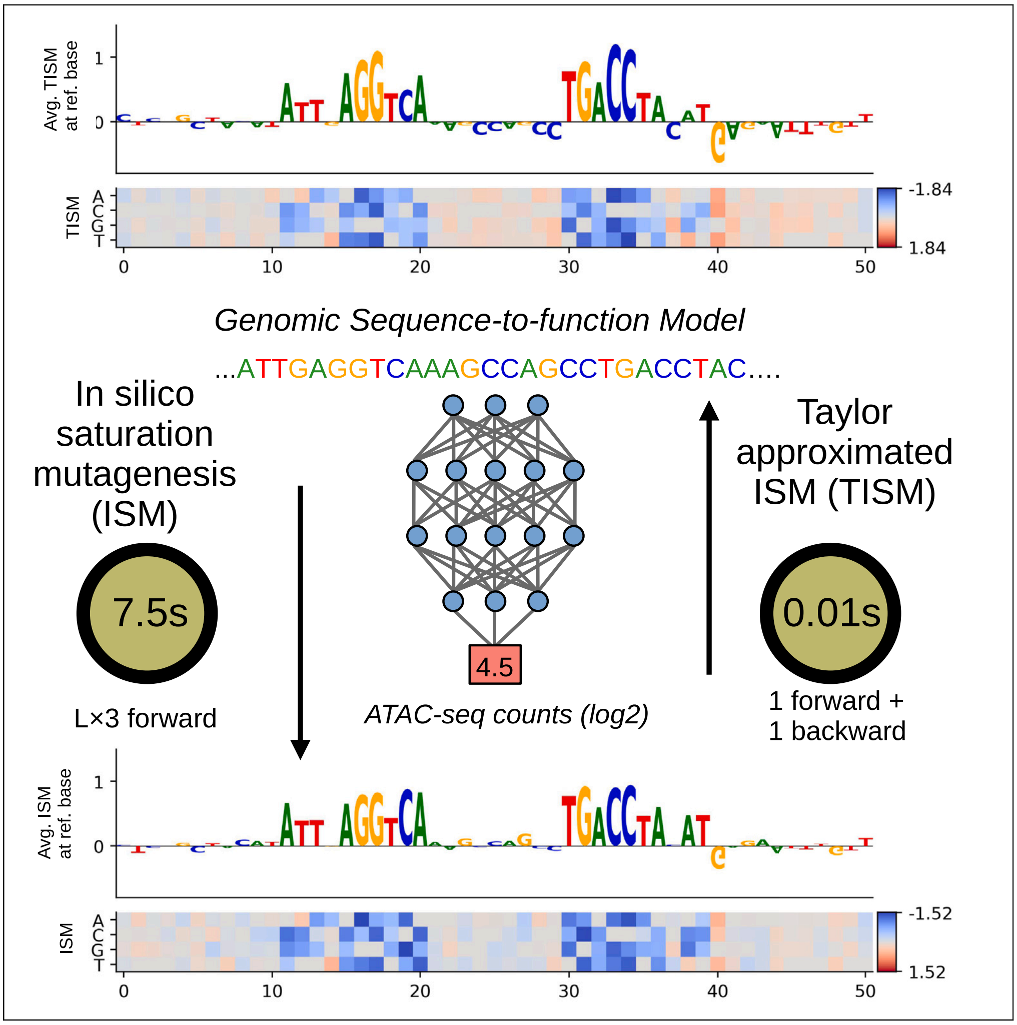
iScience

CellPress
OPEN ACCESS

Article

Quick and effective approximation of in silico saturation mutagenesis experiments with first-order taylor expansion



Alexander Sasse, Maria Chikina, Sara Mostafavi

mchikina@pitt.edu (M.C.)
saramos@cs.washington.edu (S.M.)

Highlights

In silico saturation mutagenesis (ISM) can be approximated from the model's gradient

Taylor approximated ISM improves run times proportional to the length of the sequence

TISM's similarity to ISM is within the range of ISM profiles from different models

TISM is robust to model architectures and training parameters

Sasse et al., iScience 27, 110807
September 20, 2024 © 2024
Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2024.110807

CellPress
OPEN ACCESS

Article

Quick and effective approximation of in silico saturation mutagenesis experiments with first-order taylor expansion

Alexander Sasse,[1] Maria Chikina,[2,*] and Sara Mostafavi[1,3,4,*]

SUMMARY

To understand the decision process of genomic sequence-to-function models, explainable AI algorithms determine the importance of each nucleotide in a given input sequence to the model's predictions and enable discovery of *cis*-regulatory motifs for gene regulation. The most commonly applied method is *in silico* saturation mutagenesis (ISM) because its per-nucleotide importance scores can be intuitively understood as the computational counterpart to *in vivo* saturation mutagenesis experiments. While ISM is highly interpretable, it is computationally challenging to perform for many sequences, and becomes prohibitive as the length of the input sequences and size of the model grows. Here, we use the first-order Taylor approximation to approximate ISM values from the model's gradient, which reduces its computation cost to a single forward pass for an input sequence. We show that the Taylor ISM (TISM) approximation is robust across different model ablations, random initializations, training parameters, and dataset sizes.

INTRODUCTION

Deep learning models have become the preferred tool to analyze the relationship between genomic sequence and genome-wide experimental measurements such as chromatin accessibility,[1,2] gene expression,[3–5] 3D chromatin conformation,[6–8] and other molecular data modalities.[9–11] To understand the models' decision processes, and extract the learnt genomic features, various explainable AI algorithms have been developed.[12–14] These methods estimate the importance of each nucleotide in an input sequence to the model's predictions.

The most commonly used algorithm to interpret genomic sequence-to-function models is in *silico* saturation mutagenesis (ISM).[15] ISM is very straightforward to implement and biologically highly interpretable. It can be intuitively compared to performing *in vivo* saturation mutagenesis experiments,[16] as ISM computes the change in the model's prediction as a function of a change in a single nucleotide. More formally, given a trained sequence-to-function model, at every position l along an input sequence of length L the reference base b_0 is replaced by one of the other three alternative bases $b_v \in \{A,C,G,T \mid b_v \neq b_0\}$ one at a time, and the model's predictions on the alternative sequences recorded. Thus, to compute the ISM profile for a sequence of length L, three times L forward passes are required. The differences between the predictions of these variant sequences and the prediction from the "reference" (initial) sequence is then used to define the impact or importance of the reference base and each alternative base along the sequence.

