Check for updates

Obtaining genetics insights from deep learning via explainable artificial intelligence

Gherman Novakovsky[1,2,7], *Nick Dexter*[3,4,7], *Maxwell W. Libbrecht*[4,8]✉, *Wyeth W. Wasserman*[1,8]✉ *and Sara Mostafavi*[5,6,8]✉

Abstract | Artificial intelligence (AI) models based on deep learning now represent the state of the art for making functional predictions in genomics research. However, the underlying basis on which predictive models make such predictions is often unknown. For genomics researchers, this missing explanatory information would frequently be of greater value than the predictions themselves, as it can enable new insights into genetic processes. We review progress in the emerging area of explainable AI (xAI), a field with the potential to empower life science researchers to gain mechanistic insights into complex deep learning models. We discuss and categorize approaches for model interpretation, including an intuitive understanding of how each approach works and their underlying assumptions and limitations in the context of typical high-throughput biological datasets.

该领域有可能使生命科学研究人员能够对复杂的深度学习模型获得机械的见解。

Features
Scalar inputs to a machine learning model.

Local interpretation
The task of understanding a model's prediction for a single input.

Global interpretation
The task of understanding how a model makes predictions across all inputs.

Deep learning (DL) is having a major impact on the study of genetics owing to its impressive performance; however, because of its complex nature, it is often perceived as a 'black box'. With increasingly large amounts of data being generated at low cost and complementary hardware advances, the new forms of artificial intelligence (AI) are enabling diverse prediction tasks ranging from annotation of the regulatory genome to categorization of single-cell data[1–4]. By understanding the properties in large-scale data that are the basis for such successful predictions, it is anticipated that we can obtain deeper insights into the biological processes being studied. Over the past few years, new model interpretation techniques that allow such insights have been emerging rapidly in the field of explainable AI (xAI)[5].

新的模型解释技术

Deep learning models can learn complex patterns. In determining which DNA sequences in the human genome are directing gene transcription in a specific cell type, for instance, models may be learning sequence composition (for example, GC content), the presence or absence of specific motif patterns (transcription factor (TF) binding sites), the local accessibility of chromatin, biophysical properties of DNA (for example, bendability), positional differences between sequence properties (that is, features found in the edges or middle of the region), positive or negative interactions between features, or attributes that are beyond our current state of knowledge. To learn a large and complex feature set, such models learn millions of parameters that together determine the model predictions, but without providing an explanation about how a given prediction was made. Hence, the challenge of interpreting a complex dataset is essentially converted into a challenge of interpreting a complex model, with the benefit that probing the model is limited only by computation cost.

In this Review, we present an organized overview of the key interpretation approaches with the intention of empowering researchers working across topics in genetics to incorporate xAI into their studies. We focus on the task of post-hoc interpretation, a popular class of xAI algorithms that are being increasingly used and improved for biological applications. Post-hoc interpretation operates on a trained model and aims to identify relevant combinations of input features and to quantify the importance of each feature — or combination thereof — to the model's performance. These algorithms thus provide a new and exciting approach for systematic hypothesis generation, aiding researchers to prioritize experimentation. Post-hoc interpretation methods can be divided into two types: local and global. Local interpretation aims to identify important features that affect a model's prediction on a single input example (for example, an individual DNA sequence). Global interpretation applies across all input examples to identify the combination of features that affects the model's overall performance.

使从事遗传学研究的研究人员能够将xAI用于他们的研究。

In the sections that follow, we first provide fundamental information on how deep learning methods are used in regulatory genomics, and then we categorize four approaches for interpretation: model-based

深度学习方法如何用于调控基因组学

✉e-mail: maxwl@sfu.ca; wyeth@cmmt.ubc.ca; saramos@cs.washington.edu

https://doi.org/10.1038/s41576-022-00532-2

Sequence-to-activity models
A class of learning tasks that takes a DNA sequence as input and predicts a property of the activity of that sequence, such as transcription factor binding or chromatin accessibility in a cell type of interest.

Convolutional neural networks
(CNNs). A neural network architecture that includes convolutional nodes.

Recurrent neural networks
(RNNs). A type of neural network architecture in which nodes are arranged in a chain along a sequential input such as a DNA sequence.

Layers
Sets of neural network nodes that take input from nodes of the previous layer and output to nodes of the subsequent layer.

Convolutional nodes
(Also known as filters). A type of neural network node that takes input from a short contiguous sequence of nodes, usually 3–20 bp in sequence-to-activity models.

Nodes
(Also known as units and artificial neurons). The basic units of a neural network. They take input from other nodes and output scalar values to other nodes.

interpretation, mathematical propagation of influence, identification of interactions between features and use of prior knowledge for transparent models (FIG. 1). For each approach, we provide an intuitive understanding of how it works and explain its underlying assumptions and limitations in the context of typical high-throughput biological datasets. When applicable, we highlight how local interpretations can be aggregated to reveal a global understanding of the behaviour of a model, but we note that at the time of writing, there is no general framework for aggregating information from local interpretation techniques. Although we present methods with reference to successful cases, we recognize that the insights derived from the application of xAI methods are highly dependent on experimental design, dataset properties, and the accuracy of the trained models. Throughout the Review, we use examples from the field of regulatory genomics (see BOX 1 for an introduction to gene regulation research), but the concepts presented can be generalized across a broad range of applications.

我们使用了调控基因组学领域的例子

Deep learning in regulatory genomics

Neural networks and sequence-to-activity models. Deep neural network (DNN) models have emerged as the leading type of predictive model in regulatory genomics[1,4,6]. For this Review, we focus on sequence-to-activity models based on neural networks. These models take a putative regulatory DNA sequence (usually 100–10,000 bp) as input and aim to predict some dynamic property (that is, cell or context specificity) of the sequence's activity. For example, a model may predict whether a given TF binds to that sequence in a given cell type as measured by a chromatin immunoprecipitation followed by sequencing (ChIP–seq) experiment[7–10]. Other common prediction targets include chromatin accessibility[11,12], RNA binding[13], gene expression[14–17], splicing[18,19] or aspects of chromatin 3D organization[20].

序列到活性的模型

The most widely used architectures for sequence-to-activity models are convolutional neural networks (CNNs) and recurrent neural networks (RNNs)[2], both of which were initially developed for tasks in computer vision and natural language processing (see BOX 2 for an overview of neural networks). CNNs include initial layers composed of convolutional nodes (also called filters), in addition to fully connected nodes in later layers. A convolutional node is a type of neural network node that serves as a pattern detector. It is scanned across the input, at each position evaluating whether the sequence matches a specific pattern, usually of length 3–20 bp. RNNs are composed of nodes arranged in a chain. In sequence-to-activity models, this chain is oriented along the input DNA sequence. Each node (or collection of nodes) in the chain takes a single DNA letter as input and outputs a value to the next node in the chain. Recurrent and convolutional architectures are often used together, such that sequence is passed through one or more convolutional layers before entering a recurrent layer[12,21]. No consensus currently exists regarding how to design the best neural network architecture for a given task, so researchers typically experiment with multiple architectures[21]. We point readers to previous review articles for a more complete discussion of various neural network architectures and their properties in the context of genomics datasets[1,2,4].

Why is DNN model interpretation difficult? To learn complex combinations of predictive features from data, state-of-the-art DNN models typically consist of tens of millions of free parameters that are learned during training. Although the high capacity of DNNs to encode latent feature representations results in state-of-the-art prediction accuracy, it comes with the challenge of identifying what features and feature combinations have been learned by the model.

xAI algorithms can examine the inner workings of black box models such as DNNs to reveal the basis on which predictions are made. Intuitively, almost all post-hoc model interpretation methods can be understood through a feature removal recipe, whereby trained models are probed to assess the importance of a given feature or feature combination to the model's performance[22]. From a mathematical perspective, the challenge of deep learning model interpretation stems from the need to somehow navigate a vast combinatorial search space. Many frameworks have been proposed, including gradient-based[23], perturbation-based[24] and game theory-based analysis[25,26], to intelligently evaluate the combinatorial search space to produce local interpretations. However, each of these strategies comes with assumptions and limitations, and hence there is no globally optimal strategy for model interpretation. The second challenge is in generalizing from local interpretations, which estimate feature importance on a given input example at a time, to a global understanding of important feature combinations on an entire data corpus. Another key difficulty is the inability to systematically evaluate interpretation strategies, first because there is a lack of benchmark datasets in which the true set of important features is known ahead of time, and second because the validity of the assumptions made by the various algorithms depends on both the properties of the input dataset and the biological processes involved.

