Published in final edited form as:
Nat Mach Intell. 2022 December ; 4(12): 1088–1100. doi:10.1038/s42256-022-00570-9.

Evaluating deep learning for predicting epigenomic profiles

Shushan Toneyan[1,+], **Ziqi Tang**[1,+], **Peter K. Koo**[1,*]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA

Abstract

Deep learning has been successful at predicting epigenomic profiles from DNA sequences. Most approaches frame this task as a binary classification relying on peak callers to define functional activity. Recently, quantitative models have emerged to directly predict the experimental coverage values as a regression. As new models continue to emerge with different architectures and training configurations, a major bottleneck is forming due to the lack of ability to fairly assess the novelty of proposed models and their utility for downstream biological discovery. Here we introduce a unified evaluation framework and use it to compare various binary and quantitative models trained to predict chromatin accessibility data. We highlight various modeling choices that affect generalization performance, including a downstream application of predicting variant effects. In addition, we introduce a robustness metric that can be used to enhance model selection and improve variant effect predictions. Our empirical study largely supports that quantitative modeling of epigenomic profiles leads to better generalizability and interpretability.

Deep learning (DL) has achieved considerable success in predicting epigenomic profiles from DNA sequences, including transcription factor binding[1–3], chromatin accessibility[4,5], and histone marks[6,7]. By learning a sequence-function relationship, trained DL models have been utilized on various downstream tasks, such as predicting the functional effects of single-nucleotide variants associated with human diseases[4,6,8–12].

Recently, the variety of DL models proposed to address regulatory genomic prediction tasks has increased substantially[13–21]. The diversity of proposed models, the datasets they are trained on, how the datasets are processed, and the tricks used to train the models make it challenging to assess which innovations are driving performance gains. A direct comparison of model performance cannot always be made easily due to the variations of how the prediction tasks are framed. For instance, previous approaches typically frame the

[*] koo@cshl.edu .
[+]These authors contributed equally to this work
Author contributions statement
ST, ZT, and PKK conceived of the experiments. ST and ZT wrote the code base, conducted the experiments, analyzed and interpreted the results. All authors contributed to the manuscript.

Competing interests
The authors declare no competing interests.

Code availability
The code to reproduce results and figures in this study is available at Zenodo, https://doi.org/10.5281/zenodo.6464031 [58]. The open-source project repository is available at GitHub, https://github.com/shtoneyan/gopher. Stable version of code used for generating results in paper is available through Zenodo, https://doi.org/10.5281/zenodo.6977213 [59]

task as a *binary* classification, where binary labels represent functional activity based on a peak caller. However, in collapsing the amplitude and shape of a peak into a binary label, information about differential *cis*-regulatory mechanisms encoded in these attributes is lost. Recently, *quantitative* models[22–26] have emerged, similarly taking DNA sequences as input but now directly predicting experimental read coverage as a regression task, thus bypassing the need for a peak caller and preserving quantitative information of epigenomic tracks. Since standard metrics differ across classification and regression tasks, it remains unclear how to directly compare models trained on different tasks.

To address this issue, Kelley et al[22] propose to 'binarize' their quantitative predictions, enabling a comparison of the overlap with binary labels. However, this narrowly focuses on genomic regions that have been annotated by a peak caller, which is sensitive to hyperparameter choices[27]. Alternatively, Avsec et al[25] compared the performance of a binary model with an augmented version that appends an output-head that simultaneously predicts quantitative profiles. While this measures the added benefit of quantitative modeling, is requires retraining multiple model versions, which can be sensitive to initialization and is not easily extendable to existing models.

Moreover, other modeling choices within a prediction task make it challenging to directly make fair comparisons. For instance, existing quantitative models predict different resolutions of the epigenetic profiles. Basenji[22] predicts at a resolution based on non-overlapping bins of 128 base-pairs (bp), while BPNet[25] predicts at base-resolution. Comparing models across different resolutions is not straightforward, because binning affects the smoothness of the coverage values which, in turn, can influence performance metrics. Moreover, existing methods employ different data augmentations and analyze different subsets of training and test data, further complicating any direct comparisons.

As the number of applications continues to grow, a bottleneck of modeling innovations is forming as we lack the ability to perform a critical assessment of newly proposed models. Here, we propose an evaluation framework for genomic DL models that enables a systematic comparison of prediction performance and model interpretability, irrespective of how the prediction task is framed. Using this framework, we perform a comprehensive evaluation of quantitative models and binary models on a chromatin accessibility prediction task to elucidate beneficial factors in model architecture, training procedure, and data augmentation strategies. Moving beyond predictive performance, we assess each model with additional criteria: 1) robustness of predictions to small perturbations to the input sequence, 2) variant effect predictions, and 3) interpretability of the learned representations.

Results

To gain deeper insights into the factors that drive model performance, it is critical to be able to make a systematic and fair comparison across DL models. To address this gap, we developed a python-based software package called GOPHER (GenOmic Profile-model compreHensive EvaluatoR) that consists of high-level Tensorflow/Keras-based APIs for data processing, augmentation strategies, and comprehensive model evaluation, including variant effect predictions and model interpretability, for binary and quantitative modeling of epigenomic profiles (Fig. 1).

Performance evaluation of best-in-class quantitative models

The prominent quantitative models, Basenji[22] and BPNet[25], each employ different strategies for model design, data processing, loss function, evaluation metric, and data augmentations (Supplementary Table 1). To identify the key factors that drive performance gains, we performed a systematic comparison of Basenji- and BPNet-inspired models on a multi-task quantitative prediction of chromatin accessibility ATAC-seq data across 15 human cell lines (see Methods). This dataset provides a sufficient challenge in deciphering the complexity of regulatory elements across cell types but maintains a dataset size that is amenable to the scale of comprehensive evaluations performed in this study.

For each base model, we used GOPHER to search for optimal hyperparameters using each model's original *target resolution* and *training set selections* (Supplementary Fig. 1). Target resolution defines the bin size which is used to create non-overlapping windows of coverage values, ranging from base resolution to predicting a single quantitative output. BPNet-based models were trained at base resolution (BPNet-base) on *peak-centered* data, i.e. genomic regions that contain at least 1 peak from a target cell-type. On the other hand, Basenji-based models were trained at 128 bin-resolution (Basenji-128) with *coverage-threshold* data, which consists of segmenting chromosomes into non-overlapping regions and then sub-selecting the regions that have a max coverage value above a set threshold (see Methods for additional details).

Overall, the quality of the model predictions were in line with previous studies (Fig. 2a). Using the optimized models as a baseline, we compared the performance impact of various factors, including loss function, target resolution, training set selection, and test set selection.

Loss function.—The choice of loss function for quantitative models is not as straightforward as binary models, which is binary cross-entropy; loss functions can penalize the shapes (e.g. Pearson's r) or the magnitudes (e.g. mean-squared-error (MSE)). To explore the effect of loss function on quantitative modeling, we systematically evaluated Basenji- and BPNet-based models across 5 different loss functions at 8 different target resolutions (Fig. 2b). Evidently, Poisson NLL outperformed the other losses at all tested bin-resolutions, i.e. lower MSE and higher Pearson's r on the held out test set. The combination of Pearson's r and MSE yielded the second best overall performance. For Basenji-based models, higher bin sizes tended to yield better performance up to a bin size of about 1 kb, which is roughly the width of ATAC-seq peaks. Surprisingly, BPNet-based models yielded a different trend, where base-resolution models performed the best (Extended Data Fig. 1a). This is expected (to an extent) as each model was optimized for different resolutions. This suggests that model design can be optimized for a given resolution but may not necessarily generalize across resolutions.

Target resolution.—Quantitative models that employ different target resolutions cannot be directly compared, because higher bin sizes effectively smoothens high-frequency noise, affecting correlation-based performance metrics. To explore whether the relationship between resolution and performance is due to a statistical artefact from binning, we developed an evaluation scheme that enables a direct comparison across target resolutions. Specifically, we binned the predictions of the higher resolution models to match the lower resolution predictions. This effectively provides an avenue to fairly compare the performance across target resolutions. As expected, Basenji-based models trained at a given target resolution yield a higher Pearson's r with increased smoothing, despite the underlying predictions remaining unchanged (Fig. 2c). A similar observation was made for BPNet-based models (Extended Data Fig. 1b). To further demonstrate the sensitivity of Pearson's r on smoothness properties, we systematically smoothed predictions by applying a box-car filter with window sizes that matched a lower-resolution bin and observed a similar trend (Extended Data Fig. 2).

Training and test set selection.—To assess how dataset selection affects generalizability we trained Basenji-based models at different resolutions on datasets with different coverage thresholds as well as a peak-centered dataset. By evaluating each model on the whole-chromosome test set, we found that the training set with the lowest threshold yielded the best performance, while peak-centered models performed the worst (Fig. 2d). Thus, limiting the training set to higher activity regions reduces model's whole chromosome performance.

Also, choice of test set can influence the measure of generalization performance. We generated new test sets with progressively stringent coverage thresholds to modulate between whole chromosome and peak-centered coverage. Interestingly, Basenji-based models' performance monotonically increased with the coverage threshold of the test set (Fig. 2e). In addition, we performed a more targeted evaluation of the performance at high-activity regions across models trained on either peak-centered or coverage-threshold datasets (Supplementary Table 2). We observed a trend where models trained on peak-centered data had a slight performance advantage over models trained on coverage-threshold data in terms of scaling the heights of the reads. However, models trained on coverage-threshold data yielded substantially better peak detection performance across the whole chromosome. Thus, there is a slight trade-off in scaling the predictions when training on coverage-threshold data, but it leads to lower false-positive predictions genome-wide.

Robustness test to identify fragile models

Robustness to input perturbations is a widely used criterion for evaluating the trustworthiness of DL models[28,29]. However, adversarial attacks[30] using small, targeted noise do not extend naturally to DNA sequences. Alternative perturbations, such as single-nucleotide mutations, can affect function and hence are inappropriate. Thus, we developed a robustness test to measure the sensitivity of model predictions to translational shifts of the input sequences; function is largely maintained by also shifting the target predictions. In our robustness test, a variation score is calculated by considering the variability in the predictions of randomly translated input sequences (Fig. 3a).

We evaluate the robustness of BPNet-128 and Basenji-128 across training sets (i.e. peak-centered or coverage-threshold data) and different combinations of augmentations, including random reverse-complement (RC) and random shifts of the input sequence (up to 1024 bp). Since the robustness metric is sensitive to bin-resolution, we opted to compare models at the same resolution to enable a fair comparison. Indeed, models trained with augmentations yielded improved robustness, especially when trained on peak-centered data (Fig. 3b). On the other hand, models that were trained on coverage-threshold data already benefited from the randomly-centered profiles. This explains why RC alone was sufficient for coverage-threshold models, whereas peak-centered models benefited from both augmentations. Surprisingly, we observed that prediction performance and model robustness are not strictly correlated, suggesting that robustness can be utilized as an additional metric to facilitate model selection.

Comparing quantitative model architectures

The space of architectures for quantitative models has been much more limited than binary models. Hence, we wanted to address two open questions: (1) could standard convolutional neural networks (CNNs) that were successful on binary classifications have similar success at quantitative regressions, and (2) could further exploration of the architectures improve performance?

We benchmarked 8 CNNs based on different architectures (a baseline CNN and a residual CNN with dilated convolutions[31] called ResidualBind[32]), first-layer activations (rectified linear units (ReLU) or exponential) and output heads (single or task-specific). A more comprehensive overview of the motivation for these architectural choices and results are presented in Supplementary Note 1. Briefly, we found that simple CNNs can be effective at predicting epigenomic profiles as a quantitative regression (Supplementary Table 3). Nevertheless, enhancements, such as residual blocks and task-specific output heads, can improve performance. This highlights how design considerations for quantitative modeling is nascent and can benefit from further comprehensive explorations.

