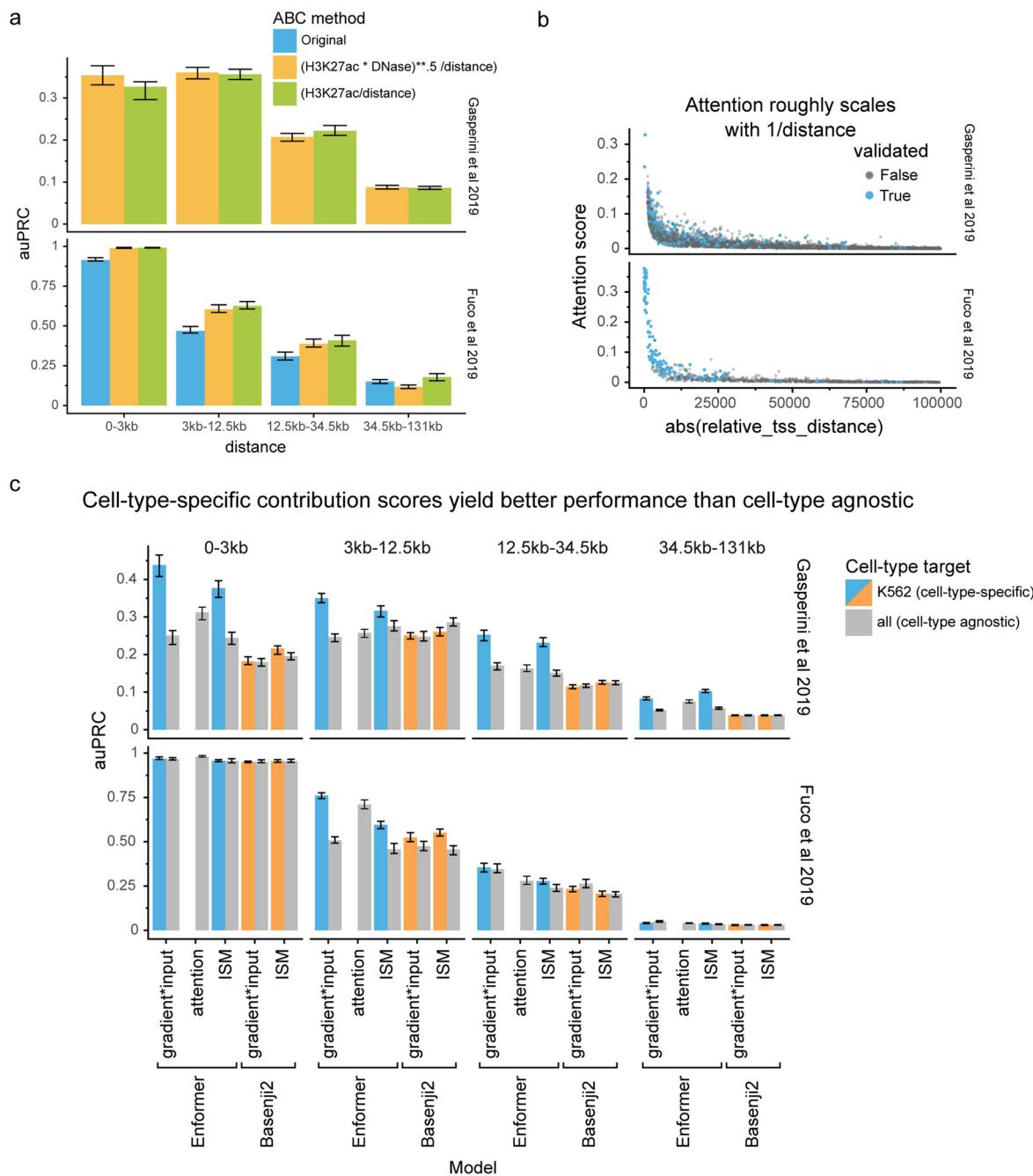
**Extended Data Fig. 4 | Enformer predicts mRNA-seq more accurately than ExPecto.** **a**) Test set predictive performance comparison of a linear model trained on top of Enformer CAGE predictions from the major TSS (y-axis) and ExPecto (x-axis) computed either across genes (first column) or across tissues (second column). Gene expression matrix was normalized across genes to have zero mean and unit variance for each tissue. Enformer was re-trained only on the human genome using the same training chromosomes for this comparison (Methods). **b**) Same as a), but using Enformer predictions averaged across all TSS of the gene. **c**) Observed versus Enformer-predicted gene expression values for all 990 test genes in 6 RNA-seq samples.