It is hoped that when applied to increasingly accurate sequence-based deep learning models, ISM can aid in solving for a comprehensive *cis* regulatory grammar and in some cases replace laborious and expensive *in vivo* saturation mutagenesis experiments.[4,9,17] However, as the state-of-the-art models continue to model larger input sequences (e.g., >100Kb), it is becoming computationally prohibitive to apply ISM. Here, we study the effectiveness of a first-order Taylor approximation to compute ISM values using the model's gradient from a single forward pass for each sequence. We show that Taylor approximated ISM (TISM) approximations speed up computations of ISM values by a factor L times three divided by the batch size. TISM derived attribution maps highly resemble attribution maps from ISM, more than the models' gradient as it is. We also derive that TISM represents the theoretical link between a recently proposed correction of the model's gradient for investigation of genomic sequences and attribution maps from ISM. Importantly, we show that TISM values are robust across different models, random initializations, training parameters, and dataset sizes.

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA
[2]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 16354, USA
[3]Canadian Institute for Advanced Research, Toronto, ON MG5 1ZB, Canada
[4]Lead contact
*Correspondence: mchikina@pitt.edu (M.C.), saramos@cs.washington.edu (S.M.)
https://doi.org/10.1016/j.isci.2024.110807

RESULTS

Approximating ISM with the model's gradient

To approximate the value of a complex function *f* at input *s*, Taylor's approximation linearly decomposes the function value *f(s)* at *s* into the value of the function at nearby position s_0, given by $f(s_0)$, and the derivative of the function $\frac{df}{ds_0}$ at s_0 multiplied by the difference between the position of interest *s* and s_0.

$$f(s) = f(s_0) + \frac{df}{ds_0}(s - s_0) + \mathrm{O}(2) \tag{Equation 1}$$

Where O(2) represents the second order term that is truncated in the linear approximation. In the case of sequence-based deep learning models the input to the function *f* is a real numbered one-hot encoded sequence tensor of size L times the number of channels (e.g., four for DNA, i.e., A, C, G, T). Gradients for these models are automatically computed with numerical methods that are implemented in the specific deep learning libraries.[18] On the other hand, the gradient $\frac{df}{ds_0}$ can also be approximated by finite differences from the sequence of interest s_0 to a sequence with a single nucleotide substitution b_0 to b_1 at position *l*, denoted by $s_0(l, b_0 \to b_1)$.

$$f(s) \approx f(s_0) + \frac{\delta f}{\delta s_0}\Delta s \approx f(s_0) + \frac{f(s_0(l, b_0 \to b_1)) - f(s_0)}{s_0(l, b_0 \to b_1) - s_0}\Delta s \tag{Equation 2}$$

In the case where the finite distance δs to approximate the gradient is equal to the distance from the reference sequence Δs, the numerator and denominator cancel each other out and we are left with the ISM value given by Equation 3.

$$= f(s_0) + \frac{f(s_0(l, b_0 \to b_1)) - f(s_0)}{s_0(l, b_0 \to b_1) - s_0}(s_0(l, b_0 \to b_1) - s_0)$$

$$= f(s_0) + f(s_0(l, b_0 \to b_1)) - f(s_0) = f(s_0) + ISM(s_0, l, b_1) \tag{Equation 3}$$

Thus, Equation 3 shows that ISM values represent the effect from a linear approximation of the deep learning model *f* to a single base change. In practice, ISM values are used in two ways: (1) as a per-nucleotide value that indicates how much the prediction changes if the reference base is replaced by the specific variant; (2) as attribution maps which indicate how important each nucleotide is for a model's prediction. To generate attribution maps from ISM values, practitioners subtract the mean at each position *l* from the ISM values to get attributions per base-pair $A_{ISM}(s_0, l, b_v)$.

$$A_{ISM}(s_0, l, b_v) = ISM(s_0, l, b_v) - \frac{1}{4}\sum_{j=0}^{4} ISM(s_0, l, b_j) \tag{Equation 4}$$

Regulatory motifs are usually identified from the values of these attribution maps at the reference base. They represent how important the present nucleotide is for the model's predictions akin to common measures of per-nucleotide sequence conservation.

$$A_{ISM}(s_0, l, b_0) = ISM(s_0, l, b_0) - \frac{1}{4}\sum_{j=0}^{4} ISM(s_0, l, b_j) = 0 - \frac{1}{4}\sum_{j=0}^{4} ISM(s_0, l, b_j) \tag{Equation 5}$$

While ISM is easy to implement, it is computationally costly, and so users often resort to using "gradient-times-input" to indicate how important the given nucleotide is for a model's prediction.[4,9] Gradient-times-input is less computationally taxing because it uses a single pass through the model to simultaneously approximate the importance of every nucleotide in the input sequence. Specifically, during model training, the gradient with respect to the parameters is computed automatically in every forward pass to enable parameter updates with back-propagation. Therefore, the gradient with respect to the input is available for "free" from just a single forward pass through the network. For model interpretation, gradient-times-input simply uses the gradient at the reference base, indicating whether it is beneficial for the model to either "change," or keep the base at this position.

Here, we propose to instead use the gradient to approximate ISM using a first-order Taylor approximation. Equating *f(s)* in Equations 1 and 3, shows that ISM can be approximated from the model's gradient:

$$f(s_0) + ISM(s_0, l, b_1) \approx f(s_0) + \frac{df}{ds_0}s_0(l, b_0 \to b_1) - \frac{df}{ds_0}s_0 = f(s_0) + TISM(s_0, l, b_1) \tag{Equation 6}$$

where TISM denotes the first-order Taylor approximation to ISM. Applying this to a one-hot encoded input in which the reference base b_0 a position *l* is replaced (set b_0 from 1 to 0) by an alternative base b_1 (set b_1 from 0 to 1), we can see that ISM at l, b_1 is equal to the gradient with respect to the reference sequence s_0 at base l, b_1 minus the gradient at base l, b_0.

$$ISM(s_0, l, b_1) \approx \frac{df^{(l, b_1)}}{ds_0} - \frac{df^{(l, b_0)}}{ds_0} = TISM(s_0, l, b_1) \tag{Equation 7}$$

This relationship allows us to quickly approximate the per nucleotide ISM values from the gradient of the input sequence using only a single forward pass through the model. This is especially useful when applying ISM to long sequences, or for comparing regulatory motifs across many sequences. Distal regulatory elements are common in genomics and estimating the correct effect size is key to determine their impact on gene regulation.

