Genome Biology

RESEARCH

Open Access

# Large sample size and nonlinear sparse models outline epistatic effects in inflammatory bowel disease

Nora Verplaetse[1*], Antoine Passemiers[1], Adam Arany[1], Yves Moreau[1] and Daniele Raimondi[1*]

*Correspondence: nora.verplaetse@kuleuven.be; daniele.raimondi@kuleuven.be

[1] Department of of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium

## Abstract

**Background:** Despite clear evidence of nonlinear interactions in the molecular architecture of polygenic diseases, linear models have so far appeared optimal in genotype-to-phenotype modeling. A key bottleneck for such modeling is that genetic data intrinsically suffers from *underdetermination* ($p \gg n$). Millions of variants are present in each individual while the collection of large, homogeneous cohorts is hindered by phenotype incidence, sequencing cost, and batch effects.

**Results:** We demonstrate that when we provide enough training data and control the complexity of nonlinear models, a neural network outperforms additive approaches in whole exome sequencing-based inflammatory bowel disease case–control prediction. To do so, we propose a biologically meaningful sparsified neural network architecture, providing empirical evidence for positive and negative epistatic effects present in the inflammatory bowel disease pathogenesis.

**Conclusions:** In this paper, we show that underdetermination is likely a major driver for the apparent optimality of additive modeling in clinical genetics today.

**Keywords:** Genome interpretation, Machine learning, Neural networks

## Background

Since the sequencing of the first whole human genome in 2003 and the advent of high-throughput sequencing techniques [1], clinical and population genetics have been blooming fields of research. Every year, an increasing number of genetic studies are published worldwide, dedicated to understanding the relationship between genotype and phenotype, which is a crucial step towards precision medicine and other health applications [2, 3]. Nevertheless, given that each individual genome contains about three million variants, together with the challenges in gathering large, homogeneous cohorts because of limited phenotype incidence, sequencing cost, and batch effects, most of these studies suffer from a limited sample size $n$ relative to the number of variants $p$ (also called the $p \gg n$ setting in statistics). This *underdetermination* of genetic datasets has indeed been

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

a crucial problem so far, limiting the statistical power of analysis approaches and contributing to puzzling results, such as the missing heritability problem [4, 5]. For this reason, the go-to methods for genomic data analysis have historically always been additive (linear) models, such as univariate analysis of association (i.e., genome-wide association studies (GWAS) [6]), polygenic risk scores (PRS) [7], and linear mixed models [8, 9].

Relying on additive models has until now appeared to be optimal in the genetic context [10], thanks to their interpretability and simplicity, which conveniently translates into robustness at unfavorable $n/p$ ratios. Researchers thus adopted this *lens* of additivity in their investigations to address the ubiquitous underdetermination of their datasets, notwithstanding biological arguments for the existence of nonlinear genetics effects, such as epistasis [5, 11–17]. With the blooming of complex machine learning (ML) methods (e.g., deep learning) in the life sciences [18–20] and the growth of the available sample sizes because of advances in sequencing technologies [1] and the accompanying decrease in cost, it is now timely to consider whether more sophisticated approaches could also benefit genetic data. Such approaches could allow us to perceive aspects of the genetics landscape that are currently masked by the omnipresent lens of additivity.

Early attempts to apply neural networks (NN) to genome interpretation (GI), namely the explicit modeling of the relationship between genotype and phenotype, did not succeed in outperforming additive models [10, 21]. The debate on the opportunity of using nonlinear models is still widely open in clinical genetics [10, 21–29] and agricultural biotechnology [30–34]. However, the current effectiveness of additive modeling in genetics raises more questions than answers [15]. Most of the molecular mechanisms producing phenotypes described by systems and molecular biology studies are replete with nonlinear interactions between the components of extremely complex systems [5, 11–17, 35]. In these disciplines, pure additive effects are as uncommon as they are widespread in current genetics literature.

From a data science perspective, we see two main limitations that could have prevented NN models for GI from outperforming additive approaches. First, most of the NNs applied on human genetic datasets of large sample sizes used SNP array data that has been prefiltered with a GWAS-based univariate variant selection [21, 26–29], thereby possibly removing a priori much of the nonadditive interaction signal from the input. Second, NN GI attempts on WES/WGS data have been only sporadic and mostly restricted to very small sample sizes [22–24, 36, 37], while it is empirically clear that deep learning methods require large datasets to perform best. If nonlinear models can outperform linear ones, a key question is then how large a dataset needs to be before a nonlinear model starts offering any advantage.

In this article, we test the hypothesis that underdetermination is one of the major drivers for the apparent optimality of additive models in genetics. To do so, we exploit the intrinsic sparsity of biological networks to build the smallest NN model possible that is still capable of nonlinear inference and apply it to one of the largest available WES case–control datasets. This is a dataset for inflammatory bowel disorder (IBD), which constitutes an ideal test case for our study given its polygenic nature and high heritability estimates.

We show that once a sufficiently large sample size is reached, NNs reliably outperform conventional additive approaches. Moreover, we show (1) how this result provides

empirical evidence that, given enough samples and a model able to detect them, epistatic effects start to emerge from the data and (2) that positive/negative epistasis plays a role in the genetic mechanism underlying IBD. Our study indicates that the main reason for the effectiveness of additive models for the analysis of genetic data is their robustness at small sample sizes, thereby recontextualizing them as a situational and temporary necessary solution instead of the undisputed *statistical model* for genetics. Our results indicate that as larger genetic datasets become available, we can envision a systematic nonlinear advantage for NN models applied to GI.

## Results and discussion

### Small, biologically sparsified NNs outperform additive and nonlinear baselines

To address the predominant underdetermination of genetic datasets, but still benefit from nonlinear modeling, we minimized the gap between the number of samples and the number of model parameters by applying the smallest possible NN (but which could still perform nonlinear inference) to one of the largest whole exome sequencing (WES) inflammatory bowel disease (IBD) cohorts available.

Similarly to [36, 37], to obtain a maximal reduction of the number of parameters in our NN, we start by using a compact, gene-centric representation of the input WES data in which each gene is represented by the observed *mutational load* it carries (see the "Material and methods" section). Genes are thus the base biological semantic entity in our models (see input data in Fig. 1). Several variations of this encoding have been explored as well (see Additional file 1: Fig. S1, Additional file 2: Table S1, Additional file 3: Table S2) without observing performance improvement.