**Extended Data Fig. 5 | Comparison to dilated convolutions.** **a)** Enformer with original transformer layers (Extended Data Fig. 1a left) performs better than Enformer with dilated convolutions (Extended Data Fig. 1a center) across different model sizes and training dataset subsets as measured CAGE gene expression correlation in the validation set (same metric as in Fig. 1b across genes). At 15 dilated layers, the model starts to reach outside of the input sequence range (receptive field of 224,263 bp). Note that all the evaluations here are limited by TPU memory preventing you from using more layers or channels. **b)** Performance comparison to Basenji2 (left) and Enformer (right) to Enformer with the same receptive field (44 kb) as Basenji2 by either allowing a fixed attention radius of 16 across all layers where query can attend to at most 16 positions away (Enformer 769: Fixed radius: 16) or by exponentially increasing the respective field in the same way as the dilation rate in Basenji2. Enformer768 was trained with the same number of 768 channels and 131kb input sequences as Basenji2, whereas Enformer uses two times more channels and 1.5 times longer sequence. Same evaluation metrics are shown as in Fig. 1.



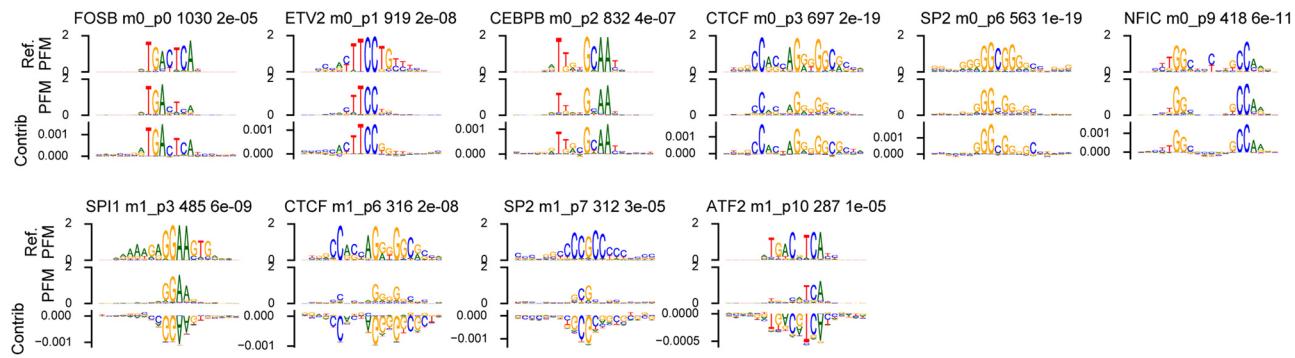
**Extended Data Fig. 6 | Custom relative positional encoding functions are required for good predictive performance.** **a)** Relative positional encoding basis function options for the transformer model. Sine/cosine basis functions are frequently used in the NLP literature for both absolute or relative positional encodings. Enformer uses a concatenation of exponential, gamma and central\_mask relative positional encodings. For each basis function, a symmetric  $f(|x|)$  and asymmetric  $\text{sign}(x) * f(|x|)$  basis function will be used to introduce directionality and thereby inform the model of what is upstream or downstream of the TSS. Each basis function is visualized with a different color. **b)** Validation set performance as measured by CAGE gene expression correlation across protein coding genes (top; same metric as in Fig. 1b across genes) or across all positions (bottom; same metric as displayed in Fig. 1c CAGE) for models trained with different classes of positional encoding functions in the transformer. Custom relative positional encodings show better performance than using standard sin/cos basis functions or using absolute positional encodings, likely because they can better capture the decreasing importance of enhancers with increased distance. Also, symmetric only ( $f(|x|)$ ) version shows much lower performance than using both, symmetric and asymmetric versions. All models use the same 96 total number of basis functions. Each positional encoding configuration was trained with multiple different random seeds. Red points denote runs with lower performance than the y-axis limits.