While gradient times input estimates the importance of the reference base from the gradient at the reference base $\frac{df^{(l,b_0)}}{ds_0}$, attribution maps derived from TISM correctly add the effect from the alternative bases to the attribution of the reference at position *l*.

$$A_{TISM}(s_0, l, b_0) \; = \; TISM(s_0, l, b_0) - \frac{1}{4}\sum_{j=0}^{4} TISM(s_0, l, b_j)$$

$$= \; 0 - \frac{1}{4}\sum_{j=0}^{4} TISM(s_0, l, b_j) \; = \; \frac{df^{(l,b_0)}}{ds_0} - \frac{1}{4}\sum_{j=0}^{4}\frac{df^{(l,b_j)}}{ds_0} \qquad \text{(Equation 8)}$$

We note here that Majdandzic et al. recently proposed the same correction for gradient-based attribution maps from a geometrical approach.[19] Briefly, the authors suggested minimizing the impact of of-simplex gradient noise by removing the random orthogonal gradient component from the input gradient. Here, we showed that this correction represents an approximation of attribution maps that are derived from ISM values. In addition, we show how these values can be biologically interpreted as ISM values and how the gradients of the model are related to ISM values.

TISM effectively approximates ISM with massive speed ups

We used a modified version of our previously published model AI-TAC (see STAR Methods) and evaluated the concordance between ISM and TISM on the per nucleotide effects in each input sequence. The model takes a DNA sequence of 251 bp around the ATAC peak (open chromatin region [OCR]) as input and predicts the normalized accessibility (i.e., the logarithm of the number of Tn5 cuts within 250 bp around the ATAC peak, corrected for sequencing depth) of that peak across 81 different cell types in a multitask fashion. The model was trained on 286,000 OCRs and ISM and TISM values were computed for 9,158 OCR sequences for all 81 cell types (i.e., 741,798 attribution maps each). In all evaluations below, we solely use regions from chromosome 19 which were entirely left out during model training and validation.

In our baseline model, we observed an average correlation value of 0.7 between TISM and ISM values (Figure 1A). Encouragingly, 87% of TISM profiles computed from test regions had a correlation value of at least 0.6. Visual inspection confirmed the high concordance between ISM and TISM profiles across different cell types and suggests that both methods detect the same motifs and predict similar changes to their effect across cell types (Figure 1B). Next, we confirmed our theoretical derivation and compared TISM to the gradient as a popular alternative to ISM. TISM's correlations to ISM are consistently higher than those of the gradient itself (Figure 1C). We also compared concordance between the mean effect per base from TISM and ISM versus the concordance of the mean effect from ISM to gradient-times-input (Figure S1). While the correlation of the mean effect per base from TISM to ISM is also consistently higher, gradient times input's correlation to ISM is closer than the correlation of gradients across all four bases.

We examined the sequences with lower correlation between TISM and ISM and noted that for most part, these correspond to regions of low average predicted chromatin accessibility and high coefficients of variations of predicted counts (Figures 1D and S2A). We did not observe a relationship between model performance and ISM to TISM correlation (Figure S2B). When we measured the running times for both methods, we confirmed the theoretical speed up of TISM over ISM (Tables 1 and 2). When we measured the speedup for different numbers of sequences, TISM was on average ∼160 times faster than ISM (Table 1). This is consistent with theoretical values from using a batch size of 20 to compute the ISM values (3 times 1,000 bp divided by batch size of 20). TISM exerts its real value for long sequences, where its speedup improves from 25 times to 8,000 times for sequences of length 251 bp to sequences with length 20,000 bp (Table 2). We note that this speedup is 2.5 times larger than expected with a batch size of 20.

Additionally, we also compared the run times between TISM, ISM, and Yuzu[20] on two untrained model architectures (Figure 2). Yuzu uses compressed sensing to speed up computation of ISM values. Yuzu's ISM values are not an approximation and therefore they are identical to ISM values. However, Yuzu is not applicable to all network architectures, modules, or long sequences. While Yuzu is 10 times faster than ISM on a shallow model (3 conv. layers, Figure 2 top; Table S1 and S2), it is only five times faster on a deeper model (8 conv. layers, Figure 2 bottom; Tables S3 and S4). Yuzu improves its speed up over ISM for larger sequences (Tables S1 and S3). However, we were not able to run sequences longer than 2 kb on our GPU. Nevertheless, for all these tests, TISM massively outperforms Yuzu, making it especially valuable for exceedingly long, or large numbers of sequences. These gains also hold true for computations on the CPU (Figure S3).

TISM robustly approximates ISM values across models

To determine how robust TISM approximations are across models, we trained our baseline model from four random parameter initializations and computed the Pearson correlation between TISM and ISM profiles from all four models (Figure 3A). On average, the correlation of ISM profiles from two separate model runs is 0.58. TISM profiles behave similarly, with an average correlation of 0.52: However, TISM and ISM profiles correlate on average with Pearson R = 0.7 when they are from the same model, showing that TISM is more concordant with ISM than ISMs between different model trainings.