Benchmarking performance across binary and quantitative models

Although quantitative models were developed with the aim of preserving more information about epigenomic profiles, directly comparing the different prediction formats between binary and quantitative tasks is not straightforward. To bridge this gap, we outline a strategy to convert between the prediction formats (Fig. 4a). To convert binary predictions to a quantitative format, we treated the pre-activation logits as the predicted coverage values. Although binary models are not trained to predict signal strength, the model's confidence can be encoded in the unbound logits. Moreover, to convert quantitative predictions to a binary format, we calculated the average coverage predictions at positive regions and negative regions based on corresponding binary-labelled data. These two distributions can be used to calculate standard classification metrics.

Using this task-conversion method, we compared the performance of various quantitative and binary models (Supplementary Table 4). Interestingly, when evaluating on peak-centered data, several binary models yielded similar performance compared to quantitative models (Fig. 4b and Extended Data Fig. 3a). However, when converting the binary models to quantitative metrics, quantitative models outperformed all binary models. All quantitative models also yielded better performance when evaluated across the whole chromosome (Fig. 4c and Extended Data Fig. 3b). Together, this demonstrates that while some binary models can be competitive with quantitative models within highly functional sites, quantitative models tend to yield better overall performance across whole chromosomes.

Out-of-distribution generalization: variant effect prediction

A major downstream application for sequence-to-function models is scoring the functional effect of mutations. We benchmarked each model on this out-of-distribution (OOD) generalization task using experimental data from the CAGI5 Competition[33,34], similar to previous studies[5,26]. The CAGI5 dataset consists of massively parallel reporter assays (MPRAs) that measure the effect size of single-nucleotide variants through saturation mutagenesis of 15 different regulatory elements across different cell-types. Instead of making a single sequence prediction, we employed robust predictions by introducing random translations to each mutated sequence and averaging the central overlapping region, similar to our robustness test (see Methods). An anecdotal visualization shows that the variant effect predictions by quantitative models are qualitatively effective despite being trained on different data (Fig. 5a).

By benchmarking various models, we observed that quantitative models consistently outperformed binary models (Fig. 5b). Importantly, whole-chromosome generalization seems to be a reliable metric for variant effect performance. As a control, we compared whether robust predictions are beneficial compared to the standard approach of a single-pass prediction. Strikingly, 50 out of 56 models performed better using robust predictions (Fig. 5c). Models that did not employ random shift augmentations benefited the most from robust predictions (Fig. 5c, inset). This suggests that robust models naturally yield more trustworthy variant effect predictions, and our method for robust predictions can rescue the efficacy of less robust models.

Model interpretation

A major downstream application of genomic DL models is interpretability analysis, which can facilitate the discovery of functional motifs and their complex interactions. Here we compare binary and quantitative models across several common interpretability approaches, including motif discovery through filter visualizations and quantitative hypothesis testing using global importance analysis (GIA)[32].

First-layer convolutional filters provide an inductive bias to learn translational patterns such as motifs. However, the extent that they actually do in practice depends on many factors including design choices[35–37]. Using activation-based alignments to visualize filters[38], we quantitatively compared each of the model's first layer representations against a database of known motifs. We found that improved predictions do not necessitate learning better motifs (Supplementary Table 4). Irrespective of how the modeling task was framed, models that employ exponential activations consistently led to learning better motif representations in first layer filters. For additional details of this analysis, see Supplementary Note 2.

GIA is an interpretability approach that enables direct testing of hypotheses of the importance of features learned by a DL model. GIA computes the effect size, or global importance score, of hypothesis patterns that are embedded within a population of background sequence, where the other positions are effectively randomized. This approach essentially marginalizes out any confounding patterns within any individual sequence, revealing the global importance of only the embedded patterns on model predictions. Using GIA, we test various hypotheses of AP-1, GATA, and ATAAA motifs, which were identified as the most prevalent motifs through attribution analysis (see Supplementary Note 3).

Using GIA, we first explore the effect of flanking nucleotides in PC-3 cells using high performing models, quantitative ResidualBind-32 and ResidualBind-binary, both with exponential activations. Strikingly, we found that flanking nucleotide combinations relative to the AP-1's core binding site can drive predictions by a factor of 2 to 3 (Fig. 6a), similar to previous observations[39,40]. On the other hand, a position-weight-matrix-based approach[41], which considers each position independently, would score many flank combinations for the AP-1 binding sites the same. A similar observation was made for other models, both quantitative and binary (Extended Data Fig. 4). This demonstrates that DL models consider complex higher-order dependencies of flanking nucleotides to be an important feature of TF binding sites, a well-known phenomenon[42,43]. Moreover, both binary and quantitative models can capture this information *de novo*.

We then explored the extent that distance between motifs influences model predictions. We performed GIA experiments where the AP-1 motif was fixed at the center of the sequences and the position of the other motif (i.e. AP-1 or ATAAA) was varied (Fig. 6b). Interestingly, we observed that two AP-1 motifs yielded a symmetric 50 bp window where predictions are plateaued, beyond which, the global importance begins to drop off for both models. On the other hand, the ATAAA motif exhibits an asymmetric distance dependence with a favorable location on the 3' end with a few nucleotide gap, beyond which, there is a precipitous drop in global importance. This was also observed across other models, both binary and quantitative, albeit with variable magnitudes in effect size but a similar trend (Extended Data Fig. 5).

We often observed multiple motifs present in combinations across accessible sites (Supplementary Note 3). To test whether the models have learned cooperativity between AP-1 and other motifs, we compared the global importance for the motifs embedded in sequences alone and in combinations with other motifs (at previously identified optimal distances). In ResidualBind-32, we observed that the sum of individual effects was less than the scenario where both motifs were present, indicating the model has learned cooperative interactions (Fig. 6c). The effect size was varied across transcription factors, with a smaller effect observed for GATA::AP-1 compared to other motifs, such as ATAAA::AP-1 and AP-1::AP-1. This suggests that cooperative interactions are strongly associated with chromatin accessibility levels. Surprisingly, a discrepancy arose for binary models, where there was no strong evidence that cooperativity was learned. These trends were also observed across other binary and quantitative models (Extended Data Fig. 6).

Instead of directly imposing patterns on background sequences, we also conducted occlusion-based interventional experiments where we identified exact instances of the core motif for AP-1 and replaced them with randomized sequences across the test set – a global importance of motif occlusion within its natural sequence context. We find that the number of AP-1 motifs indeed drives high functional activity for PC-3 (Supplementary Fig. 2).

Together, the interpretations of quantitative models appear to be more consistent with each other than binary models. Despite under-performing on generalization tasks, well-trained binary models can largely capture similar biological interpretations as quantitative models, with the exception of cooperative interactions.

Discussion

The variety of DL models that predict regulatory genomic tasks has increased substantially in recent years. The variations of architectures, prediction tasks, and training strategies make it challenging to assess which innovations are driving performance gains. To address this gap, we introduced GOPHER, an evaluation framework to perform a comprehensive and fair evaluation of DL models in regulatory genomics. GOPHER provides a framework that supports data processing for both binary classification and quantitative regression tasks, in addition to training custom DL models with various data augmentations. GOPHER also incorporates many model interpretability tools, such as first-layer filter visualization, global importance analysis, and attribution analysis. Using GOPHER, we addressed several open questions: (1) how to fairly compare binary and quantitative models; how choice of (2) loss function and (3) dataset selection influences model performance; (4) how to compare quantitative models at different resolutions; (5) how augmentation strategies influence model performance and robustness to translational perturbations; how modeling choices influence (6) functional variant effect predictions and (7) model interpretability.

While the study here focuses on ATAC-seq data, the specific claims of optimal architectures and training procedures may be nuanced across other data modalities, such as ChIP-nexus[25] and CAGE-seq[44]. In such cases, additional considerations may arise, such as GC-bias and signal normalization. Also, BPNet and Basenji were originally optimized for different datasets and scales. Nevertheless, both models performed well here upon hyperparameter optimization. Emerging architectures based on transformers[45] could provide stronger inductive bias to capture distal interactions[26], but was not explored here.

In general, our work largely supports that quantitative modeling yields better generalization (on average), both on held-out data and OOD variant effect predictions. Of course, well-tuned binary models can perform comparable to (or even better than) a poorly designed quantitative model. It remains unclear whether binary models are fundamentally limited based on their treatment of functional activity or whether incorporating more inactive regions during training would boost performance. Moreover, it is not clear whether the performance gains of quantitative models are due to learning better biological signals or whether they are just better at learning experimental noise sources. A major limitation arises as a consequence of focusing on performance – treating experimental measurements as ground truth, despite biological and technical variability across replicates (eg. Supplementary Table 5). Thus, evaluation based on important downstream tasks, such as variant effect prediction and model interpretability, provides a path to move beyond one-dimensional performance benchmarks to the beneficial use case of biological discovery.

Methods

Training data

ATAC-seq (Assay for Transposase-Accessible Chromatin with high-throughput sequencing[46]) data for human cell lines were acquired from the ENCODE database[47] – fold change over control bigWig files for quantitative analysis and IDR peak bed files for binary analysis – using experimental accessions in Supplementary Table 6. The bigWig tracks were log-fold-normalized for sequencing depth using control tracks as per the ENCODE data processing pipeline; no further processing was done. Each of the 15 cell lines were sub-selected based on a lower cross-correlation of coverage values at IDR peaks across cell lines below 0.75. Data from replicate 1 for each experiment was used to generate the train, validation, and test sets. Data from replicate 2 was used to assess the experimental ceiling of prediction performance.

Coverage-threshold data.—Each chromosome is split into equal, non-overlapping input size chunks of 3 kb and each chunk is included in the dataset if the max coverage value for any of the targets is above the threshold. By default, coverage-threshold data employed a threshold of 2, unless specified otherwise. Each sequence that passed this threshold was included as part of the dataset and down-sampled to 2 kb with a strategy that depends on data augmentations (see below). The targets were then binned with non-overlapping windows according to the specified target resolution and was calculated online during training and testing. For any given coverage value array of length L and bin size B, it was reshaped into an array with shape (B, L/B) – down-sampling was achieved according to the mean within each bin.

Peak-centered data.—For peak-centered datasets, we selected IDR bedfiles from the ENCODE experiments corresponding to the same replicate as the coverage-threshold data. The bed files of each cell line were merged into a single bed file, in a manner similar to Kelley et al[4]. The Basset data processing pipeline divides the genome into segments of length specified as the input size and merges peaks according to an overlap size parameter. Each sequence in the dataset contains at least one peak across all cell lines. Sequences containing an IDR peak for the cell line is given label ‘1’ otherwise label ‘0’.

Data splits.—We split the dataset into training, validation, and test sets using chromosome 8 for test, chromosome 9 as validation and the rest as training (excluding Y chromosome). We also removed the unmappable regions across all data splits. The same split was applied to datasets to allow a direct comparisons across experiments.

Held-out test evaluation

Pearson correlation can be calculated using the concatenated whole chromosome per cell line, which is referred to as Pearson’s r (whole), or per sequence correlation averaged across the test set when specified. The difference between these metrics manifests as a different mean in the correlation calculation; a global mean for whole chromosome versus a per sequence mean. Whole chromosome evaluation is calculated by concatenating the predictions for the entire chromosome 8 with the exception of unmappable regions.

A per sequence Pearson correlation was calculated for peak-centered data, test selection analysis, and robustness analysis, unless specified otherwise. For a compilation of all model evaluations see Supplementary Data 1.

Scaling predictions.—Predictions were scaled to address the large discrepancy between predictions and experimental values for shape-based loss functions (eg. Pearson's r). Though we found that applying it to other losses also yielded slightly better performance. This was accomplished by calculating a global scaling factor per cell line, which is computed as the ratio of the mean of experimental and predicted coverage values across the entire test chromosome, and multiplying the scaling factor to the predictions.