In Table 1, we benchmark three NN architectures specifically designed for the WES-based case–control IBD prediction against conventional methods, such as an additive model and RF. We tested several alternatives for the additive model. In Table 1, we only show the best performing additive approach, which is a logistic regression with

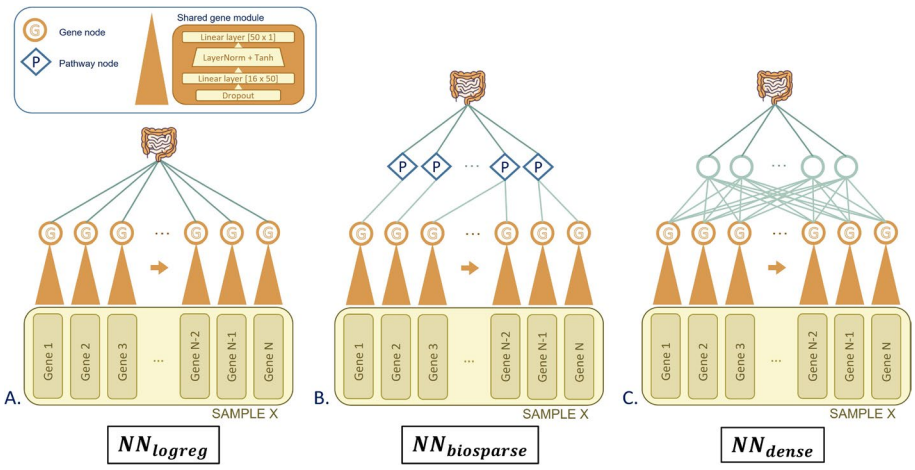


**Fig. 1** Panel overview of NN architectures, each build on top of the shared gene module with increasing complexity. Panel **A** shows $NN_{logreg}$, the simplest architecture connecting the gene neurons *G* directly to the output, implementing a LogReg of the |*G*| neurons. Panel **B** illustrates $NN_{biosparse}$ with connections between gene neurons *G* and pathway neurons *P* based on the KEGG database. The model in panel **C**, $NN_{dense}$, adds a fully connected hidden layer to increase model expressiveness

**Table 1** Benchmark of our NN architectures specifically designed for the WES-based case–control IBD prediction against conventional methods

| Model | ROC AUC[a] | Number of parameters |
|---|---|---|
| Best additive model | 0.728 (0.00599) | 1,734,301 |
| Random forest | 0.688 (0.00578) | |
| $NN_{logreg}$ | 0.753 (0.0117) | 24,379 |
| $NN_{biosparse}$ | 0.758 (0.00689) | 25,503 |
| $NN_{dense}$ | 0.743 (0.00944) | 6,515,063 |
| $NN_{linear}$ | 0.717 (0.0261) | 25,503 |

[a] Performance given as mean (standard deviation) of test set ROC AUC from 10 different full threefold cross-validation runs with identical splits across models

$L_2$ regularization. We could not add polygenic risk scores (PRS) to this benchmark, because most of the variants found by GWAS are noncoding, resulting in only 42% of IBD GWAS variants [38, 39] present in our WES IBD dataset. A PRS computed on these 42% of GWAS variants produces a test ROC AUC of 0.563. Methodological details and additional analyses on this variant selection can be found in Additional file 4: Note S1.

The architectures of the NNs we tested (as reported in Table 1) are shown in Fig. 1 by increasing complexity from left to right. In all of them, a shared gene module processes separately the features describing each gene (see the "Material and methods" section) and the differences between the architectures lie in the arrangement of the modules connecting the $|G| =$ neurons and the output neuron. These NN architectures were designed so that they represent three levels of expressivity (i.e., complexity of the decision boundaries they can learn). This translates in their increasing capability of modeling certain categories of interactions between the $|G|$ neurons. In the simplest model, called $NN_{logreg}$ (see Fig. 1A), each gene latent representation produced by the shared gene module is connected to the output neuron similarly to a logistic regression (LogReg). This model will therefore not be able to model any interactions between the $|G|$ neurons. The second model (see Fig. 1B) represents an intermediate level of expressivity whereby two gene neurons are connected to and can thus interact in at most one hidden neuron of the next layer. The choice of the connection arrangements was based on the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [40] with the hidden nodes representing the KEGG pathways and the connection for each gene being picked out of the known KEGG gene-pathway relations. Since each neuron and each connection in this architecture has a biological meaning, we refer to this model as a biologically sparsified NN ($NN_{biosparse}$). Finally, Fig. 1C shows an NN with a standard densely connected hidden layer between gene neurons and output ($NN_{dense}$). This way all gene neurons are connected to all hidden neurons, making it possible to model any kind of interaction between them (universal approximator) [41].

In Table 1, we show the ROC AUC obtained by each model on IBD case–control discrimination, alongside their number of trainable parameters. Our NN models outperform the best performing RF (two-sided $t$-test, corrected for correlation between ROC AUC measurements [42]: $p$-value $p = 5.77\text{e}{-7}$) and the best additive model (corrected two-sided $t$-test [42] $p = 0.00457$). Among the three NN architectures we presented, $NN_{biosparse}$, which has the lowest number of parameters, is also the best

performing model. A detailed table of RF performance on different data representations can be found in Additional file 5: Table S3.

### Sample size can explain the apparent optimality of additive models in genetics

Table 1 shows that our NN models outperform the best results obtained by classical additive approaches, which is not always the case in genetic studies [10, 21, 22, 28, 30–33]. To investigate in more details at which conditions this advantage appears, we show in Fig. 2 the comparison of the prediction performances of these two approaches in function of the sample size available during training. By looking at the trajectories of these models, we can see that when at least 80% (3038) of samples are used for training, our best approach ($NN_{biosparse}$) starts to outperform the additive model and the improvement eventually becomes statistically significant. This empirically demonstrates that for this dataset the optimality of additive models is not general, but sample size-dependent.

The balance between model expressivity (ability to learn nonlinear patterns) and prediction quality is nevertheless quite delicate in this ($p \gg n$) dataset, and is directly related to the bias–variance trade-off [43, 44]. Bias and variance can be considered as two competing qualities of statistical models. Bias refers to the error the model makes because of incorrect assumptions and limited expressive power, and it is linked to *underfitting*. Variance represents the ability of the model to learn small variations (and noise) in the training data, and relates to the risk of *overfitting*. Simple additive approaches have high bias but, while their ability to model complex patterns is limited, the uncertainty about optimal model parameters will decrease faster than for more complex models. Genetic datasets have thousands or even millions of possibly noisy observations (variants) for each sample, and the intrinsic simplicity of additive model helps them handle the resulting uncertainty. At small sample size, additive models can