**Extended Data Fig. 7 | Tissue-specific contribution scores are required for good enhancer-recall performance.** **a**) Enhancer-gene ranking performance comparison of different ABC score versions, including the original score which uses DNase, H3K27ac, and Hi-C data. H3K27ac / distance is a good if not even slightly better proxy for the ABC score. The alternative versions use a fixed and 2 kb wide aggregation window whereas the original uses a dynamic peak width depending on the DNase peak width. **b**) Attention-based contribution score as a function of distance at all enhancer-gene pairs in both studied datasets. **c**) Contribution scores that are cell-type-specific (shown in green, achieved by computing the contribution scores w.r.t. cell-type-specific target variables) outperform cell-type agnostic contribution scores (shown in orange). Colored bars in a and c depict the median auPRC with error bars at the 25th and 75th percentile obtained by sampling 80% of enhancer-gene pairs 100 times without replacement.

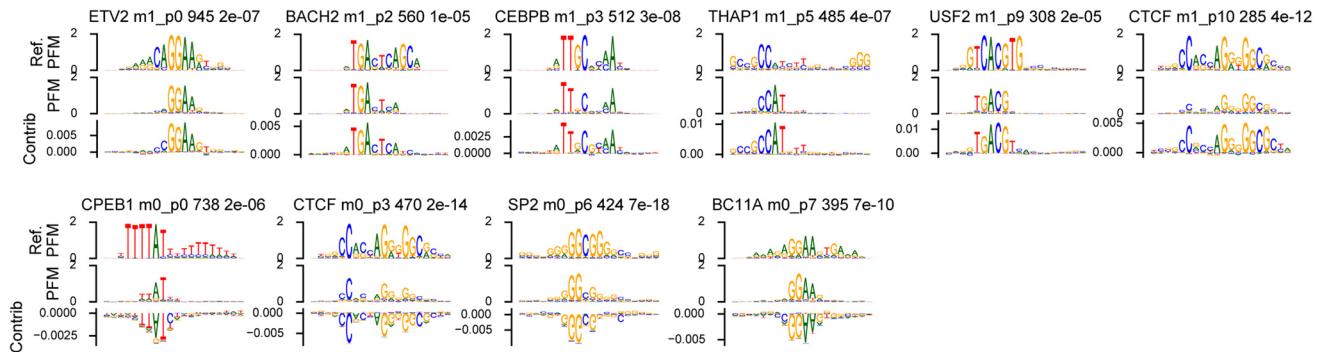
a

target=average DNase tracks

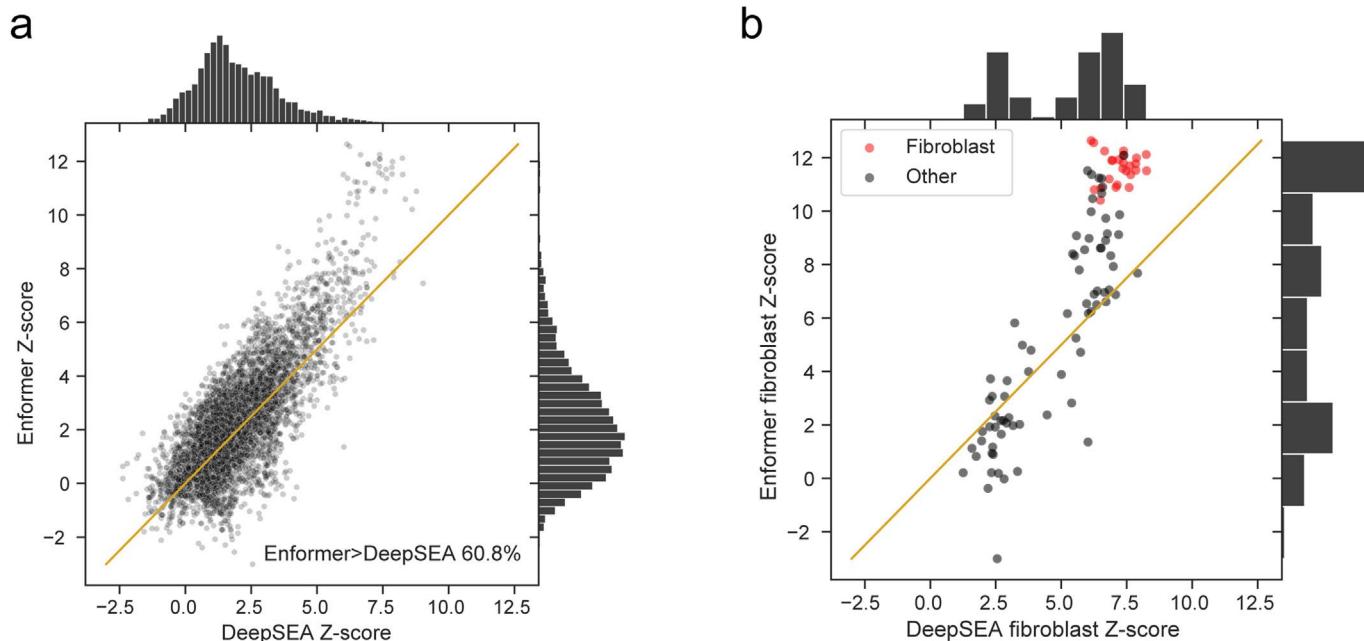


b

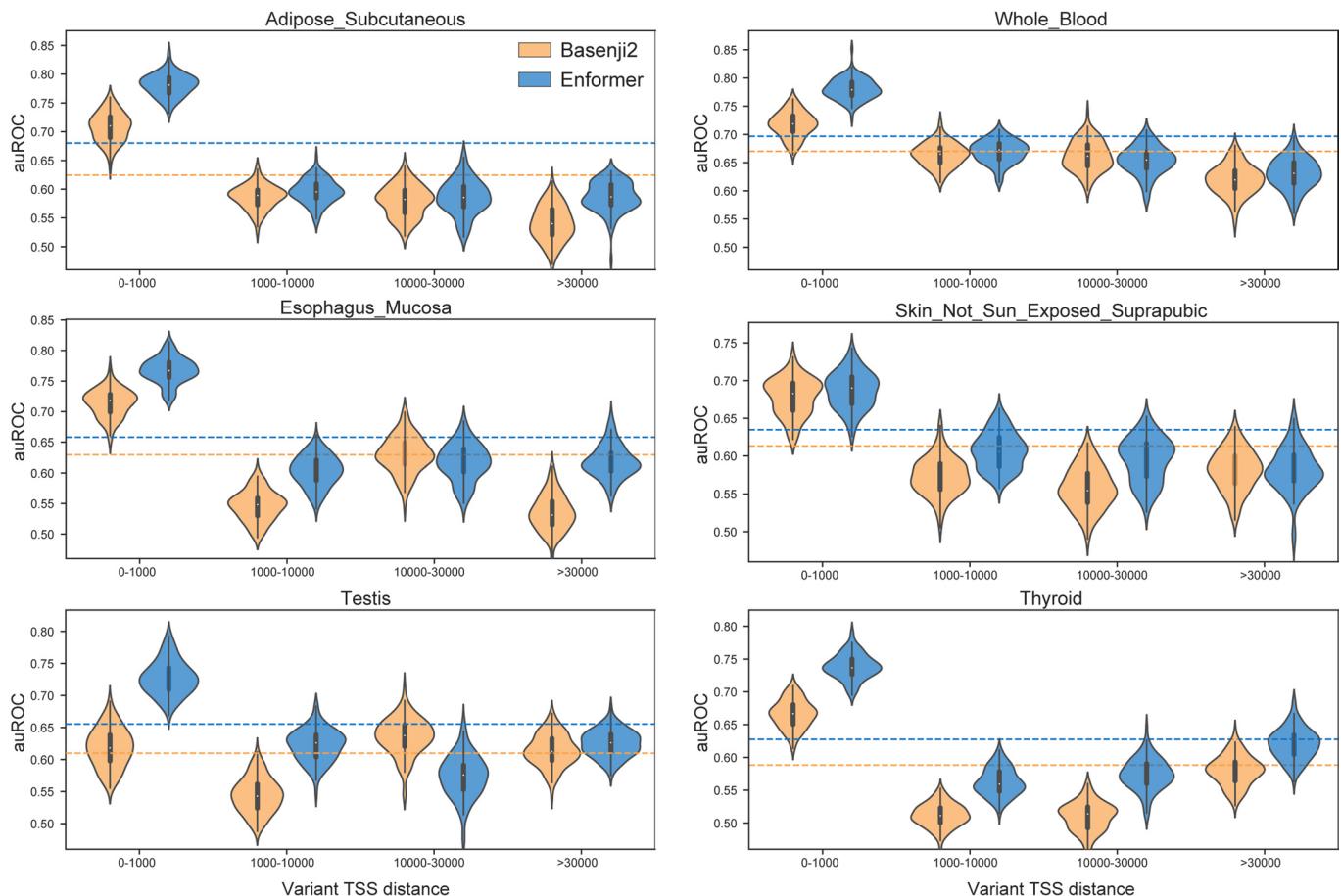
target=average CAGE tracks



**Extended Data Fig. 8 | TF-MoDISco motifs at TAD boundaries.** a,b) Motifs obtained by TF-MoDISco from gradient x input DNase (a) or CAGE (b) contribution scores at 1,500 TAD boundaries. Motif title contains: TF name of the closest motif match from HOCOMOCO v11 database, metacluster and pattern id returned by TF-MoDISco, number of seqlets supporting the motifs, and Tomtom q-value for the closest motif match (lower means better