DeepSeries: Genomic Deep Learning Model for Allelic Series

Mentors: Francesco Paolo Casale, Antonio Nappi
Students: Hanane Moha Ouchane, Chen Xi, Bálint András Lassú

School of Computation, Information and Technology, Technical University of Munich, Munich, Germany;
2024.02.07.

Abstract

Association of rare variant genetic markers with adverse health effects is still actively researched. There are multiple methods in development for identifying allelic series of rare variants and predicting their associated risk with phenotypic consequences. In this study we will introduce a machine learning model that predicts burden scores based on a combination of annotations for rare variant allelic series per gene. DeepSeries was able to fit realistic simulation data of multiple genes and reach high correlation between predicted burden scores and simulated phenotype values.

Introduction

The identification and classification of localised genetic mutations based on their possible health effects is a well researched topic, however the interaction of multiple variants and other genetic mechanisms make calculating exact effects difficult. The uncertainties increase with rare single nucleotide variants as there is less data available to establish clear population-wide risks, and often there is no insight into their underlying functional architecture [1]. With the advance of machine learning techniques, we have a new outlet for predicting the effect sizes of rare variants and finding candidate locations that can be associated with certain diseases or disorders. Here we perform analysis of rare variant allelic series from five genes using DeepSeries, a deep learning model we have developed to predict burden scores in a data-driven manner. Our goal with this study is to test the usefulness of applying a simple machine learning approach based on simple and more complex empirical functions, as well as the merit of using a mix of available annotation sources.

DeepSeries takes advantage of four different variant annotations combined using a modified weighted sum method implemented into a perceptron model. After experimentation and training we were able to recover weights and reach high correlation between simulated phenotype values and predicted burden scores. Testing with a high amount of added noise decreased model performance but comparisons showed that it is more robust than simpler methods. With our results we can confirm some of the newly forming principles of risk prediction of rare variants.

Methods

Data

The allelic series we received are simulated based on realistic allelic series models learned in the UKBB cohort of around 500,000 individuals and are viewed to have increasing deleterious phenotypic effects parallel with the number and severity of single nuclear polymorphisms (SNPs) present. The functional annotation categories are from GeneBass [2].

The initial dataset consists of a series of SNPs per gene, including position, reference and alternative alleles, allele count in the population and a functional description that categorizes the variants into three groups: synonymous, missense, and protein loss of function. Synonymous variants are found in a protein coding region, but they do not cause

alterations in the encoded amino acid sequence. While synonymous SNPs do not directly impair protein production, they can have effects on different aspects of gene expression, such as splicing. Missense SNPs result in an amino acid substitution and are directly affecting in vivo protein activity through possible modification of folding, ligand binding, allosteric regulation among others. Because of the causal connection, missense variants are usually viewed as posing higher risk to the host. Protein loss of function (PLoF) mutations also directly influence the coded proteins functions and are found to be deleterious. The last group is nominally viewed as having the highest risk towards its organism. However, it is still a major challenge to tell the exact exhibits on gene function or disease susceptibility, as it is possible for PLoFs to be benign and synonymous SNPs to cumulatively increase risks of pathogenicity.

Annotation types

Apart from the provided functional category annotations we decided to try and combine our predictions based on multiple different annotation scores reflecting possible deleteriousness. There are currently numerous annotation strategies available publicly testing different approaches, such as the use of other structural and functional features, support vector machines, artificial neural networks and further supervised and unsupervised methods. [3]

One of the first methods to try genome-wide interpretation and prioritization of variants using machine learning techniques is Combined Annotation-Dependent Depletion (CADD) [4]. CADD itself already uses multiple other pre-established genomic annotation sources to assign a score to each variant that corresponds to the likelihood of it being pathogenic or disruptive to gene function.

AlphaMissense is a combination of machine learning on population frequency data, language modelling of amino acid sequences, and the AlphaFold protein structure prediction tool [5]. AlphaMissense focuses on predicting pathogenicity scores of missense variants that conform to existing human clinical databases without explicitly training on such data.

PrimateAI is a deep residual network that uses comparisons of human and other primate species' variant effects to identify pathogenic mutations [6]. It is uniquely trying to describe rare variant effects based on these comparisons by process of elimination.

We have further experimented with AbSplice, a tissue specific aberrant splicing prediction system, however due to the poor overlap with our initial dataset leading to bias we have not included it in the final results of this study [7].