Models

Basenji.—Basenji-based model is composed of a convolutional block, max-pooling with pool size of 2 (which shrinks the representations to 1024), 11 residual blocks with dilated convolutional layers[48], followed by a final convolutional layer. The convolutional block consists of: GELU activation[49], convolutional layer with a kernel of width 15, batch normalization[50]. The residual block is composed of: GELU activation, dilated convolutional layer with a kernel of width 3 and half the number of filters and a dilation rate that grows by a factor of 1.5 in each subsequent residual block, batch normalization, GELU activation, convolutional layer with width 1 and the original number of filters, batch normalization, and dropout with a rate 0.3. Each residual block has a skipped connection to the inputs to residual block. An average pooling layer is applied to the final output convolutional layer to shrink the representations to the corresponding target resolution. A dense layer with softplus activations following the last convolutional block then outputs the predictions target. In case of base resolution, the first max-pool size is set to 1.

The original Basenji model employs multiple convolutional blocks with a max pooling of size 2 to reduce the dimensions of the sequence to 1024 units, upon which 11 residual blocks are applied. Since our input size is 2048 bps, we employed a single convolutional block to achieve the same dimensions as the original Basenji model. We performed hyperparameter search of the number of convolutional filters in each layer to optimize for the ATAC-seq dataset used in this study. For additional details of specific hyperparameters, see Supplementary Data 2.

BPNet.—BPNet consists of a convolutional layer, followed by 9 dilated convolutional layers with progressively increasing dilation rates (scaled by powers of 2) that each have a residual connection to the previous layer. Task-specific output-heads, each with a separate transpose convolution, is built upon the final residual layer. To adapt BPNet to lower resolutions, all predictions are initially made at base-resolution followed by an average pooling layer for each task-specific output-head, with a window size and stride that matches the target resolution.

A key difference with the original BPNet architecture is that the negative strand, bias track and read counts output-head was not used throughout this study. Moreover, the original loss function was not employed here as we found better success with the modified BPNet using a Poisson NLL loss. This may be attributed to the lower resolution in read coverage for bulk ATAC-seq, or due to original model targeting raw read count instead of fold change over control tracks, though further investigation is needed to understand the disparity. These modifications may have affected the performance of BPNet. We optimized hyperparameters of the model, focusing on the number of filters in each layer and the kernel size of the transpose convolution in the task-specific output heads (Supplementary Fig. 1). The specific choices of hyperparameters in BPNet can be found in Supplementary Data 2.

CNN-baseline.—The CNN baseline model is composed of 3 convolutional blocks, which consist of a 1D convolution, batch normalization, activation, max pooling and dropout, followed by 2 fully-connected blocks, which includes a dense layer, batch normalization, activation, and dropout. The first fully connected block scales down the size of the representation, serving as a bottleneck layer. The second fully-connected block rescales the bottleneck to the target resolution. This is then reshaped to match the number of bins × 8. For instance, the number of hidden units for models at 32 bin target resolution are 2048/32 = 64 × 8, then reshaped to (64, 8). Base resolution models set the hidden units to 2048 × 8 then reshaped to (2048, 8). This is followed by another convolutional block. The representations from the outputs of the convolutional block is then input into task-specific output heads or is directly fed to a linear output layer with softplus activations. For task-specific output heads, each head consists of a convolutional block followed by a linear output layer with softplus activations. The activation of the first layer is either exponential or ReLU, while the rest of the hidden layer activations are ReLU. The specific hyperparameters of each layer, including the dropout rates, are specified in detail in Supplementary Data 2.

ResidualBind-base.—ResidualBind-base builds upon the CNN-baseline models by adding a residual block after the first 3 convolutional layers. The first two residual blocks consist of 5 dilated convolutional layers and the third residual block consists of 4 dilated convolutional layers. Similar to CNN-baseline models, this is then followed by 2 fully connected blocks, which are reshaped to a shape (2048, 8), and a convolutional block. Here, another residual block that consists of 5 dilated convolutional layers was applied. This is then fed into an output head, which has the same composition as the CNN-baseline. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

ResidualBind-32.—ResidualBind-32 also builds upon the CNN-baseline models by adding a residual block after the first 3 convolutional layers, but with a few key differences from ResidualBind-base. The third residual block consists of 3 dilated convolutional layers instead of 4. Moreover, ResidualBind-32 does not go through a bottleneck layer that is prototypical of the CNN-baseline design. For task-specific output heads, the representations of the third residual block are input into a convolutional block followed by a task-specific output heads similar to the CNN-baseline models. For a single output head, the representations of the third residual block are input into a position-wise fully connected block followed by a linear output layer. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

Binary models—Four main model structures are used for binary models. One fine-tuned Basset[4] structure and three re-purposed quantitative models structures: Basenji, CNN-base, and ResidualBind-base. Basset is composed of three blocks of convolutional layer followed by batch normalization, activation and max-pooling. The output is then flattened and fed into 2 fully connected layers with dropout and an output layer with sigmoid activations. Basset hyperparameters were optimized for the binary version of the ATAC-seq dataset in a similar manner to Basenji. For Basenji-binary, CNN-binary and ResidualBind-binary, their structure highly resembles the quantitative model based on a single output-head. For CNN-binary and ResidualBind-binary, we apply a fully connected output layer with sigmoid activations to the bottleneck layer. For Basenji-binary, we take the penultimate representation and perform a global average pool, followed by a fully connected output layer. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

Training.—Each quantitative model was trained for a maximum of 100 epochs using ADAM[51] with default parameters. Early stopping was employed with a patience set to 6 epochs (monitoring validation loss as a criterion). By default, models were trained with random reverse-complement and random shift data augmentations unless specified otherwise.

Quantitative CNN and ResidualBind (base and 32 bin-resolution), along with binary versions of these models, were trained for a maximum of 100 epochs using ADAM with default parameters. Early stopping with a patience of 10 was used. The initial learning rate was set to 0.001 and decayed by a factor of 0.2 when the loss function did not improve after a patience of 3 epochs.

Data augmentations

Random shift.—Random shift is a data augmentation that randomly translates the input sequence (and corresponding targets) online during training. All datasets were generated with input size set to 3,072 bp. When random shift is used, for each mini-batch, a random sub-sequence of 2,048 bp and its corresponding target profile was selected separately for each sequence. When random shift is not used, the central 2,048 bp is selected for all sequences in the mini-batch.

Reverse-complement.—Reverse-complement data augmentation is employed online during training. During each mini-batch, half of training sequences were randomly selected and replaced by their reverse-complement sequence. For those sequences that were selected, the training target was correspondingly replaced by the reverse of original coverage distribution.

Hyperparameter search

The ATAC-seq datasets in this study differ greatly in complexity, i.e. size and coverage distribution, from the original Basenji and BPNet studies. Therefore, we performed a hyperparameter search for each base architecture for our ATAC-seq dataset (Supplementary Fig. 1). We used WandB[52] to keep track of the model choices and for visualization. We fine-tuned Basenji and BPNet at 128 bp and base resolution, respectively, which represent the original resolutions for these models. We also kept their original training set selection strategy, that is, we trained Basenji on coverage-threshold data and BPNet on peak-centered data. For Basenji, the number filters across the convolutional layers were varied as well as the presence or absence of dropout layers (fixed rate for each layer). For BPNet, we performed a hyperparameter search over the number of convolutional filters in each layer and the kernel size in the task-specific output heads. We employed the original data augmentations (i.e. random reverse-complement and random shifts for Basenji and only random reverse-complement for BPNet). For each model, we used the Poisson NLL loss function. We originally used a MSE and multinomial NLL loss for BPNet, but found that optimization using Poisson NLL yielded better performance. The models were trained for maximum of 40 epochs with an Adam optimizer[51] using default parameters. Initialization was given according to Ref.[53]. The optimal set of hyperparameters for each model was selected based on the lowest validation loss and the final architectures are given in Supplementary Data 2.

Robustness test

To measure the robustness to translational perturbations, we analyzed the sequences within the held-out test chromosome that were identified to contain a statistically significant peak for the given cell-type under investigation. This ensures that the robust predictions are only considered for genomic regions that exhibit statistically significant coverage values. Specifically, we took each 3072 bp sequence in the dataset and generated 20 contiguous sub-sequences of length 2048 bp. Each sub-sequence was sent through the model to get a prediction, and all of the predictions were aligned based on the sub-sequence. All sub-sequences contain a center 1024 bp window that overlaps. Standard deviation is calculated for each position across these 20 sequences and averaged across the length of prediction. The average sequence coverage across 20 sequences was used to normalize the average standard deviation to make it invariant to scale. Therefore variation score for each sequence is calculated as average per position standard deviation divided by average sequence coverage. A higher variation score corresponds to a less robust model, while a lower variation score corresponds to more stable predictions, irrespective of translations to the inputs. Due to binning artifacts, we only compare this robustness test for models that share the same bin-resolution.

Variant effect prediction

Dataset.—The CAGI5 challenge dataset was used to benchmark model performance on variant calling. Each regulatory element ranges from 187bp - 600bp in length. We extracted 2048 bp sequences from the reference genome centered on each regulatory region of interest and converted it into a one-hot representation. Alternative alleles are then substituted correspondingly to construct the CAGI test sequences.

Standard predictions.—For a given model, the prediction of 2 sequences, one with a centered reference allele and the other with an alternative allele in lieu of the reference allele, is made and the coverage values are summed separately for each cell-type. For each sequence, this provides a single value for each cell-type. The cell-type agnostic approach employed in this study then uses the mean across these values to calculate a single coverage value. The effect size is then calculated with the log-ratio of this single coverage value for the alternative allele and reference allele, according to: log(alternative coverage / reference coverage).

Robust predictions.—Robust predictions were calculated separately for reference and alternative alleles and the effect size was calculated based on their log2 fold change. For a given model, robust predictions were made by: 1) sampling 20 randomly shifted sequences centered on a variant-of-interest, 2) sending them through the model to get coverage predictions for each cell-type, 3) align predictions based on the shifted sub-sequences, 4) calculating the mean coverage within overlapping 1024 bp region for each cell-type, and 5) averaging the mean coverage values across cell-type. This was done separately for the reference allele and the alternative allele, and the effect size was calculated similar to the standard predictions as the log-ratio.

Evaluation.—To evaluate the variant effect prediction performance, Pearson correlation was calculated within each CAGI5 experiment between the experimentally measured and predicted effect size. The average of the Pearson correlation across all 15 experiments represents the overall performance of the model. A full list of variant effect prediction performances for models can be found in Supplementary Data 3.

Model interpretability

Tomtom.—The motif comparison tool Tomtom[54] was used to match the position probability matrix of the first convolutional layer filters (calculated via activation-based alignments[38]) to the 2022 JASPAR nonredundant vertebrates database[55]. Matrix profiles MA1929.1 and MA0615.1 were excluded from filter matching to remove poor quality hits; low information content filters then to have a high hit rate with these two matrix profiles. Hit ratio is calculated by measuring how many filters were matched to at least one JASPAR motif. Average q-value is calculated by taking the average of the smallest q-value for each filter among its matches.

Attribution analysis.—Attribution analysis was based on grad-times-input with saliency maps[56]. For a given model, gradients of the prediction with respect to a given cell-type were calculated with respect to the input sequence to yield a $L \times A$ map, where L is the length of the sequence and A is 4 – one for each nucleotide. Each saliency map was multiplied by the input sequence, which is one-hot, to obtain just the sensitivity of the observed nucleotide at each position. A sequence logo was generated from this by scaling the heights of the observed nucleotide, using Logomaker[57].

Global importance analysis.—For global importance analysis[32], we generated background sequences by performing a dinucleotide shuffle of 1,000 randomly sampled sequences from those within our coverage-threshold test set. The global importance is calculated via the average difference in predictions of background sequences with embedded patterns-under-investigation and without any embedded patterns. For quantitative models, the predictions represent the average coverage predictions for the cell-type under investigation. For binary models, the predictions represent the logits for the cell-type under investigation.