现有策略

无法获取全局最优的模型解释策略

Model-based interpretation

直观方法

An intuitive approach to model interpretation is to examine individual components of a network to understand what (hidden) patterns they represent and their

Author addresses

[1]Centre for Molecular Medicine and Therapeutics, Department of Medical Genetics, BC Children's Hospital Research Institute, University of British Columbia, Vancouver, British Columbia, Canada.
[2]Bioinformatics Graduate Program, University of British Columbia, Vancouver, British Columbia, Canada.
[3]Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada.
[4]School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada.
[5]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA.
[6]Canadian Institute for Advanced Research, Toronto, Ontario, Canada.
[7]These authors contributed equally: Gherman Novakovsky, Nick Dexter.
[8]These authors jointly supervised this work: Maxwell W. Libbrecht, Wyeth W. Wasserman, Sara Mostafavi.

contributions to the prediction performance. Although this can be effective for small networks, modern DNNs are prohibitively large and complex. However, even for deep networks, often a partial model-based interpretation can be helpful in extracting important features learned by the model. For instance, in computer vision, artificial neurons in hidden layers have the potential to learn semantically interpretable features with increasing levels of abstraction[27]. In a facial recognition model, the neurons of the first hidden layer may detect the

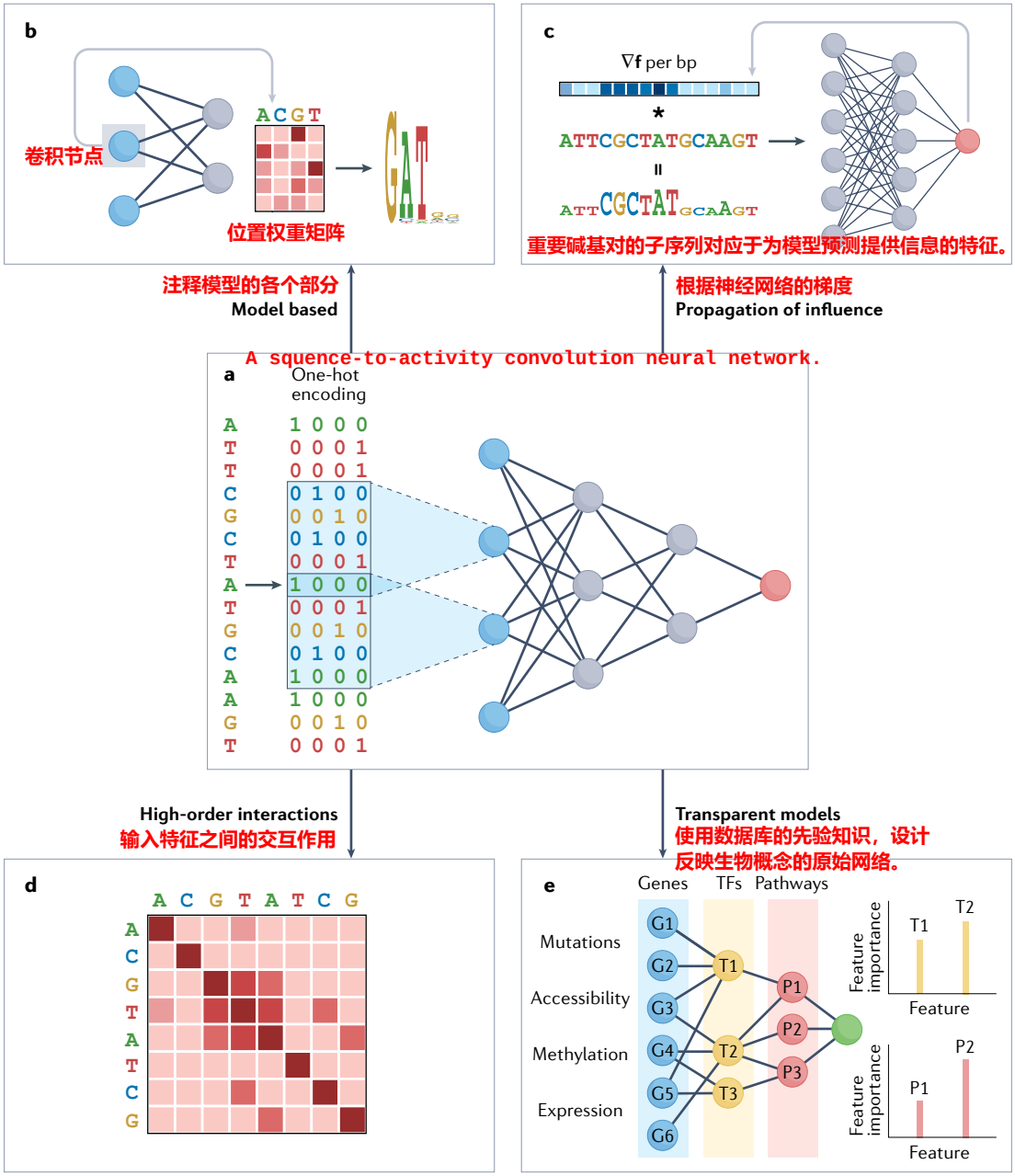
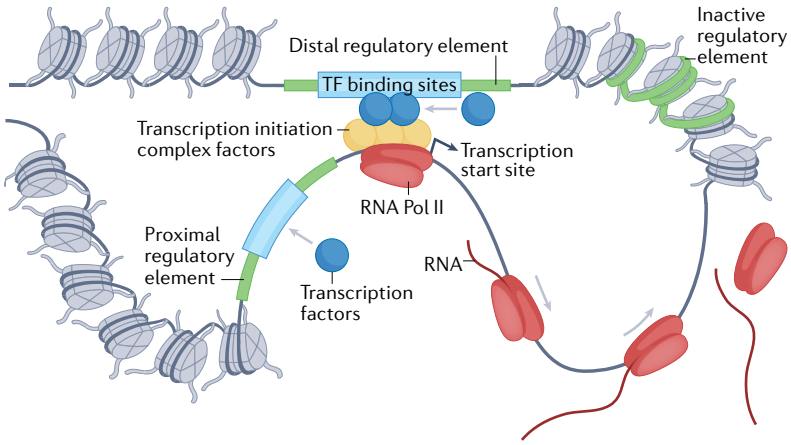


Fig. 1 | **Conceptual approaches to explainable artificial intelligence. a** | A schematic representation of a sequence-to-activity convolutional neural network. The input DNA sequence is one-hot encoded and scanned by convolutional nodes in the first layer (cyan). The activation values of convolutional nodes get passed through non-linearities introduced in subsequent layers (grey). The output layer (red) predicts a property of interest. **b** | Model-based approaches focus on annotating individual parts of the model. For example, a convolutional node (cyan) represents a position-weight matrix (red heat-map) and can be linked to a transcription factor (TF) binding site or other sequence pattern. **c** | Propagation of influence predicts the importance of each input feature (for example, base pair (bp)) according to the gradient of a neural network model ($\nabla \mathbf{f}$). Subsequences of important base pairs correspond to features that are informative to the model's prediction. **d** | As these networks learn a nonlinear function, they can be studied for the presence of interactions between input features. Interaction identification methods can output the interaction strength of each pair of input features. In the example shown, the pairwise interactions between nucleotides in the input DNA sequence are indicated by the intensities in the heat-map plot. **e** | The original network can be designed to reflect biological concepts, which also makes it transparent. Researchers use prior knowledge from databases including those of motifs or Gene Ontology subsystems.

Box 1 | **Regulatory genomics: research in gene regulation**

Understanding the control of gene transcription is crucial to understanding many aspects of cellular biology. Multiple biochemical mechanisms influence gene transcription, including the structure of chromatin, the binding of transcription factors (TFs) to DNA, the components of the RNA polymerase (RNA Pol) complex and the actions of proteins that interact with the above[87] (see the figure). At a higher level, studies explore the different types of regulatory element within the genome such as promoters, from which RNA transcription is initiated, and distal regulatory elements such as enhancers, which confer patterns of specificity upon promoters[88,89]. Owing to the crucial role of gene regulation in controlling the phenotype of cells[90], new technologies for measuring cellular properties related to gene regulation are constantly emerging, providing higher resolution and broader coverage, which have transformed the data landscape of the field. At the core of experimental studies are assays that quantify the structure of chromatin (for example, Hi-C)[91], the accessibility of chromatin (for example, DNase hypersensitivity, and assay for transposase-accessible chromatin sequencing (ATAC-seq))[89], and the presence of proteins or modified forms of proteins at specific locations along the DNA[92] (for example, chromatin immunoprecipitation followed by sequencing (ChIP–seq) for TFs or histone modifications, or many specialized forms of ChIP–seq, such as ChIP–exo). As measurements across an entire tissue capture heterogeneous mixtures of cell types, these technologies are increasingly applied to isolated cell types and more recently at single-cell level. Deep learning methods have been propelled to the forefront of the analysis of such data as their size and complexity continue to expand[1,3,4,93,94]. Common questions pursued by researchers include the identification of and interplay between regulatory sequences within the genome that are active in a particular cell or cell context[7,8,11], the locations of TF binding sites within such regions, the identity of and cooperative interplay between TFs to derive a given cell state[16], the 3D structure of chromatin in the nucleus and much more[20].



Regulatory element
Region in genomic DNA that can contribute to gene regulation.

Attention mechanism
A component of a neural network that can learn to adaptively prioritize (that is, pay attention to) certain parts of an input by weighting.

Attention weights
Weights learned by the attention mechanism.

Drop-out
A form of regularization typically used during training of neural networks in which activations from subsets of hidden units are zeroed out.

presence of edges in input images at various orientations on a fine scale, and then the deeper layers use combinations of edge detectors to identify higher-level image parts such as ears, nose and mouth[28]. Analogously, in sequence-to-activity models, first-layer neurons may learn short subsequences that represent TF binding motifs, and the deeper layers construct various combinations of these motifs as composite features that are useful for prediction of cell-type-specific chromatin states[11].

Two main method categories for model-based interpretation can be considered. The simplest approach directly examines the activity of hidden neurons to extract a set of relevant features. The second approach initially trains models with an attention mechanism, which directly yields a measure of relevance per input feature through a set of learned attention weights. In this section, we describe how these approaches can be applied to sequence-to-activity models to understand individual features that are learned by the first layer of the network, and in the 'Identifying interactions between features' section (below) we describe approaches to identify combinations of features learned in deeper network layers.