match). For each motif, the information content of the position frequency matrix (PFM) is visualized for the database motif in the top row and for the actual TF-MoDISco motif in the second row. Third row for each motif shows the contribution weight matrix (CWM) which can be negative. Shown are the top 6 motifs for each contribution score sign with sufficiently close match to a known motif (q-value < 1e-5) and support from at least 200 seqlets. Interestingly, the CTCF motif was discovered for both CAGE and DNase in both contribution score signs, suggesting that it can influence them in a positive or negative manner.



**Extended Data Fig. 9 | Enformer achieves greater and more specific SLDP concordance to GTEx than DeepSEA.** To compare Enformer and DeepSEA Beluga (convolutional neural network used in ExPecto) variant effect predictions, we manually matched DNase datasets that both models were trained on, finding 100 confident matches. We computed genome-wide statistical concordance between variant effect predictions for these DNase datasets and GTEx eQTL summary statistics using SLDP across all variants in the 1000 genomes dataset. **a**) We scatter plotted all DNase sample and GTEx tissue z-scores for the DeepSEA and Enformer predictions, observing that the Enformer scores are greater for 60.8%. Each point corresponds to (DNase sample, GTEx tissue) pair. Only some (DNase sample, GTEx tissue) pairs are biologically well-matched, while the majority are. **b**) Since many DNase samples profile fibroblasts we specifically plotted all DNase sample z-scores for the GTEx fibroblast summary statistics. We colored the DNase fibroblast samples