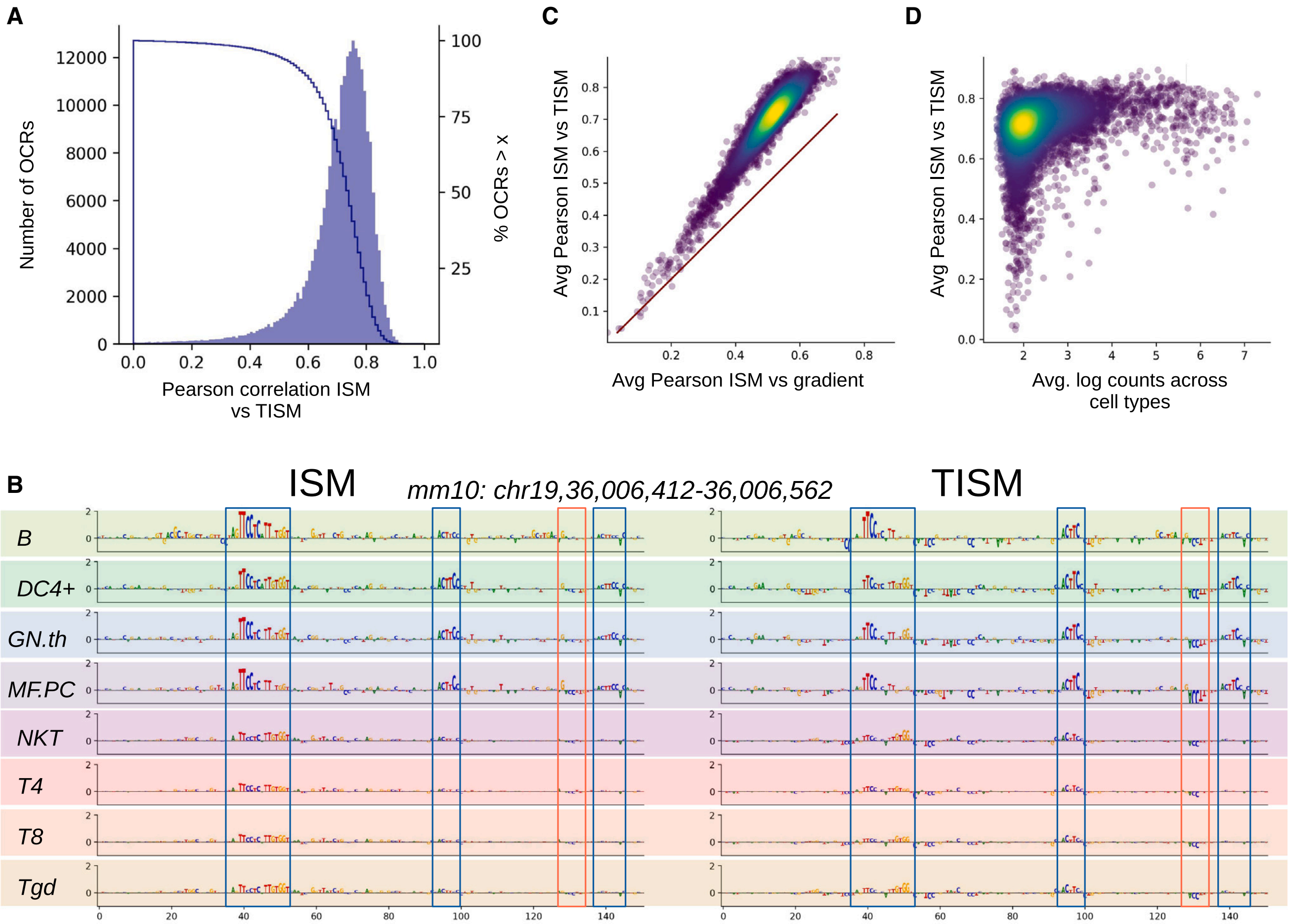


Figure 1. Comparison between ISM and Taylor approximated ISM

(A) Histogram and cumulative percentage of Pearson's correlation between ISM and TISM of 9158 chromatin regions across 81 cell types in the test set (Mean = 0.7, Median = 0.73).

(B) Attributions of reference base from ISM and TISM for "Atac.peak_251276" at chr19:36,006,362-36,006,612 across 8 cell types with differential chromatin accessibility. Selected peak's ISM and TISM correlate 0.75 across all 81 cell types. Blue squares indicate consistent motifs between TISM and ISM. Red squares indicate motifs that are inconsistent (here only present in TISM) (C) Average Pearson correlation of OCRs across all cell types between ISM and the Gradient (x axis) versus the average correlation between ISM and Taylor corrected gradient (TISM).

(D) Correlation between peaks' mean log accessibility and the correlation between ISM and TISM is R = 0.29.

Next, we examined the concordance between ISM and TISM profiles with decreasing size of training data. To do so, we sub-sampled OCRs to create a couple of different smaller training sets (1%, 5%, 10%, 20%, and 50%). For each training set size, we trained a model on the randomly selected subset of data points and evaluated the correlation between TISM and ISM on the same 3,000 peaks across 81 cell types (243,000 ISM profiles; Figure 3B). Surprisingly, we observe that the correlation between ISM and TISM is higher for models that were trained on a subset of the data points, with the largest concordance between the two at 5% of the data points, or 14,300 training data points. Simultaneously, the predictive performance of the model is decreasing as expected (Figure 3B bottom). From visual inspection, we observe that models trained on smaller datasets are missing regulatory motifs that are present when trained on larger datasets (Figure S4A, blue frames). Additionally, the TISMs from smaller datasets are also missing negative motifs that are not concordant with the ISM effects (Figures 1B and S4, red frames). We hypothesize that these discordant motifs in models trained on larger datasets are the result of non-linear effects that TISMs cannot account for by the first-order Taylor approximation, even within proximity of only single nucleotide change.

To investigate this further, we looked at the evolution of concordance between TISM and ISM during model training. We trained a model for 400 epochs and assessed the concordance between ISM and TISM after 1, 2, 3, 5, 7, 11, 20, 60, 100, and 400 epochs (Figure 3C). As expected, the test set performance (mean Pearson correlation R of OCRs across cell types) is increasing until 11 epochs and then slightly decreases afterward while the training performance continues to increase. We note that the model is not overfitting as strongly as we would expect normally. We assume that this is due to our architectural choices, the Pearson correlation loss across cell types and the random sequence shifting in particular. At the beginning of model training, we observe that the concordance between ISM and TISM profiles is similar

Table 1. Comparisons between run times of ISM and TISM for sequences of 1,000 bp

Number of sequences	*t(s)* ISM	*t(s)* TISM
10	22.41	0.15
100	320.01	2.84
500	1,483.57	8.83
1,000	2,382.91	10.63

For all experiments, a batch size of 20 was used to perform forward passes through the models. Run times to generate ISM and TISM values for 10, 100, 500, and 1,000 sequences of length 1,000 bp.

to our fully trained baseline model while its predictions are still random ($\text{mean}_{\text{Test}} = -0.02$, $\text{mean}_{\text{Train}} = 0.01$). Interestingly, the concordance between ISM and TISM increases after three epochs and reaches its optimum with a mean Pearson of 0.85 at epoch seven, when model performance is slightly less than optimal ($\text{mean}_{\text{Test}} = 0.37$). Once the model reaches its optimum performance in the test set at epoch 11 ($\text{mean}_{\text{Test}} = 0.41$) the concordance has decreased back to an average 0.71. Additional overfitting (epoch 400: $\text{mean}_{\text{Test}} = 0.34$, $\text{mean}_{\text{Train}} = 0.45$) does not affect the concordance between TISM and ISM (mean concordance 0.71). We hypothesize that the increase in correlation before reaching the optimal model performance is the result of the model learning linear relationships which the first-order approximation can well represent. However, afterward the model potentially starts learning non-linear effects that increase its performance but reduce the concordance between ISM and TISM.