Interpreting first-layer convolutional nodes. In a convolutional sequence-to-activity model, first-layer neurons (filters) capture short sequence motifs, encoded in convolutional weight matrices. Mathematically, the operations performed by application of convolutional weight matrices to sequence are equivalent to scanning the sequence with a position-weight matrix (PWM) (FIG. 2a). Thus, one can apply a simple transformation (called softmax) to a convolutional weight matrix to derive a position frequency matrix (PFM)[13] that quantifies position-dependent nucleotide frequencies, and further scaling and log-transformation enables visualization of the matrix associated with a filter as a standard PWM[29] (which quantifies the log likelihood of each nucleotide in the motif represented by the corresponding convolutional matrix). However, unconstrained learning of weight values may create scaling issues that render this approach less effective. In practice, a more common strategy is to search for subsequences that activate a given filter above a chosen threshold and directly construct a PWM based on the alignment of the set of activating subsequences (FIG. 2b). One can use the entire input dataset to search for maximally activating subsequences, or solve a more general optimization problem that searches among all possible subsequences of length m for those that maximally activate a given filter[30,31]. These PWMs can then be annotated by comparing them with known TF binding profiles in databases such as JASPAR and Cis-BP[32,33]. Although there is no guarantee that CNN filters resemble known TF binding motifs, previous work has shown that the learned PWMs typically do so. For example, when CNNs were applied to assay for transposase-accessible chromatin sequencing (ATAC-seq) data across various immune cells, most learned PWMs mapped to known TF binding motifs that have prominent roles in immune cell differentiation, including motifs for PAX5, EBF1 and LEF1 (REF.[11]). Other studies have shown similar results[8,12].

Because neural networks are over-parameterized by design, the mere presence of a PWM does not imply that it is a predictive, interesting or useful feature. Therefore, we need to measure the contribution of the PWM to the model's predictions[8,11,18]. In the node-based strategy, this is achieved by nullifying (or ablating) each filter in turn and measuring the impact of such nullification on the model's predictions (FIG. 2c). Intuitively, if an influential filter is nullified, then the network's prediction(s) should be significantly altered. The impact of filter nullification can be measured for each input example, giving a local interpretation. To form a global interpretation, the simplest approach is to average the local interpretations.

Node-based strategies are reductionist approaches to understanding a complex system, with the core assumption that individual units are independently interpretable on their own. However, because DNNs are usually trained to be robust to drop-out[34,35] of individual neurons, in practice, an important pattern may be captured

Overfitting
The case when a machine learning model is specific to its training set and does not generalize to other inputs.

Labels
The target outputs of a classification model.

One-hot encoding
The process of converting a DNA letter into a length-4 vector such that one position is set to 1 and the others are set to 0, for use as input to a neural network.

multiple times by different neurons: that is, nodes may be redundant. In this situation, nullifying a single neuron will not provide the true importance of a pattern to the model's predictions. Additionally, depending on the architecture of the model, a biologically interpretable pattern may be learned as a combination of subcomponent nodes. For example, a long TF motif may be learned by two filters, each learning some part of the motif[36].

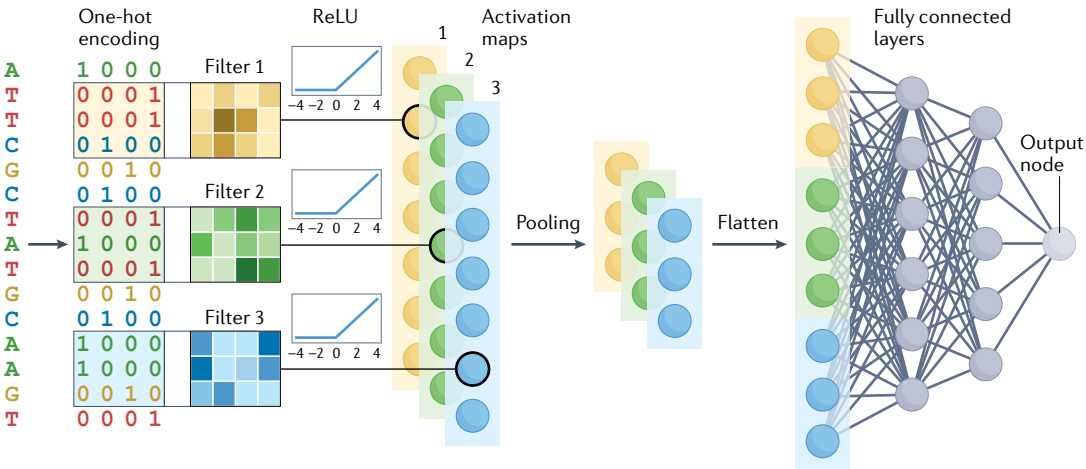
Attention mechanism weights for visualizing feature importance. Weight regularization is a general technique to help mitigate the issue of overfitting in training neural networks and often improves both performance and ability to isolate important features[35]. The simplest strategy for weight regularization is to add a term to the objective function during model training to encourage the learned weights to have certain properties, for example, to be small in magnitude or to be sparse with few non-zero elements[37,38]. However, one can also consider different types of weighting in neural network models. Attention (and its many variants) can be considered a form of weight regularization for which weights are introduced for input sequences to prioritize positions in the input as likely to hold relevant information for processing[39]. The computation of these weights is generally handled by an additional module in the network that automatically learns the attention weights during training to capture relevance between hidden states in machine learning tasks[40]. Attention mechanisms can improve both performance and interpretability of neural network models[41], and in particular address the challenge of performance degradation with respect to increasing input sequence length for RNNs[39].

As the name implies, when combined with model training, attention weights force the model to focus on a limited portion of the input while learning hidden features. In the case of sequence-to-activity models, the attention vector can be examined directly to aid in

Box 2 | **Neural networks for high-throughput genomics**

A neural network is a model that implements a mathematical function taking input features (such as a DNA sequence) and producing one or more output labels (such as predicted transcription factor (TF) binding or gene expression)[35]. Neural network models are constructed from many nodes, also known as units or artificial neurons, organized in layers (see the figure). For sequence-to-activity models, input DNA sequences of length L are converted using one-hot encoding: a binary matrix of size $4 \times L$, where each column has a single entry set to 1 indicating the DNA base (A, C, T or G) and all other entries set to 0. Neural network models taking length L one-hot encoded sequences have $4L$ input nodes. Internal nodes compute their values as $\sigma(w_1 x_1 + \cdots + w_n x_n + b)$, where $(x_1,\ldots,x_n)$ are node inputs, $(w_1,\ldots,w_n)$ are the weights, b is the bias and σ is the activation function. The output node of a neural network represents its prediction. Some neural networks have multiple output nodes, and thus solve multiple tasks at once; for example, sequence-to-activity models may simultaneously predict the binding of multiple TFs, or the same TF across multiple cell types. The training process adjusts the weights $(w_1,\ldots,w_n)$ and bias (b) of each node to maximize agreement between the output nodes and labels of a training set.

One can construct single-layer neural networks that are mathematically equivalent to linear or logistic regression, depending on the activation function. Deep neural networks (DNNs) contain additional layers of nodes other than the input and output; these nodes are hidden nodes because their values are never observed directly but instead are learned from data during training. The non-linearity of DNNs stems from the fact that they use nonlinear activation functions in at least some of the layers. A popular activation function is the rectified linear unit (ReLU). A pooling operation is typically applied to the output of a convolutional layer (that is, activation map), whereby the maximum or average of nearby elements is used to reduce the number of learned parameters in the later layers. Finally, a flattening operation combines multiple activation maps into a single vector to use as input to the fully connected layer. A DNN model with only linear activation functions would still be equivalent to linear regression. The neural network architecture refers to the number of hidden nodes and how they are connected to each other. DNNs are typically organized into layers, where the nodes of each layer take the previous layer nodes as input. The objective function is the mathematical function that the training process seeks to optimize; for example, mean squared error (m.s.e.) is the objective function typically used to train a model on continuous output. Recent reviews provide a deeper overview of machine learning methods, including DNNs, and their application to high-throughput biological datasets[1,3,4,95].