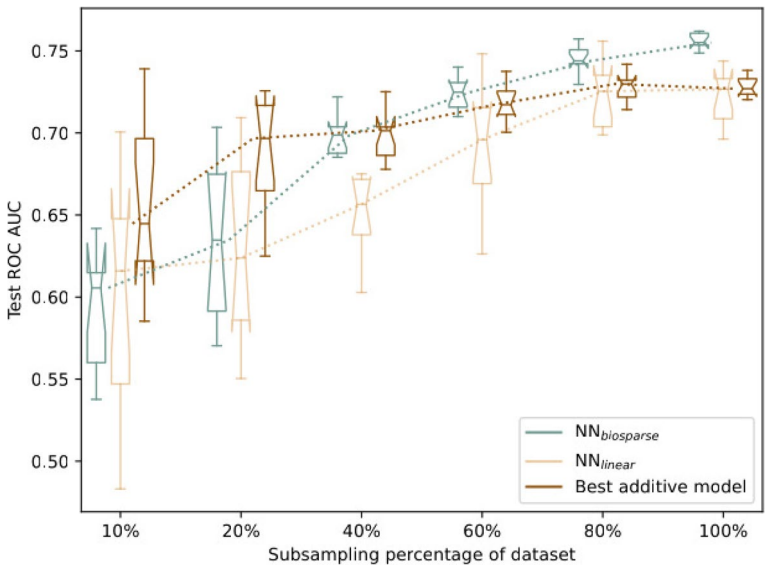


**Fig. 2** Performance using different random subsets containing 10%, 20%, 40%, 60%, 80%, and 100% of the dataset. Models shown are the best additive model ($L_2$ penalty), $NN_{biosparse}$ and $NN_{linear}$. Performances are measured using ten different threefold cross-validation runs, using identical splits for the three models

learn only a simple hypothesis from the data, while more expressive (high variance) models such as NNs are at greater risk of achieving poor generalization.

#### *Absence of evidence is not evidence of absence*

Figure 2 empirically shows that even if at low sample sizes additive models are optimal, there can exist a sample size threshold $t$ above which certain nonlinear patterns become detectable by sufficiently expressive models. In the case of this IBD cohort and our $NN_{biosparse}$, we were able to empirically identify $t \approx 3000$, since it happens to be lower than the dataset sample size ($t < n$). We can speculate that the same behavior could be observed in several other real life genetic datasets for appropriately large sample sizes. Therefore, to have a chance to be truly conclusive and comprehensive, the currently ongoing debate on the use of additive versus nonlinear models on genetic datasets should make sure to investigate the behavior of both models across a sufficiently wide range of sample sizes. This is crucial to ensure that the apparent optimality of additive models does not come from the fact that $t > n$ in the datasets under consideration. Furthermore, this might help to get a more complete molecular picture, thereby addressing the well-known missing heritability problem.

To exclude that the behavior in Fig. 2 is simply the result of the different format of the input representations between the additive model (which takes as input a vector of all the variants) and our NNs (which use our gene-centric data representation, see the "Material and methods" section), we ran an additional experiment. We add a third model to Fig. 2 to compare our $NN_{biosparse}$ with an NN with the same architecture, but with identity functions instead of the hyperbolic tangent tanh nonlinear activations ($NN_{linear}$). This makes this $NN_{linear}$ effectively equivalent to a linear model. Again, we see the same pattern as for the best additive model, indicating that the higher performances at high sample size emerge from the nonlinearity of the model and not from the different input encoding per se.

### Empirical evidence for positive/negative epistasis in IBD

Nonlinear interactions between alleles, called epistasis, are ubiquitous at the molecular level [5, 11–17]. As shown in Fig. 3D, G, epistasis between two alleles at two loci (genes, in our case) means that the phenotypic effect of the allele at Locus 1 depends on the allele present at Locus 2 [15], thus deviating from a situation in which each locus independently influences the phenotype (additivity, see Fig. 3A). In positive and negative epistasis (Fig. 3D), the epistatic loci are affecting the magnitude of each other's effect on the phenotype. Conversely, when two alleles alone have a negative phenotypic effect, but taken together they have a positive (or vice versa) effect (see lock-and-key model example in Additional file 6: Fig. S2), this is called reciprocal sign epistasis (Fig. 3G). Although our NNs produce individual-specific probability-like predictions, for illustrative purposes in Fig. 3, we associate log odds ratios to the hypothetical alleles shown, because we needed a metric that can be negative (i.e., indicating a protective effect of variants) to more clearly illustrate reciprocal sign epistasis.

In many phenotypes, especially in polygenic diseases such as IBD, the interactions between multiple genetic factors can contribute to the phenotype and the modulation of its severity [16, 17, 45, 46]. Therefore, it is unsurprising that, if the bias–variance

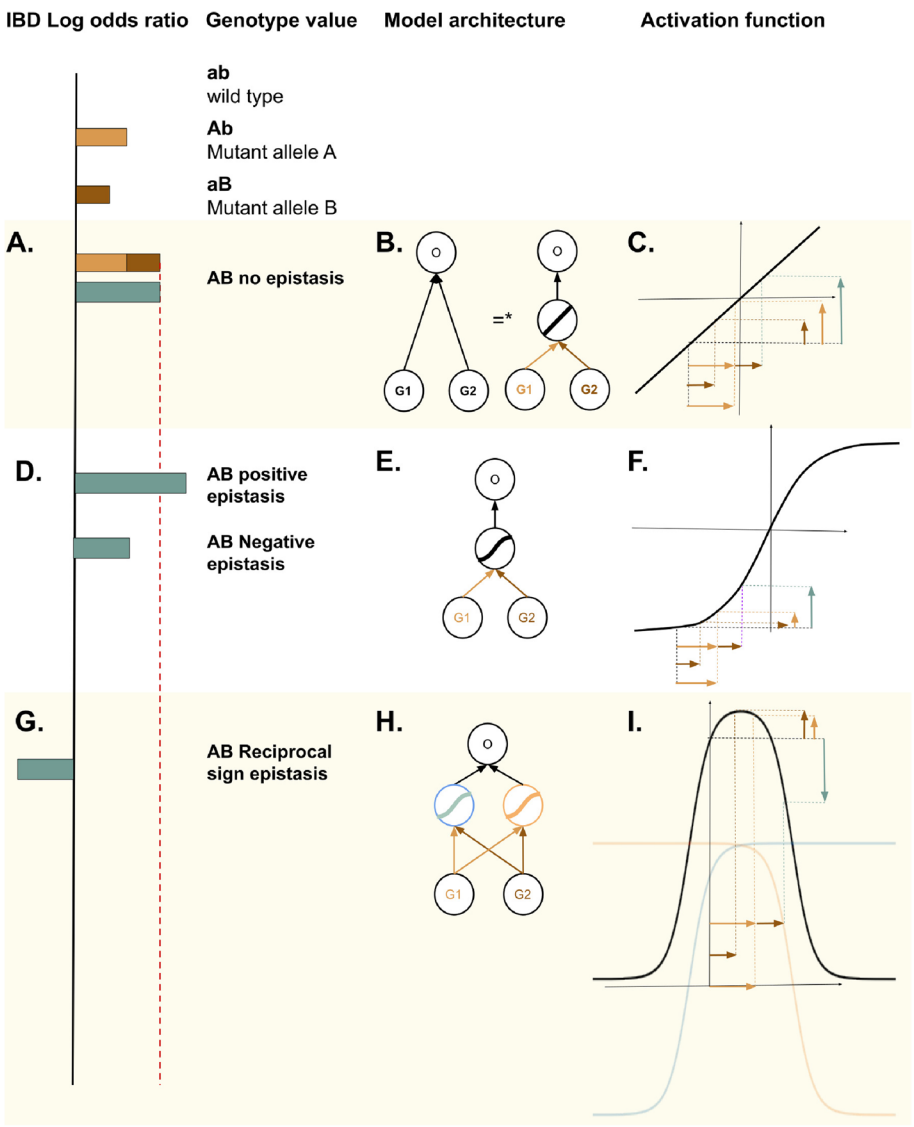


**Fig. 3** Different forms of biological epistasis (left) with minimal NN architectural requirements, illustrated by a small NN with two gene neuron inputs (middle) and the activation function of the hidden layer on top of the gene neurons (right). *A linear function of a linear function is a linear function

trade-off allows a nonlinear model to robustly detect these patterns, this nonlinear model will outperform additive ones.

### *Relating the expressivity of NN architectures to epistasis types*

From a mathematical perspective, a peculiarity of our $NN_{biosparse}$ model (see Fig. 1B) is that the arrangement of the sparse interactions between the gene neurons $G$ and the pathway neurons $P$ is less expressive than a fully connected (dense) layer (see $NN_{dense}$ in Fig. 1C), and therefore, it cannot model every pattern that is accessible to $NN_{dense}$. To draw biological conclusions from this observation coupled with the observed prediction performances shown in Table 1, we link the expressivity of the three different NN architectures (i.e., the types of nonlinear patterns they can model)

with the different forms of biological epistasis that they can detect (see Additional file 7: Table S4).

$NN_{logreg}$ (Fig. 1A) is only capable of modeling additivity between genes, because no interaction between features (gene neurons *G* in this case) is allowed in their architecture (see Fig. 3A–C). The sparse architecture of $NN_{biosparse}$, with the gene–pathway layer (see Fig. 1B) can model negative and positive epistasis between genes, because genes of the same pathway are allowed to interact via that pathway neuron, which has a nonlinear activation function (see Fig. 3E, F). Since our $NN_{biosparse}$ design particularly allows for only one connection from each gene neuron to the next layer, two genes will never interact in more than one pathway neuron, making it impossible to model *nonlinearly separable patterns*, such as reciprocal sign epistasis. A fully connected layer such as in model in Fig. 1C, however, can model such epistasis, because it can model arbitrarily complex functions, including nonlinearly separable ones, as a *composition* of nonlinear activations (see Fig. 3I).

The last column of Fig. 3 illustrates an intuitive explanation of this behavior, by showing the output of three minimal examples of NNs representing our three architectures, having just two gene neurons as input. With a linear activation (see Fig. 3C), the joint effect of the two genes will always be additive, namely the simple sum of their separate effects, making it impossible to model epistatic interaction. However, if we add a nonlinear activation, such as tanh, to the same model (Fig. 3F), the same shift on the $x$-axis can cause different shifts on the $y$-axis, depending where on the $x$-axis we are located. In Fig. 3F, we show an example of how such a model could learn a positive epistatic interaction between two genes, where the effect of the two genes together is larger than the sum of their separate effects. Nevertheless, tanh is a monotonically increasing function, making it impossible to model reciprocal sign epistasis, because the *joint* effect of both inputs cannot have opposite direction with respect to their separate effects. To allow this, we will need to break the monotonicity of the curve by allowing the NN architecture to learn how to compose the activation functions of more than one hidden neuron, thus needing more than one hidden neuron via which the genes can interact (see Fig. 3I). Intuitively, reciprocal sign epistasis is similar to the classical XOR classification problem in ML [47], that also needs at least two hidden neurons, representing an OR and NOT AND gate, to be modeled (see Additional file 6: Fig. S2).

#### *Reciprocal sign epistasis is not detectable on the current dataset*

From the performance in Table 1, we see that, given the currently available sample size, there is no advantage in investing more parameters (higher model complexity) towards training NNs capable of modeling reciprocal sign epistasis ($NN_{dense}$) on this IBD cohort, since the significant increased model complexity produces lower AUC. Given the currently available samples in this IBD cohort, a model expressivity sufficient to address positive/negative epistasis seems therefore to be optimal. Similarly to what we previously discussed between additive and nonlinear models, the explanation for this fact may be that either (1) reciprocal sign epistasis does not have a major role in IBD pathogenesis or (2) the current sample size is insufficient to detect it.

To empirically verify that this inability is not just the result of the over-parametrization of $NN_{dense}$ with respect to the dataset size, causing a suboptimal positioning in

the bias–variance spectrum, we ran an additional experiment building two additional biologically sparsified architectures. The first architecture, shown in Additional file 8: Fig. S3A, has an identical number of neurons as $NN_{biosparse}$ but allows for more than one connection for each gene, representing all the gene-pathway relations present in the KEGG database [40]. The second additional model, shown in Additional file 8: Fig. S3B, contains an extra sparse layer that mimics known gene–gene interactions from the Interactome database [48] before the gene–pathway layer. In both models each pair of genes can possibly reach the output neuron following more than one path (see Fig. 3H), thereby achieving the requirements necessary to model reciprocal sign epistasis as well. Notwithstanding the lower number of parameters compared to $NN_{dense}$ (see Additional file 7: Table S4), none of these models outperform $NN_{biosparse}$ (with mean (std) ROC AUC 0.756 (0.00829) and 0.729 (0.0110), respectively), suggesting that indeed either reciprocal sign epistasis is not present or not robustly detectable at this sample size. It will be interesting to further investigate these more complex architectures in future larger cohorts.

#### Random sparsity outperforms biologically meaningful and learned sparsity

Table 1 shows that the sparse gene–pathway layer in our $NN_{biosparse}$ is instrumental towards its performance, because it allows an optimal level of expressivity while avoiding over-parametrization. But does the biologically meaningful arrangement of these connections between neurons mimicking genes and pathways also play a role towards prediction accuracy? In Fig. 4, we compare the performance of the knowledge-based sparsity pattern extracted from KEGG with three other connection arrangements using the same number of connections, thus without increasing the number of parameters.

