UAB
Universitat Autònoma
de Barcelona

Deep Learning and its Application to Predicting Quantitative Phenotypes from Genomic Data

VÍCTOR FERNÁNDEZ OLIVERAS

Supervised by Dr Antonio Barbadilla
Genetics Degree
Bachelor Thesis (TFG)

2025-05-22

Contents

1 Objectives and Methodology

This bachelor thesis aims to:

- Describe neural network architectures as well as its training process, with its key challenges (*e.g.*, previous data curation, overfitting) and how to address them.
- Address model interpretability in deep learning (DL) to determine its importance and review current methods.
- Contextualize each component of deep learning models in the prediction of quantitative phenotypes from genomic data.
- Critically review the current situation and propose relevant future research directions specific to genomics prediction models.
- Offer an open-access educational tool that implements the theoretical concepts discussed.

The work is based on a systematic and critical review of relevant literature related to deep learning in genomics, prioritising recent publications (2020-2025) while including older key references. Searches were principally conducted using PubMed and Nature search engines with keywords such as “deep learning”, “neural network”, “machine learning", “interpretability”, “genomics”, “phenotype prediction” and “complex trait”. The review was divided into three stages: (1) searching high-impact articles addressing DL in genomics broadly, to structure the work and define its principal sections, (2) selecting specific studies in which DL was applied or novel DL-based software was presented, and (3) targeted searches to clarify some specific sections or complement them. Furthermore, the practical example was performed by programming in Python.

2 Introduction

Inside artificial intelligence (AI) methods, machine learning (ML) comprises statistics-based algorithms (*e.g.*, support vector machines, random forests)[17] designed to mimic the human brain function for diverse tasks[15]. However, the data quantity and complexity that a typical ML method can handle is limited[10].

In genomics, high-throughput sequencing technologies generate vast amounts of data, which are not only extensive but also extremely complex[8,18,29]. Deep learning emerges as a subtype of ML specialized in such even more complex massive datasets[2,27], such as those containing nonlinear genotype-phenotype relationships[11]. DL field is growing at an overwhelming pace and its methods are currently outperforming human capacity in solving many specific problems[4], representing the state-of-the-art for performing predictions in genomics[31].

Unlike unsupervised learning, which identifies inherent patterns to detect data structure, DL predictive models are based on supervised learning, as they are trained with labeled examples to predict unlabeled data[33]. This includes modified data, enabling the prediction of the effect of variants absent in current populations[3,29].

A typical dataset consists of data pairs / examples, whose terminology varies significantly across literature. They are formed by (1) a sample / feature, which is a DNA sequence, a SNV array, a set of CNVs, or others; and (2) an associated label / value / outcome[6,28], which represents a molecular or macromolecular phenotype like gene expression[21,23,24] or treatment response[32].

That said, it is important to clarify that DL is not inherently superior to other ML methods. For some problems, shallow ML may outperform DL by generating better results or by being easier to implement and interpret[10,16,22]. DL is not the "evolution" or the final solution for AI-based problem-solving, but another powerful tool for specific cases[13]. For instance, for predicting the phenotype of a purely additive quantitative trait, a simpler ML method would be a better option, while DL is preferred for loci involving environmental and epistatic effects[18]. Therefore, DL can better predict complex traits in humans[5,26,29,30], other animals[7] and plants[21,27]. These quantitative traits are crucial in several fields such as biomedicine[11,32], evolutionary genomics[19], agrigenomics[7,15,18] or biotechnology[3].

3 Neural Networks

The core framework of deep learning is based on neural networks (NNs), where the basic unit is the neuron, a computational node that processes inputs to generate outputs[10]. DL models consist of numerous neurons clustered in diverse interconnected layers.

3.1 Data Encoding, General Architecture and Information Flow

In order to make genomic data readable by models, it must be encoded. In this work, nucleotide sequence data will be the analyzed case. For these inputs, one-hot encoding transforms length-L strings formed by A, C, T and G into 4xL matrices encoded as: A = [1,0,0,0], C = [0,1,0,0], T = [0,0,1,0] and G = [0,0,0,1]; or with other combinations[3,20] (Figure 1). Final encoded data contains these matrices (one per sequence) forming multidimensional matrices, called tensors[33].

The input encoded data is analysed by an input layer, which applies simple initial filters (kernels) to compute a vector of weights based on the input features[3,33]. This output is passed to the following layer as an input with an associated weight that specifies the importance of the node-node connection, and so on with the subsequent hidden layers (layers whose values are not directly observed by the modeller)[29,33]. The presence of many hidden layers leads to the NN being defined as a Deep Neural Network (DNN)[6], but in genomics predictive models, usually less than 5 hidden layers are sufficient[33]. These layers perform nonlinear transformations of their inputs using activation functions, the most common of which is the Rectified Linear Unit (ReLU). However, other functions are used depending on the model and even simple thresholds can be useful[2,27]. The capacity of prediction of the network emerges from working as a whole entity[6] and culminates in the final prediction computation in the output layer. These general structure, information flow and node computation are represented in Figure 2.

3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the most commonly used architecture in genomics prediction models[8,21,29,31], where they are usually coupled with a following set of fully-connected layers[22]. Novakovski *et al.*[20] perfectly explained CNNs functioning. First, an individual scanning of each position in the input data by the first layer is performed. In this step, nodes calculate local weighted sums by k-mers, generating outputs called activation maps. Then, a pooling operation is applied to these outputs, so as the authors describe, "maximum or average of nearby elements is used to reduce the number of learnt parameters in the later layers". After that, the pooled activation maps are combined (flatten operation) and passed to the first hidden layer[10,30,33]. This convolutional architecture is represented in Figure 3.

4 Model Training

Deep learning models require training on specific problems to optimize their predictive performance. During the training process, the model iteratively "learns" about the data to minimize prediction errors. Data is splitted into three subdatasets: (1) a training set used for parameter optimization through iterative "learning", (2) a validation set used to evaluate its training performance at the end of each iteration/epoch, and (3) a test set to determine the generalization performance of the final model[33].

4.1 Predictions and Error Calculation

DNA sequences are represented as X_i and their corresponding phenotype as Y_i. Each pair (X_i, Y_i) is processed by the network to generate a predicted phenotype Y_i', quantifying the prediction error with the loss function $L(Y_i', Y_i)$[33]. For continuous phenotypes, the mean squared error (MSE) is the most commonly employed loss function[20]. Although predictions and hence errors are calculated on both training and validation sets, the performance used to evaluate the model during training is that shown in the validation set.

4.2 Parameter Update: Back-propagation

At the beginning of the training, parameters (weights and biases, whose number can reach millions) are usually randomly initialized[33], except when applying transfer learning to reuse parameters of prior models[13]. After each iteration, the model adjusts its parameters in order to improve its performance in the training set[33].

Even though several updating strategies exist, back-propagation remains the predominant approach. This algorithm computes the partial derivative of $-L(Y_i', Y_i)$ with respect to each parameter, propagating this calculus backwards from the output to the input layers. Parameters are then updated through small changes in the direction of these derivatives (Figure 4), with higher absolute values indicating parameters that highly influence the model's performance[33].

Eraslan *et al*[6] exposed that a key advantage of back-propagation is that increasing the data volume affects computation time, but not the memory requirements. Furthermore, explained that for enhanced optimization, back-propagation can be implemented within a stochastic gradient descent (SGD) framework, where random subsets (batches) of the training set are used in each epoch. An SGD-based method adapted to very noisy tasks called Adam[12] is the preferred method in the these models[21,22,30,32].

4.3 Overfitting and Underfitting

An incorrect number of iterations during the training process can lead to the following phenomena. Underfitting occurs when the model undergoes insufficient training, failing to "learn" enough from the data[33]. Conversely, overfitting emerges when training continues for too many epochs, causing the model to excessively adapt its predictions to the training set[3] by learning spurious correlations[33].

Overfitting can be detected by observing the decoupling of the reductions between the training and validation sets[24] (Figure 5) and several strategies exist to address it. The most basic approach is manually stopping training or repeating it after identifying at which number of epochs overfitting starts. Other more sophisticated techniques include L2 regularization of weights, which penalizes models with excessively high weight values by adding the sum of the squares of these parameters to the loss function[22]; dropout, where nodes are randomly ignored in each iteration[10]; and early-stopping, the automatic counterpart of manually stopping training[27].

5 Hyperparameter Tuning

In deep learning models, the term "hyperparameter" refers to all the structural and operational elements of the network that are explicitly defined by the modeller rather than "leart" by the model during training[11,32]. Some examples of hyperparameters are: the number of layers, the number of neurons in each layer, the activation functions, the optimization method...

The hyperparameter setting significantly impacts model performance and therefore requires careful configuration. Two approaches are exposed by Wang *et al.*[27] for this process: try-and-error, which is easier to implement but produces worse results; and automatic machine learning (AutoML), which offers better results at higher computational cost. AutoML algorithms systematically evaluate multiple hyperparameter combinations within a defined search space to identify configurations that minimize prediction error[31]. The number of possible combinations can be enormous, as each layer can have a different activation function, number of neurons, etc.

6 Data Curation

Genomic data frequently present inherent structure and internal relations that violate model assumptions, such as the independent and identical distribution (IID) of the examples[28]. Models cannot automatically detect these issues, so it must be the modeller who carefully curates the data to minimize biases that could influence model's performance[31]. Two of these main possible pitfalls are presented in this section.

6.1 Distributional Differences

The distribution of features P(X), outcomes P(Y) or phenotype-genotype pairs P(Y|X) may be different within or between subsets, causing the model to over or under "learn" from some data[10,28]. For instance, when studying aberrant gene expression as a disease cause, Hölzlwimmer *et al.*[9] had to deal with the number of these samples with extreme values being drastically lower than those with standard values.

These differences are frequently detected by simple data visualization with scatter plots or histograms (Figure 6), but there also exist statistical tests, like Kolmogorov-Smirnov test, that compare group distributions[28]. After its detection, the main strategies to mitigate this pitfall are: downsampling overrepresented examples[21], oversampling underrepresented examples[28], weighting pairs depending on their relative frequencies[28] or using performance measure methods that consider these differences[33]. It is crucial to note that these approaches should only be applied to the training set, so evaluation data preserve the real distribution[13]. Finally, although distributional imbalances are generally seen and assessed as an obstacle, Whalen *et al.*[28] propose that they could also be treated as an opportunity to discover insights on your data.

6.2 Correlated Samples

Another important pitfall is the presence of correlated data, such as sequences belonging to familiarly-related individuals[13,16] or repeated pairs[21]. According to Whalen *et al.*[28], this can occur within or between subsets, causing information leakage and impeding correct independent training. They further expose that principal strategies to address this pitfall include downsampling correlated pairs or explicitly acknowledging its presence and adopting measures to reduce overfitting. These authors highlight that statistical methods that consider covariances between samples remain not extended for large-scale genomic data, neither available in common DL tools.

7 Interpretability

Unlike more traditional statistical models with simpler interpretation due to more direct input-output relations[8,17], deep learning is often criticized for its lack of inherent interpretability or explainability[5,18] and perceived as a "black box"[25,26]. Nevertheless, there exists a whole AI subarea called explainable artificial intelligence (xAI)[20], dedicated to determine the relative importance of each datum and model parameter in the output generation[4] (Figure 7). For example, the application of xAI techniques enabled Peleke *et al.*[21] and MacNish *et al.*[15] to identify functional genomic regions such as transcription start/termination sites or lncRNAs, respectively.

Model interpretation is essential in many genomics applications, even sometimes being more important than the predictions themselves[22,27], as they are or could be applied in sensitive decision-making contexts[5,8] such as genomic profile-based treatment assigning[32]. However, this subfield faces several challenges. For example, the redundancy and nonlinearity of the nodes in the network, which furthermore do not learn invariant motifs[6]. In addition, an incredibly vast combinatorial search space and the different limitations and assumptions in each method, whose validity depends on both the properties of the dataset and which biological processes are influencing it[20].

Even though the back-propagation process described in 4.2 Parameter Update: Back-propagation, also known as backward propagation of influence, can be used as a low computational cost interpretability method[20,26], other important techniques address the interpretability issue from diverse perspectives.

7.1 Perturbation-based Methods (Forward Propagation of Influence)

These methods (such as DeepLift or DeepSHAP)[24] systematically introduce perturbations in the input dataset and measure the degree of change in the model's predictions[26]. Methods using this forward propagation of influence are considered to be model-agnostic, because they are widely applied with independence of the NN structure[20,24]. The process is analogous to in silico mutagenesis (ISM), as small changes are iteratively introduced while and the rest of the data remains untouched[3,26]. For instance, these techniques substitute each nucleotide with its 3 alternatives and assign them an individual importance / attribution / saliency score based on their change on model's performance[6,21,33]. Results are typically represented as sequence logos and/or heat maps determined by the scores of all nucleotide changes in an individual sequence (attribution map) or across all them (global attribution map)[17,20,30] (Figure 8).

These methods require high computational resources due to the high number of replacements that are performed[24,33]. To reduce them, the analysis can be focused on a subset of examples expected to contain important insights[22].

7.2 Position-weighted Matrices

Another relevant interpretability strategy is applying position-weighted matrices (PWMs) to convolutional nodes. These matrices indicate which regions or features have activated the most their nodes during training[20], principally enabling motif detection[26]. Nevertheless, high PWMs values do not necessarily correlate with the predictive importance of the node, causing the necessity of performing nullification of nodes[20], a process analogous to dropout. However, this technique has limitations since, as Novakovski *et al.*[20] note, "an important pattern may be captured multiple times by different neurons [...] nullifying a single neuron will not provide the true importance of a pattern to the model's predictions".

7.3 Epistatic Analysis: Self-attention

Novakovski *et al.*[20] also detailed an important approach used to detect epistatic interactions: self-attention matrices. According to the authors, these matrices are applied within a framework called deep feature interaction maps (DFIM), which focuses on certain sample regions or input subsets, combining forward and backward propagation of influence to measure how changes in nucleotides affect the effect of other nucleotides in the model's performance. Thus, it enables the calculation of feature interaction scores (FIS)[20] and, according to An *et al.*[2], the improvement of accuracy of following predictions. Currently, several programs such as that published by Seitz *et al.*[24], SQUID, use this approach in their interpretability framework. Apart from matrices, complete self-attention layers are used in transformer architectures coupled with CNNs, following the same logic and sharing objectives with self-attention matrices[3].

7.4 Transparent Models

Visible Neural Networks (VNNs)[25] belong to an specialized class of interpretable models, transparent models, that are constructed by assigning real subsystems like genes or biological pathways to nodes or layers[8,30]. VNNs are an exceptional approach for research with highly detailed and prior knowledge, which unfortunately is not very frequent in functional genomics[20]. However, some software, like P-NET[5], are currently employing this approach.

8 Downside of Deep Learning

As previously mentioned, deep learning has limitations and disadvantages that must be considered when deciding whether to implement it in experiments and during its application.

8.1 Data and Bioethics

Research faces a "lack of unbiased experimentally and clinically confirmed variant effects that can be used as gold standards in evaluations"[29]. This issue stems partially from historically skewed practices related to discrimination to social minorities and rare diseases in large-scale initiatives[25]. For instance, principal genomic databases lack an equal representation of data for all human ethnicities. As mentioned above: models trained with skewed data generate skewed predictions.

8.2 Local Minima and Maxima

During training, loss function and accuracy reach local minima and maxima, respectively. Therefore, although the models obtained can have relatively low error, they will not be optimal models. However, it is generally assumed that all local extrema have a very similar value. Moreover, pursuing global optima increases the risk of overfitting and the computational requirements, which are not specially low in DL. These factors explain why some authors do not consider this point as an alarming limitation[18].

8.3 Downside of Interpretability

It is crucial to take into account that explainable AI methods simply identify mathematical correlations between the dataset/model elements and the predictions, which may or may not have a biological meaning. Validating this often requires non-*in silico* experimentation[13,17]. As an example, an unobserved SNV in a plant with an associated predicted phenotype could be verified by generating a transgenic organism with CRISPR edition[15]. In addition, there is currently no consensus on which xAI method is better and the majority opinion is that it strongly depends on the model architecture, the characteristics of the dataset and the specific biological question being addressed[20].

8.4 No-free-lunch Theorem, Domain Knowledge and Expectations

In the context of predictive models in genomics, the no-free-lunch theorem can be interpreted as that a model trained for one task cannot typically be extrapolated to another task[18], at least not with the same performance[28]. In consequence, every new experiment requires a whole new model investigation. Moreover, as seen in this work, this model tuning involves a virtually infinite number of options[28], so rigorous model construction, training and implementation require proper domain expertise on both neural networks and genomics.

In addition, although deep learning models are expected to improve in tasks-range capacity and accuracy in the near future[22], generative AI boom has induced a general thought that DL can solve more problems than it actually can.

9 Conclusions and Future Perspectives

Literature has plenty of successful cases in which deep learning has outperformed simpler machine learning algorithms in complex phenotype prediction, making it a powerful and promising tool for crucial fields like personalized medicine, genomic selection, biotechnology or evolutionary genetics. In functional genomics prediction, Convolutional Neural Networks are the most established architecture, using back-propagation for parameter optimization during training. This process requires rigorous data curation (mainly to avoid distributional differences and sample correlations) and overfitting prevention through techniques such as early-stopping, regularization methods or dropout layers. Additionally, hyperparameters (*e.g.*, activation functions, number of neurons in each layer) require systematic tuning to detect combinations that minimize loss function. In the near future, more systematic comparisons of DL elements are needed in order to guide modellers. Although each experiment needs a different model, reducing the search space can reduce computational resource use and time wasting.

In order to make decisions using the information provided by DL predictions, it is important to apply interpretability methods to understand which elements of the model or the data have an important role in prediction-making. Several interesting approaches have been applied for interpretation, like backward and forward-propagation of influence, PWMs, self-attention matrices and layers for epistatic analysis or transparent models construction. However, in the future, “xAI will have an even more central role in genomics”[20]. The ultimate goal should be to reach fully comprehensive models that could accurately predict quantitative phenotypes using whole genomes (coupled with other information such as transcriptomics, proteomics or epigenomics data) and help elucidate their genetic and environmental components.

DL is not a panacea and several challenges arise when applying it: (1) genomic data is usually structured and biased, (2) model interpretation is extremely complex and does not guarantee real biological association, (3) computation costs, which are yet high in DL, increase because of the necessity of specific model tuning in each research, (4) the DL boom and its overhype induces its usage without the required expertise and sometimes when other tools (such as simpler ML methods) are better options, reducing the exit probabilities and insightful research…

Other research directions such as ensemble learning[7] or multimodal learning[11,15], based on combining diverse methods and models, respectively, could be interesting approaches to obtain better results in the not-so-distant future.

Finally, as some authors have noted[10,22,29], the development and improvement of libraries (Pytorch, TensorFlow, Keras, Scikit-learn…), frameworks[11,24] or other tools should enable non expert investigators to engage with DL, so the field can grow and advance. Sometimes, biological background researchers face the problem of not having sufficient mathematical knowledge or informatics skills to properly understand and use these networks[10], so software developers should work hard on creating user-friendly tools[1,10,31]. It must not be forgotten that this field is composed of not only computer scientists and mathematicians but equally importantly, biologists.

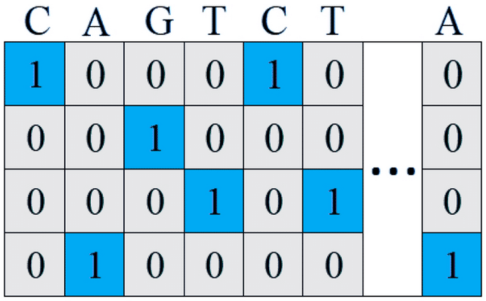
10 Practical Example

This work is accompanied by a GitHub repository (https://github.com/victor-fdz/Deep-Learning-for-Quantitative-Phenotype-Prediction-from-Genomic-Data) containing a Keras-based naive model that exemplifies key concepts discussed in this manuscript. This simple framework serves as educational material, but it may also function as an introductory resource for DL application in genomics research. The example includes: simulation of random genomic data (representing genomic continuous regions or sets of genomic positions) with an associated fictitious complex trait, implementation of a CNN architecture with fully-connected layers for phenotype prediction, and visualization of model accuracy and interpretability through various graphs.

11 Acknowledgments

Finally, I want to acknowledge people that have helped in this bachelor thesis in some way. Firstly, I would like to thank my family for supporting me on all my decisions and my friends for continuously asking me questions that challenge my understanding of genetics. Third, I want to specially thank Eric for being my constant source of competitive motivation, as well as to Jose and Simon for introducing me to the world of deep learning. Finally, I would like to thank my supervisor, Dr Barbadilla, for his advice and vision through this work.

Figures



C	A	G	T	C	T		A
1	0	0	0	1	0		0
0	0	1	0	0	0	…	0
0	0	0	1	0	1		0
0	1	0	0	0	0		1

Figure 1 | **One-hot encoded DNA sequence forming a 4xLenght matrix.** In this case, the sequence is encoded as: A = [0,0,0,1], C = [1,0,0,0], T = [0,0,1,0] and G = [0,1,0,0]. Modified from Liu *et al.*[14]

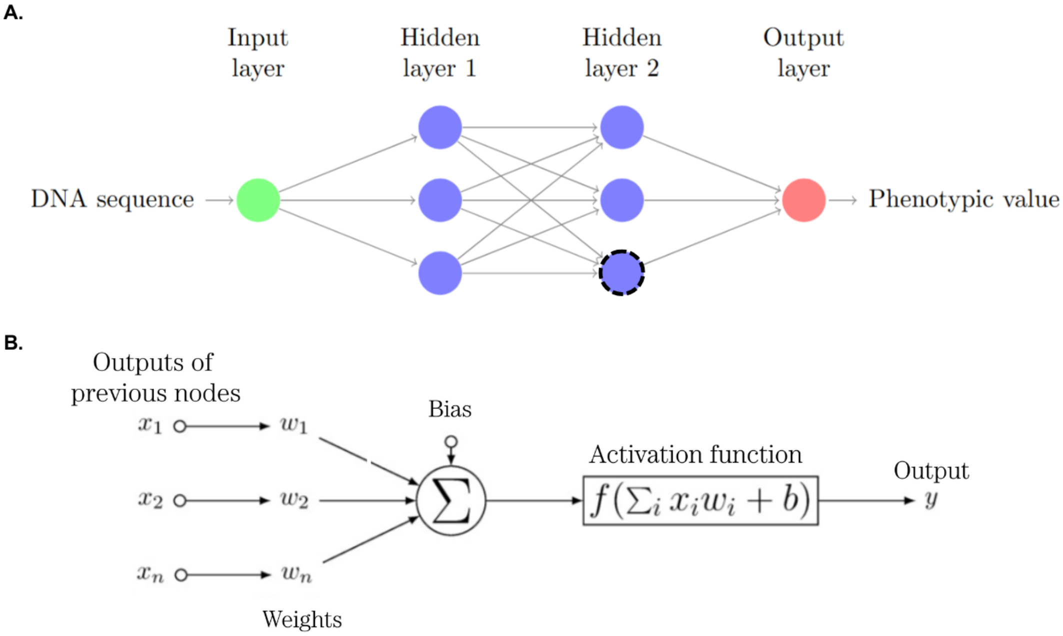


Figure 2 | **General architecture, information flow and node computation in neural networks. A.** Example of a neural network basic architecture, with an input layer, 2 hidden fully-connected layers and a final output layer. The input is a DNA sequence and the output is a phenotypic value. Drawn using online resource https://texample.net/neural-network/. **B.** Schematic of the mathematical processing in the node marked in A. The outputs of the neurons connected to the analysed node are multiplied by their weights and a bias parameter is introduced by the node. Then, this linear transformation is nonlinearly processed by the activation function to generate its output. Modified from Pérez-Enciso *et al.*[22].

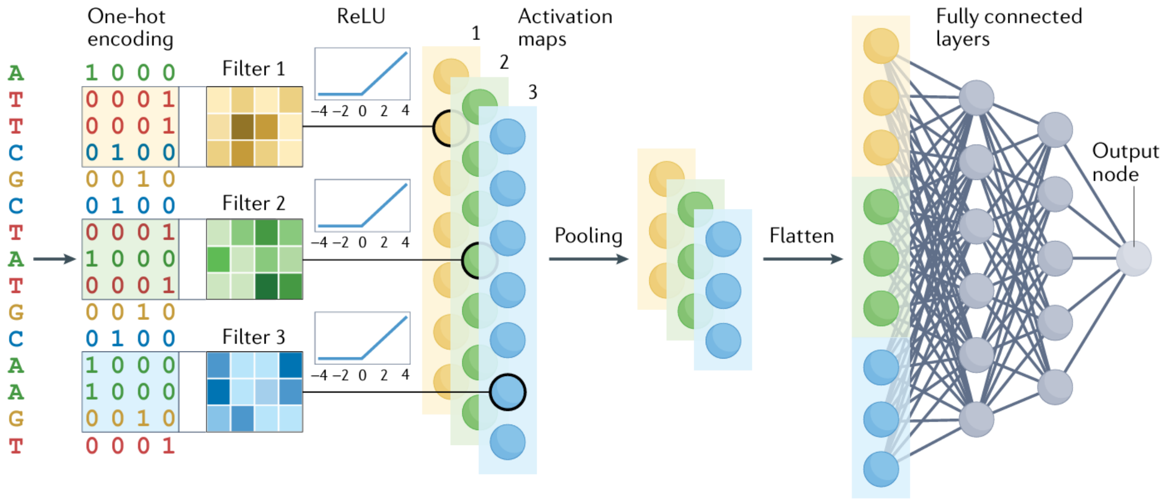


Figure 3 | **Convolutional Neural Network structure.** Initial filters are applied to the one-hot encoded sequence and ReLU functions generate activation maps that are posteriorly pooled and flattened to generate the input that enters the fully-connected network. Extracted from Novakovski *et al.*[20].

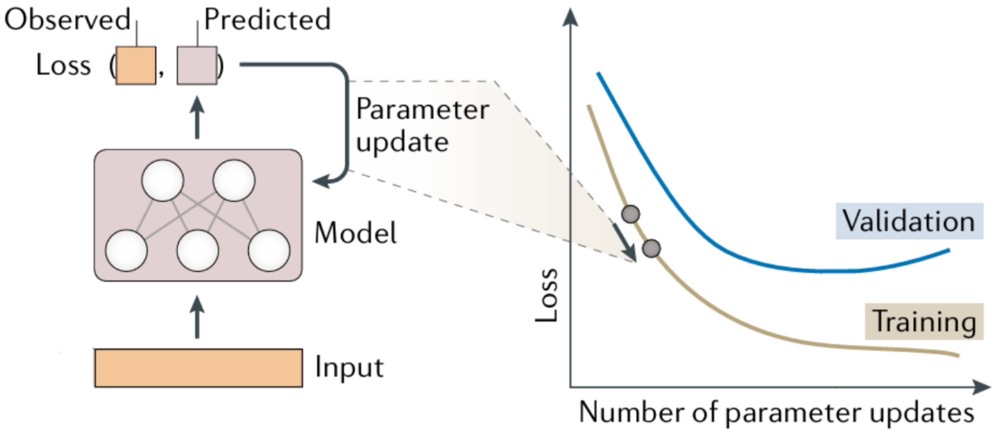


Figure 4 | **Schematic of model training.** The model predicts the labels of the inputs and calculates loss. After that, it uses an optimization method to update its parameters and generate another more accurate prediction. This change reduces the loss in the training set, which is represented in the graph. Modified from Eraslan *et al.*[6].

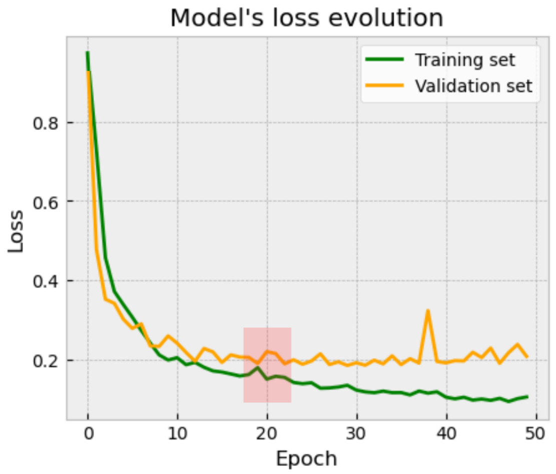


Figure 5 | **Example of overfitting detection through visual analysis of the loss change in training and validation sets during model training**. From approximately epoch 20, the functions decouple, so the parameter update is only improving the performance of the model on the prediction of the training set, not of the validation set. This enables overfitting detection.

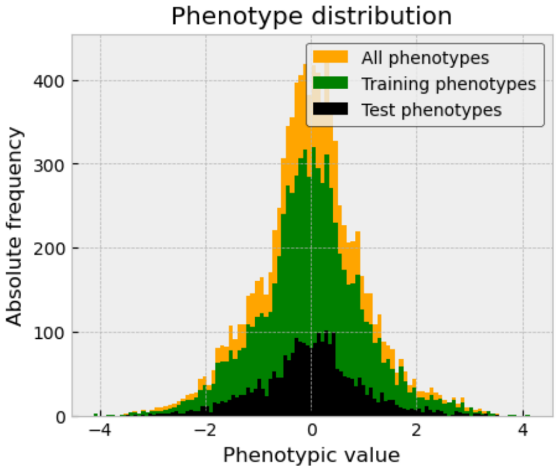


Figure 6 | **Example of distributional differences detection through a histogram.** In the graph, extreme phenotypes are less frequent than medium phenotypes, following a normal-like distribution, frequent in complex traits. In this case, there are apparently no distributional differences between training and test sets.

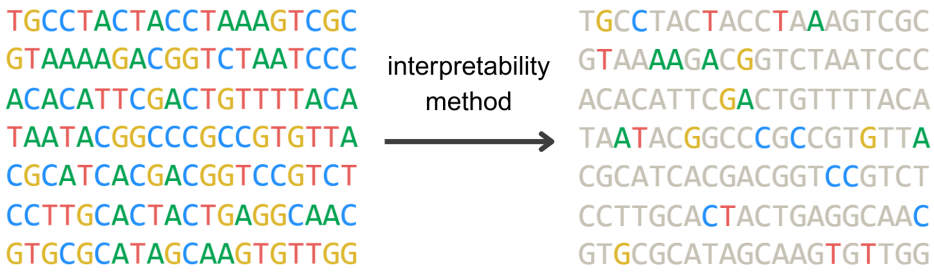


Figure 7 | **Example of interpretability applied to genomic data.** A set of nucleotides (representing a consecutive sequence or a selection of genomic positions) is illustrated. Interpretability methods can enable the infer of variants with a possible phenotypic effect by detecting a nucleotide's high importance in the model's prediction.

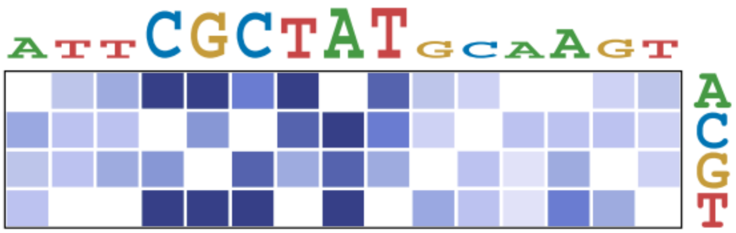


Figure 8 | **Example of a sequence logo and an attribution map for model interpretability.** Characters' height in the logo define the relative contribution of each position of the sequence and color gradients in the map indicate the importance of each nucleotide. In this example, an important region for the model's predictions is detected in positions 4-9. Modified from Novakovski *et al.*[20].

References

[1] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou. “An online platform for interactive feedback in biomedical machine learning”. In: *Nature Machine Intelligence* 2.2 (2020), pp. 86–88. DOI: 10.1038/s42256-020-0147-8.

[2] U. An, A. Pazokitoroudi, M. Alvarez, L. Huang, S. Bacanu, A. J. Schork, K. Kendler, P. Pajukanta, J. Flint, N. Zaitlen, N. Cai, A. Dahl, and S. Sankararaman. “Deep learning-based phenotype imputation on population-scale biobank data increases genetic discoveries”. In: *Nature Genetics* 55.12 (2023), pp. 2269–2276. DOI: 10.1038/s41588-023-01558-w.

[3] L. Barbadilla-Martínez, N. Klaassen, B. van Steensel, and J. de Ridder. “Predicting gene expression from DNA sequence using deep learning models”. In: *Nature Reviews Genetics* (2025). DOI: https://doi.org/10.1038/s41576-025-00841-2.

[4] A. L. Boulesteix and M. Wright. “Special issue: Artificial intelligence in genomics”. In: *Human Genetics* 141.9 (2022), pp. 1449–1450. DOI: 10.1007/s00439-022-02472-7.

[5] H. A. Elmarakeby, J. Hwang, R. Arafeh, J. Crowdis, S. Gang, D. Liu, S. H. AlDubayan, K. Salari, S. Kregel, C. Richter, T. E. Arnoff, J. Park, W. C. Hahn, and E. M. van Allen. “Biologically informed deep neural network for prostate cancer discovery”. In: *Nature* 598.7880 (2021), pp. 348–352. DOI: 10.1038/s41586-021-03922-4.

[6] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis. “Deep learning: new computational modelling techniques for genomics”. In: *Nature Reviews Genetics* 20.7 (2019), pp. 389–403. DOI: 10.1038/s41576-019-0122-6.

[7] L. L. Gu, R. Q. Yang, Z. Y. Wang, D. Jiang, and M. Fang. “Ensemble learning for integrative prediction of genetic values with genomic variants”. In: *BMC Bioinformatics* 25.1 (2024). DOI: 10.1186/s12859-024-05720-x.

[8] A. van Hilten, S. Katz, E. Saccenti, W. J. Niessen, and G. V. Roshchupkin. “Designing interpretable deep learning applications for functional genomics: A quantitative analysis”. In: *Briefings in Bioinformatics* 25.5 (2024). DOI: 10.1093/bib/bbae449.

[9] F. R. Holzlwimmer, J. Lindner, G. Tsitsiridis, N. Wagner, F. P. Casale, V. A. Yepez, and J. Gagneur. “Aberrant gene expression prediction across human tissues”. In: *Nature Communications* 16.1 (2025), p. 3061. DOI: 10.1038/s41467-025-58210-w.

[10] X. Huang, A. Rymbekova, O. Dolgova, O. Lao, and M. Kuhlwilm. “Harnessing deep learning for population genetic inference”. In: *Nature Reviews Genetics* 25.1 (2024), pp. 61–78. DOI: 10.1038/s41576-023-00636-3.

[11] F. Khodaee, R. Zandie, and E. R. Edelman. “Multimodal learning for mapping genotype-phenotype dynamics”. In: *Nature Computational Science* 5 (2025), pp. 333–344. DOI: 10.1038/s43588-024-00765-7.

[12] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint* (2014). URL: http://arxiv.org/abs/1412.6980.

[13] B. D. Lee, A. Gitler, C. S. Greene, S. Raschka, F. Maguire, A. J. Titus, M. D. Kessler, A. J. Lee, M. G. Chevrette, P. A. Stewart, T. Britto-Borges, E. M. Cofer, K. H. Yu, J. J. Carmona, E. J. Fertig, A. A. Kalinin, B. Signal, B. J. Lengerich, T. J. Triche, and S. M. Boca. “Ten quick tips for deep learning in biology”. In: *PLoS Computational Biology* 18.3 (2022). DOI: 10.1371/journal.pcbi.1009803.

[14] X. Liu, Y. Xu, Y. Luo, and L. Teng. “Prokaryotic and eukaryotic promoters identification based on residual network transfer learning”. In: *Bioprocess and Biosystems Engineering* 45.5 (2022), pp. 955–967. DOI: 10.1007/s00449-022-02716-w.

[15] T. R. MacNish, M. F. Danilevicz, P. E. Bayer, M. S. Bestry, and D. Edwards. “Application of machine learning and genomics for orphan crop improvement”. In: *Nature Communications* 16.1 (2025), p. 982. DOI: 10.1038/s41467-025-56330-x.

[16] J. Mbatchou, L. Barnard, J. Backman, A. Marcketta, J. A. Kosmicki, A. Ziyatdinov, C. Benner, C. O’Dushlaine, M. Barber, B. Boutkov, L. Habegger, M. Ferreira, A. Baras, J. Reid, G. Abecasis, E. Maxwell, and J. Marchini. “Computationally efficient whole-genome regression for quantitative and binary traits”. In: *Nature Genetics* 53.7 (2021), pp. 1097–1103. DOI: 10.1038/s41588-021-00870-7.