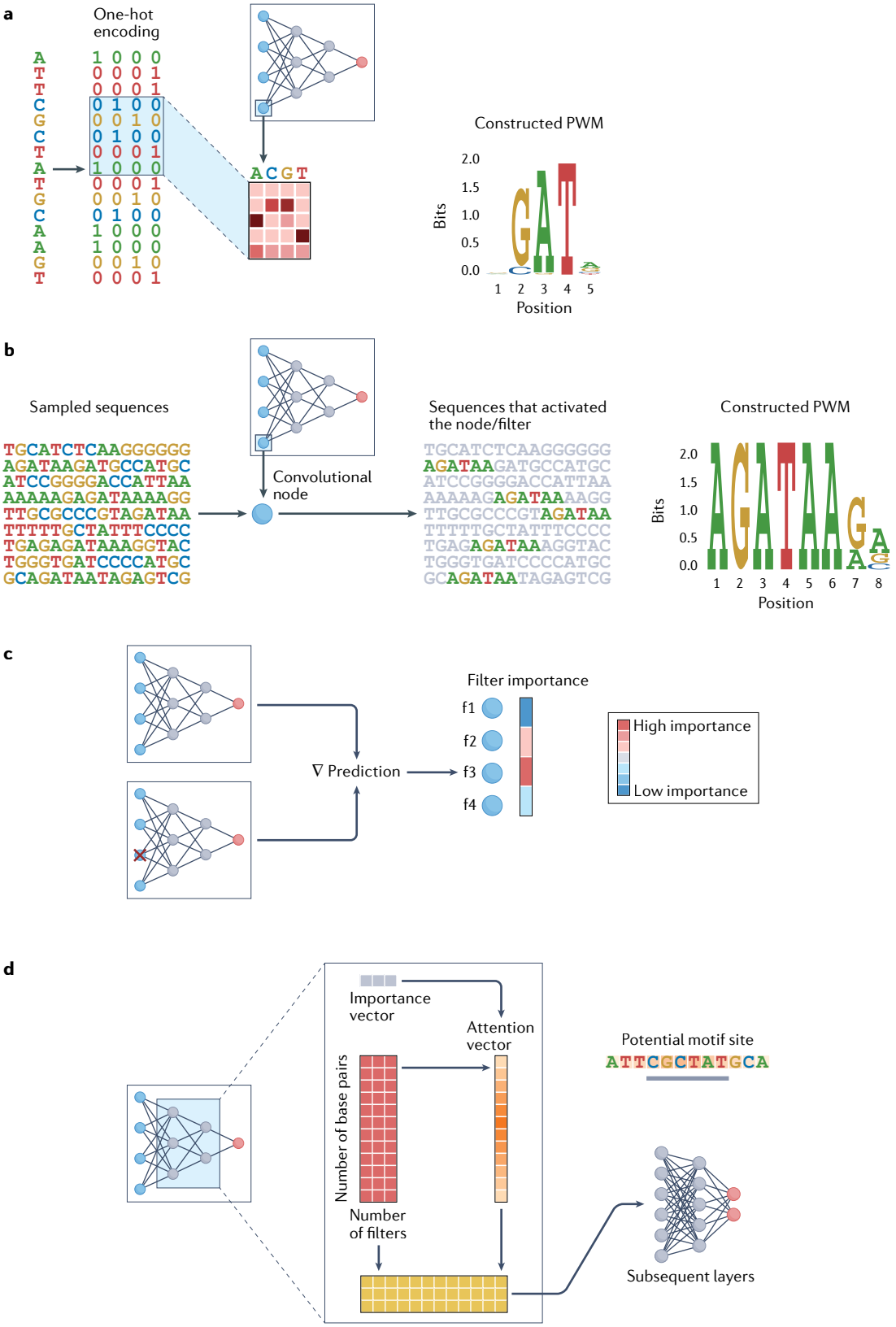


Fig. 2 | **Approaches to model-based interpretation. a** | Individual convolutional filters act as position-weight matrices (PWMs) that scan an input sequence for a pattern. **b** | Each hidden node of the network learns a certain sequence pattern. Finding subsequences that activate the given nodes above a threshold enables researchers to annotate them through PWM construction. **c** | Ablating a filter or node shows how important that node is for the final prediction. **d** | Attention is an extra layer in the network that helps the network to focus only on the most relevant input features. Visualizing attention weights gives an impression of important features that are used to make predictions.

identifying parts of the input that played a key part in the model's internal representation (FIG. 2d).

Previous empirical work has shown some promise in directly interpreting attention weights in the sequence-to-activity setting. For example, attention weights were shown to correlate with the expected location of the TF binding site in the sequence[42], and coincide with DNase footprint locations[43]. However, other studies caution against direct interpretation of attention weights[44]. For example, redundancy and multicollinearity in input features, which are typically abundant in high-throughput data, can result in instability in the estimation of attention weights.

Mathematical propagation of influence

In contrast to probing components of a trained neural network, another class of algorithms operates directly on input examples by propagating perturbed data through the model and observing the effect on predictions. These propagation-based feature attribution methods are thought to be more model-agnostic, as to some extent they can bypass the side-effects of specific model architectures (for example, redundancies in learned filters) when determining feature importance. They also solve the problem of feature identification and quantification of the importance of each feature simultaneously. Propagation-based attribution methods can be divided into two major groups: forward and backward. In this section we discuss the conceptual foundations behind the most popular attribution methods, their shortcomings and mitigation strategies in the context of genetic data.

Forward propagation of influence. In computer vision, perturbation strategies have been used for decades. The simplest form is pixel flipping, whereby one or multiple pixels of an input image are modified to identify features in the image that are relevant for the prediction results[45]. If changing the value of a pixel has a large effect on the classification result, then it may correspond to a feature (or part of a feature) that the model has identified as relevant for making predictions. Analogous to image pixel flipping one can consider flipping elements corresponding to nucleotides of biological sequences to determine feature importance for trained models (FIG. 3a). This strategy is termed in silico mutagenesis (ISM)[7], owing to the parallels between this approach and in vitro mutagenesis of DNA.

To perform ISM in practice, given an input DNA sequence X of length L, each index i (nucleotide) in the input sequence is selected, and for each of the three alternative nucleotides a new sequence is generated by changing only the entry in the ith position. The difference between the model's prediction on the alternative sequence and the original sequence is often referred to as the attribution score. Repeating for all nucleotides results in a $4 \times L$ matrix called an attribution map[46], which can be visualized as a sequence logo (FIG. 3a). The approach has been shown to deliver superior results to other attribution-based approaches[47,48]. Although ISM is typically used to record the perturbation effect on the final layer output, it can also be applied to discover attributions for hidden neurons.

Applying ISM at a single-nucleotide level across many input examples results in a large computational overhead (order of $3L$ per input sequence). To save computation, one can limit the analysis to a subset of input sequences of greatest expected insight value (such as sequences with strong predictions). Other strategies take advantage of specific architectures of neural networks to bypass some redundant computations, for example, only computing values in the receptive field of a given CNN convolution filter. These include fastISM[49] and accelerated ISM[50]. Applying these strategies enables efficient application of single-nucleotide-level ISM, but it is not computationally practical to exhaustively examine all possible combinations of nucleotides.

Instead of per-nucleotide ISM, larger stretches of the input sequence can be altered to identify important motifs that depend on a combination of important base pairs (akin to laboratory-based scanning mutagenesis) (FIG. 3b). Such forms of occlusion were used to find enhancer–gene pairs[16], CTCF (CCCTC-binding factor) sites with certain orientations[20] and boundaries of cis-regulatory elements[51]. Instead of random occlusion, greater interpretability may be gained by mutating known TF motifs (also called motif mutagenesis), but this approach requires a priori knowledge of the motif sites. For instance, one can scan existing TF PWMs on a given sequence to get a per-nucleotide scanning score, and then use ISM to alter the sequence of high-scoring segments.

This idea of partial occlusion can further be generalized to scanning models on synthetic sequences that contain specific subsequences (k-mers) at specific sequence positions[52,53], to gain higher-resolution insight into sequence properties, including the importance of motif position, spacing or flanking sequences[54], without having to rely on our incomplete knowledge about TF binding patterns. As one may expect on the basis of the underlying biology, the sequence context onto which such subsequences are positioned tends to have an impact on the results. Computationally, less-biased results may be obtained by preserving dinucleotide composition and motif positional information in the synthetic sequences[55,56].

Backward propagation of influence. Forward propagation methods are computationally expensive owing to the large number of forward passes required to generate accurate statistics. Back-propagation methods were developed to address this issue. These methods approximate ISM by evaluating the derivative of the model F at a given input sequence to compute the impact of infinitesimally small changes to the sequence on the model's prediction[46] (FIG. 3c).

As neural networks are constructed from the composition of multiple nonlinear functions, the gradient of the model F needs to be computed with the chain rule using the back-propagation procedure (backward propagation of partial derivatives through the network). This yields a function that then needs to be evaluated at a specific input to produce a gradient vector. The resulting vector has size equal to the number of input features. An element-wise product between the gradient

Attribution score
An importance score assigned to a given input feature by a post-hoc local interpretation method.

Attribution map
(Also known as saliency map or relevance map). An estimate of how much each input feature contributes to the output, produced by certain local interpretation methods.