Phenotypic data simulations

The received dataset was converted by us into sparse binary tensor matrices to reflect individual level population data using pytorch' random permutation [8]. This enabled us to simulate person to person phenotypical scores to measure deleteriousness of our variants. As mentioned above, the prediction of these effects is difficult, and we lack a common method for calculating them. Currently researched strategies include burden tests, which assumes that deleteriousness can be additively handled, variance component tests, that assumes the effect sizes will follow a described distribution and other novel techniques often relying on machine learning [9]. In our study we implemented three approaches and compared how our network performs in the permutation of different circumstances. The first method is the simplest and is entirely based on the given functional category annotations,

building on the assumption that PLoF is more deleterious than missense variants and missense more than synonymous SNPs. The second is a weighted sum approach using the additional annotation sources filtered with a threshold value and assumes that rarer variants might have larger effect sizes compared to more common ones in general[1]. The third simulation is similar, but additionally it considers the different annotation scores to be characteristically sensitive to allele frequencies. All three methods also include sigmoid function for scaling, bias and a pre-selected ratio for gaussian noise as a modifier.

a.: $yg = \sum_{i=1}^{3}\left(w_i fA_i\right)$ **b.:** $yg = \left(\sum_{i=1}^{3}\left(w_i \chi_{[t_{i'},\infty)}(A_i)\right) + w_{AF}AF + w_{AF}(1 - AF)\right) * w$

c.: $yg = \sum_{i=1}^{2}\left(w_i \chi_{[t_{i'},\infty)}(A_i)AF\right) + w_3 \chi_{[t_{3'},\infty)}(A_3)(1 - AF) + w_{fA}$

d.: $y = \sigma(yg + \beta) * \sqrt{\frac{vg}{Var(\sigma(yg+\beta))}} + yn * \sqrt{\frac{1-vg}{Var(yn)}}$

Figure 1.: Three proposed functions for phenotypic data simulation used in comparing our findings and the common part of the calculations. Method ***a.:*** *Our simplest implementation uses a weighted (w_i) sum of only functional annotation categories (fA_i) which are converted to one-hot-encoding format; Method* ***b.:*** *The second function includes a weighted (w_i) sum of annotation scores (A_i, CADD, PrimateAI, AlphaMissense) with a threshold applied ($\chi_{[t_{i'},\infty)}, t_i$) as well as allele frequencies (AF) then applying a scaling impact based on the functional annotation categories (w_{fA}); Method* ***c.*** *The last calculation uses allele frequency (AF) in conjunction with weights (w_i) to sum up annotation scores (A_i) with the exception of PrimateAI scores (A_3) where we use the inverse of the allele frequency (1 - AF) based on that PrimateAI states that their score especially considers rare genetic variants* [6]*. This method uses thresholds ($\chi_{[t_{i'},\infty)}, t_i$) for the annotation values as well. We also include here a linear addition of weight scores based on the functional annotation categories (w_{fA}).* ***d.:*** *To achieve the final phenotype scores (y) we apply a scaling sigmoid (σ) function and bias (β) to our genetic component (yg) and modify it such that part of its variance (Var) can be explained by gaussian noise (yn) based on a predetermined ratio (vg).*

DeepSeries network description

Based on the above described three functions we implemented three perceptron models with the help of the Pytorch library [8]. The simulated data was split into train and test parts equally and converted to tensor datasets. Data loaders with batch size 32 were used. For the loss function we chose to use mean square error (MSE) as it has one of the simplest interpretations, conforms well to gaussian distributions which we used as noise in our input data and is efficient in optimizing regression tasks [10]. For the optimizer we used the pytorch implementation of Adam with a learning rate of $1e^{-4}$ based on experimentation [11]. For weight initialization we used the Xavier Glorot method [12]. We trained each model ten times for fifty epochs to be able to see the average results. Repeated trains with varied

noise ratio for the simulated training data, different gene datasets and permutation of simulation and model calculation methods were run for comparisons.

Results

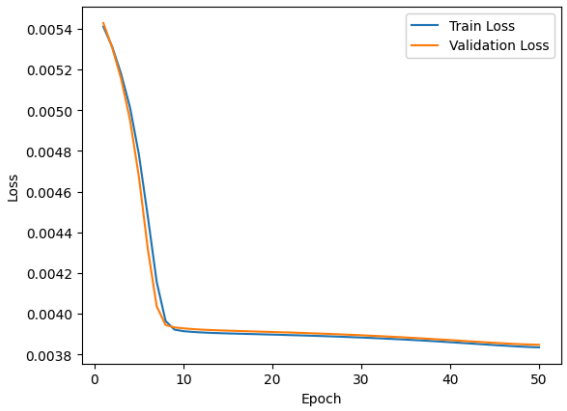
Simple model recovers simulated weights for variant annotations

Training the simple model using only the initial datasets functional annotations with its respective simulation data was done as a baseline. The results reflect that the model was able to fit the simulation data, but with higher noise ratios the learning became more difficult. This can also be seen by the increase in MSE and reducement in spearman correlation values between the predicted and real burden scores. From the curves of the changes in training and validation loss values we can see that there is significant improvement until around ten epochs after which the learning plateaus. Adjusting hyperparameters did not provide a significantly better outcome. This tells us that while a simple model in optimal circumstances can provide useful results, with the noise, imitating real-life uncertainty it becomes unreliable.