[17] C. Molnar, G. Casalicchio, and B. Bischl. “Interpretable Machine Learning - A Brief History, State-of-the-Art and Challenges”. In: *ECML PKDD 2020 Workshops*. Springer, 2020, pp. 417–431.

[18] O. A. Montesinos-López, A. Montesinos-López, C. M. Hernandez-Suarez, J. A. Barrón-López, and J. Crossa. “Deep-learning power and perspectives for genomic selection”. In: *Plant Genome* 14.3 (2021). DOI: 10.1002/tpg2.20122.

[19] M. Nachman. “An interview with Michael Nachman”. In: *eVOLUCIÓN* 19.11 (2024). Interview by Antonio Barbadilla.

[20] G. Novakovsky, N. Dexter, M. W. Libbrecht, W. W. Wasserman, and S. Mostafavi. “Obtaining genetics insights from deep learning via explainable artificial intelligence”. In: *Nature Reviews Genetics* 24.2 (2023), pp. 125–137. DOI: 10.1038/s41576-022-00532-2.

[21] F. F. Peleke, S. M. Zumkeller, M. Guitas, A. Schmitt, and J. Szymanski. “Deep learning the cis-regulatory code for gene expression in selected model plants”. In: *Nature Communications* 15.1 (2024). DOI: 10.1038/s41467-024-47744-0.

[22] M. Perez-Enciso and L. M. Zingaretti. “A guide for using deep learning for complex trait genomic prediction”. In: *Genes* 10.7 (2019), p. 553. DOI: 10.3390/genes10070553.

[23] G. Schiavo, F. Bertolini, S. Bovo, G. Galimberti, M. Muñoz, R. Bozzi, M. Candek-Potokar, C. Óvilo, and L. Fontanesi. “Identification of population-informative markers from high-density genotyping data through combined feature selection and machine learning algorithms: Application to European autochthonous and cosmopolitan pig breeds”. In: *Animal Genetics* 55.2 (2024), pp. 193–205. DOI: 10.1111/age.13396.

[24] E. E. Seitz, D. M. McCandlish, J. B. Kinney, and P. K. Koo. “Interpreting cis-regulatory mechanisms from genomic deep neural networks using surrogate models”. In: *Nature Machine Intelligence* 6 (2024), pp. 701–713. DOI: 10.1038/s42256-024-00851-5.

[25] A. I. Stevens, D. Pacia, J.-C. Belisle-Pipon, J. A. Parker, T. Ideker, V. Ravitsky, and I. Stevens. “The Ethics of Research at the Intersection of Functional Genomics and Artificial Intelligence”. In: *Hastings Center Issue Briefs* (2025).

[26] A. Talukder, C. Barham, X. Li, and H. Hu. “Interpretation of deep learning in genomics and epigenomics”. In: *Briefings in Bioinformatics* 22.3 (2021). DOI: 10.1093/bib/bbaa177.

[27] K. Wang, M. A. Abid, A. Rasheed, J. Crossa, S. Hearne, and H. Li. “DNNGP, a deep neural network-based method for genomic prediction using multi-omics data in plants”. In: *Molecular Plant* 16.1 (2023), pp. 279–293. DOI: 10.1016/j.molp.2022.11.004.

[28] S. Whalen, J. Schreiber, W. S. Noble, and K. S. Pollard. “Navigating the pitfalls of applying machine learning in genomics”. In: *Nature Reviews Genetics* 23.3 (2022), pp. 169–181. DOI: 10.1038/s41576-021-00434-9.

[29] A. K. Wong, R. S. G. Sealfon, C. L. Theesfeld, and O. G. Troyanskaya. “Decoding disease: from genomes to networks to phenotypes”. In: *Nature Reviews Genetics* 22.12 (2021), pp. 774–790. DOI: 10.1038/s41576-021-00389-x.

[30] Q. Yan, D. E. Weeks, H. Xin, A. Swaroop, E. Y. Chew, H. Huang, Y. Ding, and W. Chen. “Deep-learning-based prediction of late age-related macular degeneration progression”. In: *Nature Machine Intelligence* 2.2 (2020), pp. 141–150. DOI: 10.1038/s42256-020-0154-9.

[31] Y. Zhang, Y. Liu, and X. S. Liu. “Neural network architecture search with AMBER”. In: *Nature Machine Intelligence* 3.5 (2021), pp. 372–373. DOI: 10.1038/s42256-021-00350-x.

[32] Y. Zhao, Z. Fu, E. J. Barnett, N. Wang, K. Zhang, X. Gao, X. Zheng, J. Tian, H. Zhang, X. T. Ding, S. Li, S. Li, Q. Cao, S. Chang, Y. Wang, S. V. Faraone, and L. Yang. “Genome data based deep learning identified new genes predicting pharmacological treatment response of attention deficit hyperactivity disorder”. In: *Translational Psychiatry* 15.1 (2025), p. 46. DOI: 10.1038/s41398-025-03250-5.

[33] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, and A. Telenti. “A primer on deep learning in genomics”. In: *Nature Genetics* 51.1 (2019), pp. 12–18. DOI: 10.1038/s41588-018-0295-5.