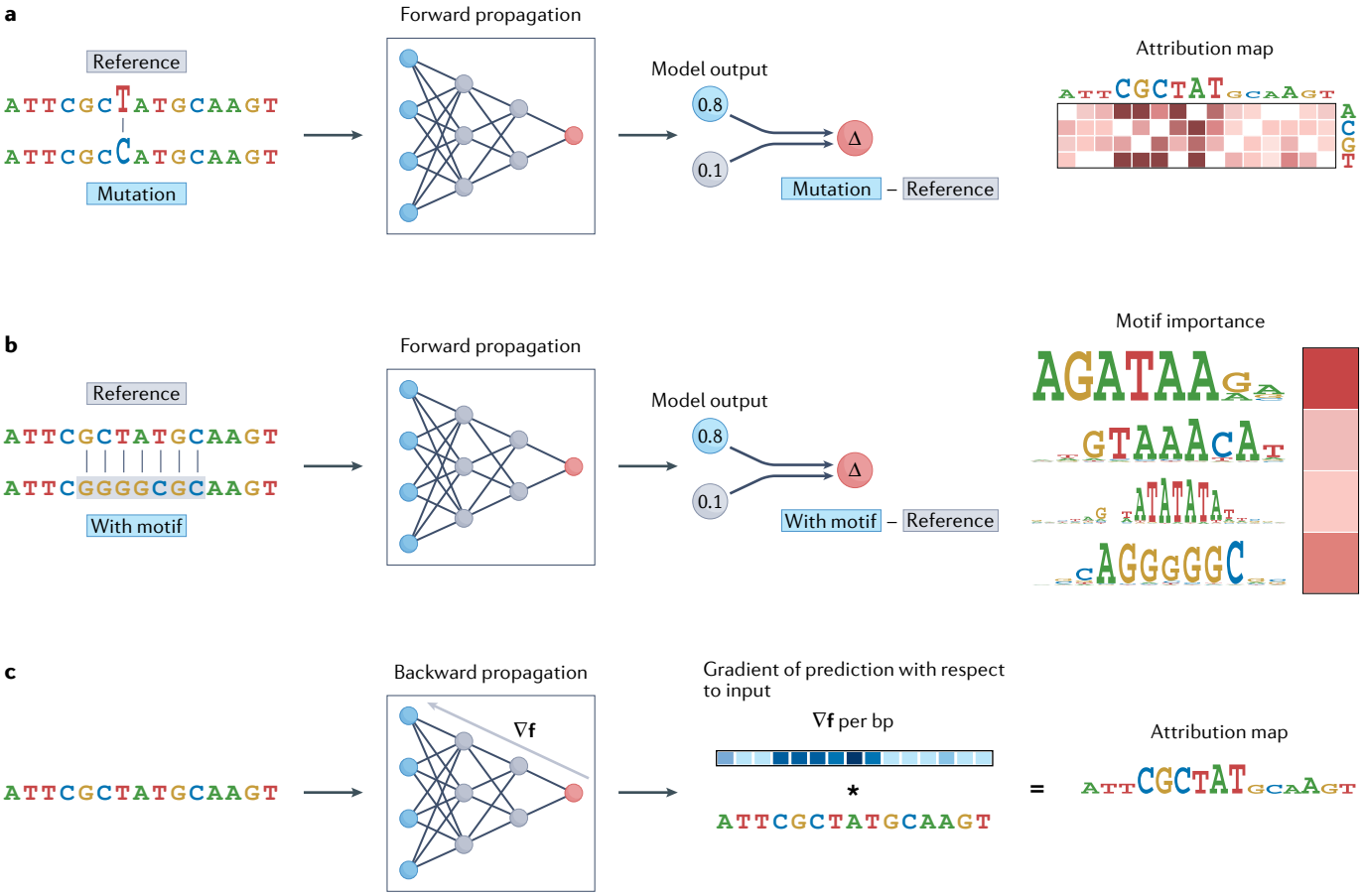


Fig. 3 | **Approaches to propagation-based interpretation. a** | In silico mutagenesis setting. Mutation of a target base pair to each of the alternative three nucleotides leads to a change in the model's prediction. These changes are collected for all three options and used to construct an attribution map (heat-map on the right). Areas with higher performance changes correspond to informative features, such as transcription factor motifs. **b** | The importance of a motif of interest can be inferred by embedding it in a random sequence and checking the difference in the model's predictions from the original sequence. **c** | Calculating the gradient of the model's output with respect to the input sequence using backward propagation approximates the importance of each position for the final prediction. This gradient is usually multiplied by the input sequence to focus only on present nucleotides (right). Part **a** adapted from REF.[4], Springer Nature Limited.

vector for a given input and the input values produces a gradient-based attribution map (FIG. 3c). More recent approaches, such as GradCAM[57] and its variants[47], attempt to improve the quality of gradient approximation through additional local weighting of feature importance.

A major shortcoming of gradient-based methods is the so-called saturation problem. Redundancy in input features can result in underestimation of importance values. For instance, a regulatory sequence may contain multiple copies of the same binding motif. The overall sensitivity of the model is distributed across the multiple copies. Reference-based gradient methods, which include the widely used integrated gradients[23,58], are designed to address this problem. Instead of evaluating the gradient at a single point, the integrated gradients method considers integrating the gradient of the model F along the line $l(\alpha) = \alpha X + (1 - \alpha)\underline{X}$ where $\underline{X}$ represents a baseline (also called the reference) version of X and $\alpha \in [0, 1]$. This procedure allows for a more sensitive assessment of input perturbation on model predictions. For genomic sequences, a mononucleotide or dinucleotide shuffled input sequence provides a reasonable baseline[59,60]. The integrated gradients method does not have a saturation problem because gradients are computed with respect to the baseline $\underline{X}$, which, in the TF binding example, will lack both binding sites. An improved variant of integrated gradients, termed enhanced integrated gradients (EIG), was proposed to additionally consider all nonlinear paths between the baseline and the sample. EIG was benchmarked against various types of baseline, and used to identify A1CF as a novel regulator of the liver splicing programme[61].

A general issue with gradient-based methods is numerical instability in computing gradients of DNN models via back-propagation. Also, even for a continuous neural network model F, the gradient of F may possess discontinuities, for example, if it is composed of rectified linear unit (ReLU) activations. This can result in large jumps in the gradient of F for small changes in the input, leading to unreliable gradient estimation. DeepLIFT is a back-propagation method that directly addresses this shortcoming[60]. DeepLIFT was previously used to reveal key markers of pluripotency[10], and to

Rectified linear unit
(ReLU). A common type of nonlinear activation function applied to the output of hidden units, which zeros-out the negative part of the output.

identify novel TF binding motifs that contribute to epidermis differentiation[9].

Unlike integrated gradients, DeepLIFT operates on the differences between the input and the reference at every node in the network. Attribution scores approximating the gradient of F are obtained by back-propagating these differences in one pass from the output to input nodes. DeepLIFT has multiple rules for assigning attribution scores, one of which can be shown to be an approximation of the Shapley values[26]. Shapley values provide a sound mathematical framework for assigning feature importance proportional to their marginal contribution to a prediction result in cooperative game theory. They were adapted for deriving feature attributions and were shown to provide a theoretical basis for unifying several feature attribution algorithms including DeepLIFT and DeepSHAP[22,26,62].

From local propagation results to a global interpretation. To generalize from per-sequence attribution maps that are produced by propagation-based methods to reveal a global understanding of important motifs, the results across many input examples need to be aggregated. TFMoDisco is one such approach that is specifically designed for DNA input sequences and relies on clustering across attribution maps to identify globally important sequence motifs[63]. As one example, TFMoDisco was used to identify sequence features that are important for telomerase reverse transcriptase (*TERT*) promoter function[64]. However, the typical challenges of clustering remain in this aggregation problem as well, including uncertainty in setting the number of clusters, defining similarity metrics and adjusting the resolution of the clustering. A few recent approaches have been specifically designed for aggregating the results of partial occlusion-based methods that provide additional statistical guarantees[65,66]. Koo and colleagues[65] showed that such methods can reveal sequence features that are important for predicting RNA–protein interactions.

Identifying interactions between features

So far, we have highlighted approaches for identifying individually important features, such as TF motifs, that contribute to the model's prediction(s). However, the power of neural networks stems from the ability to model nonlinear interactions between features. In the context of gene regulation, it is widely recognized that interactions such as cooperativity between TFs can account for activity beyond that attached to each TF in isolation. Researchers are therefore motivated to detect interactions between features identified by the neural network that are likely to represent such cooperative TF behaviour. In this section, we describe how the methods for producing local interpretation described previously can explain interactions between features.

Model-based identification of interactions. As deeper layers of neural networks hierarchically assemble features learned in lower layers, one obvious strategy for identifying interactions is to examine deeper layer neurons. For instance, by examining sequences that activate second-layer filters, Bogard and colleagues[67] identified and validated interaction effects between RNA binding proteins involved in alternative polyadenylation. For this type of task in computer vision applications, optimization-based methods that search among a large set of random inputs (such as images) for those that maximally activate a given hidden neuron tend to work best[68].

A model that uses self-attention, a popular variant of the general attention mechanism[40,69], directly represents interactions between base pairs in a sequence-to-activity model[43,70]. Self-attention can be used to replace the pooling layer in a CNN model. In this layer, the model produces an $L \times L$ attention matrix that represents the attention paid between each pair of the L base pairs of the input sequence. This attention matrix can be used as a representation of cooperativity between base pairs. For example, in a model that aims to predict the expression of a gene from a wide (for example, 200 kb) region around a transcription start site, self-attention may represent enhancer–promoter interactions responsible for regulating that gene[16,71]. In another example, in a model that aims to predict TF binding, self-attention between pairs of positions in the input sequence may represent cooperative TF binding[16,72] or more generally highlight regions in input sequences that are important for TF binding[73].

Explaining interactions through mathematical propagation. Propagation approaches — both forward and backward — can be tailored for interpretation of interactions within models.

ISM becomes prohibitively computationally expensive when used to estimate potential interactions. This is because a separate input must be tested for each pair of features, and thus the number of tests grows quadratically as the number of features increases linearly. However, ISM can be feasibly applied in a restricted fashion, for instance, for a targeted analysis of sequences that contain a specific motif pair[9] or more generally through insertion of two motifs into random sequence. In this setting, ISM mutates one motif — either a single position or the whole motif — and compares the resulting prediction with that from the intact sequence (FIG. 4a). Applying this approach allows for discovery of additive and non-additive effects between motifs, and also for assessment of the influence of motif spacing[9,54]. For example, to assess motif spacing dependencies, two features (TF motifs or k-mers) can be inserted into random genomic sequence, then one feature can be held in a fixed position (for example, at the centre of the sequence) while the second feature is slid iteratively across the sequence. This approach was effective in identifying position-dependent interactions between NANONG and SOX2 during stem cell differentiation[10]. This procedure can also be applied to two identical motifs to test multimerization. To overcome noise or statistical instability, the process can be applied to multiple sequences. Another variant of this idea involves combining forward and backward propagation methods, in an approach termed deep feature interaction maps (DFIM)[53] (FIG. 4b). DFIM represents a trade-off between forward and backward propagation algorithms,

Self-attention
A type of attention mechanism in which every part of the input is compared with every other part, including itself.