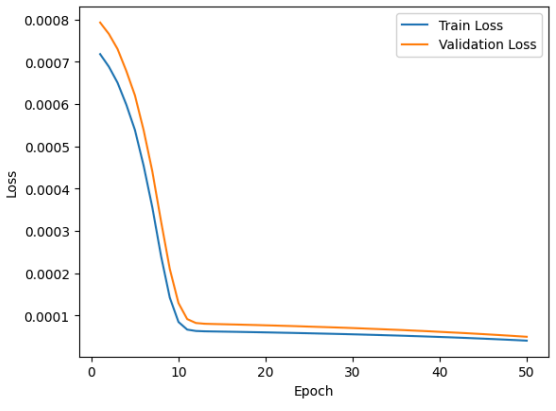
vg	*Prediction mean*	*Prediction median*	*MSE*	*Spearman correlation*
1	*8.60E-03 ± 4.91E-05*	*2.47E-03 ± 0*	*4.04E-05 ± 1.35E-05*	*9.99E-01 ± 4.21E-05*
0.75	*1.15E-02 ± 2.80E-04*	*2.47E-03 ± 0*	*3.83E-03 ± 1.75E-05*	*3.95E-01 ± 3.96E-03*
0.5	*1.45E-02 ± 3.22E-04*	*2.47E-03 ± 0*	*1.20E-02 ± 8.51E-04*	*2.75E-01 ± 1.86E-03*

Table 1.: The above values are from training the simple model on its respectively simulated dataset with different noise modifier (vg, lower value indicates larger noise ratio). The training was done on the "APOB" gene data ten times for each row. The average errors are indicated by ± x. From the mean and median values, we can tell that most of our predictions are close to or zero. The exact median values indicate that while the extremes change the majority of our predictions is this value regardless of noise, this could be an error in the architecture but we can also interpret it as the zero line for our predictions, meaning the lowest deleteriousness score possible. The MSE values show a clear increase with the introduction of noise, and the spearman correlation values inversely follow in a similar manner, telling us that the model is better able to fit with less noise and starts to struggle against noisy data.

a.: *Loss curves; vg = 0.5*



b.: *Loss curves; vg = 1*



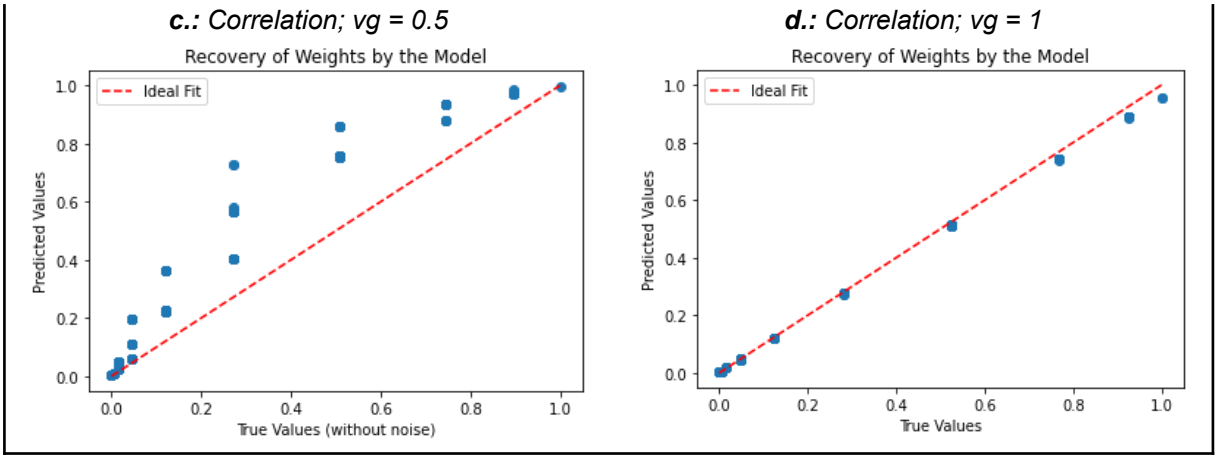


*Figure 2.: Plotted curves of the changes in training- and validation loss values during training, by epochs. Plot **a.** shows that after an initial strong learning period the training plateaus out with no further significant improvement. The graph shows training done on the 'APOB' genes simulated data for fifty epochs with vg = 0.5, meaning a 50% gaussian noise influence on the input data. Plot **b.** shows the same architecture but with a vg = 1, eliminating noise. From the loss curves alone, we can see that the scale of the differences is much lower without noise, indicating a better fit. The overall shapes of the curves show optimal learning rate and hyperparameters. Plots **c.** and **d.** show simplified graphs of correlation between the simulated true values and the predictions of the model. We can observe on both plots the higher point density close to low deleteriousness scores, telling us that the majority of the predictions are close to zero, this correlates with our findings from the low median values. Comparing the two graphs we can see the noise makes it more difficult to fit the model.*