Lastly, we trained twelve different models, each with a single ablation to the baseline model. We used these models to study the impact of model architecture on the accuracy of TISMs approximations (Figure 3D, STAR Methods). Most training and architectural choices did not affect the concordance between ISM and TISM (i.e., exponential activation, no dropout, L1 on sequence kernels, forward strand input only). While removing batch norm, and max pooling did not affect the performance of the model, both choices slightly decreased the concordance between TISM and ISM ($\text{mean}_{\text{NoBatchnorm}} = 0.68$, $\text{mean}_{\text{Maxpool}} = 0.66$). The shallow CNN *CNN0* performed slightly worse in performance but has the same mean concordance as the baseline model (Mean = 0.7).

We observed the worst concordance between ISM and TISM from a model that used ReLU activations across the network ($\text{mean}_{\text{ReLU}} = 0.61$), while this choice did not result in worse predictions. On the other hand, using AdamW (mean = 0.82), MSE loss (mean = 0.76), and no sequence shifting (mean = 0.74) during model training, resulted in worse performance but higher concordance between ISM and TISM. Visual inspection of ISM and TISM profiles generated by a model trained with AdamW suggest that the updates with weight decay led to smaller motif effects (Figures 1B and S4B, blue frames) and less varying gradients outside the well-defined motifs (Figures 1B and S4B, red frames). Since non-linear effects are rare in the data, weight decay, in addition to reducing the size of the linear effects, removes rare non-linear effects entirely, leading to more concordant TISM but less accurate predictions.

DISCUSSION

Here, we provide the theoretical link between ISM and gradient-based interpretation methods for sequence-to-function models, which we call Taylor approximated ISM (TISM). We use TISM to generate 741,798 sequence attribution maps of length 251 bp for 22 models and assess the concordance between the computationally efficient approximation and the directly computed values across sequences. We find that TISM is highly concordant with ISM (mean correlation ∼0.7, see Figures 1A and 1B). The motifs that appear in these attribution maps are highly similar (Figures 1B and S4). We showed that TISM and ISM values from the same model have a higher correlation value than ISM values between different model initializations, suggesting that TISM's approximations are within the uncertainty of over-parameterized deep neural networks. In fact, the majority of TISM (89%, >0.58) values correlates well above ISM values from different model initializations, suggesting that TISM is sufficient to understand the model's learned regulatory grammar and predict effects of sequence variants across different loci.

Concordance between the two gets worse for model architectures that use ReLU and max-pooling layers which potentially make it more challenging to accurately compute the model's gradients. Counterintuitively, we also observe that models trained on fewer data points, model architectures with worse predictive performance, or not fully trained models possess higher concordance between ISM and TISM.

Table 2. Comparisons between run times of ISM and TISM for 10 sequences of different lengths

Sequence length	*t(s)* ISM	*t(s)* TISM
251	3.29	0.13
1,000	22.41	0.15
5,000	238.41	0.23
20,000	6,126.65	0.75

For all experiments, a batch size of 20 was used to perform forward passes through the models. Run times of models that take as input sequences of length 251, 1,000, 5,000, and 20,000 bp to compute $n = 10$ ISM and TISM values.

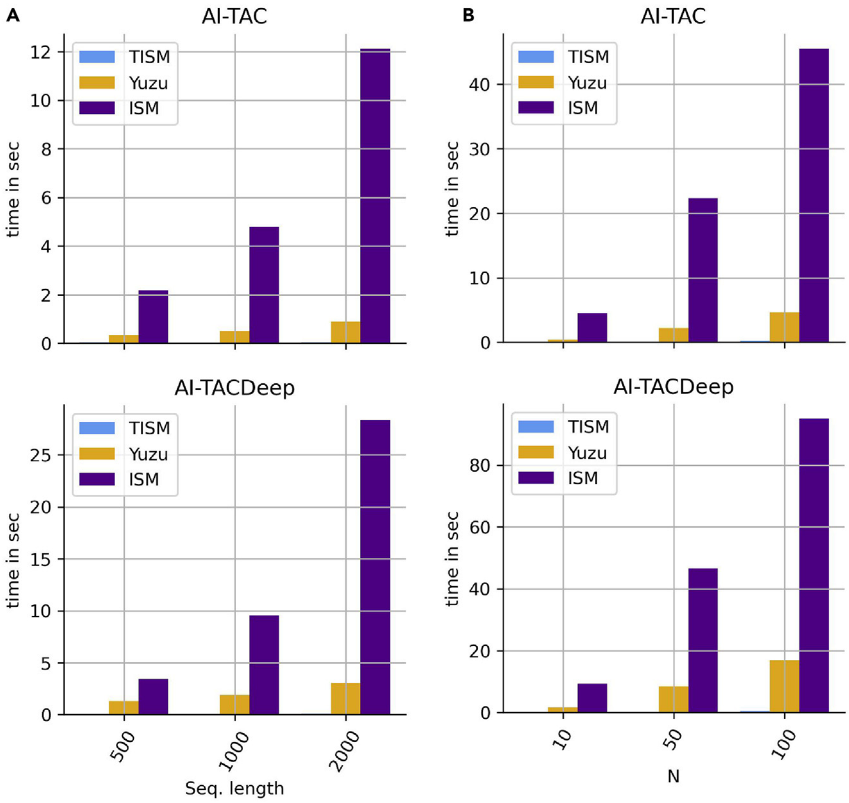


Figure 2. Comparisons between run times of ISM, Yuzu, and TISM on GPU for sequences of different length and different numbers of sequences

Run times are reported for ISM, Yuzu-ISM, and TISM for (A) 10 sequences of length 500, 1,000, and 2,000bp, as well as (B) 10, 50, and 1000 sequences of length 1,000bp. The top row shows run times for a standard AI-tac model (3 conv. layers +2 fully connected layers) and the bottom row shows run time for a deeper AI-tac model (8 conv. layers +3 fully connected layers).