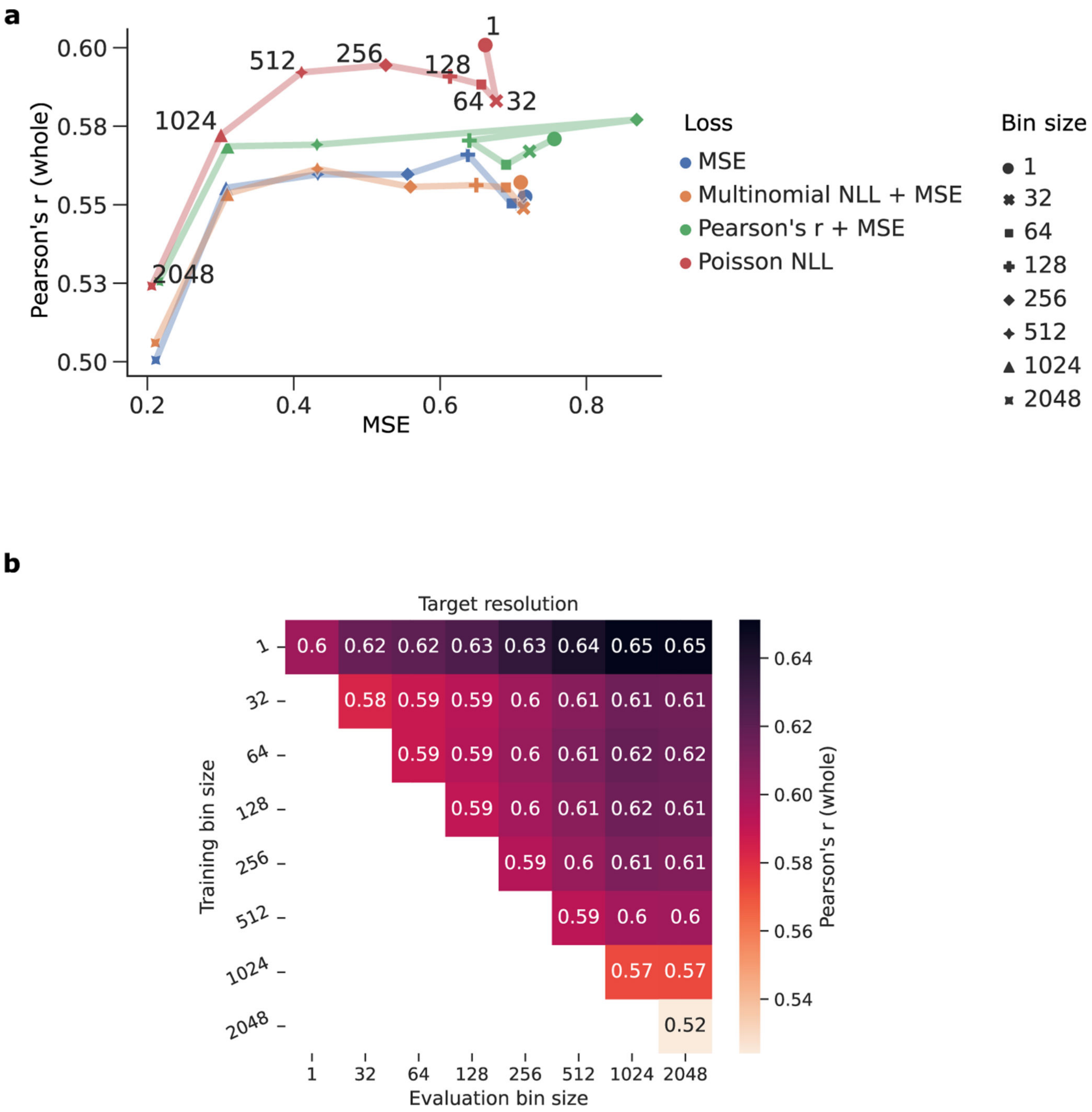
GIA for flanking nucleotides.: We fixed the core motif at the center of all background sequences, i.e. starting at position 1024, and varied the 2 flanking nucleotides on each side (and the central nucleotide for only AP-1) by separately performing a GIA experiment for all possible combinations of flanking nucleotides.

GIA for distance-dependent motif interactions.: To quantify the functional dependence of the distance between 2 motifs with optimized flanks, we fixed the position of 1 motif at the center of the sequence, i.e. starting at position 1024, and then systematically performed a GIA experiment with the second motif at different locations ensuring no overlap. This experiment provides a global importance score for the 2 motifs at different distances in both positive and negative directions.

GIA for motif cooperativity.: To quantify whether motifs are cooperatively interacting, we inserted each motif (with optimized flanks) at the corresponding position (1024 for motif 1 and best position for interaction for motif 2 based on the distance-dependent GIA experiments) individually and in combinations. We then compared the global importance when both motifs are embedded in the same sequence versus the sum of the global importance when only one motif is embedded.

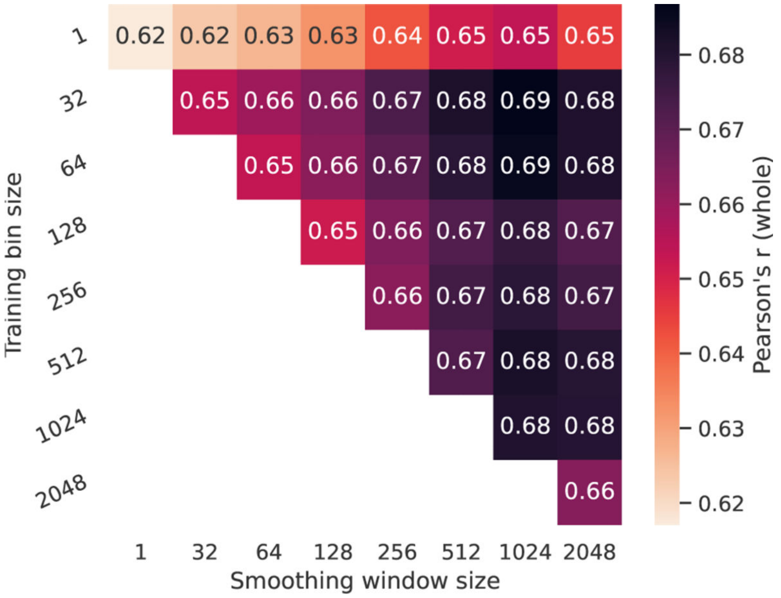
Occlusion-based experiments.—We randomly sampled 10,000 sequences from those within our coverage-threshold test set. We performed a string search looking for exact matches to the core motif of AP-1, i.e. TGA-TCA, where the - can be any nucleotide. For each cell-type, we grouped the sequences according to the number of instances that the core AP-1 motif was observed – 1 observed motif, 2 observed motifs, and 3 or more observed motifs. For each group, we replaced the core motif with randomized sequences. Due to spurious patterns from randomized sequences, we performed a GIA experiment where 25 randomized sequences were embedded in lieu of the core binding site and the model predictions were averaged – first across the coverage for the cell-type under investigation, then across the 25 randomized sequences. This effectively marginalizes out the impact of the motif for a given sequence. This occlusion-based (or conditional) GIA experiment was done for each sequence in each group.

Extended Data



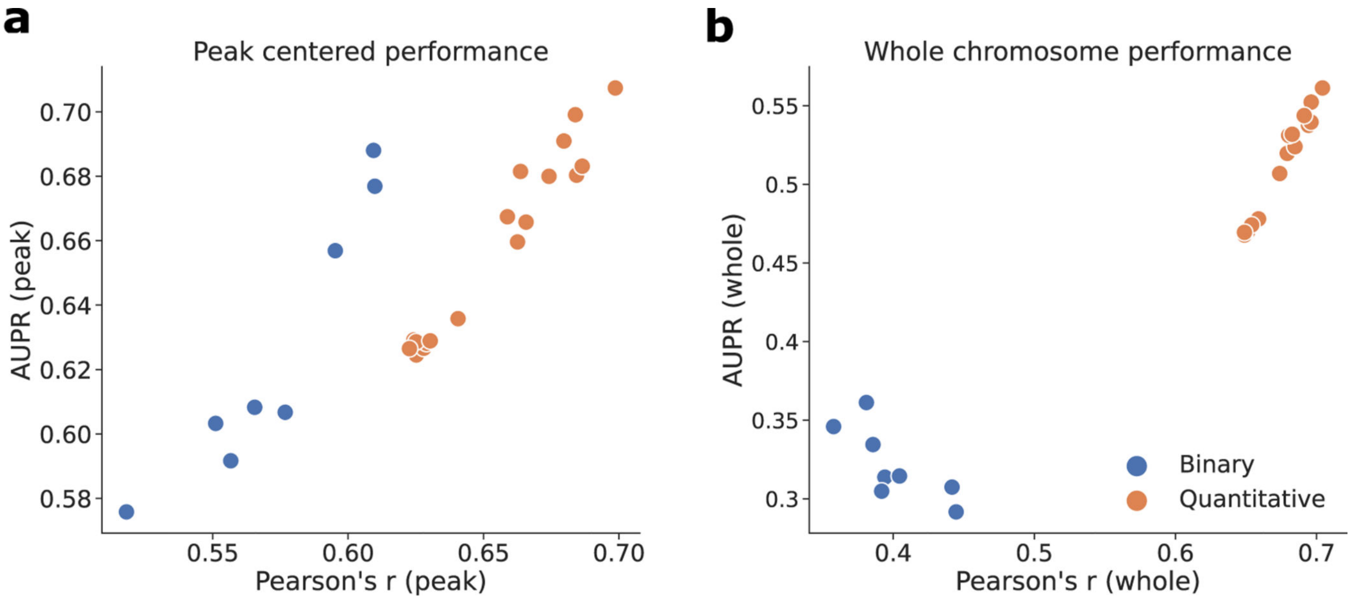
Extended Data Figure 1.

Evaluation of BPNet-based quantitative models. (**a**) *Loss function analysis*. Scatter plot of the whole-chromosome Pearson's r versus the MSE for different loss functions (shown in a different color) and different target resolutions (shown in a different marker). The results for the scaled Pearson's r loss function was removed due to poor training runs. (**b**) *Bin resolution analysis*. Plot of the whole-chromosome Pearson's r for models trained on a given bin size (*y*-axis) with predictions that were systematically down-sampled to a lower resolution for evaluation (*x*-axis). (**a,b**) Pearson’s r represents the average across cell lines.