Complex model B learns alternative weights for variant annotations

Complex model B was tested similarly as the simple model and found to fit well with low noise modification to the respective simulated data. However, initial MSE and spearman correlation values show higher errors compared to the simple model test. Increasing input noise lowered correlation between true- and predicted less than in the case of the simple model promising more robustness. Closer look at the exact predictions showed that it has a tendency to inflate burden scores for some rare variants with higher annotation input scores.

vg	*Prediction mean*	*Prediction median*	*MSE*	*Spearman correlation*
1	*1.11E-02 ± 4.11E-04*	*2.47E-03 ± 0*	*6.25E-04± 6.84E-05*	*9.91E-01± 9.14E-03*
0.75	*1.90E-02 ± 1.38E-03*	*2.47E-03 ± 0*	*1.31E-02 ± 2.41E-03*	*7.03E-01 ± 6.04E-04*
0.5	*1.93E-02 ± 3.75E-03*	*2.47E-03 ± 0*	*1.28E-02 ± 5.20E-03*	*5.65E-01 ± 4.79E-03*

Table 2.: results from training and fitting complex model B ten times for each noise level (vg, lower value indicates higher noise ratio) onto respective simulated data of "APOB" genetic input with fifty epochs. Average errors indicated by ± x. We can observe the increase in mean values with noise, indicating higher predicted deleteriousness scores. The MSE and spearman correlation values reflect the difficulty in fitting to noisy data but it has a lower impact on prediction performance compared to the simple model. There is especially little change in mean and MSE values between vg=0.75 and vg=0.5 noise ratios.