We hypothesize that this can be explained by these underperforming models having learned only simple motif grammar that misses interaction terms between bases. Further exploiting this observation, we hypothesize that one could use the discordance between TISM and ISM to detect sequences that harbor strong base-pair interactions.

Issues with interpreting attribution maps result from a limited understanding of what these values functionally mean. ISM is biologically interpretable but can become computationally challenging for large sets of long sequences that are processed by deep networks. While other backpropagation-based methods can help with this, their values are often harder to interpret and therefore hard to compare across, positions, sequences, and models. The recently developed geometrical correction to the model's gradient by Majdandzic et al.[19] shows empirical and anecdotal evidence for improving motif identification but do not provide a theoretical link to ISM attribution maps or a biological explanation of what these geometrically corrected motif values represent.

Here, we show how one can approximate ISM from the model's gradient. Approximating ISM enables the analysis of both large sets of sequences and long sequences. TISM's strength especially comes through for long sequences (e.g., >20kb), and therefore it is extremely useful to detect, extract, and compare regulatory motifs across sequences and tasks.[4] While not as accurate as FastISM[21] or Yuzu[20] (because these are not approximations), TISM, in contrast, is applicable to any network written in any code base, any number of sequences, and only requires a few lines of code to turn the model's gradient into TISM values.

Limitations of the study

Taylor's approximation uses the gradient around an infinitesimally small region around the sequence of interest to determine the attributions for each base. Other backpropagation-based methods, e.g., DeepLIFT[12] or DeepSHAP[22] approximate the behavior of the non-linear neural network function in a larger region between the sequence of interest and a baseline sequence, i.e., a sequence with neutral signal. These methods avoid saturation effects because they estimate the effect of every base independent of the effect from surrounding bases. Similar gene knockouts, where single gene perturbations can be compensated by paralogs, many bases may only reveal strong effects conditional on other changes. On the other hand, these methods require the definition of new backpropagation rules for the deep learning libraries that are used. Current implementations of the latter are limited to standard activation functions or require a high degree of expertise to work for complex modules that use SoftMax normalizations, such as attention.

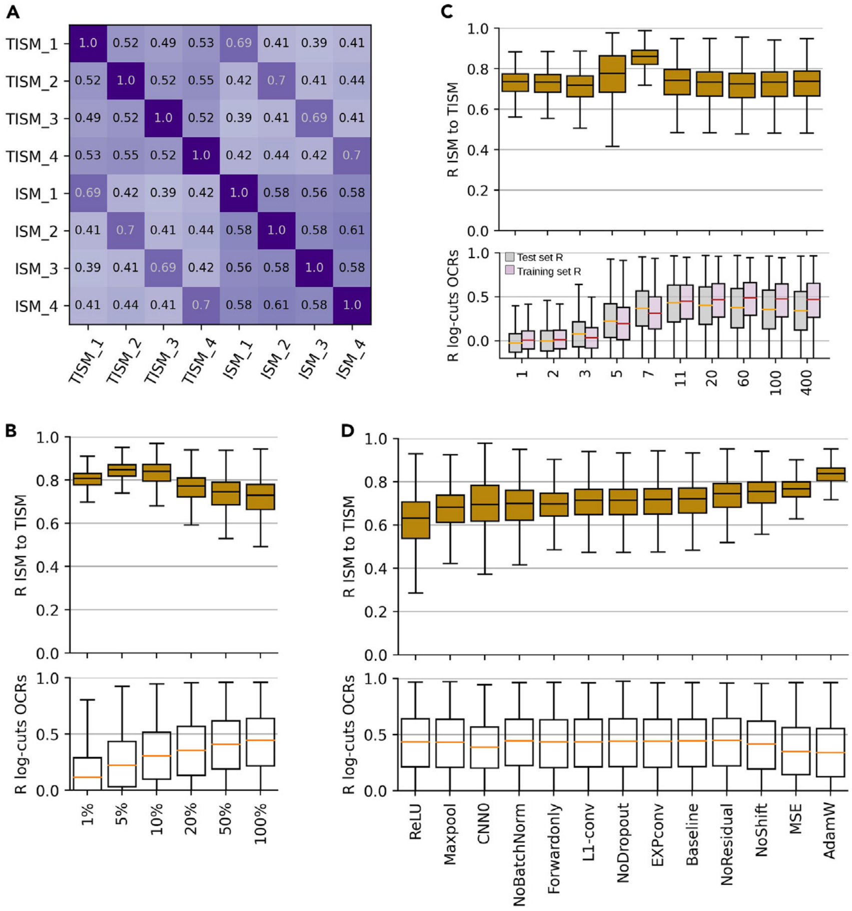


Figure 3. Comparison between ISM and Taylor approximated ISM across models

(A) Average correlation between ISM and TISM values across 3,000 regions and 20 cell types from four random model initializations. The AI-TAC model was trained four times from different random parameter initialization.

(B) Pearson correlation between TISM and ISM for 3,000 test set regions across 81 cell types for six models trained on different percentages of the training set. Bottom shows the Pearson correlation between predicted and measured log-counts of OCRs in the test set across 81 cell types. The data are represented by the median, and boxes extend from the lower to upper quartile values (Q1-Q3).

(C) Pearson correlation between TISM and ISM for 3,000 test set regions across 81 cell types after training for different epochs. Pearson correlations between predicted and measured log-counts of OCRs across 81 cell types is shown for test and the training set sequences. Boxes represent data as described in (B).

(D) Pearson correlation between TISM and ISM for different ablations of the baseline model on top, and Pearson correlation between predicted and measured log-counts of the twelve models for 9158 chromatin regions across 81 cell types. Boxes represent data as described in (B).