The first surprising result is that the knowledge-based sparsification does not outperform the random one (see Fig. 4). Using biological networks to sparsify gene–pathway interactions in an NN layer seemed an obvious choice, since it assumed that genes

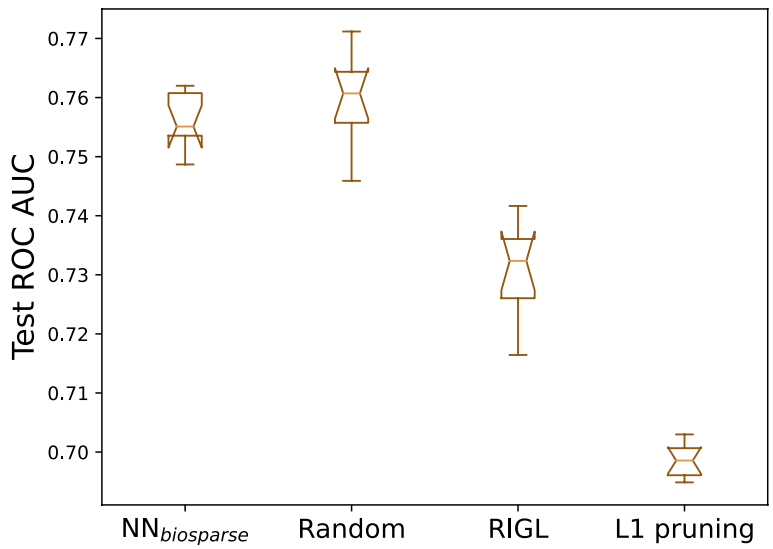


**Fig. 4** Comparison of four sparsification methods: **A** based on biological pathways (KEGG), **B** randomly, **C** learned using RigL algorithm, and **D** learned using heavy $L_1$ regularization. For fair comparison, all models shown have the same number of connections and hidden units

participating in the same pathway could be more prone towards modulating each other in an epistatic fashion. However, databases such as KEGG are still far from complete: only one third of the genes in our dataset could be connected to a KEGG pathway (as described in the "Material and methods" section, we joined them to a "dummy neuron" to avoid discarding them).

In Fig. 4, we also benchmarked two ways to *learn* optimal sparsity arrangements from the data, using the RigL [49] method and imposing $L_1$ regularization on the gene–pathway connections. In both cases, the learned sparsity yielded lower performance. We hypothesize that by learning which connections to make and thus optimizing sparsity during training, we offer the model another way to overfit the noisy training data, in contrast to the strong regularizing effect of random sparsity. For more details on the RigL method and its benchmark settings, see Additional file 9: Fig. S4 and Additional file 10: Suppl. Method.

Since the prediction performance does not seem to benefit from learned or biologically meaningful arrangements of the hidden layer sparsity, we investigate whether the *level* of sparsity is playing a more important factor role. In Fig. 5, we compare the performance of several random sparsity degrees, ranging from only connecting 25% of the genes to a pathway hidden neuron, to a fully connected layer with 281 connections for each gene. The figure shows an optimum at 75% of the genes connected, suggesting that the model could be regularized even further by dropping some genes completely (although the difference in performance for 75% and 100% of the genes connected is not statistically significant).

This shows that irrespectively of the *arrangement* of the connections, the main driver of the model performance is the degree of sparsity, and the effect it has on its complexity (i.e., its position in the bias–variance spectrum).

### Further exploiting random connections with ensemble learning leads to the best predictions

Given the results obtained by random sparsification in our models, we decided to exploit this behavior to build an ensemble learner of randomly sparsified NNs to solve the IBD prediction task. Ensemble methods follow the intuition that combining a set of (possibly weak) base learners with low correlation between them will lead to better prediction, more robust to noise [50]. The crucial step in this process is minimizing the

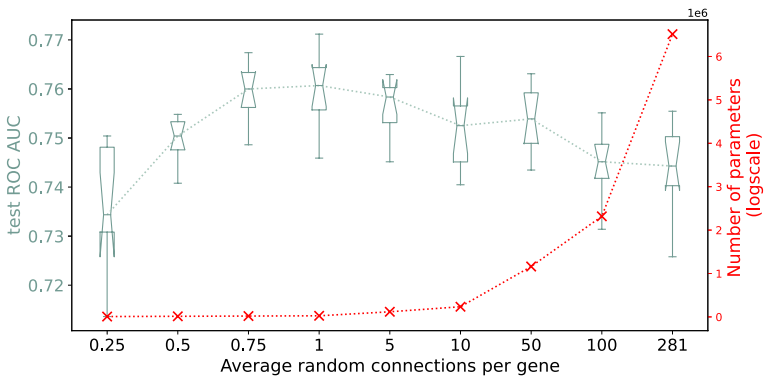


**Fig. 5** Effect of degree of sparsity on performance, going from only 25% of the genes randomly connected to a fully connected layer. Red crosses indicate the number of parameters for each model

correlation between the base learners by injecting a random element in each of them. Conventionally, the randomness is introduced through features and input sample subsampling. Here, we randomized instead the connection arrangements in our sparse NN layer. We thus train 100 randomly sparsified NNs on the full training set and average their prediction. In Table 2, we show that the ensemble model has an almost 2% increase in ROC AUC with respect to $NN_{biosparse}$ (corrected two-sided *t*-test [42] $p = 0.04146$).

For completeness, we performed additional experiments aiming at predicting Crohn's disease and ulcerative colitis, the two main IBD subtypes, separately, notwithstanding the consequently smaller sample sizes available for training (see Additional file 11: Fig. S5, Additional file 12: Table S5, Additional file 13: Note S2).

## Conclusions

In this article, we zoomed in on the role that the widespread underdetermination of genetic datasets ($p \ll n$) plays on the performance of different modeling methods, depending on their position in the bias–variance spectrum. We thereby showed that underdetermination is plausibly one of the major drivers for the apparent optimality of additive modeling in clinical genetics today. We showed that when we reach more favorable $n/p$ ratios by constraining the NN complexity and providing enough training data, NNs outperform conventional additive models in WES-based IBD case–control discrimination. To do so, we proposed a biologically meaningful sparsified NN architecture, but further experiments showed that the degree of sparsity is more decisive for predictive performance than the biological meaningfulness of the connections, again emphasizing the importance of underdetermination. Furthermore, by linking the expressivity of our NN architectures with different forms of biological epistasis, we provided empirical evidence that positive/negative epistatic effects are present in the genetic architecture of IBD. Our result suggests that larger cohorts will allow further improvement through more complex modeling architectures in the near future, thereby enabling a new, nonadditive *lens* on genome interpretation and contributing to a more complete molecular picture of all kinds of phenotypes.

## Materials and methods

### Dataset

We analyzed the data from the inflammatory bowel disease (IBD) exome sequencing study (dbGaP phs001076.v1.p1) [51], a case–control study containing whole exome sequences (WES) for 3318 IBD cases and 480 controls. The 3318 cases consist of the

**Table 2** Ensemble model

| Model | ROC AUC[a] | Parameters |
|---|---|---|
| $NN_{biosparse}$ | 0.758 (0.00689) | 25,503 |
| Randomly sparsified (100% genes connected) | 0.759 (0.00778) | 25,503 |
| Randomly sparsified (75% genes connected) | 0.760 (0.00571) | 19,708 |
| Ensemble (100% genes connected) | 0.774 (0.00276) | 25,503 |
| Ensemble (75% genes connected) | 0.776 (0.00265) | 19,709 |

[a] Performance given as mean (standard deviation) of test set ROC AUC from 10 different full threefold cross-validation runs

two main subtypes of IBD: 2036 Crohn's disease (CD) patients and 1215 ulcerative colitis (UC) patients. For 67 cases, the IBD subtype is unknown (indeterminate colitis). In the control group, 39.4% of the participants are male compared to 46.7% of the cases. The data is provided as a VCF file listing the observed variants.