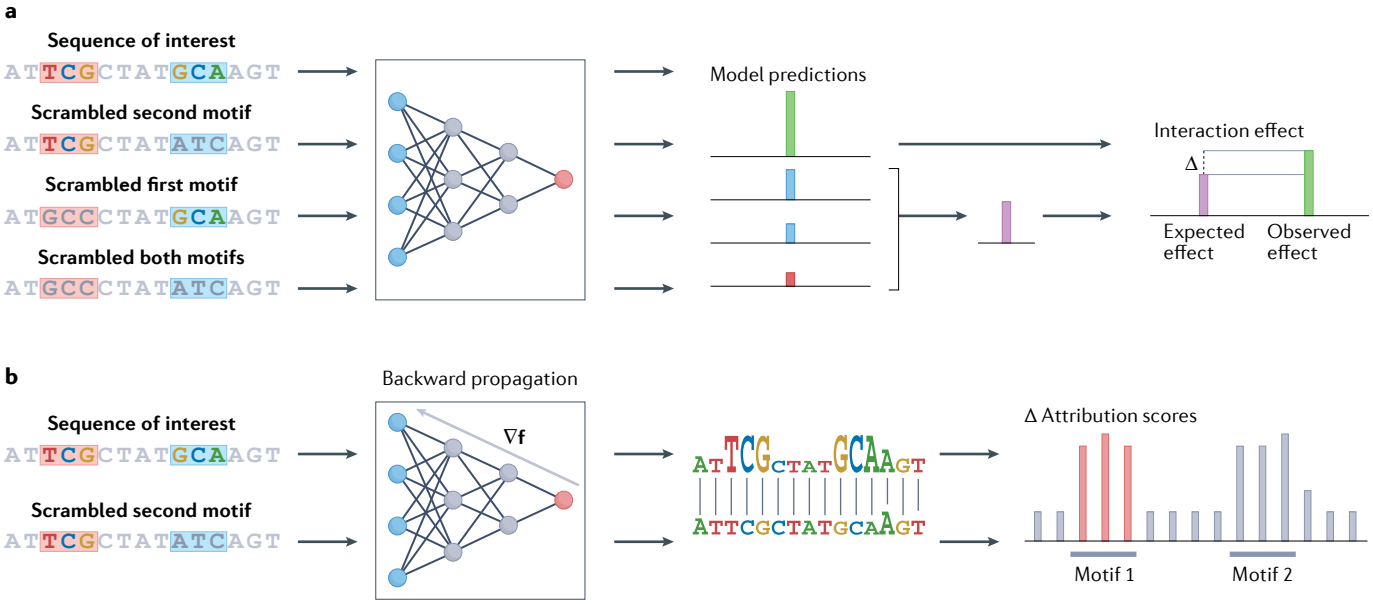


Fig. 4 | **Approaches to reveal interactions between features in model performance. a** | A scheme of the combinatorial motif in silico perturbation analysis. A given sequence with two motif instances is changed so that either one or both motifs are mutated. The network's output is recorded for each alternative and compared with the reference. The prediction change for the joint mutation is compared with the sum of changes for the marginal mutations to infer the potential interaction effect. **b** | Deep feature interaction maps (DFIMs) compare the attribution scores computed for the reference sequence with both motif instances with the scores of an altered sequence, in which one of the motifs is scrambled. If the scores differ significantly, it might imply dependency of the motifs on each other.

as it requires several passes through the network, which grows linearly in the number of features, in contrast to the quadratic growth of naive second-order ISM. In the DFIM approach, a feature interaction score (FIS) between any pair of features (source feature and target feature) in an input DNA sequence is computed as the change in the importance score of the target feature when the source feature is perturbed, while keeping all the other features in the sequence intact.

Back-propagation-based methods such as integrated gradients can also be extended to exhaustively assess pairwise feature interactions. Instead of first-order derivatives, one can assess the second-order derivative of the model's prediction, termed integrated Hessians[74]. This approach yields a matrix of second derivatives H, in which $H(i, j)$ corresponds to the degree to which feature i modulates the influence of feature j or vice versa. A high value of $H(i, j)$ indicates an interaction between these two features. Notably, integrated Hessians cannot be applied to neural networks that use an activation function without an informative second derivative, such as the widely used ReLU activation function[75]. Using integrated Hessians could plausibly enable efficient evaluation of interaction effects — for example, revealing the interactions between every pair of base pairs in a DNA sequence — but they have yet to be systematically tested for genomics applications.

Using prior knowledge for transparent models

In comparison with the input nodes, interpreting hidden nodes in deeper layers is more challenging because each corresponds to a complex nonlinear function of the inputs and may not correspond to any observable quantity. A transparent neural network model is one in which the hidden nodes are constructed to physically correspond to biological units at a level of granularity that is helpful for human understanding[76] (FIG. 5).

To build models that have inherently interpretable units, one needs to use prior knowledge to design the network architecture. For example, one can initialize filters according to known TF binding motifs[12] (FIG. 5a). The deeper layers construct combinations of these input features to model parts of biological systems at higher levels of abstraction. For example, the second layer may represent co-binding relationships between motifs, a higher layer may correspond to biological pathways and so on. Although prior knowledge about interactions between TFs and pathway memberships are sparse in general, previous work has used partial knowledge from sources such as Gene Ontology[77] to either hard-code these as binary edges between neurons or more softly encourage edges between units across different layers through regularization[78]. Inspecting a model trained in this way can grant insight into the presence of such prior interactions in a given context (FIG. 5b). Although building 'transparent' models, such as probabilistic graphical models, was a focus before the recent popularity of neural networks, there are two reasons that can motivate the creation of such models with neural networks. First, accompanying optimization algorithms can be used to learn nonlinear interactions between interpretable units, and second, computational infrastructure can be re-used to optimize and apply models on large datasets that would not be feasible otherwise.

One of the first transparent neural network models in biology to use neural network computational

Activation function
A function applied to the output of neurons, typically to model non-linearity.

Regularization
A common machine learning scheme that controls model expressivity by including a term in the objective function that penalizes model complexity.

infrastructure is called DCell[76]. DCell models the relationship between genotype and growth rate in yeast. The structure of this model consists of first-layer nodes that represent genes and second-layer nodes that are constructed as functional groups defined hierarchically according to the Gene Ontology database. The model's transparency assisted in explaining the link between mutations that disrupt the genes *PMT1* and *IRE1* that impact growth rate. Both genes were linked to more than 200 Gene Ontology subsystems, but only the endoplasmic reticulum unfolded protein response node was substantially affected compared with the wild type. Experimental validation confirmed the correlation between this subsystem and the double mutants. More recently, the P-NET[79] model was developed to create transparent representations of molecular variables from multi-omic datasets. The model's transparency helped the authors to discover novel biomarkers (such as MDM4) of metastatic prostate cancer. In a similar spirit to DCell and P-NET, others have created neural networks in which units and parameters have a biophysical interpretation[80,81].

Recently Agarwal and colleagues[82] introduced a simplified neural network architecture called neural additive models (NAMs), which sacrifice some of the capacity of traditional neural network models to gain direct parameter interpretation. Although they are only able to detect linear relationships between features (which can still be learned de novo), their early application to genomics shows competitive performance relative to more complex models in some settings[83]. However, the main drawback is the inability of such models to learn potential interactions between features (such as nonlinear relationships between pairs of motifs, or dependence of motif effect on spacing and flanking sequence).

The primary limitation of this transparent modelling approach is that it requires the user to have systematic prior information. It may not be directly applicable to tasks for which the entities or their hierarchical structure are poorly characterized, or to tasks for which such entities are not assayable. The user must also take caution in interpreting the nodes of transparent models, because the approaches described here may fail to enforce the equivalence between nodes and biological entities. For example, a CNN initialized with TF motifs may see its filters drift away from those motifs during training. Also, techniques for enforcing transparency can degrade performance. For example, hard-coding of pairwise edges to encode TF–TF interactions or gene–pathway memberships are likely to reduce accuracy relative to an unconstrained model (although this reduction may be less severe when using softer regularization techniques).