ISM represents a well understood concept with clear experimental interpretations. It can be applied to any model architecture without specialized packages. Similarly, TISM uses the standard functions to compute the gradients which any deep learning libraries are equipped with to train the models' parameters. In this work, we only assess the concurrence between ISM and TISM, and note that the study by Majdandzic et al.[19] provides a solid benchmark of the learnt motifs. We demonstrated that attributions computed from centered TISM values result in the same gradient corrections as those proposed by Majdandzic et al.,[19] and therefore we point the reader to their motif benchmarking results instead of replicating those here. Lastly, while we tested a dozen model ablations, and observed only minor discrepancies between ISM and TISM for models that use ReLU activations and max pooling, these analyses are not exhaustive; other modules or combinations of modules and data may have a stronger impact on their concordance.

RESOURCE AVAILABILITY

Lead contact

Further information should be directed to and will be fulfilled by the Lead Contact, Sara Mostafavi (saramos@cs.washington.edu).

Materials availability

This study did not generate any reagents.

Data and code availability

- This paper analyzes existing, publicly available data. These accession numbers for the datasets are listed in the key resources table.
- All original code has been deposited at Zenodo and is publicly available at https://github.com/LXsasse/TISM as of the date of publication. DOIs are listed in the key resources table.
- Processed ATAC-seq data and called peaks can be found at: https://sharehost.hms.harvard.edu/immgen/ImmGenATAC18_AllOCRsInfo.csv or https://www.dropbox.com/s/r8drj2wxc07bt4j/ImmGenATAC1219.peak_matched.txt?dl=0, https://www.dropbox.com/s/7mmd4v760eux755/mouse_peak_heights.csv?dl=0.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Jacob Schreiber for the insightful discussions around sequence attribution scores.

AUTHOR CONTRIBUTIONS

Conceptualization: A.S., S.M., and M.C. methodology: A.S., and S.M., and M.C.; data curation: A.S. software: A.S.; investigation: A.S.; formal analysis: A.S.; visualization: A.S.; validation: M.C. and S.M.; writing: A.S., S.M., and M.C.; supervision: S.M. and M.C.

DECLARATION OF INTERESTS

The authors declare no competing interest.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - ○ Baseline model training
  - ○ Model variants
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ Assessing run times

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.110807.

Received: July 19, 2024
Revised: August 8, 2024
Accepted: August 20, 2024
Published: August 23, 2024

REFERENCES

1. Kelley, D.R., Reshef, Y.A., Bileschi, M., Belanger, D., McLean, C.Y., and Snoek, J. (2018). Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. *28*, 739–750.
2. Zhou, J., and Troyanskaya, O.G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. Nat. Methods *12*, 931–934.
3. Zhou, J., Theesfeld, C.L., Yao, K., Chen, K.M., Wong, A.K., and Troyanskaya, O.G. (2018). Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat. Genet. *50*, 1171–1179.
4. Linder, J., Srivastava, D., Yuan, H., Agarwal, V., and Kelley, D.R. (2023). Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. Preprint at bioRxiv. https://doi.org/10.1101/2023.08.30.555582.
5. Agarwal, V., and Shendure, J. (2020). Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks. Cell Rep. *31*, 107663.
6. Fudenberg, G., Kelley, D.R., and Pollard, K.S. (2020). Predicting 3D genome folding from DNA sequence with Akita. Nat. Methods *17*, 1111–1117.
7. Schwessinger, R., Gosden, M., Downes, D., Brown, R.C., Oudelaar, A.M., Telenius, J., Teh, Y.W., Lunter, G., and Hughes, J.R. (2020). DeepC: predicting 3D genome folding using megabase-scale transfer learning. Nat. Methods *17*, 1118–1124.
8. Zhou, J. (2022). Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. Nat. Genet. *54*, 725–734.
9. Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-Barwinska, A., Taylor, K.R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D.R. (2021). Effective gene expression prediction from sequence by integrating long-range interactions. Nat. Methods *18*, 1196–1203.
10. Agarwal, V., and Kelley, D.R. (2022). The genetic and biochemical determinants of mRNA degradation rates in mammals. Genome Biol. *23*, 245.
11. Jaganathan, K., Kyriazopoulou Panagiotopoulou, S., McRae, J.F., Darbandi, S.F., Knowles, D., Li, Y.I., Kosmicki, J.A., Arbelaez, J., Cui, W., Schwartz, G.B., et al. (2019). Predicting Splicing from Primary Sequence with Deep Learning. Cell *176*, 535–548.e24.
12. Shrikumar, A., Greenside, P., and Kundaje, A. (06–11 Aug 2017). Learning Important Features Through Propagating Activation Differences. In Proceedings of the 34th International Conference on Machine Learning Proceedings of Machine Learning Research, D. Precup and Y.W. Teh, eds. (PMLR), pp. 3145–3153.
13. Chen, H., Lundberg, S.M., and Lee, S.-I. (2022). Explaining a series of models by propagating Shapley values. Nat. Commun. *13*, 4512.
14. Sundararajan, M., Taly, A., and Yan, Q. (06–11 Aug 2017). Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning Proceedings of Machine Learning Research, D. Precup and Y.W. Teh, eds. (PMLR), pp. 3319–3328.
15. Alipanahi, B., Delong, A., Weirauch, M.T., and Frey, B.J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat. Biotechnol. *33*, 831–838.
16. Patwardhan, R.P., Lee, C., Litvin, O., Young, D.L., Pe'er, D., and Shendure, J. (2009). High-resolution analysis of DNA regulatory elements by synthetic saturation mutagenesis. Nat. Biotechnol. *27*, 1173–1175.