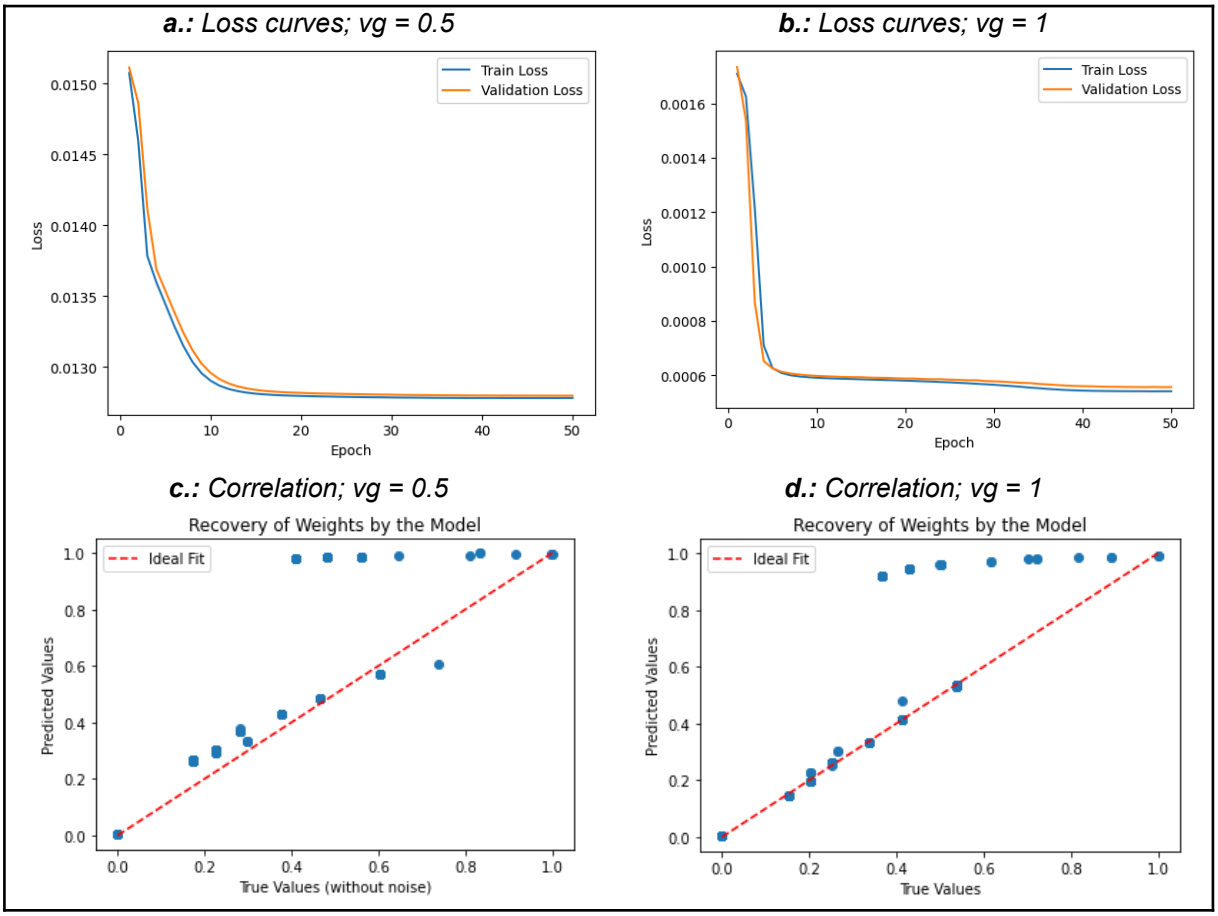


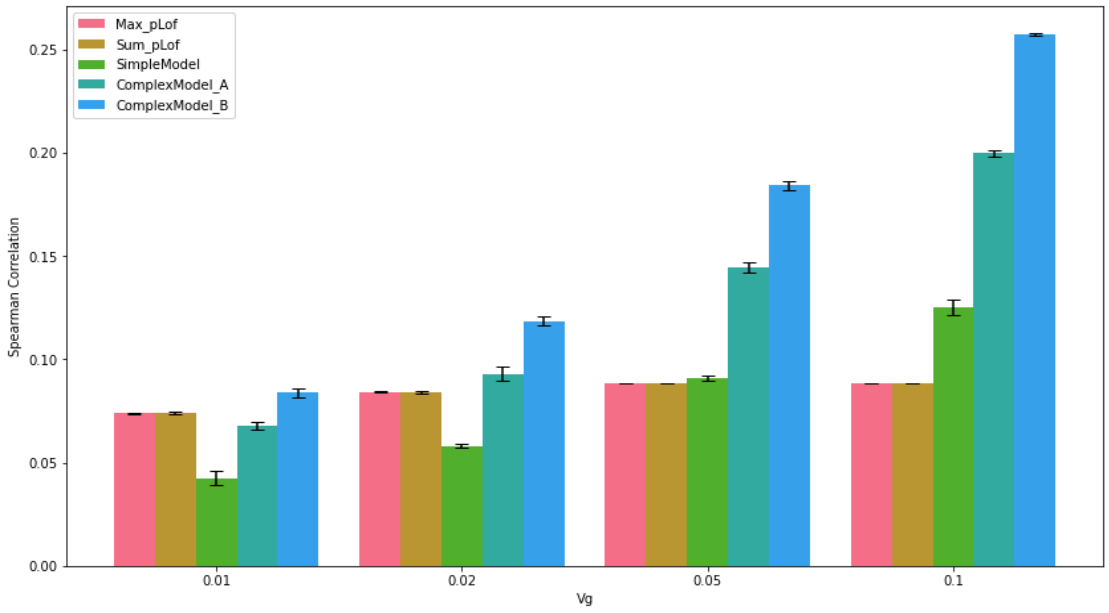
*Figure 3.: Graphs showing training curves and correlations for complex model B with different noise ratios. Plot **a.** Training and validation loss curves over 50 epochs with vg=0.5 using the "APOB" dataset. The large scale and more rounded curve reflects a more difficult learning process compared to the no-noise plot, reaching a plateau after around 15-20 epochs. Plot **b.** learning curves for the same training process with vg=1 (no noise). We can observe a sharp drop of fast initial learning and a smaller y scale for the loss values from which we expect the model to fit easier to the clean data. Plots **c., d.** show the correlation between predicted and true values for the original and noisy data. Although the clean data test shows a better fit of the model, there are clear similarities in the deleterious values. Our model seems to show a pessimistic approach, rating some rare variants with almost maximal risk if they have shown mild deleteriousness.*

As the level of noise decreases, the correlation between predicted burden scores and true phenotype values increases across different genes

After training our different models with high noise modification on every provided genes' simulated dataset we were able to confirm that the complex models show a higher correlation between the predicted and true values. For reference we also tested here the correlation results for only taking the sum and the maximum of the number of PLoF variants. Recent studies already showed that a combination of many different annotation scores show more promising results in predicting associated risk of rare variants [13] and that through the nuanced interactions of genetic mechanisms this prediction requires complex simulations, leading to the increased use of deep learning techniques in the field [14]. The above can also be observed in our study, in how the models with more annotation values and slightly

more complex models applied on their respective simulated datasets lead to better correlations. Comparing performance on different genes show slight differences, especially for the “APOB” gene but overall, we can determine that the results are consistent across genes.