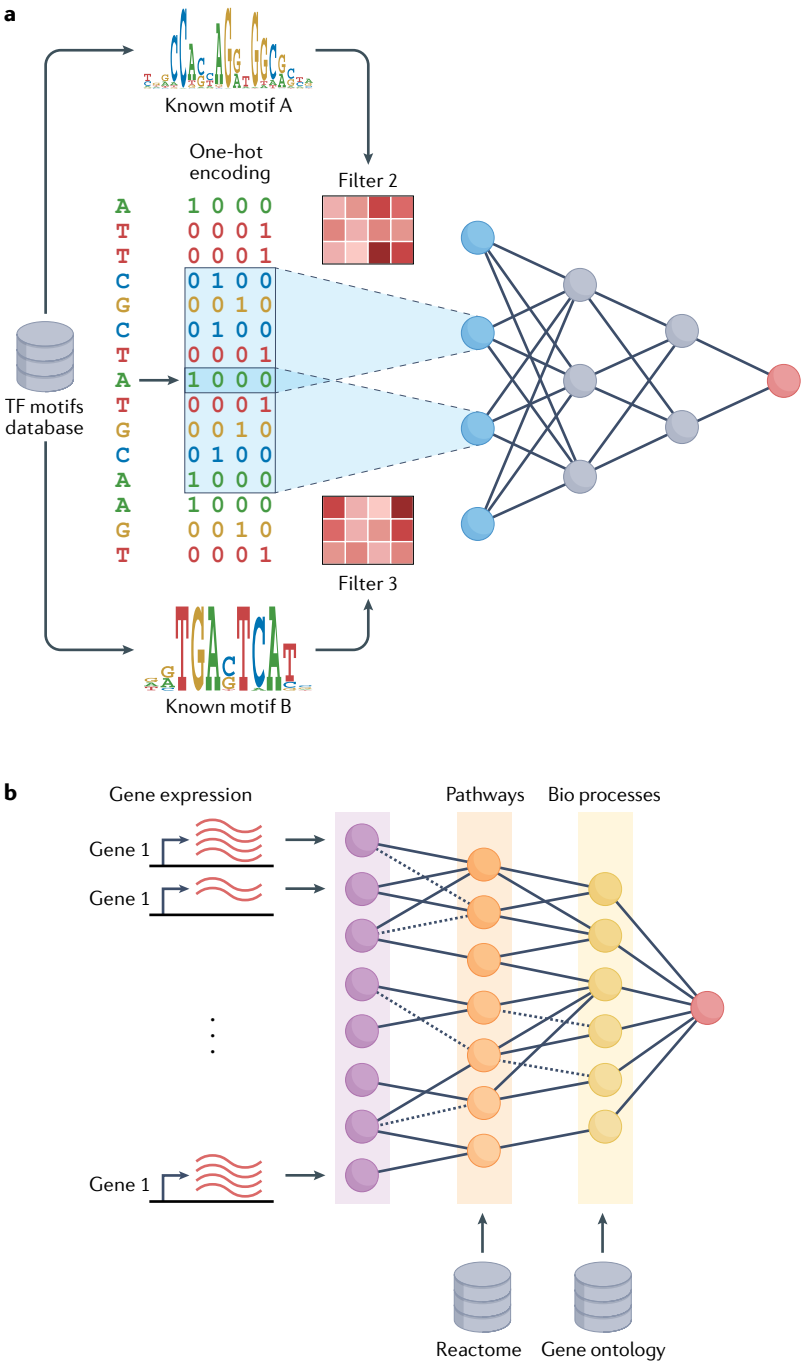


Fig. 5 | **Using prior knowledge to construct transparent neural networks.** **a** | Initializing first-layer convolutional filters (cyan circles) with known motifs from existing databases helps to make convolutional neural networks (CNNs) internally more interpretable. **b** | Transparent models are built using biological prior knowledge about gene regulatory hierarchies and functional annotations. Models are typically initialized with sparse edges (solid lines) representing biologically established connections (for example, a pathway and its gene member). The model may learn additional connections (dashed lines) representing novel associations. TF, transcription factor.

Conclusion and future perspectives

In this Review we focus on sequence-to-activity models, yet the described xAI methods apply broadly across applications of deep learning in genomics, including to models of phenotype, gene expression and other multi-omics measurements, and single-cell measurements. However, interpretation may be more challenging in cases in which, unlike sequence models, the direction of causality is unclear.

The goal and utility of model interpretation depend strongly on the target application, which should therefore guide the choice of interpretation approach. When a user needs an overall understanding of a whole biological process, such as when aiming to make experimentally testable mechanistic hypotheses, a global interpretation

approach is appropriate. When a user needs to understand an individual prediction made by a model, such as when identifying genetic risk of disease, a local interpretation approach is appropriate.

Interpretation methods ultimately amount to identifying and extrapolating from predictive patterns discovered within the training data. Spurious correlations present in the training data can present themselves in interpretation of the downstream model. In particular, when the model is trained on a dataset that is small or contains artefacts, a predictive model can achieve high accuracy by learning non-biological 'shortcuts', and thus interpretation will not yield meaningful biology. Interpretations can provide a way to identify such spurious correlations; for example, in a medical imaging study, model interpretation enabled the researchers to identify hidden batch effects that were being used by the model to make predictions[84]. Thus, interpretation can also be a tool for debugging predictive models.

The quality of biological insight granted by model interpretation depends crucially on the prediction accuracy of the interpreted model. For example, when performing ISM, a perfect model would by definition yield the correct consequence of a given mutation, whereas a low-accuracy model may yield spurious inferences. Interpretations should be used with care when approaching tasks in which state-of-the-art prediction accuracy is far from perfect, such as when predicting gene expression or phenotype. This problem is particularly acute when interpreting higher-order effects of a model such as feature interactions, because any combination of feature values occurs in just a small number of training examples.

The reliability of interpretation is hampered by the 'un-identifiability' of the interpreted model[85,86]. That is, because DNN models typically contain far more parameters than training examples, the training procedure is not guaranteed to find the best possible model. This means that the model parameters are sensitive to random selection of training examples and the initialization parameters. Model-based interpretations are most sensitive to this un-identifiability issue; however, we believe this phenomenon affects all interpretation techniques to varying degrees. Thus, interpretation must be used with care and with the understanding that some properties of a model, and the resulting features, may arise owing to chance. This issue can be alleviated by comparing multiple datasets and multiple training initializations, but this is sometimes too costly to perform.

At the time of writing, there is no consensus regarding which xAI approaches are most effective. Part of this variability stems from the variety of goals held by users of these approaches. However, even when the goal is fixed, different approaches can each yield different insights. Interpretation methods sometimes yield results that are biologically nonsensical, perhaps for the reasons above or others that are not yet understood. We expect that as this field matures, best practices will become established and integrated into accessible analysis tools.

As the size and availability of biological datasets grows, it becomes more important to investigate complex relationships between features using models. Distilling insight from these models requires effective xAI methods. Thus, xAI will have an ever-more central role in genomics.

Published online 3 October 2022

1. Angermueller, C., Pärnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Mol. Syst. Biol.* **12**, 878 (2016).
2. Zou, J. et al. A primer on deep learning in genomics. *Nat. Genet.* **51**, 12–18 (2019).
3. Ching, T. et al. Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 20170387 (2018).
4. Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* **20**, 389–403 (2019).
This review paper provides a succinct overview of deep learning in genomics, suitable for biomedical researchers.
5. Molnar, C., Casalicchio, G. & Bischl, B. Interpretable machine learning – a brief history, state-of-the-art and challenges. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2010.09337 (2020).
This textbook provides an overview of approaches for interpreting machine learning models.
6. Toneyan, S., Tang, Z. & Koo, P. K. Evaluating deep learning for predicting epigenomic profiles. Preprint at *bioRxiv* https://doi.org/10.1101/2022.04.29.490059 (2022).
7. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–934 (2015).
One of the first papers to use a sequence-to-activity neural network for a broad class of regulatory genomics tasks.
8. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
One of the first papers to use a sequence-to-activity neural network for a broad class of regulatory genomics tasks.
9. Kim, D. S. et al. The dynamic, combinatorial cis-regulatory lexicon of epidermal differentiation. *Nat. Genet.* **53**, 1564–1576 (2021).
10. Avsec, Ž. et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
A pioneering paper that shows how non-linear relationship between motifs and context-dependent spacing can be derived using various post-hoc model interpretation techniques.
11. Maslova, A. et al. Deep learning of immune cell differentiation. *Proc. Natl Acad. Sci. USA* **117**, 25655–25666 (2020).
12. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **44**, e107 (2016).
A paper that proposes one of the first hybrid CNN–RNN models in genomics applications.
13. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
This study introduces the application of CNNs to genomics.
14. 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).
15. Kelley, D. R. et al. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.* **28**, 739–750 (2018).
16. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
A first paper that introduces transformers and attention mechanism for improved prediction of gene expression from large input sequences.
17. Tasaki, S., Gaiteri, C., Mostafavi, S. & Wang, Y. Deep learning decodes the principles of differential gene expression. *Nat. Mach. Intell.* **2**, 376–386 (2020).
18. Xiong, H. Y. et al. RNA splicing. The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347**, 1254806 (2015).
19. Leung, M. K. K., Xiong, H. Y., Lee, L. J. & Frey, B. J. Deep learning of the tissue-regulated splicing code. *Bioinformatics* **30**, i121–i129 (2014).
20. Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3D genome folding from DNA sequence with Akita. *Nat. Methods* **17**, 1111–1117 (2020).
21. Lanchantin, J., Singh, R., Wang, B. & Qi, Y. Deep motif dashboard: visualizing and understanding genomic sequences using deep neural networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1608.03644 (2016).
22. Covert, I., Lundberg, S. & Lee, S.-I. Explaining by removing: a unified framework for model explanation. *J. Mach. Learn. Res.* **22**, 1–90 (2021).
This paper presents a unified framework for understanding feature attribution methods.
23. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1703.01365 (2017).
24. Ivanovs, M., Kadikis, R. & Ozols, K. Perturbation-based methods for explaining deep neural networks: a survey. *Pattern Recognit. Lett.* **150**, 228–234 (2021).
25. Rozemberczki, B. et al. The Shapley value in machine learning. in *Proc. 31st Int. Jt Conf. Artificial Intelligence* (ed. De Raedt, L.) 5572–5579 (IJCAI, 2022).
26. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In *Proc. 31st Int. Conf. Neural Information Processing Systems* (eds von Luxburg, U. et al.) vol. 30 4768–4777 (NIPS, 2017).
This paper presents a unified framework for interpretation and presents DeepSHAP.
27. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
28. Bau, D. et al. Understanding the role of individual units in a deep neural network. *Proc. Natl Acad. Sci. USA* **117**, 30071–30078 (2020).
29. Luo, X., Tu, X., Ding, Y., Gao, G. & Deng, M. Expectation pooling: an effective and interpretable pooling method for predicting DNA-protein binding. *Bioinformatics* **36**, 1405–1412 (2020).
30. Cuperus, J. et al. Deep learning of the regulatory grammar of yeast 5′ untranslated regions from

500,000 random sequences. Preprint at *bioRxiv* https://doi.org/10.1101/137547 (2017).
31. Min, X. et al. Predicting enhancers with deep convolutional neural networks. *BMC Bioinform.* **18** (Suppl. 13), 478 (2017).
32. Castro-Mondragon, J. A. et al. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.* **50**, D165–D173 (2022).
33. Weirauch, M. T. et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell* **158**, 1431–1443 (2014).
34. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1207.0580 (2012).
35. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
A machine learning textbook that focuses on DNN models.
36. Koo, P. K. & Ploenzke, M. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *Nat. Mach. Intell.* **3**, 258–266 (2021).
37. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**, 301–320 (2005).
38. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.* **18**, 851–869 (2017).
39. Chaudhari, S., Mithal, V., Polatkan, G. & Ramanath, R. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol.* **12**, 1–32 (2021).
40. Vaswani, A. et al. Attention is all you need. in *Proc. 31st Int. Conf. Neural Information Processing Systems* (eds von Luxburg, U., Guyon, I., Bengio, S., Wallach, H. & Fergus, R.) vol. 30 5998-6008 (NIPS, 2017).
41. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1409.0473 (2014).
42. Park, S. et al. Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Sci. Rep.* **10**, 13413 (2020).
43. Mao, W., Kostka, D. & Chikina, M. Modeling enhancer–promoter interactions with attention-based neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/219667 (2017).
44. Serrano, S. & Smith, N. A. Is attention interpretable? In *Proc. 57th Annual Meeting of the Association for Computational Linguistics* (eds Korhonen, A et al.) 2931–2951 (Association for Computational Linguistics, 2019).
45. Samek, W., Binder, A., Montavon, G., Bach, S. & Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 2660–2673 (2017).
46. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1312.6034 (2013).
47. Zheng, A. et al. Deep neural networks identify sequence context features predictive of transcription factor binding. *Nat. Mach. Intell.* **3**, 172–180 (2021).
48. Cochran, K. et al. Domain-adaptive neural networks improve cross-species prediction of transcription factor binding. *Genome Res.* **32**, 512–523 (2022).
49. Nair, S., Shrikumar, A. & Kundaje, A. fastISM: performant in-silico saturation mutagenesis for convolutional neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/2020.10.13.337147 (2020).
50. Schreiber, J., Nair, S., Balsubramani, A. & Kundaje, A. Accelerating in-silico saturation mutagenesis using compressed sensing. Preprint at *bioRxiv* https://doi.org/10.1101/2021.11.08.467498 (2021).
51. Washburn, J. D. et al. Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence. *Proc. Natl Acad. Sci. USA* **116**, 5542–5549 (2019).
52. Yuan, H. & Kelley, D. R. scBasset: sequence-based modeling of single cell ATAC-seq using convolutional neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/2021.09.08.459495 (2021).
53. Greenside, P., Shimko, T., Fordyce, P. & Kundaje, A. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics* **34**, i629–i637 (2018).
A first paper describing how occlusion can be used to detect significant motif–motif epistasis.
54. de Almeida, B. P., 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).
55. Prakash, E. I., Shrikumar, A. & Kundaje, A. Towards more realistic simulated datasets for benchmarking deep learning models in regulatory genomics. In *Proc. 16th Machine Learning in Computational Biology meeting* (eds Knowles, D. A. et al.) vol. 165, 58–77 (PMLR, 2022).
56. Finnegan, A. & Song, J. S. Maximum entropy methods for extracting the learned features of deep neural networks. *PLoS Comput. Biol.* **13**, e1005836 (2017).
57. Selvaraju, R. R. et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput Vis.* **128**, 336–359 (2020).
58. Sundararajan, M., Taly, A. & Yan, Q. Gradients of counterfactuals. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1611.02639 (2016).
59. Huang, D. et al. Weakly supervised learning of RNA modifications from low-resolution epitranscriptome data. *Bioinformatics* **37**, i222–i230 (2021).
60. Shrikumar, A., Greenside, P. & Kundaje, A. Learning important features through propagating activation differences. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) vol. 70, 3145–3153 (PMLR, 2017).
A technical paper that describes the DeepLIFT feature attribution method, one of the most widely used propagation-based methods in genomics.
61. Jha, A., K Aicher, J., Gazzara, M. R., Singh, D. & Barash, Y. Enhanced Integrated Gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome Biol.* **21**, 149 (2020).
62. Jethani, N., Sudarshan, M., Covert, I., Lee, S.-I. & Ranganath, R. FastSHAP: real-time Shapley value estimation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2107.07436 (2021).
63. Shrikumar, A. et al. Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5.6.5. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1811.00416 (2018).
64. Sahu, B. et al. Sequence determinants of human gene regulatory elements. *Nat. Genet.* **54**, 283–294 (2022).
65. Koo, P. K., Majdandzic, A., Ploenzke, M., Anand, P. & Paul, S. B. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Comput. Biol.* **17**, e1008925 (2021).
66. Hammelman, J. & Gifford, D. K. Discovering differential genome sequence activity with interpretable and efficient deep learning. *PLoS Comput. Biol.* **17**, e1009282 (2021).
67. Bogard, N., Linder, J., Rosenberg, A. B. & Seelig, G. A deep neural network for predicting and engineering alternative polyadenylation. *Cell* **178**, 91–106.e23 (2019).
68. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. & Lipson, H. Understanding neural networks through deep visualization. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1506.06579 (2015).
69. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
70. Tao, Y. et al. Interpretable deep learning for chromatin-informed inference of transcriptional programs driven by somatic alterations across cancers. Preprint at *bioRxiv* https://doi.org/10.1101/2021.09.07.459263 (2021).
71. Karbalayghareh, A., Sahin, M. & Leslie, C. S. Chromatin interaction-aware gene regulatory modeling with graph attention networks. *Genome Res.* **32**, 930–944 (2022).
72. Ullah, F. & Ben-Hur, A. A self-attention model for inferring cooperativity between regulatory features. *Nucleic Acids Res.* **49**, e77 (2021).
73. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* **37**, 2112–2120 (2021).
74. Janizek, J. D., Sturmfels, P. & Lee, S.-I. Explaining explanations: axiomatic feature interactions for deep networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2002.04138 (2020).
75. Dombrowski, A.-K. et al. Explanations can be manipulated and geometry is to blame. *Adv. Neural Inf. Process. Syst.* **32**, 13567–13578 (2019).
76. Ma, J. et al. Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).
This paper presents one of the first 'transparent neural network' models in genomics.
77. The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29 (2000).
78. Fortelny, N. & Bock, C. Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. *Genome Biol.* **21**, 190 (2020).
79. Elmarakeby, H. A. et al. Biologically informed deep neural network for prostate cancer discovery. *Nature* **598**, 348–352 (2021).
80. Tareen, A. & Kinney, J. B. Biophysical models of cis-regulation as interpretable neural networks. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2001.03560 (2019).
81. Liu, Y., Barr, K. & Reinitz, J. Fully interpretable deep learning model of transcriptional control. *Bioinformatics* **36**, i499–i507 (2020).
82. Agarwal, R. et al. Neural additive models: interpretable machine learning with neural nets. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2004.13912 (2020).
83. Novakovsky, G., Fornes, O., Saraswat, M., Mostafavi, S. & Wasserman, W. W. ExplaiNN: interpretable and transparent neural networks for genomics. Preprint at *bioRxiv* https://doi.org/10.1101/2022.05.20.492818 (2022).
84. DeGrave, A. J., Janizek, J. D. & Lee, S.-I. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nat. Mach. Intell.* **3**, 610–619 (2021).
85. Heil, B. J. et al. Reproducibility standards for machine learning in the life sciences. *Nat. Methods* **18**, 1132–1135 (2021).
86. Haibe-Kains, B. et al. Transparency and reproducibility in artificial intelligence. *Nature* **586**, E14–E16 (2020).
87. Leman, D. V., Parshikov, A. F., Georgiev, P. G. & Maksimenko, O. G. Organization of the *Drosophila melanogaster* SF1 insulator and its role in transcription regulation in transgenic lines. *Russ. J. Genet.* **50**, 341–347 (2014).
88. Lambert, S. A. et al. The human transcription factors. *Cell* **172**, 650–665 (2018).
89. Klemm, S. L., Shipony, Z. & Greenleaf, W. J. Chromatin accessibility and the regulatory epigenome. *Nat. Rev. Genet.* **20**, 207–220 (2019).
90. Carter, B. & Zhao, K. The epigenetic basis of cellular heterogeneity. *Nat. Rev. Genet.* **22**, 235–250 (2021).
91. Rowley, M. J. & Corces, V. G. Organizational principles of 3D genome architecture. *Nat. Rev. Genet.* **19**, 789–800 (2018).
92. Stormo, G. D. & Zhao, Y. Determining the specificity of protein-DNA interactions. *Nat. Rev. Genet.* **11**, 751–760 (2010).
93. Xu, C. & Jackson, S. A. Machine learning and complex biological data. *Genome Biol.* **20**, 76 (2019).
94. Koo, P. K. & Ploenzke, M. Deep learning for inferring transcription factor binding sites. *Curr. Opin. Syst. Biol.* **19**, 16–23 (2020).
95. Whalen, S., Schreiber, J., Noble, W. S. & Pollard, K. S. Navigating the pitfalls of applying machine learning in genomics. *Nat. Rev. Genet.* **23**, 169–181 (2022).

Acknowledgements
N.D. acknowledges the support of the Pacific Institute for the Mathematical Sciences (PIMS) Postdoctoral Fellowship program. W.W.W. acknowledges support from Natural Sciences and Engineering Research Council of Canada (NSERC) and the British Columbia (BC) Children's Hospital Foundation. S.M. acknowledges support from the Canadian Institute for Advanced Research (CIFAR). M.W.L. acknowledges support from Genome Canada, Genome BC, NSERC and Health Research BC. The authors thank W. Stafford Noble for helpful comments on the manuscript.

Author contributions
All authors contributed to all aspects of the article.

Competing interests
The authors declare no competing interests.

Peer review information
Nature Reviews Genetics thanks Shaun Mahony and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© Springer Nature Limited 2022