17. Celaj, A., Gao, A.J., Lau, T.T., Holgersen, E.M., Lo, A., Lodaya, V., Cole, C.B., Denroche, R.E., Spickett, C., Wagih, O., et al. (2023). An RNA foundation model enables discovery of disease mechanisms and candidate therapeutics. Preprint at bioRxiv. https://doi.org/10.1101/2023.09.20.558508.
18. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic Differentiation in PyTorch.
19. Majdandzic, A., Rajesh, C., and Koo, P.K. (2023). Correcting gradient-based interpretations of deep neural networks for genomics. Genome Biol. *24*, 109.
20. Schreiber, J., Nair, S., Balsubramani, A., and Kundaje, A. (2022). Accelerating in silico saturation mutagenesis using compressed sensing. Bioinformatics *38*, 3557–3564.
21. Nair, S., Shrikumar, A., Schreiber, J., and Kundaje, A. (2022). fastISM: performant in silico saturation mutagenesis for convolutional neural networks. Bioinformatics *38*, 2397–2403.
22. Lundberg, S., and Lee, S. (2017). A unified approach to interpreting model predictions. Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1705.0787.
23. Yoshida, H., Lareau, C.A., Ramirez, R.N., Rose, S.A., Maier, B., Wroblewska, A., Desland, F., Chudnovskiy, A., Mortha, A., Dominguez, C., et al. (2019). The cis-Regulatory Atlas of the Mouse Immune System. Cell *176*, 897–912.e20.
24. Maslova, A., Ramirez, R.N., Ma, K., Schmutz, H., Wang, C., Fox, C., Ng, B., Benoist, C., and Mostafavi, S.; Immunological Genome Project (2020). Deep learning of immune cell differentiation. Proc. Natl. Acad. Sci. USA *117*, 25655–25666.
25. Koo, P.K., and Ploenzke, M. (2021). Improving representations of genomic sequence motifs in convolutional networks with exponential activations. Nat. Mach. Intell. *3*, 258–266.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Raw and processed ATAC-seq data	Yoshida et al. 2019[23] https://doi.org/10.1016/j.cell.2018.12.036	GSE100738
Filtered ATAC-peak locations	Maslova et al. 2020[24]	https://www.dropbox.com/s/r8drj2wxc07bt4j/ImmGenATAC1219.peak_matched.txt?dl=0
Normalized ATAC-peak counts	Maslova et al. 2020[24]	https://www.dropbox.com/s/7mmd4v760eux755/mouse_peak_heights.csv?dl=0
Software and algorithms		
Python version 3.8	Python Software Foundation	https://www.python.org
Pytorch 2.3	The PyTorch Foundation	https://pytorch.org/
TISM	Github	https://github.com/LXsasse/TISM https://doi.org/10.5281/zenodo.13290074
AI-TAC	Github	https://github.com/smaslova/AI-TAC
DRG	Github	https://github.com/LXsasse/DRG

METHOD DETAILS

Baseline model training

To determine how well TISM reproduces the results computed by ISM under different conditions, we trained various versions of our previous AI-TAC model[24] on the same ATAC-seq data for 286,000 open chromatin regions (OCRs, regions that were determined to be open in at least one cell type) across 81 mouse immune cells.[23] All models use a 251bp one-hot encoded sequence around the center of the ATAC-seq peaks and predict $\log_2(x+2)$ transformed Tn5 cuts within the peak region. We train on OCR sequences from 16 out of 19 chromosomes and use OCRs on chromosome 8 and 11 for validation, and those on chromosome 19 as an independent test set. We use early stopping and select our final model based on the highest mean Pearson's correlation coefficient between model's prediction of accessibility and the ground truth across cell types for OCRs (computed on the validation set).

All the models use 298 kernels of length 19bp, with a GELU activation function, and SoftMax weighted mean pooling of size 2. The models apply 4 residual convolutional blocks with batch normalization, kernel size of 7, 298 kernels, and subsequent SoftMax weighted mean pooling of size 2. The resulting representation is flattened and condensed to a size 512 tensor with a linear layer, followed by two fully connected layers with GELU activation, before the model heads predict the log transformed counts for the 81 cell types in a multi-task fashion. All models use a dropout of 0.1 in all the fully connected layers. We use a mixture of MSE and mean correlation of OCRs across 81 cell types as a loss function and update the models' parameters with SDG with 0.9 times momentum and a learning rate of 1e-5. The learning rate is exponentially warmed up in 7 epochs, and fine tuning is performed for five iterations on the best performing parameters with gradually reduced learning rates. All models use the forward and the reverse strand of the sequence and perform kernel specific max-pooling along the aligned forward and backward activations from both sequences, only forwarding the highest activation of a kernel from the two strands at a given position. In addition to the sequence centered at the ATAC-peak, we also perform data augmentation by including shifted versions of each sequence,[9] where we randomly shift the genomic location of a given sequence by a number between -10 to 10 base pairs.

Model variants

AI-TAC is a standard CNN model trained with correlation loss function. Because the attribution methods we use here are agnostic to the specific model architectures, the results presented should generalize more broadly to other models and training datasets. However, we also repeated these experiments with various modifications to the model's architecture to examine the generalizability of our results. Specifically, to test how concordant TISM values are to ISM values across different modeling choices/architectures, we trained different models and used ablation to investigate the effect of various modeling choices. First, we trained four versions of the above "baseline" model with different random initializations. Second, we trained the model on five different percentages of the training set to assess how training set size influences TISM's concordance to ISM. Third, we stopped training after 1, 2, 3, 5, 7, 11, 20, 60, 100, and 400 epochs and assessed how the concordance changes during training. Lastly, we performed ablation analysis of the baseline model as follows: 1) We trained solely using the MSE loss, 2) we used ReLU activation throughout the model, 3) we used an exponential activation function after the first convolution,[25] 4) we used max-pooling instead of the weighted mean pooling, 5) we trained without the reverse complement sequence, 6) we trained without residual connections, 7) we trained without dropout, 8) we trained without batch norm in the residual convolutional layers, 9) we used L1-regularization of the 298 kernels in the first layer, 10) we trained with AdamW instead of SGD, 11) we trained without randomly shifted sequences, and 12) we trained a shallow CNN that only uses one convolutional layer, 70 bp wide weighted mean pooling, which is flattened and then directly given to the linear prediction head.

QUANTIFICATION AND STATISTICAL ANALYSIS

Assessing run times

To empirically determine the speedup of TISM over ISM, we trained three additional model architectures that used sequences of length 1,000bp, 5,000bp, and 20,000bp as input. For these models, we used the same number of convolutional layers but adjusted the size of the SoftMax weighted mean pooling to account for the larger sequence windows. We then determined the time to compute ISM and TISM values for models with four different input lengths and for one model for four different numbers of sequences. Specifically, to compute ISM values, we measured the total time to generate a set of one-hot encoded variant sequence tensors that contain a single base-pair change to the original sequence, make predictions with the model using a batch size of 20, and finally generate the ISM tensor from these predictions. For TISMs, the measured time includes the forward pass through the model, backpropagation through the model to the input sequence to obtain the gradient, and finally subtraction of the gradient at the reference base from each position to get TISM values. All the measurements were performed on a single NVIDIA RTX A4000 GPU with 16GB Memory.