in red, revealing that they are the highest scoring and most improved in the Enformer model relative to DeepSEA. This suggests that Enformer variant effect predictions are more tissue specific, since one would expect to obtain the highest Z-score for these matched samples shown in red.



**Extended Data Fig. 10 | Enformer outperforms Basenji2 on eQTL sign prediction.** For each of the GTEx tissues, we manually matched FANTOM5 CAGE sample descriptions to choose a single matched dataset (Methods). We then arranged a classification task to discriminate between fine-mapped causal eQTLs in which the minor allele increases gene expression versus eQTLs in which the minor allele decreases gene expression. We computed auROC statistics by ranking causal variants by their signed prediction for the corresponding sample. To consider the influence of variant distance to TSS, we compute auROC in four bins of roughly equal size. Across tissues and TSS distances, Enformer predictions usually achieve more accurate classification of eQTL sign than Basenji2 predictions. We display six example tissues with large numbers of fine-mapped eQTLs and with clear correspondence between CAGE and GTEx tissues. Violin plots show the auROC distribution of 100 bootstrap samples from the full set of variants. (The white dot represents the median, the thick gray bar in the center represents the 25%-75% percentile range and the thin line represents the entire data range.) Dashed lines represent the mean auROC over all distances. Both models struggle with variants beyond the promoter (TSS distance > 1,000), highlighting an important problem for future research.

## Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

### Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size ( $n$ ) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided  
*Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g.  $F$ ,  $t$ ,  $r$ ) with confidence intervals, effect sizes, degrees of freedom and  $P$  value noted  
*Give P values as exact values whenever suitable.*
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's  $d$ , Pearson's  $r$ ), indicating how they were calculated

*Our web collection on [statistics for biologists](#) contains articles on many of the points above.*

### Software and code

Policy information about [availability of computer code](#)

Data collection	No software was used for data collection.
Data analysis	All components of our core algorithm, including the full model architecture and example code to train and evaluate the model are available under the open source Apache 2.0 license at the following URL: <a href="https://github.com/deepmind/deepmind-research/tree/master/enformer">https://github.com/deepmind/deepmind-research/tree/master/enformer</a> . We used TensorFlow (v2.4.0) and Sonnet ( <a href="https://github.com/deepmind/sonnet">https://github.com/deepmind/sonnet</a> , v2.0.0) to train and evaluate the model. We used the following software in python 3.6 to analyze and visualize data: pandas (v1.1.5), numpy (v1.16.4), scikit-learn (v0.23.2), scipy (v1.2.1), pyBigWig (v0.3.17, <a href="https://github.com/deeptools/pyBigWig">https://github.com/deeptools/pyBigWig</a> , git hash: 705b07495ea78740eb7ea94fb0082e23661d9929), pyranges ( <a href="https://github.com/biocore-ntnu/pyranges">https://github.com/biocore-ntnu/pyranges</a> , v0.0.76), nucleus ( <a href="https://github.com/google/nucleus">https://github.com/google/nucleus</a> , v0.5.8), TF-MoDISco (v0.5.6.5), seaborn (v0.11.1), matplotlib (v3.0.3), and plotnine (v0.7.1). We also used the UCSC liftOver tool (Jan 2020 release).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research [guidelines for submitting code & software](#) for further information.

### Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Gene annotation was obtained from <https://www.gencodegenes.org/> (v32). Basenji2 training, validation, and test data was obtained from [https://console.cloud.google.com/storage/browser/basenji\\_barnyard/data](https://console.cloud.google.com/storage/browser/basenji_barnyard/data). Processed CRISPRi data for Fulco et al 2019 was obtained from supplementary material and for

Gasperini et al 2019 from GEO accession GSE120861. H3K27ac ChIP-seq data in K562 used for analysis in Fig. 2 was obtained from <https://www.encodeproject.org/> with file accession ENCF779QTH and DNase with file accessions ENCF413AHU and ENCF936BDN. TAD boundaries processed by Fudenberg et al 2020 were obtained from [https://console.cloud.google.com/storage/browser/basenji\\_hic/insulation](https://console.cloud.google.com/storage/browser/basenji_hic/insulation). Fine-mapped eQTLs are available from the supplementary material of Wang et al 2021. We acquired training and test sets as well as the predictive accuracies of individual competition participants from the CAGIS competition (M. Kircher, personal communication, <https://genomeinterpretation.org/content/expression-variants>). For comparison to ExPecto, we used the provided data from <https://github.com/FunctionLab/ExPecto/tree/master/resources>.

## Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences       Behavioural & social sciences       Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://nature.com/documents/nr-reporting-summary-flat.pdf)

## Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We used 5,313 human datasets and 1,643 mouse datasets to train and evaluate the Enformer model (Supplementary Table 2,3). This dataset (and therefore the number of samples) was compiled and already used by the previous best model (Basenji2).
Data exclusions	No data were excluded from analyses.
Replication	No experimental findings were disclosed, hence no replication was performed.
Randomization	Randomization was not relevant for this study as it involved a reanalysis of published datasets.
Blinding	Not applicable. No new data were collected.

## Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems		Methods	
n/a	Involved in the study	n/a	Involved in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies	<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Eukaryotic cell lines	<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology	<input checked="" type="checkbox"/>	<input type="checkbox"/> MRI-based neuroimaging
<input checked="" type="checkbox"/>	<input type="checkbox"/> Animals and other organisms		
<input checked="" type="checkbox"/>	<input type="checkbox"/> Human research participants		
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data		
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern		