Opening the black box: interpretable machine learning for geneticists

Christina B. Azodi[1,2,3¶], Jiliang Tang[4], Shin-Han Shiu[1,2,5¶]

[1] Department of Plant Biology, Michigan State University, East Lansing, MI, USA

[2] The DOE Great Lakes Bioenergy Research Center, Michigan State University, East Lansing, MI, USA

[3] Bioinformatics and Cellular Genomics, St. Vincent's Institute of Medical Research, Fitzroy, Victoria, Australia

[4] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA

[5] Department of Computational Mathematics, Science, and Engineering, Michigan State University, East Lansing, MI, USA

Corresponding authors:

Christina B. Azodi
St. Vincent's Institute of Medical Research
9 Princes Street
Fitzroy, Victoria, 3065, Australia
Tel: +61 04 3396 7476
E-mail: cazodi@svi.edu.au

Shin-Han Shiu
Michigan State University
Plant Biology Laboratories
612 Wilson Road, Room 166
East Lansing, MI 48824-1312, USA
Tel: +1-517-353-7196
E-mail: shius@msu.edu

Key words: interpretable machine learning, deep learning, predictive biology

Abstract

Machine learning (ML) has emerged as a critical tool for making sense of the growing amount of genetic and genomic data available because of its ability to find complex patterns in high dimensional and heterogeneous data. While the complexity of ML models is what makes them powerful, it also makes them difficult to interpret. Fortunately, recent efforts to develop approaches that make the inner workings of ML models understandable *to humans* have improved our ability to make novel biological insights using ML. Here we discuss the importance of interpretable ML, different strategies for interpreting ML models, and examples of how these strategies have been applied. Finally, we identify challenges and promising future directions for interpretable ML in genetics and genomics.

Highlights

- Machine learning (ML) has emerged as a powerful tool for harnessing big biological data.
- The complex structure underlying ML models means that their inner logic is not readily intelligible to a human, hence the common critique of ML models as black boxes.
- However, advances in the field of interpretable ML have made it possible to identify important patterns and features underlying a ML model using various strategies.
- These interpretation strategies have been successfully applied by researchers in genetics and genomics to derive novel biological insights from ML models.
- This area of research is becoming increasingly important as more complex and difficult to interpret ML approaches (i.e. deep learning) are being adopted by biologists.

doi:10.20944/preprints202002.0239.v1

Glossary

Algorithm: The procedure taken to solve a problem/build a model.

Decision tree: A model made up of a series of branching true/false questions.

Deep Learning: A subset of ML algorithms inspired by the structure of the brain that can find complex, nonlinear patterns in data.

Feature: An explanatory (i.e. independent) variable during modeling.

Global interpretation: A ML interpretation that explains the overall relationship between the features and the label for all instances.

Instance: A single example from which the model will learn or be applied to.

Interpretable: Capable of being understood by a human.

Label: The variable to be predicted (i.e. the dependent variable).

Local interpretation: A ML interpretation that explains the relationship between the features and the label for one or a subset of instances.

Machine learning: Computational models that learn from data without being explicitly programmed.

Model: The set of patterns learned for a specific problem, where given input (i.e. instances and their features) the model will generate an output (i.e. prediction).

Model performance: A quantitative evaluation of the model's ability to correctly predict labels.

Parameters: Variables in an ML model whose values are estimated/optimized during training.

Perturbing strategies: A family of interpretation strategies that measure how changes in the input data impact model predictions or performance.

Probing strategies: A family of interpretation strategies that involve inspecting the structure and parameters in a trained model.

Surrogate strategies: A family of interpretation strategies that involve training an inherently interpretable model (e.g. a linear model) using the same data as a black-box model to serve as the black-box model's surrogate.

Training: The process of identifying the best parameters to make up a model – the learning part in ML.

Importance of interpretable machine learning

Biological Big Data [1,2] has driven progresses in fields ranging from population genetics [3] to precision medicine [4]. Much of this progress is possible because of advances in **machine learning** (see Glossary; ML; **Box 1)** [5–10], "[a] field of study that gives computers the ability to learn without being explicitly programmed" [11]. ML works by identifying patterns in data in the form of a **model** that can be used to make predictions about new data. While powerful, ML also presents new challenges. For example, a common criticism is that the ML models are "black boxes", meaning their internal logic cannot be easily understood *by a human* [12]. Luckily, strategies to demystify the inner working of ML models are available and ever improving.

There are three major reasons – troubleshooting, novel insights, and trust – why **interpretable** ML model, or the ability to understand what logic is driving a model's prediction, is important (**Figure 1A**, Key Figure). First, ML models rarely perform well without tweaking or troubleshooting. Understanding how predictions are made is essential for identifying mistakes or biases in the input data and issues with how the model is **trained**. Second, an ML model with impressive performance may have identified biologically novel patterns. However, such insights will only be available if the model can be interpreted. Finally, we are unlikely to trust a prediction if we do not understand why it was made. For example, a doctor may not trust a ML diagnosis with no supporting justification out of concern that the model may be capturing artifacts or have unknown biases or limitations [13].

Overview of strategies for interpretable machine learning

A wide range of strategies for interpretable ML have been developed and applied to problems in genetics and genomics [14–16]. These strategies can be characterized based on if they are applicable to all ML **algorithms** (i.e. model-agnostic) or only to one or a subset of algorithms (i.e. model-specific). They can also be characterized based on if they provide **global** or **local** interpretations. Global interpretations involve explaining the overall relationship between **features** and **labels**. While local interpretations focus on explaining the prediction of an individual **instance**. For example, imagine you train an ML model to predict if a gene (an instance) is up-regulated after some treatment (the label) based on the presence or absence of a set of regulatory sequences (the features). A global interpretation strategy will tell you how important regulatory sequence X is for predicting up-regulation across all genes in your dataset. While a local interpretation strategy will tell you how important regulatory sequence X is for predicting gene Y as up-regulated. This means that the type of interpretation strategy you select will dictate what you will learn from your ML model, with different strategies possibly telling different stories. We should also emphasize that ML models identify association through correlation, thus ML interpretation strategies do not identify causal relationships between input features and labels. Instead, interpretations should be used to generate new hypotheses that can be tested experimentally. We will review three general ML interpretation strategies: **probing**, **perturbing**, and **surrogate strategies** (**Figure 1B**; [14,16]).

Probing strategies dissect the inner structure of ML models

Training an ML model involves identifying the set of **parameters** best able to predict the label of an instance (e.g. gene Y is up-regulated). After training, these parameters can be probed (or inspected) to better understand what the model learned. Probing strategies provide global interpretations with some exceptions (e.g. DeepLIFT, see below). Because type of parameters and structure of how they connect to each other varies by algorithm, probing strategies are model-specific. While probing strategies are straightforward for some ML algorithms (e.g. Support Vector Machine; SVM; and **decision tree**-based algorithms), this is not the case for more complex ML algorithms (e.g. **deep learning**).

Probing Support Vector Machine models

SVM is an algorithm that finds the hyperplane that best separates instances by their label when they are plotted in n-dimensional space (n = number of features). Training an SVM model to predict gene up-regulation using regulatory sequences as features means learning the combination of weights to apply to each regulatory sequence (i.e. coefficient weight) in order to make the best hyperplane (**Figure 2A**). SVM models can be trained to learn either linear or non-linear relationships between features and labels. While there are advanced methods for probing non-linear SVM models [17,18], in most biological applications of SVM, only linear SVM models are probed.

A trained linear SVM model is probed by extracting the coefficient weights that define the hyperplane (**Figure 2A**), where features assigned a higher absolute weight have a stronger relationship with the label and thus are more important for driving the prediction. For example, a linear SVM model was trained to classify simulated populations as being under positive or negative selection using genetic markers as features [19]. Genetic markers with large, positive coefficient weights in the SVM model were the same as those associated with positive selection using classical population genetics statistical tests (e.g. Tajima's *D*).

Importantly, SVM probing strategies (like other strategies discussed below), can provide an incomplete picture of feature importance. For example, two highly correlated features will split the weight between them, reducing their perceived importance. Or a feature with a strong non-linear relationship with the label may not be assigned a large weight by a linear SVM model and will therefore be missed when the trained model is probed.

Probing decision tree-based models

A decision tree is a set of true/false questions nested in a hierarchical structure. They are inherently interpretable because the content and order of each question can be directly observed. How well a true/false question separates instances by their label can also be quantified using metrics such as the mean decrease in node impurity. In **Figure 2B**, using the presence/absence of regulatory sequence "AACGT" to separate up- from down-regulated genes results in a decrease in the mean node impurity. Because single decision trees tend to perform poorly at predicting complex patterns, ensemble approaches (e.g. Random Forest, Gradient Tree Boosting [20]), where many decision trees are combined to generate one prediction, are often used. Ensemble decision-tree models can be probed by calculating the mean decrease in node impurity for each feature across all trees in the ensemble. This approach was used determine which DNA motifs were the most important for predicting if a gene would be differentially expressed under salt stress conditions in *Arabidopsis thaliana* [21].

The hierarchical structure of decision tree-based models means that interactions between features an be readily probed. For example, using a tool for finding stable feature interactions in Random Forest models [21] , Vervier and Michaelson identified interactions between genomic, transcriptomic, and epigenomic features that were predictive of deleterious genetic variants [23]. Specifically, that an interaction between the local GC content and the distance to the nearest expression Quantitative Trait Loci was important for predicting deleterious variants.

As with coefficient weights from SVM models, mean decrease impurity scores can be misleading when features are highly correlated. This score also tends to inflate continuous over categorical features, categorical features with a larger number of categories, and continuous features with a larger numeric range and should therefore be interpreted with caution when feature space is not uniform [24].

Probing deep learning networks

While the classical ML algorithms described above are readily interpretable, deep learning (**Box 2**) algorithms are being applied more and more in the ML community because they frequently outperform classical ML algorithms at modeling complex systems [25–27] and they can learn from raw data (e.g. whole DNA sequence) rather than user defined features (e.g. known regulatory sequences). However, there is often a tradeoff between predictability and interpretability [28], and this is certainly the case for deep learning [29]. Fortunately, there has been a substantial effort to develop new methods to interpret these complex models. First we describe three general approaches to calculate feature importance scores by probing deep learning models: connection weights-based, gradient-based, and activation level-based approaches (**Figure 2C**) [15].

Connection weight-based feature importance scores quantify the global relationship between each feature and the output by summing the learned weights assigned to connections between nodes in input-to-hidden, one or more hidden-to-hidden, and hidden-to-output layers for each input feature [30,31]. Following the path through the example artificial neural network (**Figure 2C)**, the connection weights (represented by line widths) between some features (e.g. f_1) and the output layer are larger than the connection weights between other features (e.g. f_3) and the output layer, indicating f_1 is more important for that model. This approach was used to determine which microRNA features were the most important for predicting the expression level of Smad7, a gene involved in disrupting a signaling process up-regulated in patients with breast cancer [32]. Connection weight-based feature importance scores can be misleading when feature are on different scales, when positive and negative connection weights cancel each other out, or when a connection has a large weight but is rarely activated (i.e. the nodes is rarely turned on) [33].

The gradient-based feature importance scores (a.k.a. Saliency) also quantify the global relationship between a feature and the output, but do so by calculating the gradient, or the change in the predicted output (e.g. the likelihood a gene is up-regulated) as small changes are made to the input feature (e.g. the frequency of regulatory sequence X). The gradient is calculated using a handy calculus trick, the partial derivative [34]. This approach was used to identify putative distal regulatory sequences in genomic regions where positive and negative gradient-based importance score peaks represented enhancer and silencer regions, respectively [35]. This approach is not useful when input features are categorical or when small changes in the feature value do not change the output prediction [33].

Finally, the activation level refers to the output value from a node after it has passed through a non-linear function (i.e. the activation function; see **Box 2**). Activation level-based feature importance scores provide a local interpretation for an instance of interest by comparing how much each feature activates nodes in the trained network compared to the feature values from a reference instance. A reference instance for an image classification model could be one that is solid white, while a reference for a model using a DNA sequence as instances could be an instance with the background nucleotide frequency at every site. This approach (coined DeepLIFT [33]), has been used in multiple biological studies [36–38]. For example, Zuallaert *et al.* used DeepLIFT to find nucleotide sequences important for predicting splice sites [37]. Because DeepLIFT probes activation levels rather than connection weights, it avoids the pitfall of the connection weight-based approach. Further, because it compares a specific instance to a reference, it also avoids the pitfalls of the gradient-based approach.

Another way to probe deep learning models is to learn what pattern each node in the network learned to identify (**Figure 2C**). This can be done by finding real or simulated instances that maximally activate that node, then the properties of those real or simulated instances can be used to interpret that node. For example, if the 10 DNA sequences that maximally activate node X (i.e. cause node X to have the maximum possible output value after passing through the activation function) all contain the motif ACGGTC, one could infer that node trained to find the ACGGTC motif. Because probing every node in every layer may produce results that are still too complex to interpret, dimensionality reduction techniques can be used to ease interpretation. For example, Esteva *et al.* used a dimensionality reduction technique to visualize the nodes in the last hidden layer of a convolutional neural network (see **Box 2**) trained to diagnose different types of skin cancer from photos [39]. This allowed them to visualize how well their convolutional neural network learned to separate different types of carcinomas.

Perturbing strategies for interpreting machine learning models

Perturbing strategies involve modifying the input data and observing some change in the model output. Because modifications to the input data can be made regardless of the ML algorithm used, perturbing strategies are generally model-agnostic. We discuss two general perturbation-based strategies: sensitivity analysis and what-if methods (**Figure 3**).

Sensitivity Analysis

Sensitivity analysis involves modifying an input feature and measuring the impact on **model performance** (**Figure 3A**). Feature modification typically means removing (i.e. leave-one-feature-out) or permuting (e.g. set all values to the mean) one feature at a time. The decrease in model performance after a feature is removed or permuted is an intuitive score for each feature indicating its contribution to the predictions (**Figure 3A**). Because perturbing a feature not only impacts that feature but also other features that interact with it, sensitivity analysis also captures interaction effects for each feature. However, sensitivity analysis can miss important features if correlation exists in the feature set. For example, if features X and Y are highly correlated, feature Y could compensate when X is removed or permuted, masking its potential importance.

Che *et al.* used the leave-one-feature-out approach to find that genomic region length was the most important feature for identifying genomic regions that contain clusters of genes acquired by horizontal gene transfer [40]. Leave-one-feature-out analysis is computationally expensive because it requires training a new model for every perturbed dataset. Therefore, it is typically not used to interpret deep learning model (which are already computing intensive) except when there are few input features. For example, leave-one-feature-out was used to determine that, of five histone marks, removing H3K4me3 resulted in the largest decrease in a deep learning model's ability to predict TF binding sites [41].

Permutation strategies determine feature importance score by measuring how the performance of an ML model changes when different features are randomly permuted. They are more computationally efficient than leave-one-feature-out strategies because only one model needs to be trained. This strategy is particularly intriguing for genetic studies because its logic is similar to DNA mutagenesis experiments. It was demonstrated that *in silico* mutagenesis (i.e. computationally permuting DNA sequence) could identify which nucleotides impact tissue specific gene expression the most [42]. A permutation-based strategy used in image analysis is called occlusion sensitivity. Here different regions in images are grayed out and the resulting change in performance is measured. For example, occlusion of regions of blood smear images confirmed that a malaria classification model performed worst when parasitized regions were grayed out [43].

What-if Analysis

The what-if approach (a.k.a. counterfactuals [44]) measures how the prediction of a particular instance changes (rather than the overall model performance) when the input value for one or more features is changed. Thus, what-if analysis provides local interpretations while sensitivity analysis provides global interpretations. Here we focus on two what-if methods: partial dependency plots (PDPs) and individual conditional expectation (ICE) plots (**Figure 3B**; [16]).

PDPs show how a prediction changes when the input value for a feature of interest is changed, marginalizing (i.e. ignoring) the effects of all other features [45]. Imagine we trained a ML model that predicts the likelihood that a sequence will be bound by a certain transcription factor (TF). A PDP would show, for example, how the TF-binding likelihood would change if the nucleotide at position of interest is changed from C to A, G or T (left panel, **Figure 3B**). This approach was used to demonstrate the impact of sequence features (e.g. amino acid identity, conservation) on the predicted efficacy of a guide RNA for CRISPR-Cas9 [46]. PDPs can miss important features when there are interactions between features. For example, imagine if a C at position #3 increased TF binding affinity when position #2 contained a T but decreased binding affinity if position #2 contained an A. Because position #2 is marginalized in the position #3's PDP, the interaction may mask the importance of position #3.

ICE plots were proposed to address this limitation of PDPs [47]. ICE plots are essentially PDPs generated for every individual instance in the dataset. For example, an ICE plot for position #3 would show that the presence of a C at position #3 only increases the TF binding likelihood in the subset of sequences, which with further investigation we find are the sequences with a T in position #2 (right panel, **Figure 3B**). Because this strategy does not require model re-training, it is well suited for interpreting deep learning models. For example, ICE plots were used to better understand what patterns of gene expression an adversarial deep learning model (see

doi:10.20944/preprints202002.0239.v1

Box 2, Figure IIB) learned were characteristic of single cell data [48]. By varying the expression level of individual genes (the feature) within the single cell (the instance), they found the genes with the biggest impact on the prediction (real or not) were genes known to be markers for particular cell-type states (e.g. IvI, Krt10, and Krt14 for epidermal cell state).

What-if analyses can provide highly detailed and intuitive interpretations of ML models, including the magnitude, direction, and non-linearities in the relationships between features and the output label. A limitation is that PDP and ICE plots can only be visualized for one or two features at a time, so they are typically only generated for models with few features or with a subset of features deemed important by another interpretation strategy or from domain knowledge [49].

Surrogate strategies for interpreting machine learning models

Image you have an ML model that is truly a black box—meaning that it cannot be probed and perturbations strategies do not provide useful information. In such a case, one can train an inherently interpretable model (e.g. linear model or a decision tree) to act as a surrogate for the black box model. For example, to generate a surrogate model for a black box model that can predict gene up-regulation using regulatory elements as features, we would first apply the black box model to a set of genes, *G*, and extract the black box predicted label (i.e. up- or down-regulated) for those genes (**Figure 1B**). Then we would use the same set of genes *G* as the instances and the black box predicted labels as the labels to train an interpretable surrogate model.

One major limitation of surrogate models is that black box models are often highly complex (e.g. highly non-linear, many higher order interactions), and thus, cannot be fully learned by an interpretable surrogate. To overcome this, one approach is to generate a surrogate to learn just a portion of the black box model, known as a Local Interpretable Model-agnostic Explanations (LIME; [50]). While the complex logic underlying the whole model may be too much for a surrogate model to learn, the logic for one instance or a group of similar instances (e.g. co-expressed genes) may be simple enough. For example, LIME was used to better understand why some patients (i.e. instances) were misclassified by a black box model predicting survival after cardiac arrest [51]. A LIME model for a patient that was mis-predicted to survive showed that the black box model was too heavily influenced by certain features (e.g. healthy neurologic status, lack of chronic respiratory illness) and did not place sufficient weight on other features that are also important (e.g. elevated creatinine, advanced age).

Concluding Remarks

Interpretability is critical for applications of ML in genetics and beyond and will therefore see substantial advances in the coming years. Just as there is no one universally best ML algorithm, there will not likely be one ML interpretation strategy that works best on all data or for all questions. Rather, the interpretation strategy should be tailored to what you want to learn from the ML model and confidence in the interpretation will come when multiple approaches tell the same story. Luckily, many user-friendly tools have already been developed to facilitate interpreting ML models using the strategies described in this review and more (**Table 1**). The insights that can be learned from interpreting a ML model are constrained by the content, quality, and quantity of the data used to generate the model. Care should be taken when selecting data and features to avoid introducing technical or biological artifacts into the models, and thus into the interpretations.

There are still many challenges to interpreting machine learning models in genetics and genomics (see **Outstanding Questions**). These challenges, while not necessarily unique to genetics or genomics, represent opportunities for computational biologists to innovate and contribute novel solutions. They also highlight the importance of training the next generation of biologists able to work at the intersection of computer and biological science.

Acknowledgement

We thank Yuying Xie for his insightful feedback on this review. This work was partly supported by the National Science Foundation (NSF) Graduate Research Fellowship (Fellow ID: 2015196719) to C.B.A.; NSF (IIS-1907704, IIS-1845081, CNS- 1815636) grants to J.T.; and the U.S. Department of Energy Great Lakes Bioenergy Research Center (BER DE-SC0018409) and NSF (IOS-1546617, DEB-1655386) grants to S.-H.S.

Outstanding Questions

- How can we interpret ML models trained on heterogeneous (e.g. multi-omic) and high dimensional (number of features >> number of instances) data? ML algorithms are well suited to take advantage of the large-scale multi-omic data for generating predictive models. However, interpreting ML models trained on high dimensional and heterogenous data remains challenging. These challenges are exasperated when features are highly correlated and of different types (e.g. continuous verses binary).
- What ML modeling and interpretation strategies are best for studying complex biological systems? Given the importance of non-linear effects in biology (e.g. epistasis, feedback loops, community dynamics, synergistic/antagonistic effects), interpretation strategies that can identify features that have important but complex effects are critical.
- How can we compare ML interpretation strategies and results? The strategies used to interpret an ML model are able to identify different aspects of the logic underlying that model. How can we benchmark new and established interpretation strategies for applications in genetics and genomics? Further, how could we join the findings from multiple strategies into a fuller, yet still coherent, interpretation of that model?
- How can interpretable ML become an accessible tool for biologists? Implementing ML interpretation strategies can require extensive computational knowledge. What roles will interdisciplinary training (e.g. computer science, data science) and the user-friendly-software play in encouraging the interpretation of ML models in genetics and genomics?
- How can researchers ensure that model interpretability will continue to be an area of development for folks working in the artificial intelligence field? As the power and precision of ML models improves, more and more trust will likely be placed in them. What role can researchers play in shaping the future of AI?

Text Boxes

Box 1: A crash course in machine learning.

Machine Learning (ML) is when a computer uses data to learn a model for predicting a value, where the relationship between the data and the value is not explicitly provided. The data is composed of instances (i.e. samples) and feature (i.e. independent variables) that describe those instances. For example, if our instances are genes, features describing those genes could be the GC content, the presence or absence of a specific functional domain, or its level of conservation across species. If the values being predicted are not known a priori for any instance, then unsupervised ML approaches (e.g. clustering) can be applied to extract previously unknown patterns. If the values being predicted are known for some of the instances, these values are referred to as labels and one can learn from these labels and turn the problem into a supervised ML problem. Further, if the known labels are categorical (e.g. is the gene up-regulated or down-regulated), it is a classification problem, while if the labels are continuous (e.g. gene expression levels), it is a regression problem.

A common supervised ML workflow involves four steps: training, applying, scoring, and interpretation (**Figure I**). First, input data made up of features and labels for many instances are divided into a training set and a testing set. The features and labels from the training set are then used to train the ML model. During training, the ML model learns the combination of internal parameters that minimize the error in the predictions of the labels. Second, the trained ML model is applied to the testing set features to generate predicted labels. A trained ML model can also be applied to unlabeled instances to make predictions. Third, the performance of the ML models is scored by comparing the predicted labels with the known labels from the test set. Many different performance metrics are used in the ML field, where the best metric depends on the type of ML problem and the nature of the question being asked. A performance metric not only informs the quality of a model, but also provides a quantitative measure of how much we known about the biological phenomenon in question given the features used. Finally, the ML model is interpreted to provide a better, quantitative understanding on how the input features contribute to the predictions.

Figure I. A supervised machine learning workflow.

Box 2: A crash course in deep learning.

ML algorithms inspired by the structure of the brain make up a subfield of ML called Deep Learning (DL). DL is promising for biology because DL models can 1) learn highly complex nonlinear patterns, 2) continue to improve when given more training data (“shallow”

ML models tend to plateau), and 3) they can learn from raw data without user defined features [52]. A DL model is made up of multiple layers of nodes connected by edges of different connection weights (w_x) (**Figure IIA**). The nodes in the input layer contain the feature values (f_x) for an instance. The nodes in the hidden layers (hidden nodes) represent the sum of the nodes from the previous layer multiplied by their associated connection weights ($\sum w_x f_x$). The node value from that summation is then passed through an activation function (represented as a light switch), which determines the extent to which that node gets turned on (i.e. activated). A DL models are able to learn nonlinear relationships when the activation function used is nonlinear (e.g. the sigmoid function). The output node (i.e. the predicted label) is the sum of the nodes from the last hidden layer and can be compared to the true label to calculate the error in the model. A DL model is trained by propagating that error back through the model and updating the learned connection weights (i.e. backpropagation of the error) until that error is minimized.

While this type of DL algorithm, often referred to as a fully-connected artificial neural network, is useful for modeling complex, nonlinear relationships. Other DL algorithms many be useful for addressing different biological questions (**Figure IIB**). For example, convolutional neural networks learn spatial patterns making them ideal for identifying sequence motifs and patterns in images, while recurrent neural networks remember earlier predictions and are therefore ideal for sequential data analysis.

Figure II. Graphical explanations of deep learning algorithms. (A) An example fully-connected artificial neural network. (B) Uses, graphical explanations, and example biological applications for three additional deep learning algorithms: Convolutional Neural Networks, Recurrent Neural Networks, and Adversarial Learning.