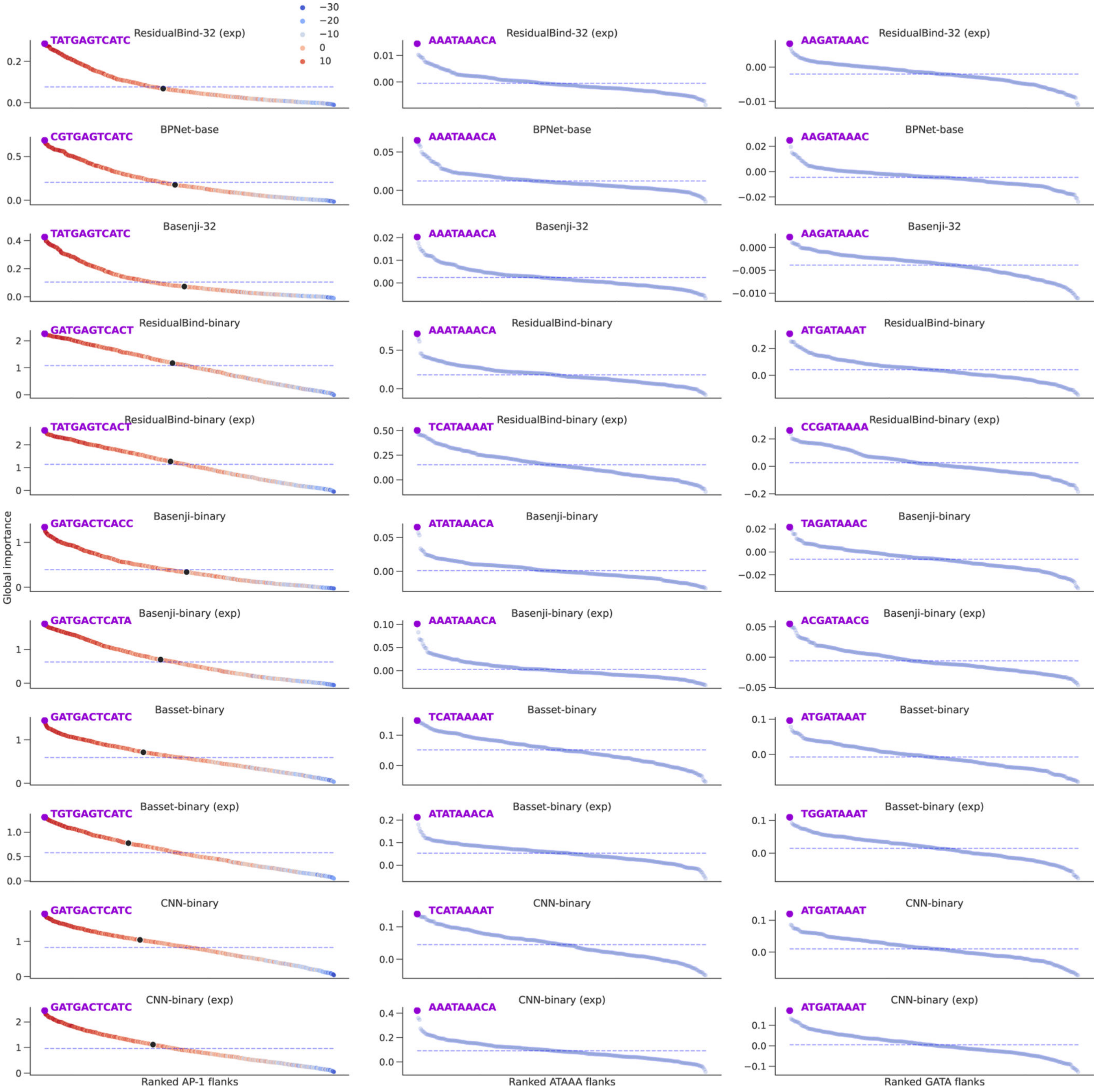
Extended Data Figure 2.

The effect of smoothing coverage on performance. Basenji-based models were trained on target resolutions (*y*-axis) and evaluated using different levels of smoothing with a box-car filter. For each higher resolution model, a box-car filter was applied to both predictions and experimental coverage values with various kernel sizes prior to calculating the average Pearson’s r (*x*-axis). Pearson’s r represents the average across cell lines.



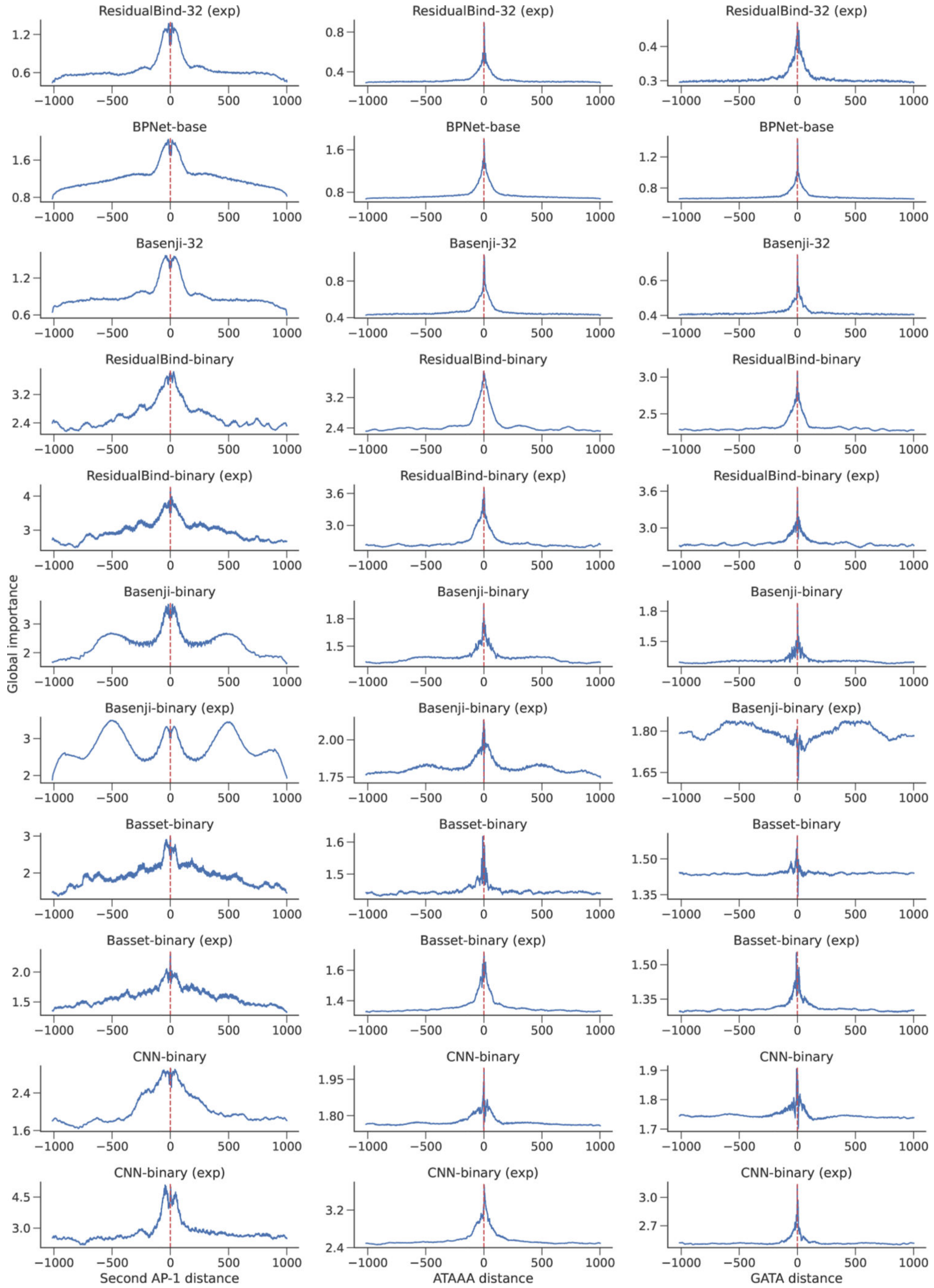
Extended Data Figure 3.

Performance comparison between quantitative and binary models. Scatter plot of the classification-based AUPR versus the regression-based Pearson’s r for various binary models (blue) and quantitative models (orange) on peak-centered test data (left) and whole-chromosome test data (right). Metrics represent the average across cell lines.



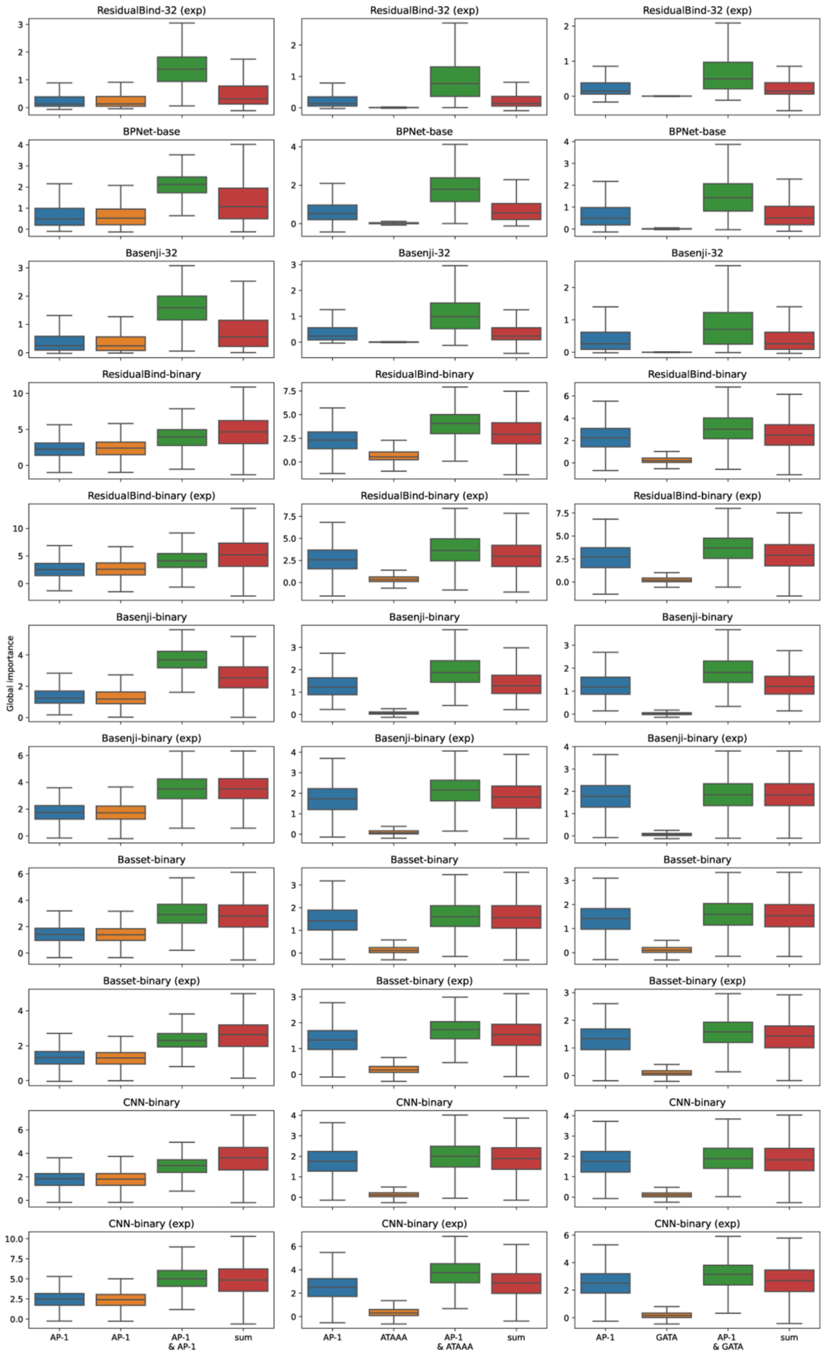
Extended Data Figure 4.

GIA for optimal flanking nucleotides of motifs in PC-3 cell line for various models. Ranked plot of the global importance for each tested flank for AP-1 motif (left column), ATAAA motif (middle column) and GATA (right column) for different models (shown in a different row). Dashed line represents the global importance of the core motif with random flanks. The hue in the first column represents the position-weight-matrix score for an AP-1 motif from the JASPAR database (ID: MA0491.1). The first 3 rows are quantitative models, the rest are binary models (with (exp) in the name indicating that the first layer ReLU activation has been replaced with an exponential function). For binary models, the results are based on the logits before the output sigmoid activation. The hue in the first column plots represents the PWM score for an AP-1 motif from the JASPAR database (ID: MA0491.1). The black dot in each plot (in the first column) indicates “TGTGATTCATG”, which has a high PWM score (12.800) but yields a global importance close to the core motif with randomized flanks.



Extended Data Figure 5.

GIA for distance dependence between AP-1 and other motifs for PC-3 cell line for various models. Global importance plot for sequences with an AP-1 motif fixed at the center of the sequence and another motif that is systematically placed in different locations. Positive and negative values represent the first positions the motifs w/ optimized flanks were embedded to be non-overlapping. First column shows results where the second motif is an identical AP-1 motif, the center column shows results for ATAAA motif and right column for the GATA motif. All the motifs were embedded with optimized flanks. Red vertical dashed lines indicate the 1024bp position. Each row corresponds to a different trained model, the first 3 are quantitative models, the rest are binary models (with (exp) in the name indicating that the first layer ReLU activation has been replaced with an exponential function). For binary models, the results are based on the logits before the output sigmoid activation.



Extended Data Figure 6.