a.: *All model correlations in relation to noise*



b.: *All model correlations across different gene datasets*

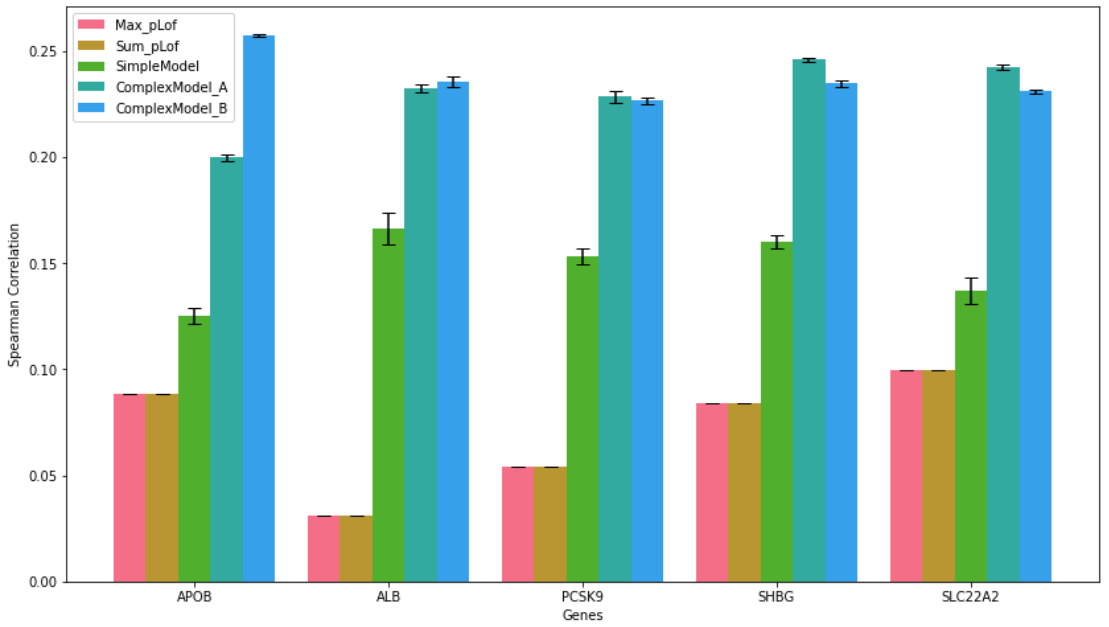
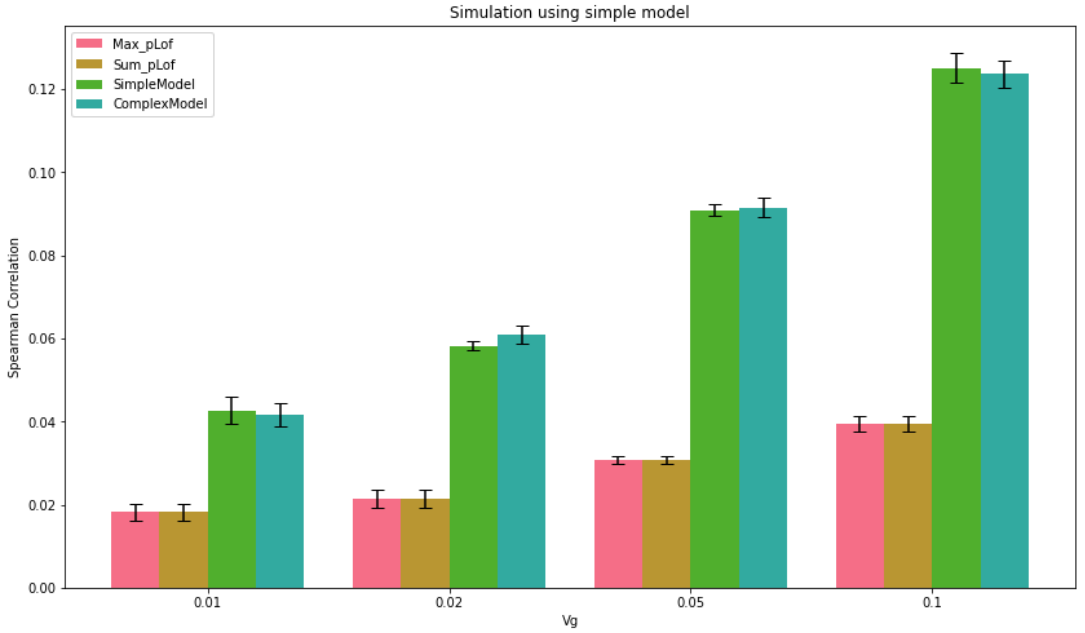