Figure Legends

Figure 1. Overview of ML model interpretation strategies

(A) Understanding the inner logic of a machine learning (ML) model (i.e. model interpretability), is important for troubleshooting during model training, generating biological insights, and instilling trust in the predictions made. **(B)** There are three general strategies for interpreting a ML model: probing, perturbing, and surrogates. Probing strategies involve inspecting the structure and parameters learned by a trained ML model (e.g. a deep learning model pictured here) in order to better understand what features or combination of features are important for driving the model's predictions. Perturbing strategies involve changing values of one or more input features (e.g. setting all values to zero) and measuring the change in model performance (sensitivity analysis) or on the predicted label of a specific instance (what if analysis). Finally, an easily interpretable model (e.g. linear regression or decision tree) can be trained to predict the predictions from a ML models, acting as a surrogate.

Figure 2. Probing a trained machine learning model.

An ML model that classifies up- (green) from down-regulated (yellow) genes using regulatory sequences (purple) as features can be probed to find what regulatory sequences are most important for predicting differential expression. **(A)** A support vector machine model learns the combination of coefficient weights (*w*; orange) that form the decision boundary (dotted line) best able to separate up- from down-regulated genes, where the features assigned the higher *w* are more important. The decision boundary is a hyperplane represented by the equation shown. **(B)** A decision tree-based model learns the most predictive series of true/false questions about the features. Here we zoom in on a node where the regulatory sequence "AACGT" is used as the feature. How well AACGT separates up- from down-regulated genes is quantified by calculating the mean decrease in node impurity after AACGT is used. Large impurity scores (here calculated as the Gini Impurity) mean the node contains a mix of up and down-regulated genes, while an impurity score equal to zero would indicate the node only contains up or down-regulated genes. **(C)** Deep learning models train to learn what combinations of connection weights (gray lines) across all nodes and layers results in the network best able to classify up- from down-regulated genes. A trained deep learning models can be probed by inspecting the size of the connection weights (gray line thickness), measuring the gradient of the output with respect to the input [i.e. ∂Out(in)/∂ (in)], and quantifying the extent to which different features cause a node to activate (represented by the light switch).

Figure 3. Perturbing the input to a machine learning model.

An example ML model predicting if a Transcriptional Factor (TF) may bind (i.e. the label) to a specific sequence (i.e. the features) can be interpreted with perturbing strategies. **(A)** Sensitivity analysis. Leave-one-feature-out means a new ML model is trained on the same input data with one feature (e.g. position 3) removed. Then the overall performance of the original model and the new model are compared. Permutation means the original model is applied to input data with the values shuffled for one feature at a time. The performance of the model applied to the original and the shuffled data are compared. Both sensitivity analyses on position 3 shown here resulted in a decrease in performance, leading to the interpretation that position 3 is important for TF binding. **(B)** What-If analysis. The partial dependency plot (left) shows the TF binding likelihood if position 3 was an A, C, G, or T, ignoring the effects of nucleotides at other positions. This plot shows that a C at position 3 increases the likelihood of TF binding. The individual conditional expectation plot (right) shows the TF binding likelihood score for every instance (dot) in the dataset when position 3 is A, C, G, or T. This plot shows when position 3 is C, the binding likelihoods have a bimodal distribution which is due to interaction with position 2 in this hypothetical example.

Table Legends

Table 1. Platforms and software available for interpretable machine learning

Name	Strategy	Use	Scope	Description	Platform
CamurWeb [53]	Probing	Decision tree-based models	Global	Interpret decision rules from Classifier with Alternative and MUltiple Rule (Camur) models	web tool
DeepExplain [54]	Probing, perturbing	Deep Learning	Global, local	Toolbox for implementing multiple interpretation methods	Tensorflow, Keras
DeepTRIAGE [55]	Probing	Attention-based Deep Learning	Local	Deep learning for the Tractable Individualized Analysis of Gene Expression	Python package

iml: interpretable ML [16]	Probing, perturbing	Model agnostic	Global, Local	Toolbox for implementing multiple interpretation methods.	R package
iNNvestigate [56]	Probing	Deep Learning	Global, Local	Toolbox for implementing multiple interpretation methods.	Keras
iRF [22]	Probing	Random Forest	Global	Decision tree based method to identify significant feature interactions.	R package
LIME [50]	Surrogate	Model Agnostic	Local	A tool to generate local surrogate models for Black-Box models.	Python package
Lucid (github.com/tensorflow/lucid)	Probing	Deep Learning	Global, local	Toolbox of methods for visualizing and interpreting neural networks.	Tensorflow
NeuralNetTools [57]	Probing, perturbing	Deep Learning	Global, local	Toolbox for implementing multiple interpretation methods.	R package
SpliceRover [37]	Probing	Deep Learning	Local	Tool to interpret which nucleotides contribute most predicting splice sites using DeepLIFT	web tool
The What-If Tool (https://pair-	Probing, perturbing	Model Agnostic	Global, local	Code free toolbox for assessing, comparing, and interpreting Tensorflow/python-based ML models	TensorBoard, Jupyter, Colaboratory notebooks

code.github.io/what-if-tool/index.html)					

References

1 Marx, V. (2013) Biology: The big challenges of big data. *Nature* 498, 255–260

2 Stephens, Z.D. *et al.* (2015) Big Data: Astronomical or Genomical? *PLOS Biol.* 13, e1002195

3 Schrider, D.R. and Kern, A.D. (2018) Supervised Machine Learning for Population Genetics: A New Paradigm. *Trends Genet.* 34, 301–312

4 Alyass, A. *et al.* (2015) From big data analysis to personalized medicine for all: challenges and opportunities. *BMC Med. Genomics* 8, 33

5 Angermueller, C. *et al.* (2016) Deep learning for computational biology. *Mol. Syst. Biol.* 12, 878–16

6 Chicco, D. (2017) Ten quick tips for machine learning in computational biology. *BioData Min.* 10, 35

7 Cuperlovic-Culf, M. (2018) Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling. *Metabolites* 8, 4

8 Libbrecht, M.W. and Noble, W.S. (2015) Machine learning applications in genetics and genomics. *Nat. Publ. Group* 16, 321–332

9 Ma, C. *et al.* (2014) Machine learning for Big Data analytics in plants. *Trends Plant Sci.*

10 Tarca, A.L. *et al.* (2007) Machine Learning and Its Applications to Biology. *PLOS Comput. Biol.* 3, e116

11 Samuel, A.L. (1959) Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.* 3, 210–229

12 Lipton, Z.C. (2018) The Mythos of Model Interpretability. *ACM Queue* 16,

13 Miller, T. (2019) Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267, 1–38

14 Guidotti, R. *et al.* (2018) A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 1–42

15 Montavon, G. *et al.* (2018) Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* 73, 1–15

16 Molnar, C. (2019) *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 1st edition.Christoph Molnar.

17 Barakat, N. and Bradley, A.P. (2010) Rule extraction from support vector machines: A review. *Neurocomputing* 74, 178–190