GIA for cooperative interactions between AP-1 and other motifs for PC-3 cell line for various models. Each column corresponds to a motif pair between two copies of AP-1, ATAAA and AP-1 and AP-1 and GATA. Each row corresponds to a different trained model, the first 3 are quantitative models, the rest are binary models (with (exp) in the name indicating that the first layer ReLU activation has been replaced with an exponential function). For binary models, the results are based on the logits before the output sigmoid activation. Blue and orange box-plots show the global importance scores for the 1000 sampled sequences when motif 1 or motif 2 is individually embedded. Green box-plot shows the case when both motifs are embedded in the same sequence. Red box-plot shows the sum of the green and blue boxes as an estimate of the global importance if there is no interaction. The pairs were embedded at the optimal distance specified from the distance dependence GIA experiments. Box plots show the first and third quartiles, central line is the median, and the whiskers show the range of data with outliers removed. For each motif pair experiment n=1000 independent samples were drawn from the test set sequences.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This work was supported in part by funding from NIH grant R01HG012131 and the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory. This work was performed with assistance from the US National Institutes of Health Grants S10OD028632-01. The authors would like to thank Jakub Kaczmarzyk and other members of the Koo lab for helpful discussions.

Data availability

The processed ATAC-seq data, JASPAR 2022 core motifs for vertebrates dataset and CAGI5 challenge dataset used that support the findings of this study are available in Zenodo, https://doi.org/10.5281/zenodo.6464031 [58].

References

1. Quang D. & Xie X. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. Methods 166, 40–47 (2019). [PubMed: 30922998]
2. Li H, Quang D. & Guan Y. Anchor: trans-cell type prediction of transcription factor binding sites. Genome Res. 29, 281–292 (2019). [PubMed: 30567711]
3. Zheng A. et al. Deep neural networks identify sequence context features predictive of transcription factor binding. Nat. Mach. Intell 3, 172–180 (2021). [PubMed: 33796819]
4. Kelley DR, Snoek J. & Rinn JL Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 26, 990–999 (2016). [PubMed: 27197224]
5. Minnoye L. et al. Cross-species analysis of enhancer logic using deep learning. Genome Res. 30, 1815–1834 (2020). [PubMed: 32732264]
6. Zhou J. & Troyanskaya OG Predicting effects of noncoding variants with deep learning–based sequence model. Nat. Methods 12, 931–934 (2015). [PubMed: 26301843]
7. Yin Q, Wu M, Liu Q, Lv H. & Jiang R. DeepHistone: a deep learning approach to predicting histone modifications. BMC Genomics 20 (2019).
8. Dey KK et al. Evaluating the informativeness of deep learning annotations for human complex diseases. Nat. Commun 11, 1–9 (2020). [PubMed: 31911652]
9. Cheng J, Çelik MH, Kundaje A. & Gagneur J. Mtsplice predicts effects of genetic variants on tissue-specific splicing. Genome Biol. 22, 1–19 (2021). [PubMed: 33397451]

10. Zhou J. et al. Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. Nat. Genet 51, 973–980 (2019). [PubMed: 31133750]
11. Park CY et al. Genome-wide landscape of rna-binding protein target site dysregulation reveals a major impact on psychiatric disorder risk. Nat. Genet 53, 166–173 (2021). [PubMed: 33462483]
12. 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). [PubMed: 30013180]
13. Kim DS et al. The dynamic, combinatorial cis-regulatory lexicon of epidermal differentiation. Nat. Genet 53, 1564–1576 (2021). [PubMed: 34650237]
14. Novakovsky G, Saraswat M, Fornes O, Mostafavi S. & Wasserman WW Biologically relevant transfer learning improves transcription factor binding prediction. Genome Biol. 22, 1–25 (2021). [PubMed: 33397451]
15. Atak ZK et al. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. Genome Res. 31, 1082–1096 (2021). [PubMed: 33832990]
16. Li J, Pu Y, Tang J, Zou Q. & Guo F. DeepATT: a hybrid category attention neural network for identifying functional effects of dna sequences. Briefings Bioinforma. 22, bbaa159 (2021).
17. Karbalayghareh A, Sahin M. & Leslie CS Chromatin interaction–aware gene regulatory modeling with graph attention networks. Genome Res. 32, 930–944 (2022). [PubMed: 35396274]
18. Chen KM, Wong AK, Troyanskaya OG & Zhou J. A sequence-based global map of regulatory activity for deciphering human genetics. Nat. Genet 1–10 (2022). [PubMed: 35022602]
19. Janssens J. et al. Decoding gene regulation in the fly brain. Nature 1–7 (2022).
20. Vaishnav ED et al. The evolution, evolvability and engineering of gene regulatory dna. Nature 1–9 (2022).
21. Zhou J. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. Nat. Genet 54, 725–734 (2022). [PubMed: 35551308]
22. Kelley DR et al. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 28, 739–750 (2018). [PubMed: 29588361]
23. Kelley DR Cross-species regulatory sequence activity prediction. PLoS Comput. Biol 16, e1008050 (2020).
24. Maslova A. et al. Deep learning of immune cell differentiation. Proc. Natl. Acad. Sci 117, 25655–25666 (2020).
25. Avsec Ž et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat. Genet 53, 354–366 (2021). [PubMed: 33603233]
26. Avsec Ž et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat. Methods 18, 1196–1203 (2021). [PubMed: 34608324]
27. Koohy H, Down TA, Spivakov M. & Hubbard T. A comparison of peak callers used for DNase-seq data. PLoS ONE 9, e96303 (2014).
28. Madry A, Makelov A, Schmidt L, Tsipras D. & Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv 1706.06083 (2017).
29. Cohen J, Rosenfeld E. & Kolter Z. Certified adversarial robustness via randomized smoothing. In International Conference on Machine Learning, 1310–1320 (PMLR, 2019).
30. Goodfellow IJ, Shlens J. & Szegedy C. Explaining and harnessing adversarial examples. arXiv 1412.6572 (2014).
31. Yu F. & Koltun V. Multi-scale context aggregation by dilated convolutions. arXiv 1511.07122 (2015).
32. Koo PK, Majdandzic A, Ploenzke M, Anand P. & Paul SB Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. PLoS Comput. Biol 17, e1008925 (2021).
33. Kircher M. et al. Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. Nat. Commun 10, 1–15 (2019). [PubMed: 30602773]
34. 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). [PubMed: 31106481]
35. Koo PK & Eddy SR Representation learning of genomic sequence motifs with convolutional neural networks. PLoS Comput. Biol 15, e1007560 (2019).

36. Koo PK & Ploenzke M. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. Nat. Mach. Intell 3, 258–266 (2021). [PubMed: 34322657]
37. Ghotra R, Lee NK, Tripathy R. & Koo PK Designing interpretable convolution-based hybrid networks for genomics. bioRxiv (2021).
38. Alipanahi B, Delong A, Weirauch MT & Frey BJ Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat. Biotechnol 33, 831–838 (2015). [PubMed: 26213851]
39. Mauduit D. et al. Analysis of long and short enhancers in melanoma cell states. Elife 10, e71735 (2021).
40. de Almeida BP, Reiter F, Pagani M. & Stark A. Deepstarr predicts enhancer activity from dna sequence and enables the de novo design of synthetic enhancers. Nat. Genet 54, 613–624 (2022). [PubMed: 35551305]
41. Stormo GD, Schneider TD, Gold L. & Ehrenfeucht A. Use of the 'perceptron'algorithm to distinguish translational initiation sites in e. coli. Nucleic Acids Res. 10, 2997–3011 (1982). [PubMed: 7048259]
42. Le DD et al. Comprehensive, high-resolution binding energy landscapes reveal context dependencies of transcription factor binding. Proc. Natl. Acad. Sci 115, E3702–E3711 (2018).
43. Levo M. et al. Unraveling determinants of transcription factor binding outside the core binding site. Genome Res. 25, 1018–1029 (2015). [PubMed: 25762553]
44. Kodzius R. et al. CAGE: cap analysis of gene expression. Nat. Methods 3, 211–222 (2006). [PubMed: 16489339]
45. Vaswani A. et al. Attention is all you need. Adv. Neural Inf. Process. Syst 30 (2017).
46. Buenrostro JD, Wu B, Chang HY & Greenleaf WJ ATAC-seq: a method for assaying chromatin accessibility genome-wide. Curr. Protoc. Mol. Biol 109, 21–29 (2015).
47. An integrated encyclopedia of DNA elements in the human genome. Nature 489, 57–74 (2012). [PubMed: 22955616]
48. Yu F, Koltun V. & Funkhouser T. Dilated residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 472–480 (2017).
49. Hendrycks D. & Gimpel K. Gaussian error linear units (GeLUs). arXiv 1606.08415 (2016).
50. Ioffe S. & Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, 448–456 (2015).
51. Kingma D. & Ba J. Adam: A method for stochastic optimization. arXiv 1412.6980 (2014).
52. Biewald L. Experiment tracking with weights and biases (2020). Software available from wandb.com.
53. He K, Zhang X, Ren S. & Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778 (2016).
54. Gupta S, Stamatoyannopoulos JA, Bailey TL & Noble WS Quantifying similarity between motifs. Genome Biol. 8, 1–9 (2007).
55. Castro-Mondragon JA et al. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. Nucleic Acids Res. 50, D165–D173 (2021).
56. Simonyan K, Vedaldi A. & Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv 1312.6034 (2013).
57. Tareen A. & Kinney JB Logomaker: beautiful sequence logos in python. Bioinformatics 36, 2272–2274 (2020). [PubMed: 31821414]
58. Toneyan S, Tang Z. & Koo P. Evaluating deep learning for predicting epigenomic profiles, DOI: 10.5281/zenodo.6464031 (2022).
59. Toneyan S, Tang Z. & Kaczmarzyk J. shtoneyan/gopher: stable, DOI: 10.5281/zenodo.6977213 (2022).

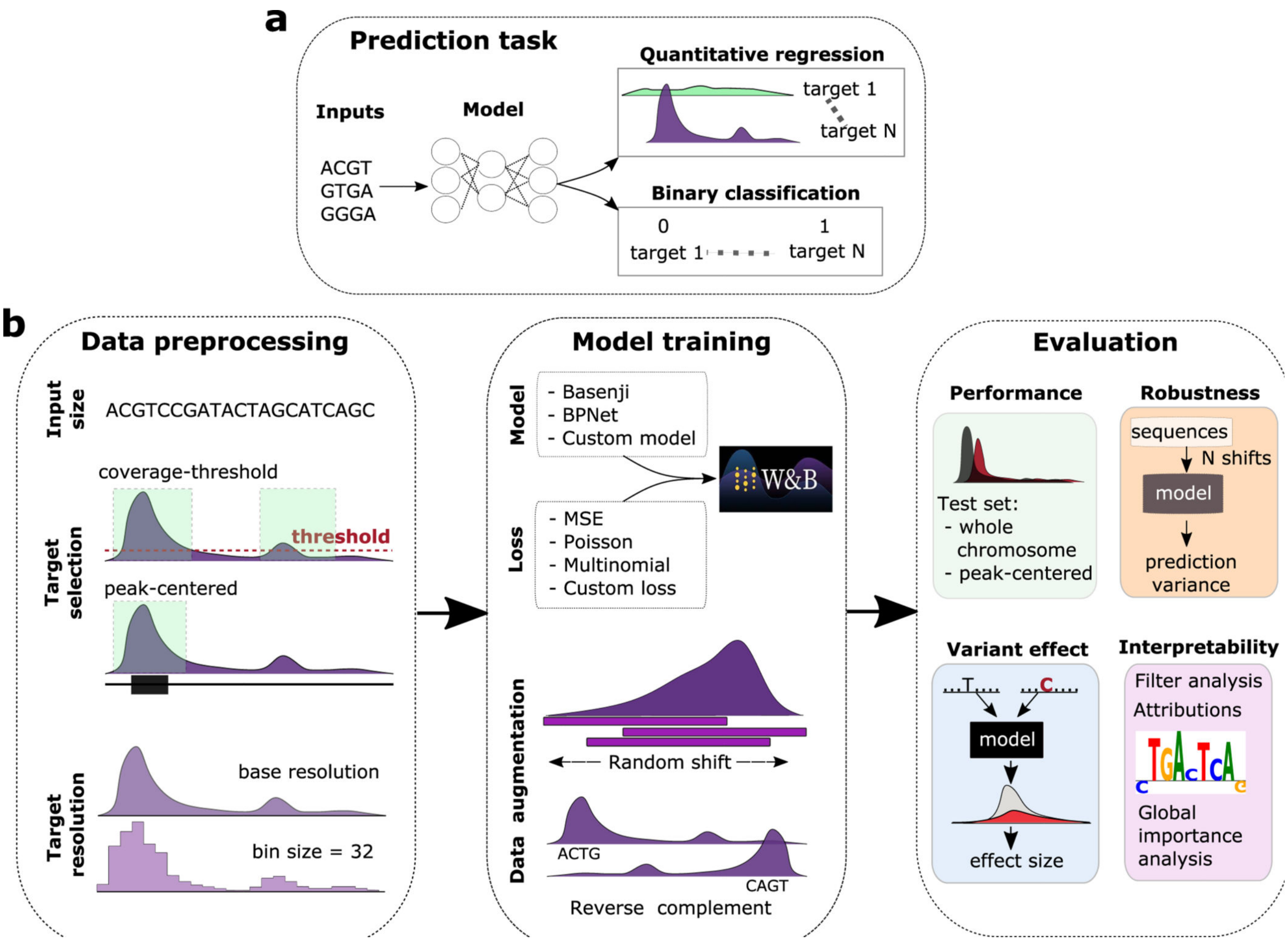


Figure 1.