*Figure 4.: Bar plot **a.** shows our different models as well as two reference techniques which only consider the sum and the maximum number of PLoF variants in the datasets predicting burden scores based on their respective simulated data. The models were all trained ten times each with every available genetic dataset for fifty epochs and different noise (vg) modifiers. Average error across tests is indicated on the top of the bars in black. We can observe a clear increase in model performance in favour of the complex models when lowering noise of the input data. Bar plot **b.** shows the same approaches’ results for each gene dataset provided for us with vg=0.1. We can observe that the two complex models perform similarly across genes with the slight exception of “APOB” dataset showing the biggest differences. Based on this we estimate that further hyperparameter tuning specialized for each gene could lead to small improvements.*

Simple and complex models outperform conventional burden scores in different simulation scenarios

After Varying the combination of simulated input data and the model used for burden score prediction shows a different aspect of our conclusion. Our prediction models exhibit similar performance, which was consistently higher than the baseline PLoF burden score calculations on all input datasets but switching to the data simulated by our complex functions showed increase in correlation on all noise levels, indicating that the use of a selection of existing annotation methods can boost our analysis results, even if we implement them with simple approaches.

a.: *Comparison of model performance on simple simulation data*



b.: *Comparison of model performance on complex simulation data*

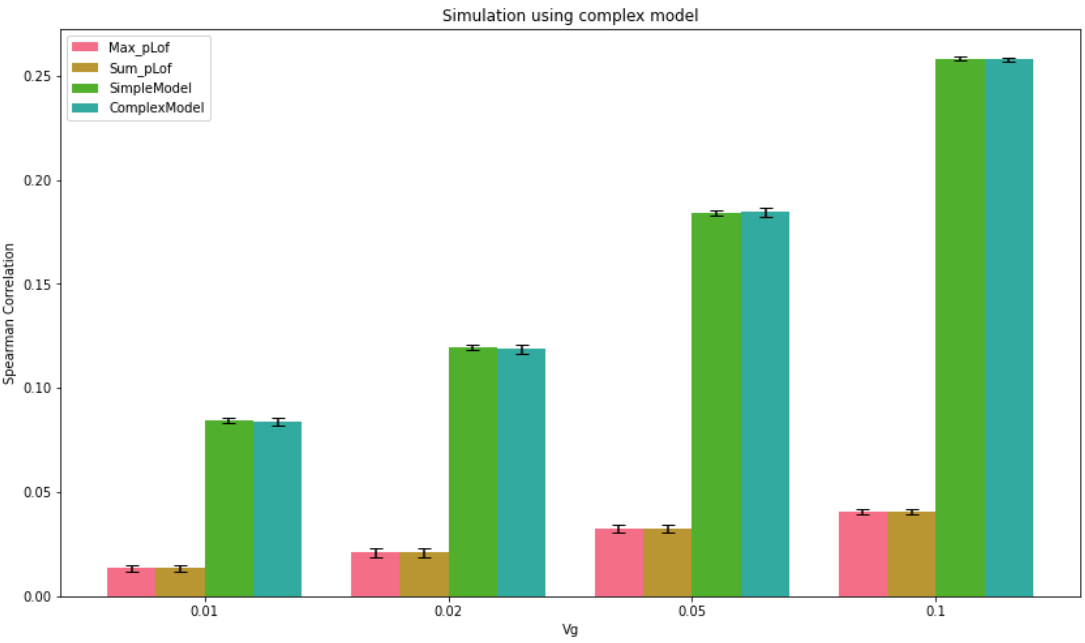


Figure 5.: Shows comparison of model performances trained and tested on the same phenotype simulation values with different noise (vg) modifiers. Plot ***a.*** *shows the simple models simulation data*

*as input while plot **b.** the complex model B's simulation data. We can observe that both models similarly outperform the baseline PLoF methods, and their results produce a similar correlation to the test data. Comparing the plots we can see that using complex method B's simulation data achieved higher correlation values for both models.*

Discussion

In our group study we implemented three machine learning models with increasingly complex empirical functions. DeepSeries network was capable of recovering weights for simulated genetic and phenotypic data and predict burden scores with high correlation in low noise scenarios. With the addition of noise the model performance decreased, however it stayed above the more simple calculation methods only relying on a single annotation and considering deleteriousness for PLoF variants. The predictions were found to be consistent across the five gene datasets that were available for us with minor fluctuation. Comparing the three model performances on their respective phenotype simulations resulted in better correlations for the complex models, while training models on the same phenotype simulation data produced burden scores with high similarity between models, however using the simulation input data based on the complex models modified weighted sum function also provided higher correlation values.