18 Rasmussen, P.M. *et al.* (2011) Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. *NeuroImage* 55, 1120–1131

19 Ronen, R. *et al.* (2013) Learning Natural Selection from the Site Frequency Spectrum. *Genetics* 195, 181–193

20 Breiman, L. (2001) Random Forests. *Mach. Learn.* 45, 5–32

21 Uygun, S. *et al.* (2019) Cis-regulatory code for predicting plant cell-type transcriptional response to high salinity. *Plant Physiol.* DOI: 10.1104/pp.19.00653

22 Basu, S. *et al.* (2018) Iterative random forests to discover predictive and stable high-order interactions. *Proc. Natl. Acad. Sci.* 115, 1943–1948

23 Vervier, K. and Michaelson, J.J. (2018) TiSAn: estimating tissue-specific effects of coding and non-coding variants. *Bioinformatics* 34, 3061–3068

24 Strobl, C. *et al.* (2007) Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8,

25 Banerjee, S. *et al.* (2017) , Performance of Deep Learning Algorithms vs. Shallow Models, in Extreme Conditions - Some Empirical Studies. , in *Pattern Recognition and Machine Intelligence*, pp. 565–574

26 Guo, Y. *et al.* (2016) Deep learning for visual understanding: A review. *Neurocomputing* 187, 27–48

27 LeCun, Y. *et al.* (2015) Deep learning. *Nature* 521, 436–444

28 Kuhn, M. and Johnson, K. (2013) *Applied Predictive Modeling*,

29 Schmidhuber, J. (2015) Deep learning in neural networks: An overview. *Neural Netw.* 61, 85–117

30 Garson, D.G. (1991) Interpreting neural network connection weights. *AI Expert* 6, 46–51

31 Olden, J.D. and Jackson, D.A. (2002) Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks. *Ecol. Model.* 154, 135–150

32 Manzanarez-Ozuna, E. *et al.* (2018) Model based on GA and DNN for prediction of mRNA-Smad7 expression regulated by miRNAs in breast cancer. *Theor. Biol. Med. Model.* 15,

33 Shrikumar, A. *et al.* (2017) Learning Important Features Through Propagating Activation Differences. *Proc. 34 Th Int. Conf. Mach. Learn.* at <http://proceedings.mlr.press/v70/shrikumar17a/shrikumar17a.pdf>

34 Simonyan, K. *et al.* (2013) Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *Int. Conf. Learn. Represent.* at <http://arxiv.org/abs/1312.6034>

35 Kelley, D.R. *et al.* (2018) Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.* 28, 739–750

36 Washburn, J.D. *et al.* (2019) Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence. *Proc. Natl. Acad. Sci.* 116, 5542–5549

37 Zuallaert, J. *et al.* (2018) SpliceRover: interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics* 34, 4180–4188

38 Kim, J.-S. *et al.* (2018) RIDDLE: Race and ethnicity Imputation from Disease history with Deep LEarning. *PLOS Comput. Biol.* 14, e1006106

39 Esteva, A. *et al.* (2017) Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 115–118

40 Che, D. *et al.* (2010) Classification of genomic islands using decision trees and their ensemble algorithms. *BMC Genomics* 11, S1

41 Jing, F. *et al.* (2019) An integrative framework for combining sequence and epigenomic data to predict transcription factor binding sites using deep learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* DOI: 10.1109/TCBB.2019.2901789

42 Zhou, J. *et al.* (2018) Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.* 50, 1171–1179

43 Rajaraman, S. *et al.* (2018) Understanding the learned behavior of customized convolutional neural networks toward malaria parasite detection in thin blood smear images. *J. Med. Imaging* 5, 1

44 Wachter, S. *et al.* (2018) Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. J. Law Technol.* 31,

45 Friedman, J.H. (2001) Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29, 1189–1232

46 Schoonenberg, V.A.C. *et al.* (2018) CRISPRO: identification of functional protein coding sequences based on genome editing dense mutagenesis. *Genome Biol.* 19, 169

47 Goldstein, A. *et al.* (2015) Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *J. Comput. Graph. Stat.* 24, 44–65

48 Ghahramani, A. *et al.* (2018) Generative adversarial networks simulate gene expression and predict perturbations in single cells. *bioRxiv* DOI: 10.1101/262501

49 Liu, Z. and Yang, J. (2014) Quantifying ecological drivers of ecosystem productivity of the early-successional boreal Larix gmelinii forest. *Ecosphere* 5, art84

50 Ribeiro, M.T. *et al.* (2016) , “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. , in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, San Francisco, California, USA, pp. 1135–1144

51 Nanayakkara, S. *et al.* (2018) Characterising risk of in-hospital mortality following cardiac arrest using machine learning: A retrospective international registry study. *PLOS Med.* 15, e1002709

52 Eraslan, G. *et al.* (2019) Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* 20, 389–403

53 Weitschek, E. *et al.* (2018) CamurWeb: a classification software and a large knowledge base for gene expression data of cancer. *BMC Bioinformatics* 19, 354

54 Ancona, M. *et al.* (2018) Towards better understanding of gradient-based attribution methods for Deep Neural Networks. *ArXiv171106104 Cs Stat* at <http://arxiv.org/abs/1711.06104>

55 Beykikhoshk, A. *et al.* (2019) DeepTRIAGE: Interpretable and Individualised Biomarker Scores using Attention Mechanism for the Classification of Breast Cancer Sub-types. *bioRxiv* DOI: 10.1101/533406