GOPHER overview. (**a**) Comparison of binary and quantitative prediction tasks for regulatory genomics. (**b**) Illustration of the 3 main components of DL analysis: data preprocessing (i.e. input size, target selection and resolution), model training (i.e. model architecture, loss and data augmentations) and evaluation (i.e. generalization performance, robustness, interpretability and variant effect predictions).

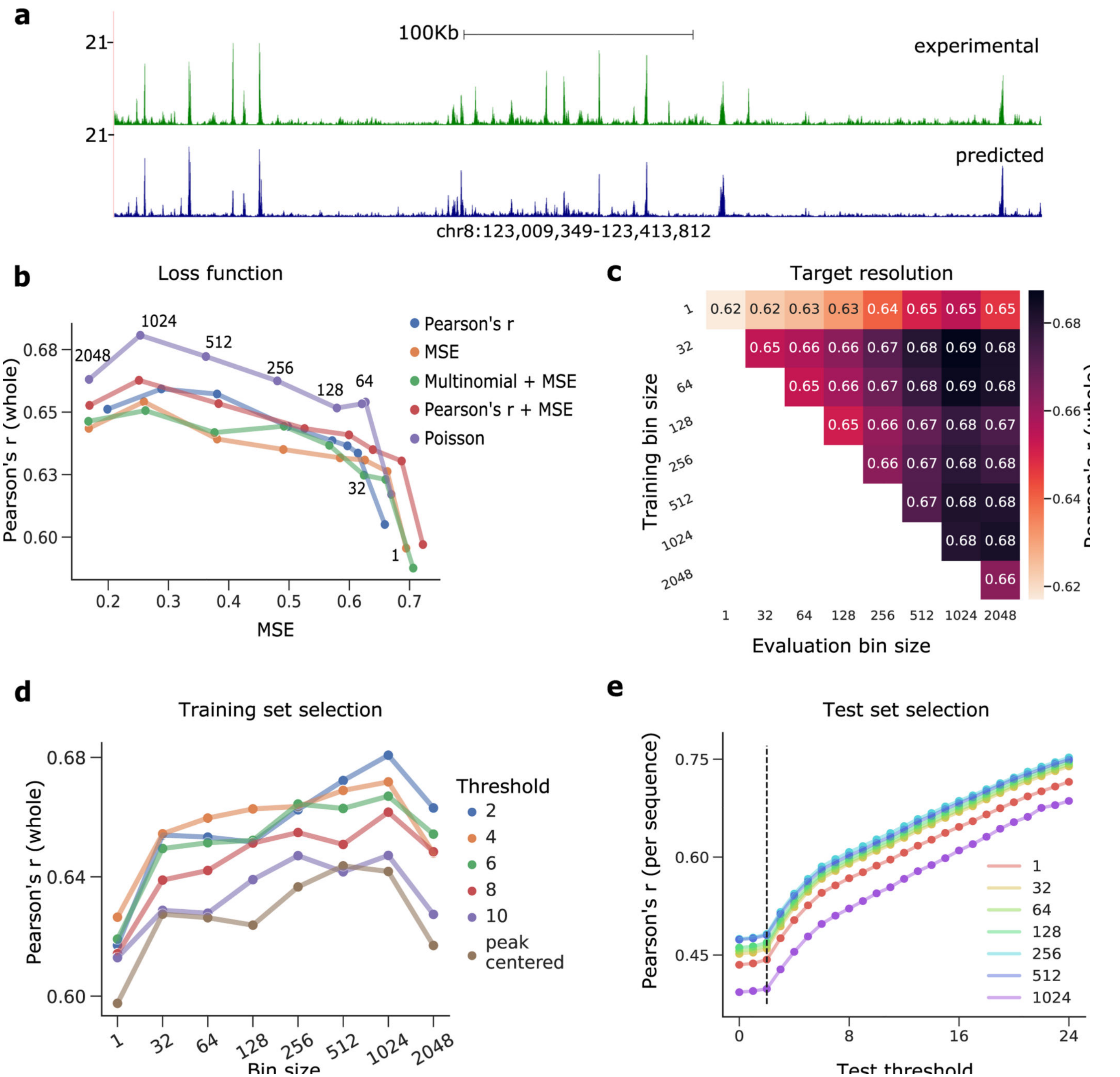


Figure 2.

Evaluation of Basenji-based quantitative models. (**a**) Example visualization of bigWig tracks for experimental measurements (top) and model predictions (bottom) for a given target cell line (GM23338) on a held-out test chromosome for Basenji-128. (**b**) *Loss function analysis*. Scatter plot of the whole-chromosome Pearson’s r versus the MSE for different loss functions (shown in a different color) and target resolutions (annotated bin sizes). Predictions were scaled for all models using ground truth mean coverage (see Methods). Lines serve as a guide-to-the-eye. (**c**) *Target resolution analysis*. Heatmap of the whole-chromosome Pearson’s r for models trained on a given bin size (*y*-axis) with predictions that were systematically down-sampled to a lower resolution for evaluation (*x*-axis). (**d**) *Training set selection analysis*. Scatter plot of whole-chromosome Pearson's r versus different target resolutions (i.e. bin size) for Baenji-based models trained on datasets with a different coverage threshold applied to the training set (shown in a different color). Peak-centered represents when the data is trained only on genomic regions identified as a peak. (**e**) *Test set selection analysis*. Scatter plot of the thresholded Pearson's r, which is average of per sequence correlation in the thresholded test set, versus different coverage thresholds applied to the test set for various Basenji-based models (of different resolutions) trained on default data, i.e. coverage-threshold data with a threshold of 2 (indicated with the black dotted vertical line). (**b-e**) Pearson's r represents the average across cell lines.

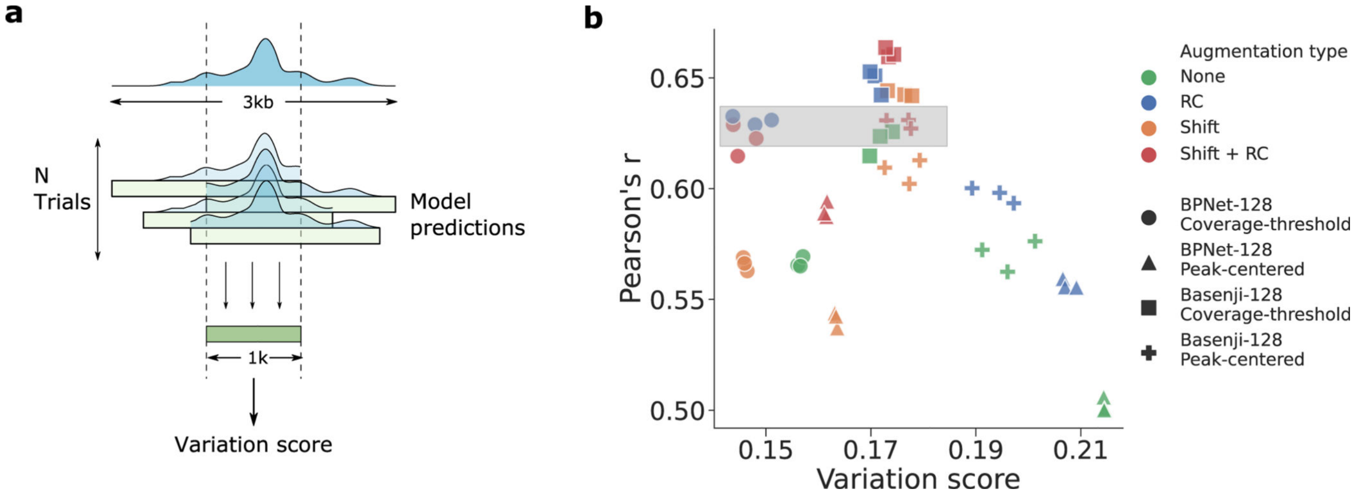


Figure 3.

Testing model robustness against translational shifts. (**a**) Schematic overview of robustness test. For each 3 kb sequence, N random 2 kb sub-sequences were extracted, and the standard deviation across predictions within the overlapping regions is calculated. Average variation score of predictions (i.e. average per-position standard deviation of coverage values normalized by the total mean coverage value) was used as a measure of model robustness. (**b**) Scatter plot of the Pearson's r (averaged across a per-sequence analysis) versus the robustness variation score across models with different augmentation methods (shown in a different color). Each 128 bin-resolution model (shown in a different marker) was trained 3 times with different random initializations. Pearson's r represents the average across cell lines.