Based on our results we find that the use of more complex methods and the use of multiple annotation sources are a necessity for valuable prediction of deleteriousness of rare variants, as both accuracy and robustness against noise, imitating real life data, can be more achievable by trying to emulate the intricacies of genetic mechanisms. The models we used in our experiments are simpler and therefore more interpretable compared to currently found in literature, namely DeepRVAT. Moreover, our models are capable of learning gene-level burden scores and without needing to be trained on large datasets encompassing different genes. While established methods exist for deleteriousness score calculations, eg.: burden tests, variance component tests, complex neural networks, we believe that a simple architecture consisting of a single layer perceptron and a carefully chosen deleteriousness function can learn to predict burden scores in a data-driven manner.

Acknowledgements

We would like to thank Francesco Paolo Casale and Antonio Nappi for their helpful guidance and patience through this group project as well as all the other members of the Computational Modelling for System Genetics class lecturers (2023-24 WS) for providing us with valuable insight into the field.

References

[1] Z. R. McCaw *et al.*, 'An allelic-series rare-variant association test for candidate-gene discovery', *The American Journal of Human Genetics*, vol. 110, no. 8, pp. 1330–1342, Aug. 2023, doi: 10.1016/j.ajhg.2023.07.001.

[2] K. J. Karczewski *et al.*, 'Systematic single-variant and gene-based association testing of thousands of phenotypes in 394,841 UK Biobank exomes', *Cell Genomics*, vol. 2, no. 9, p. 100168, Sep. 2022, doi: 10.1016/j.xgen.2022.100168.

[3] M. C. Lopes *et al.*, ‘A Combined Functional Annotation Score for Non-Synonymous Variants’, *Hum Hered*, vol. 73, no. 1, pp. 47–51, 2012, doi: 10.1159/000334984.
[4] M. Schubach, T. Maass, L. Nazaretyan, S. Röner, and M. Kircher, ‘CADD v1.7: using protein language models, regulatory CNNs and other nucleotide-level scores to improve genome-wide variant predictions’, *Nucleic Acids Research*, vol. 52, no. D1, pp. D1143–D1154, Jan. 2024, doi: 10.1093/nar/gkad989.
[5] J. Cheng *et al.*, ‘Accurate proteome-wide missense variant effect prediction with AlphaMissense’, *Science*, vol. 381, no. 6664, p. eadg7492, Sep. 2023, doi: 10.1126/science.adg7492.
[6] L. Sundaram *et al.*, ‘Predicting the clinical impact of human mutation with deep neural networks’, *Nat Genet*, vol. 50, no. 8, pp. 1161–1170, Aug. 2018, doi: 10.1038/s41588-018-0167-z.
[7] N. Wagner *et al.*, ‘Aberrant splicing prediction across human tissues’, *Nat Genet*, vol. 55, no. 5, pp. 861–870, May 2023, doi: 10.1038/s41588-023-01373-3.
[8] A. Paszke *et al.*, ‘PyTorch: An Imperative Style, High-Performance Deep Learning Library’, 2019, doi: 10.48550/ARXIV.1912.01703.
[9] R. Monti *et al.*, ‘Identifying interpretable gene-biomarker associations with functionally informed kernel-based tests in 190,000 exomes’, *Nat Commun*, vol. 13, no. 1, p. 5332, Sep. 2022, doi: 10.1038/s41467-022-32864-2.
[10] M. Nießner, C. Yujin, M. Dahnert, G. Gafni, and S. Weitz, ‘Introduction to Deep Learning, (IN2346); Lecture 7: Training Neural Networks part 2, slide 15.’, School of Computation, Information and Technology, Technical University of Munich, Munich, Germany, Winter semester 2023.
[11] D. P. Kingma and J. Ba, ‘Adam: A Method for Stochastic Optimization’, 2014, doi: 10.48550/ARXIV.1412.6980.
[12] X. Glorot and Y. Bengio, ‘Understanding the difficulty of training deep feedforward neural networks’, in *International Conference on Artificial Intelligence and Statistics*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:5575601
[13] X. Li *et al.*, ‘Dynamic incorporation of multiple in silico functional annotations empowers rare variant association analysis of large whole-genome sequencing studies at scale’, *Nat Genet*, vol. 52, no. 9, pp. 969–983, Sep. 2020, doi: 10.1038/s41588-020-0676-4.
[14] B. Clarke *et al.*, ‘Integration of variant annotations using deep set networks boosts rare variant association genetics’, Bioinformatics, preprint, Jul. 2023. doi: 10.1101/2023.07.12.548506.

Supplement

Code availability

All codes to run DeepSeries, including the methods described and figure drawing used in this study are available at: https://github.com/AIH-SGML/allelic-series-b.

Data availability

The genetic and phenotypic data used in this study is simulated based on realistic allelic series models learned in the UKBB cohort of around 500,000 individuals and real annotations from GeneBass [2]. Further population level data simulation is present in the code as described in the methods section. The data are available at: https://github.com/AIH-SGML/allelic-series-b.