56 Alber, M. *et al.* (2018) iNNvestigate neural networks! *ArXiv180804260 Cs Stat* at <http://arxiv.org/abs/1808.04260>

57 Beck, M.W. (2018) NeuralNetTools: Visualization and Analysis Tools for Neural Networks. *J. Stat. Softw.* 85, 1–20

Figure 1

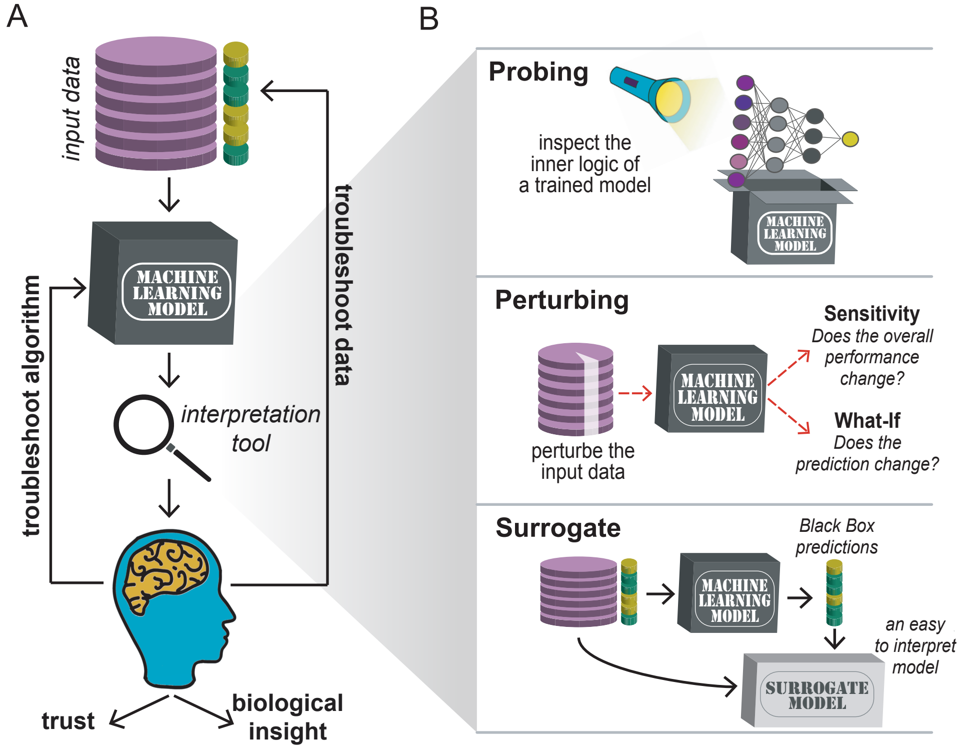

A
input data
MACHINE LEARNING MODEL
interpretation tool
troubleshoot algorithm
troubleshoot data
trust
biological insight
B
Probing
inspect the inner logic of a trained model
MACHINE LEARNING MODEL
Perturbing
perturbe the input data
MACHINE LEARNING MODEL
Sensitivity
Does the overall performance change?
What-If
Does the prediction change?
Surrogate
MACHINE LEARNING MODEL
Black Box predictions
an easy to interpret model
SURROGATE MODEL

Figure 2

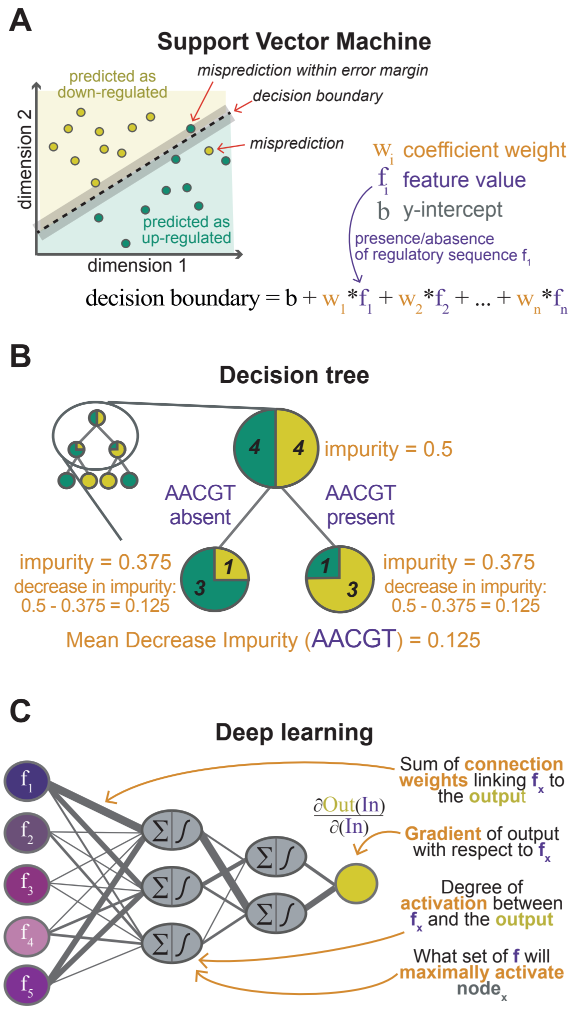
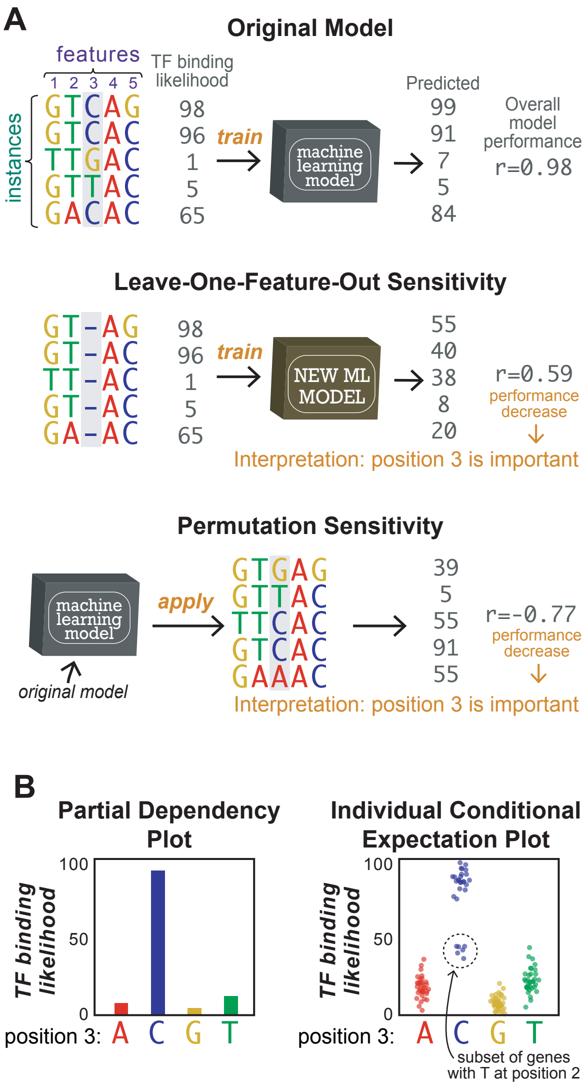