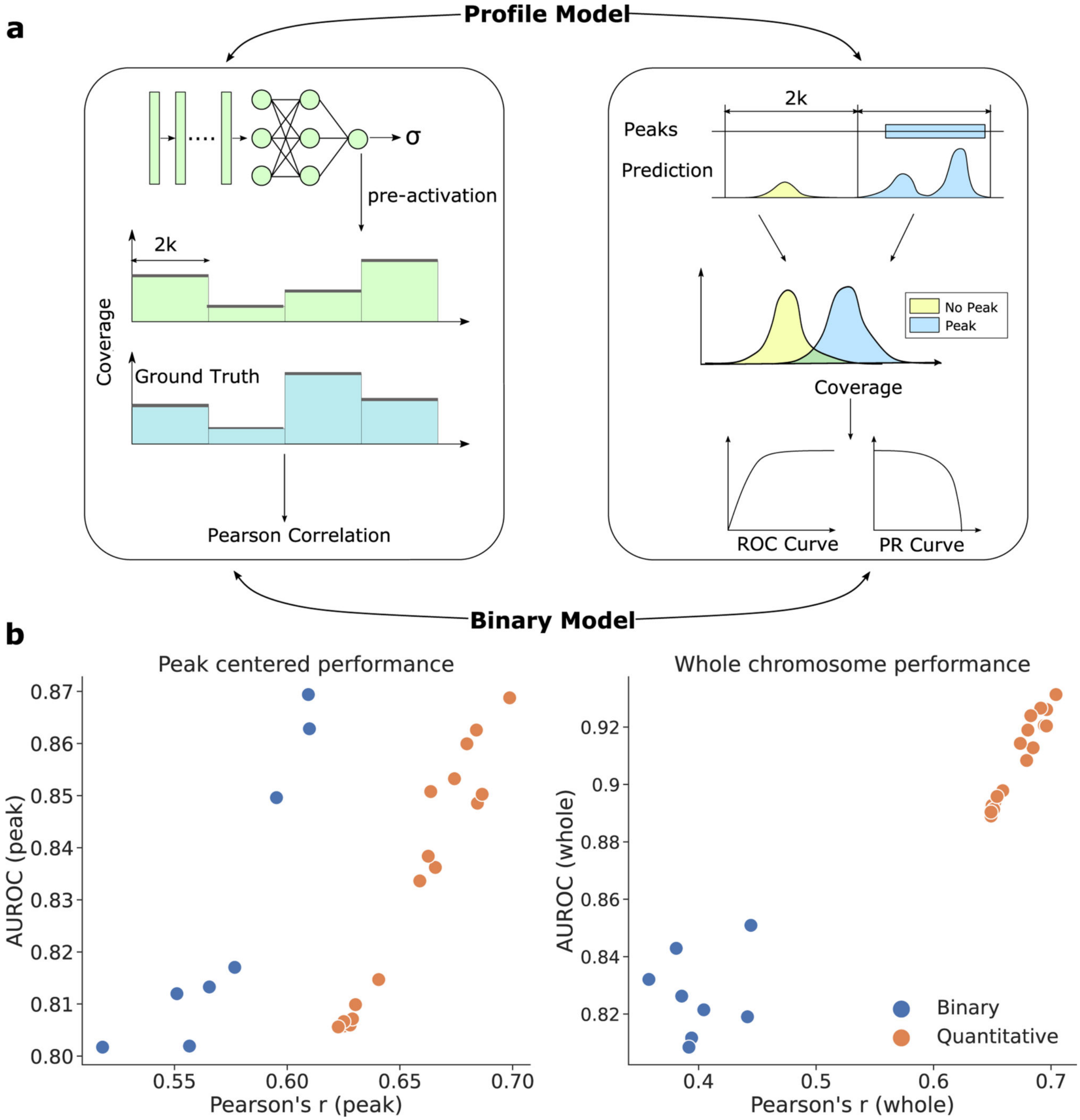


Figure 4.

Performance comparison between binary and quantitative models. (**a**) Schematic overview of prediction task conversion. Binary models can use logits to generate continuous ‘coverage-like’ values to calculate regression metrics. On the other hand, the coverage predictions of quantitative models can be grouped according to binary labels (i.e. peak and no peak groups) to calculate standard classification metrics. (**b**) Scatter plot of the classification-based AUROC versus the regression-based Pearson’s r for various binary models (blue) and quantitative models (orange) on peak-centered test data (left) and whole-chromosome test data (right). Metrics represent the averaged value across cell lines.

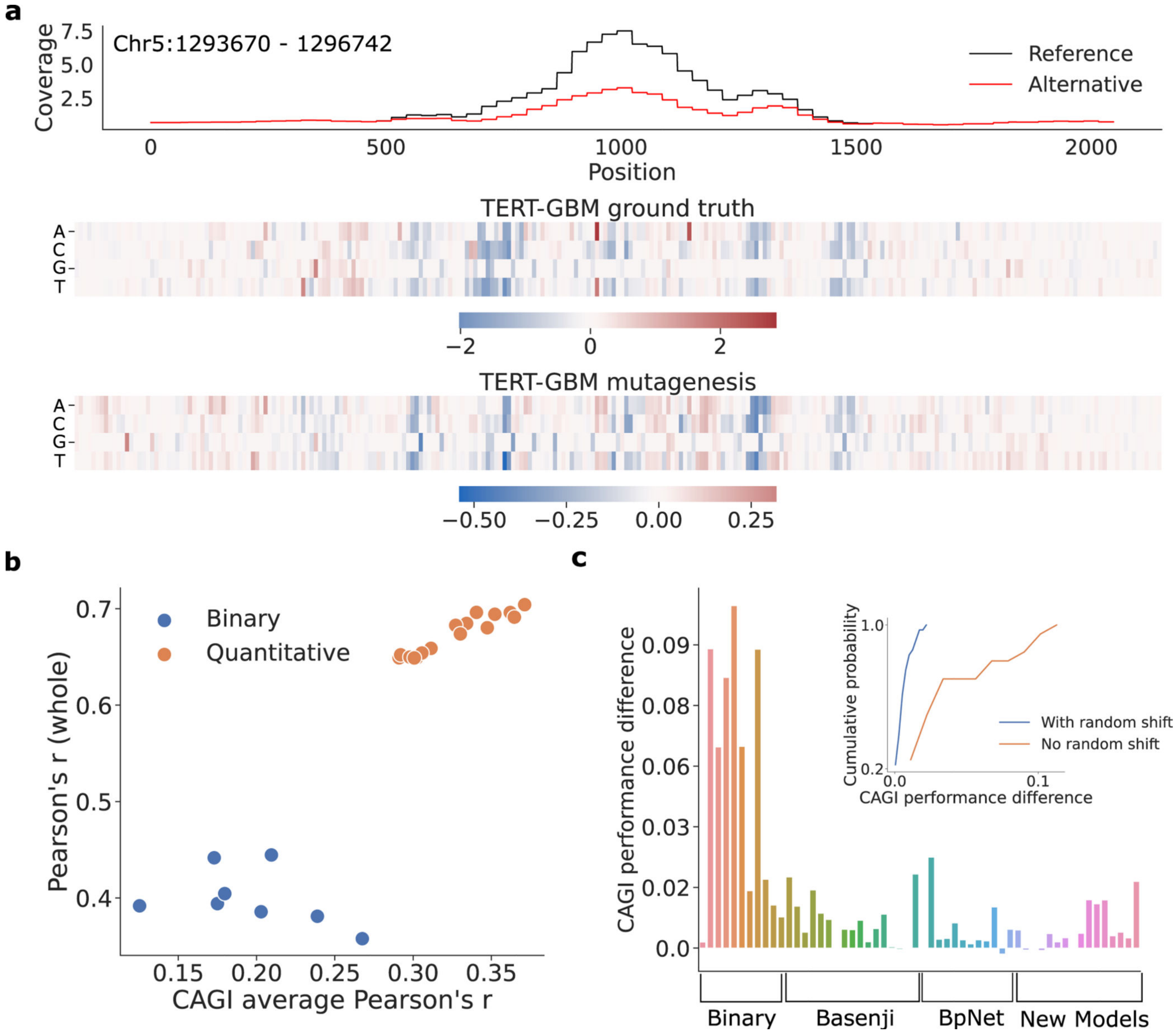


Figure 5.

Comparison of functional effect predictions. (**a**) Example visualization of predictions of a sequence with a reference allele (black curve) and an alternative allele (red curve) for a given mutation. Below, heat maps show the experimental measurements of variant effects for the TERT promoter in a GBM cell line (ground truth) and the predicted variant effects from ResidualBind-32. (**b**) Scatter plot of the prediction performance across whole-chromosome test set (*y*-axis) and the average CAGI5 performance (*x*-axis). Each dot represents a different model. (**c**) Bar plot shows the CAGI5 performance difference between robust predictions minus standard predictions. Each bar represents a different model. Groups of models represent different training strategies or target resolutions. Inset shows the cumulative distribution of variant effect performance differences for models trained with and without random shift data augmentation.

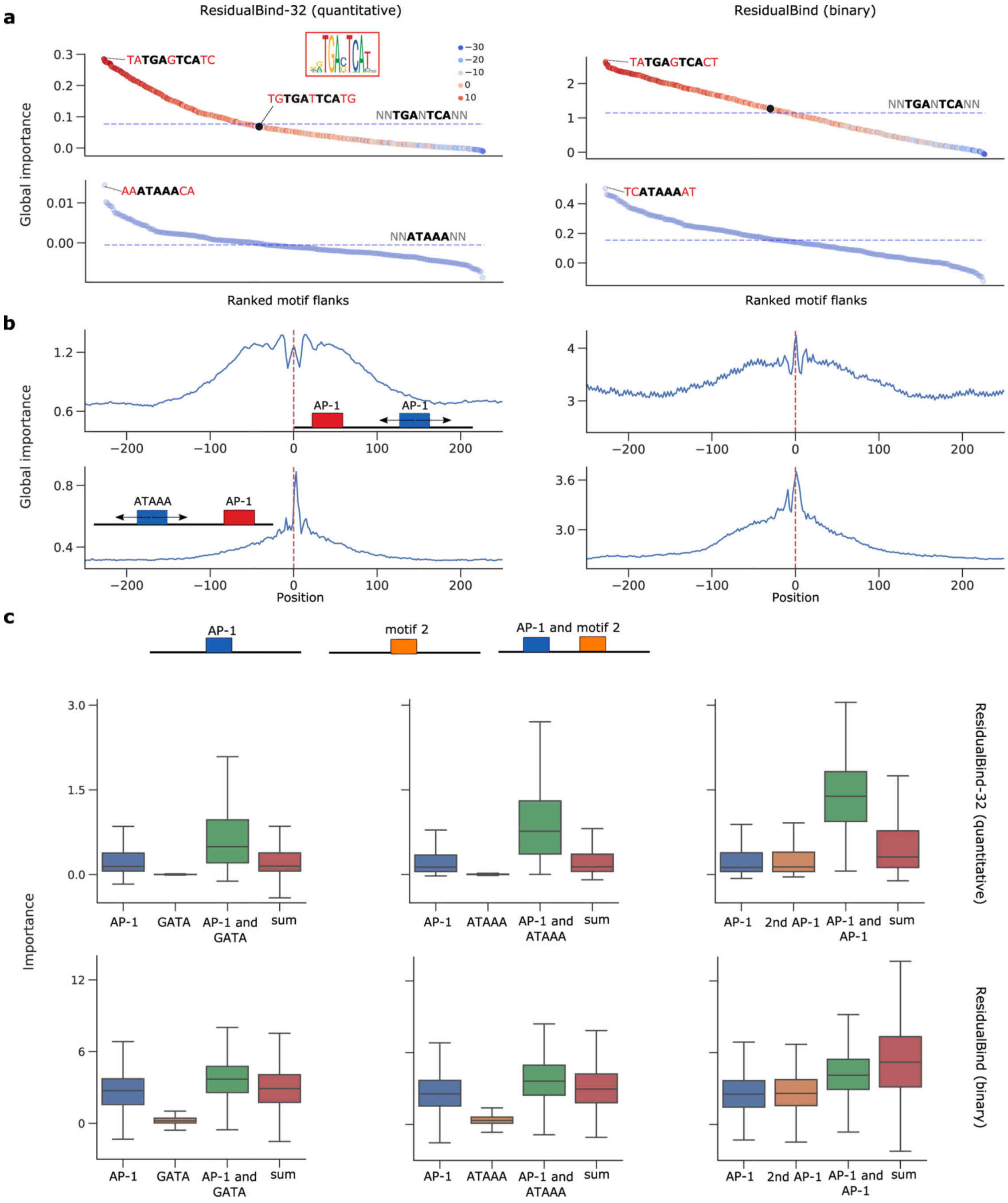


Figure 6.

GIA for ResidualBind-32 on PC-3 cell line. GIA for ResidualBind-32 on PC-3 cell line. (**a**) Ranked plot of global importance for each flank. Dashed line is global importance with random flanks. The hue represents the AP-1 position-weight-matrix score (JASPAR ID: MA0491.1). The black dot (‘TGTGATTCATG’) has a high position-weight-matrix score but yields a global importance close to the core motif. (**b**) Distance between two motifs. Global importance plot for sequences with AP-1 motif at the center and another motif placed in different locations. Positive and negative values represent the offset from the central motif. (**c**) Cooperative interactions between AP-1 and another motif. Boxplots of importance for sequences with motifs embedded individually and in combinations. The sum of each individual motif is also shown (sum). For each GIA experiment, *n* = 1000 independent samples were drawn from the test set. Box-plots show the first and third quartiles, central line is median, and whiskers show range of data with outliers removed.