### Encoding the exome variants into feature vectors

Encoding WES data into ML-ready feature vectors is not trivial, since each individual carries an arbitrary number of variants (on average 47,403 in our dataset), and the entire dataset covers a large pool of variants (1,733,480 unique variants in our dataset). To overcome this issue, we followed the approach we adopted in our previous works [36, 37], which we briefly summarize here.

We first annotated all variants in the VCF with Annovar [52], assigning each of them to a gene and to one of the following 16 functional classes: UTR3, UTR5, splicing, upstream, downstream, intronic, intronic ncRNA, exonic ncRNA, splicing ncRNA, exonic non-frameshift insertion, exonic frameshift insertion, exonic non-frameshift deletion, exonic frameshift deletion, exonic stoploss, exonic stopgain, exonic startloss and exonic nonsynonymous. We then summarized the annotated variants by aggregating them at the gene level [36, 37], making the gene the base semantic entity in our feature encoding. To do so, we count for each gene how many variants of a specific functional class map to it. In this way, each gene's *mutational load* is quantified by a 16-dimensional histogram. Each WES sample is thus described by a matrix of size $16 \times 23,177$ with 23,177 being the total number of genes in the dataset. We refer to this representation as *gene-centric encoding* [36, 37]. It is possible to further enrich this representation using gene-level and variant-level scores as additional features [37]. To avoid numerical issues, we standardized the counts of each type of variant across all genes.

### A biologically meaningful sparse NN architecture for GI

#### *A shared gene module decouples the number of parameters from the number of genes*

After defining our gene-centric feature encoding for WES data, we describe the NN architectures we devised to process this data and address the $p \gg n$ issue. Similarly to our previous work [36, 37], we process each gene histogram in our gene-centric encoding with a shared gene module (see Fig. 1). By sharing the same NN module across all the genes, we minimize the number of parameters needed to transform the (16, 23177) input matrix representing each sample into a compact latent vector of length 23,177, where each gene is described by just 1 value (the output of the shared gene module, mentioned as neuron $G_i$ for the $i^{th}$ gene, with $1 \leq i \leq 23,177$ and the total set of gene neurons being $G = \{G_1, \ldots, G_{23,177}\}$). In all the models presented here, we used the same architecture for this shared gene module, with 1 hidden layer of 50 neurons (see Fig. 1)) (chosen as the smallest architecture needed to perfectly fit the training data after a coarse grained tuning phase), which brought the total number of trainable parameters to a modest 850 weights plus 51 biases.

#### *Exploiting the small-worldness of biological networks to sparsify the NN architecture*

Figure 1 illustrates the three different NN achitecture prototypes built on top of the shared gene module described above. The simplest approach connects the $|G| = 23,177$ neurons to the output prediction, implementing a logistic regression (LogReg) of the gene neuron activations (see Fig. 1A). We refer to this model as $\text{NN}_{\text{logreg}}$. Adding a dense hidden layer (see Fig. 1C) between the gene layer and the output allows us to surpass the limited expressivity of the simple LogReg and to capture nonlinear interactions between the genes, but at the same time it causes a combinatorial explosion of the number of parameters. We refer to this model as $\text{NN}_{\text{dense}}$.

To obtain the best of both worlds, meaning nonlinear inference and maximal parameter reduction, in Fig. 1B, we sparsify the connections between the $|G|$ neurons and the next layer. To do so in a *biologically meaningful* manner, we randomly pick for each gene one of the KEGG pathways it contributes to [40]. Two thirds of the genes in the dataset (15,219 out of 23,177 genes) do not belong to any known KEGG pathway and they were connected to one *dummy* pathway neuron, to avoid discarding them. The idea behind this architecture is that we assign a biological meaning to each NN module, since each neuron in Fig. 1B represents a biological unit, such as a Gene $G_i$ or a Pathway $P_j$. This way we preserve the interpretability of the model, while requiring significantly fewer parameters than an NN with a dense hidden layer (Fig. 1B). We refer to this model as the biologically sparsified NN ($\text{NN}_{\text{biosparse}}$).

Biologically sparsified layers, like any other NN module, can be stacked on top of each other. For example, in Additional file 8: Fig. S3, we built an NN with two $\text{NN}_{\text{biosparse}}$ layers, mimicking gene–gene [48, 53] and gene–pathway interactions, thus allowing even more complex interactions between neurons, and thus a more expressive NN model, at the cost of an increased number of parameters, yet still lower than $\text{NN}_{\text{dense}}$.

### Implementation details

All NNs in this paper have been implemented with PyTorch [54] and are trained with the Adam optimizer, a learning rate of 0.001, a batch size of 128, and a weighted focal binary cross-entropy loss function ($\alpha = 480/(480 + 3,318) = 0.12638$, $\gamma = 2$) to address the class imbalance. Weight decay and dropout layers were used as regularization techniques. Epochs ranged between 20 and 75 depending on the model architecture. The hyperparameter search grid for the different models is supplied in Additional file 14: Table S6. The source code is freely available at https://bitbucket.org/noraver/ibd_gi/src/master/ [55]. All models in this paper were evaluated using 10 repetitions of threefold cross-validation with stratified splits. We report the mean and standard deviation of the area under the ROC curve (AUC) of the test set. The same splits were used to compare models in Table 1 and Figs. 2, 4, and 5. The choice to use three folds was motivated by computational time. To verify that the same patterns occur in other cross-validation approaches, a tenfold cross-validation was performed for the best linear model and neural network, with results shown in Additional file 15: Table S7. The performances on the individual cross-validation runs of Fig. 2 are shown in more detail in Additional file 16: Fig. S6.

### Baseline additive and nonlinear models

We used logistic regression with $L_1$ and $L_2$ penalty as additive baseline models. We implemented it using log loss with the SGDClassifier from the scikit-learn library [56]. As they were applied to the individual variants, these models have 1,733,481 parameters and needed regularization ($\alpha = 1$ for $L_2$ penalty, $\alpha = 0.01$ for $L_1$ penalty). As a nonlinear baseline method, we used the Random Forest (RF) model from scikit-learn. The RF on the individual variants uses the 1,733,480 dimensional feature vector described above, it has 1000 estimators of a maximum depth of 1000. The RF on the summed gene vectors uses as input the sum of the above-described 16-dimensional *mutational load* histograms for each gene and 10,000 estimators of a maximum depth of 3. The RF on the NN learned gene activations uses the gene activation values extracted from the first hidden layer of the fully trained $NN_{logreg}$, $NN_{biosparse}$ or $NN_{dense}$ as input, and 10,000 estimators of a maximum depth of 3.

## Supplementary information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-023-03064-y.

**Additional file 1: Figure S1.** Inclusion of only known IBD genes.

**Additional file 2: Table S1.** Inclusion of gene- and variant-level scores in input representation.

**Additional file 3: Table S2.** Minor allele frequency (MAF) based preselection.

**Additional file 4: Note S1.** Analysis of GWAS variants.

**Additional file 5: Table S3.** Random Forest Classifier performances on different data representations.

**Additional file 6: Figure S2.** Exclusive OR (XOR) as a well known nonlinearly separable problem in machine learning.

**Additional file 7: Table S4.** Model overview with number of parameters and interaction patterns that can be captured.

**Additional file 8: Figure S3.** Biologically sparsified model with gene-gene interaction and gene-pathway layer.

**Additional file 9: Figure S4.** Benchmarking of sparsification using RigI and random connections.

**Additional file 10.** Supplementary Method: RigI.

**Additional file 11: Figure S5.** Subtype analysis: additive versus neural network on different subsamples.

**Additional file 12: Table S5.** Subtype analysis on Crohn's Disease and Ulcerative Colitis.

**Additional file 13: Note S2.** Subtype analysis on Crohn's Disease and Ulcerative Colitis.

**Additional file 14: Table S6.** Model hyperparameters.

**Additional file 15: Table S7.** Tenfold cross-validation performance of best linear model and neural net.

**Additional file 16: Figure S6.** Details of individual cross-validation runs for Fig. 2 on the performance of different random subsets containing 10%, 20%, 40%, 60%, 80%, and 100% of the dataset.

**Additional file 17.** Review history.

#### Acknowledgements

The inflammatory bowel disease exome sequencing study (dbGaP phs001076.v1.p1), which provided the data, was supported by the National Institutes of Health (NIH) grant, NHGRI Large Scale Sequencing U54 HG003067 (S. Gabriel, E. Lander). We acknowledge the clinicians and organizations that contributed to the samples used in this study. Finally, we are grateful to the many families whose participation made this study possible. The authors are grateful to Tiziana Sanavia for the constructive discussion.

#### Review history

The review history is available as Additional file 17.

#### Peer review information

Anahita Bishop and Wenjing She were the primary editors of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

#### Authors' contributions

NV, AP, AA, YM, and DR conceived the experiments. NV and DR developed the methods. NV performed the analyses. NV, DR, and YM wrote the manuscript. All authors reviewed, contributed to, and approved the manuscript.

**Funding**
Antoine Passemiers is funded by a Research Foundation - Flanders (FWO Fonds Wetenschappelijk Onderzoek) doctoral fellowship. Adam Arany is funded by the Advanced machine learning for Innovative Drug Discovery (AIDD) project. Daniele Raimondi is funded by a Research Foundation - Flanders (FWO Fonds Wetenschappelijk Onderzoek) post-doctoral fellowship, grant number 12Y5623N.

**Availability of data and materials**
All the code described in this paper is available at https://bitbucket.org/noraver/ibd_gi/src/master/ [55]. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. No custom scripts and software were used other than those mentioned in the "Material and methods" section. The code is deposited in Zenodo as well with DOI 10.5281/zenodo.8324997 [57]. The data that support the findings of this study are available from dbGaP (Inflammatory Bowel Disease Exome Sequencing Study, dbGaP Study Accession: phs001076.v1.p1) but restrictions apply to the availability of these data. Access to the data can be requested through dbGaP [51].

## Declarations

**Ethics approval and consent to participate**
Not applicable (secondary data use).

**Consent for publication**
Not applicable (secondary data use).

**Competing interests**
The authors declare that they have no competing interests.

Received: 21 March 2023 Accepted: 20 September 2023
Published online: 05 October 2023

**References**

1. Dijk E, Auger H, Jaszczyszyn Y, Thermes C. Ten years of next-generation sequencing technology. Trends Genet TIG. 2014;30(9):418–26.
2. Haendel M, Chute C, Robinson P. Classification, Ontology, and Precision Medicine. New N Engl J Med. 2018;379(15):1452–62.
3. Ashley E. Towards precision medicine. Nat Rev Genet. 2016;17:507–22.
4. Genin E. Missing heritability of complex diseases: case solved? Hum Genet. 2020;139(1):103–13.
5. Zuk O, Hechter E, Sunyaev S, Lander E, Zuk O, Hechter E, Sunyaev SR, Lander ES. The mystery of missing heritability: genetic interactions create phantom heritability. Proc Natl Acad Sci USA. 2012;109:1193–8.
6. Uffelmann E, Huang Q, Munung NS, Vries J, Okada Y, Martin A, et al. Genome-wide association studies. Nat Rev Methods Prim. 2021;1(1):59.
7. Vega F, Bustamante C. Polygenic risk scores: A biased prediction? Genome Med. 2018;10(1):1–3.
8. Yang J, Benyamin B, McEvoy B, Gordon S, Henders A, Nyholt D, et al. Common SNPs Explain a Large Proportion of the Heritability for Human Height. Nature genetics. 2010;42:565–9.
9. Porto A, Peralta J, Blackburn N, Blangero J. Reliability of genomic predictions of complex human phenotypes. BMC Proc. 2018;12(9):157–61).
10. Montesinos-López O, Montesinos A, Pérez-Rodríguez P, Barron Lopez J, Martini J, Fajardo-Flores S, et al. A review of deep learning applications for genomic selection. BMC Genomics. 2021;22, 1-23.
11. Lehner B. Molecular mechanisms of epistasis within and between genes. Trends Genet TIG. 2011;27:323–31.
12. Kogenaru M, de Vos M, Tans S. Revealing evolutionary pathways by fitness landscape reconstruction. Crit Rev Biochem Mol Biol. 2009;44:169–74.
13. Bloom J, Ehrenreich I, Loo W, Lite T, Kruglyak L. Finding the sources of missing heritability in a yeast cross. Nature. 2013;494(7436):234–7.
14. Sittig L, Carbonetto P, Engel K, Krauss K, Barrios-Camacho C, Palmer A. Genetic Background Limits Generalizability of Genotype-Phenotype Relationships. Neuron. 2016;91(6):1253–9.
15. Mackay T. Epistasis and Quantitative Traits: Using Model Organisms to Study Gene-Gene Interactions. Nat Rev Genet. 2013;15(1):22–33.
16. Strange A, Capon F, Spencer C, Knight J, Weale M, Allen M, et al. A genome-wide association study identifies new psoriasis susceptibility loci and an interaction between HLA-C and ERAP1. Nat Genet. 2010;42(11):985–90.
17. Cortes A, Pulit S, Leo P, Pointon J, Robinson P, Weisman M, et al. Major histocompatibility complex associations of ankylosing spondylitis are complex and involve further epistasis with ERAP1. Nat Commun. 2015;6:7146.
18. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596:1–11.
19. Chowdhury R, Bouatta N, Biswas S, Floristean C, Kharkar A, Roy K, et al. Single-sequence protein structure prediction using a language model and deep learning. Nat Biotechnol. 2022;40:1–7.
20. Sapoval N, Aghazadeh A, Nute M, Antunes D, Balaji A, Baraniuk R, et al. Current progress and open challenges for applying deep learning across the biosciences. Nat Commun. 2022;13:1728.
21. Bellot P, Campos G, Pérez-Enciso M. Can Deep Learning Improve Genomic Prediction of Complex Human Traits? Genetics. 2018;210(3):809–19.

22. Pal L, Kundu K, Yin Y, Moult J. CAGI4 Crohn's exome challenge: Marker SNP versus exome variant models for assigning risk of Crohn disease. Hum Mutat. 2017;38(9):1225–34.
23. Laksshman S, Bhat R, Viswanath V, Li X. DeepBipolar: Identifying genomic mutations for bipolar disorder via deep learning. Hum Mutat. 2017;38(9):1217–24.
24. Wang Y, Miller M, Astrakhan Y, Löscher BS, Schreiber S, Franke A, et al. Identifying Crohn's disease signal from variome analysis. Genome Med. 2019;11.
25. van Hilten A, Kushner S, Kayser M, Ikram M, Adams H, Klaver C, et al. GenNet framework: interpretable deep learning for predicting phenotypes from genetic data. Commun Biol. 2021;4:1094.
26. Badre A, Zhang L, Muchero W, Reynolds J, Pan C. Deep neural network improves the estimation of polygenic risk scores for breast cancer. J Hum Genet. 2020;66.
27. Peng J, Li J, Han R, Wang Y, Han L, Peng J, et al. A Deep Learning-based Genome-wide Polygenic Risk Score for Common Diseases Identifies Individuals with Risk. medRxiv preprint 101101/2021111721265352. 2021.
28. Huang S, Ji X, Cho M, Joo J, Moore J. DL-PRS: a novel deep learning approach to polygenic risk scores. Research Square preprint. 2021. https://doi.org/10.21203/rs3rs-423764/v1.
29. Gyawali P, Le Guen Y, Liu X, Tang H, Zou J, He Z. Improving genetic risk prediction across diverse population by disentangling ancestry representations. arxiv preprint arXiv:2205.04673v1. 2022.
30. Grinberg N, Orhobor O, King R. An evaluation of machine-learning for predicting phenotype: studies in yeast, rice, and wheat. Mach Learn. 2020;109:251–77.
31. Ma W, Qiu Z, Song J, Li J, Cheng Q, Zhai J, et al. A deep convolutional neural network approach for predicting phenotypes from genotypes. Planta. 2018;248:1307–18.
32. Gill M, Anderson R, Hu H, Bennamoun M, Petereit J, Valliyodan B, et al. Machine learning models outperform deep learning models, provide interpretation and facilitate feature selection for soybean trait prediction. BMC Plant Biol. 2022;22(1):1–8.
33. Sandhu K, Patil SS, Pumphrey M, Carter A. Multitrait machine-and deep-learning models for genomic selection using spectral information in a wheat breeding program. Plant Genome. 2021;14(3):e20119.
34. Ubbens J, Parkin I, Eynck C, Stavness I, Sharpe AG. Deep neural networks for genomic prediction do not estimate marker effects. Plant Genome. 2021;14(3):e20147.
35. Huang W, Mackay T. The Genetic Architecture of Quantitative Traits Cannot Be Inferred from Variance Component Analysis. PLoS Genet. 2016;12(11):e1006421.
36. Raimondi D, Corso M, Fariselli P, Moreau Y. From genotype to phenotype in Arabidopsis thaliana: in-silico genome interpretation predicts 288 phenotypes from sequencing data. Nucleic Acids Res. 2021;50(3):e16.
37. Raimondi D, Simm J, Arany A, Fariselli P, Cleynen I, Moreau Y. An interpretable low-complexity machine learning framework for robust exome-based in-silico diagnosis of Crohn's disease patients. NAR Genomics Bioinforma. 2020;2(1):Lqaa011.
38. Tanigawa Y, Qian J, Venkataraman G, Justesen J, Li R, Tibshirani R, et al. Significant Sparse Polygenic Risk Scores across 813 traits in UK Biobank. PLoS Genet. 2022;18:e1010105.
39. Lambert S, Gil L, Jupp S, Ritchie S, Xu Y, Buniello A, et al. The Polygenic Score Catalog as an open database for reproducibility and systematic evaluation. Nature Genetics. 2021;53(4):420–5.
40. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 2000;28(1):27–30.
41. Chen T, Chen H, Liu RW. Approximation Capability in by Multilayer Feedforward Networks and Related Problems. IEEE Trans Neural Netw. 1995;6:25–30. https://doi.org/10.1109/72.363453.
42. Nadeau C, Bengio Y. Inference for the Generalization Error. Mach Learn. 2003;52:239–81. https://doi.org/10.1023/A:1024068626366.
43. Kohavi R, Wolpert D. Bias Plus Variance Decomposition for Zero-One Loss Functions. In ICML. 1997;96:275–83.
44. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: with Applications in R. Springer; 2013.
45. Mortlock S, Lord A, Montgomery G, Zakrzewski M, Simms L, Krishnaprasad K, et al. An extremes of phenotype approach confirms significant genetic heterogeneity in patients with ulcerative colitis. J Crohns Colitis. 2023;17(2):277–88.
46. Lange K, Moutsianas L, Lee J, Lamb C, Luo Y, Kennedy N, et al. Genome-wide association study implicates immune activation of multiple integrin genes in inflammatory bowel disease. Nat Genet. 2017;49(2):256–61.
47. Minsky M, Papert S. Perceptrons: An Introduction to Computational Geometry. Cambridge: MIT Press; 1969.
48. Luck K, Kim DK, Lambourne L, Spirohn K, Begg BE, Bian W, et al. A reference map of the human binary protein interactome. Nature. 2020;580(7803):402–8.
49. Evci U, Gale T, Menick J, Rivadeneira PSC, Elsen E. Rigging The Lottery: Making All Tickets Winners. In: International Conference of Machine Learning. 2020;2943–52. PMLR.
50. Breiman L. Random Forests. Mach Learn. 2001;45:5–32.
51. DbGaP. Inflammatory Bowel Disease Exome Sequencing Study. 2017. https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs001076.v1.p1. Accessed Sept 2023.
52. Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res. 2010;38(16):e164.
53. Nassar L, Barber G, Benet-Pagès A, Casper J, Clawson H, Diekhans M, et al. The UCSC Genome Browser database: 2023 update. Nucleic Acids Res. 2023;51(D1):D1188–95.
54. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc.; 2019. p. 8024–8035.
55. Verplaetse N, Passemiers A, Arany A, Moreau Y, Raimondi D. Large sample size and nonlinear sparse models outline epistatic effects in inflammatory bowel disease. Github. 2023. https://bitbucket.org/noraver/ibdgi/src/master/. Accessed Sept 2023.

56. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2012;12:2825–30.
57. Verplaetse N, Passemiers A, Arany A, Moreau Y, Raimondi D. Large sample size and nonlinear sparse models outline epistatic effects in inflammatory bowel disease. Zenodo. 2023. https://doi.org/10.5281/zenodo.8324997.

## Publisher's Note

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

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

**Learn more** biomedcentral.com/submissions