A
Support Vector Machine
predicted as down-regulated
misprediction within error margin
decision boundary
misprediction
dimension 2
dimension 1
predicted as up-regulated
w_i coefficient weight
f_i feature value
b y-intercept
presence/abasence of regulatory sequence f_i
decision boundary = $b + w_1*f_1 + w_2*f_2 + ... + w_n*f_n$
B
Decision tree
4
4
impurity = 0.5
AACGT absent
AACGT present
impurity = 0.375
decrease in impurity: 0.5 - 0.375 = 0.125
3
1
1
3
impurity = 0.375
decrease in impurity: 0.5 - 0.375 = 0.125
Mean Decrease Impurity (AACGT) = 0.125
C
Deep learning
f_1
f_2
f_3
f_4
f_5
Σ ∫
Sum of connection weights linking f_x to the output
$\frac{\partial Out(In)}{\partial (In)}$
Gradient of output with respect to f_x
Degree of activation between f_x and the output
What set of f will maximally activate $node_x$

Figure 3


A
Original Model
features
1 2 3 4 5
instances
GTCAG
GTCAC
TTGAC
GTTAC
GACAC
TF binding likelihood
98
96
1
5
65
train
machine learning model
Predicted
99
91
7
5
84
Overall model performance
r=0.98
Leave-One-Feature-Out Sensitivity
GT-AG
GT-AC
TT-AC
GT-AC
GA-AC
98
96
1
5
65
train
NEW ML MODEL
55
40
38
8
20
r=0.59
performance decrease
Interpretation: position 3 is important
Permutation Sensitivity
machine learning model
original model
apply
GTGAG
GTTAC
TTCAC
GTCAC
GAAAC
39
5
55
91
55
r=-0.77
performance decrease
Interpretation: position 3 is important
B
Partial Dependency Plot
TF binding likelihood
100
50
0
position 3: A C G T
Individual Conditional Expectation Plot
TF binding likelihood
100
50
0
position 3: A C G T
subset of genes with T at position 2

doi:10.20944/preprints202002.0239.v1

Figure I

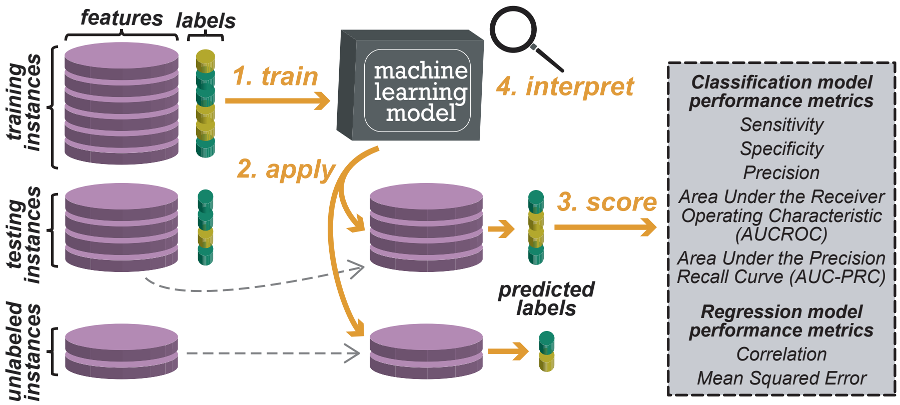
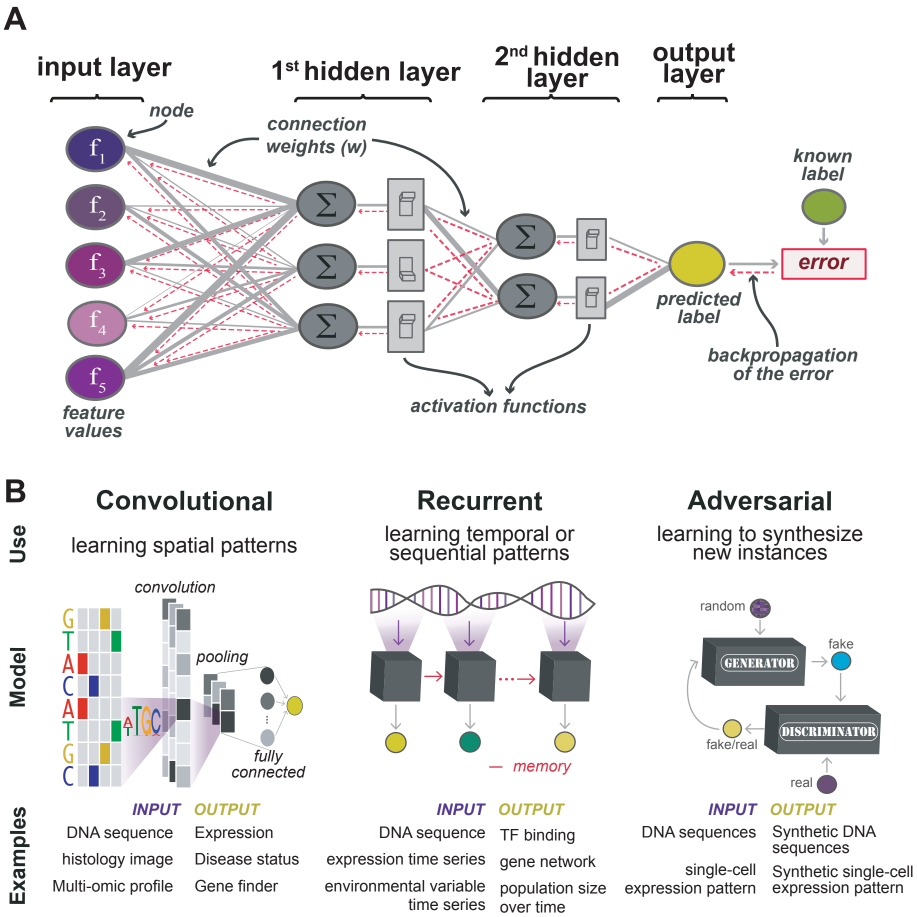


Figure II


A
input layer
1st hidden layer
2nd hidden layer
output layer
node
connection weights (w)
f1
f2
f3
f4
f5
feature values
Σ
Σ
Σ
Σ
Σ
activation functions
known label
error
predicted label
backpropagation of the error
B
Convolutional
Recurrent
Adversarial
Use
learning spatial patterns
learning temporal or sequential patterns
learning to synthesize new instances
Model
G T A C A T G C
ATGC
convolution
pooling
fully connected
memory
random
GENERATOR
fake
DISCRIMINATOR
fake/real
real
Examples
INPUT
OUTPUT
DNA sequence
Expression
histology image
Disease status
Multi-omic profile
Gene finder
INPUT
OUTPUT
DNA sequence
TF binding
expression time series
gene network
environmental variable time series
population size over time
INPUT
OUTPUT
DNA sequences
Synthetic DNA sequences
single-cell expression pattern
Synthetic single-cell expression pattern