These aren't the SNPs you're looking for: Jedi deepMIND tricks for eye color prediction

Adam Klie, James Talwar

Abstract

Phenotypic trait prediction based on genotype information is a powerful tool that has applications that range from the agricultural industry to human health. Traditional models for addressing this task are built on linear correlations assuming additive interactions of a subset of high effect size SNPs. Neural networks are powerful computational systems, capable of learning non-linear interactions between large feature sets. Here we present a feedforward neural network approach to predict eye color based on input genotype array data. We trained a variety of network architectures on a set of 276,563 SNPs from 1000Genomes individuals and compared model performances with IrisPlex eye color predictions on an independent test set from openSNP. Test set accuracy and AUC for our best neural net (55.1% and 0.74) was significantly worse than IrisPlex (81.2% and 0.88). Reducing our feature set to the 1000 SNPs most highly correlated with phenotype increased test accuracy and AUC to 68.6% and 0.80 respectively, but still fell short of IrisPlex performance. These results indicate the importance of feature selection techniques in model training and illustrate the need for large, well-phenotyped datasets for neural network implementations.

1. Introduction

Models that can make predictions of phenotypic traits based on genetic data are becoming powerful tools in genomic research. Already in use in the agricultural industry on crops and livestock such as maize[1] and cattle[2], such models have the potential to make a large impact in understanding and treating human disease. Current approaches to phenotype prediction often rely on a variety of machine learning methods on feature sets derived from genome-wide association studies (GWAS)[1,2,3]. These large scale studies perform correlation analyses between single-nucleotide polymorphisms and phenotypic traits, assigning statistically determined effect sizes to each SNP measuring a SNPs effect on the observed phenotype.

These GWAS studies, however, have several limitations. It has previously been shown that GWAS do not capture all of the heritability observed for a variety of traits[4,5]**.** These studies, and the models built upon them, focus on only variants of large effect size and may miss a large portion of heritability in low effect size or rare SNPs. Furthermore, GWAS assumes an additive effect of SNPs on phenotype and therefore does not capture gene-gene interactions that contribute to phenotype. These limitations can have a significant impact on predictive power and a model that could capture a larger proportion of heritable variation and catalog the interactions between these variants could lead to greatly improved prediction performance and biological understanding.

The winds of change are blowing in bioinformatics. Deep learning has shown significant potential in analyzing large and complex biological data sets such as those provided by next generation sequencing (NGS) technologies. Specifically they have led to significant improvements in variant calling with DeepVariant[6] and identifying functional effects of non-coding variants with DeepSEA[7]. You might ask what makes deep learning so promising for

demystifying complex genotype-phenotype relationships (and if so excellent question reader!)? The answer is two-fold. The first is intrinsic to the mechanisms of neural-network activations, which transfer inputs into non-linear space. The second is unique to reformatting data to capitalize on state-of-the-art architecture approaches. In the context of bioinformatics approaches, you can imagine that data (e.g., reads) are a square peg but modern architectures require circular inputs. The success of the aforementioned tools is that they both transform their inputs into a “circular” form that maximizes the power of a specific architecture known as a convolutional neural network.

Despite these successes, neural networks are not some magic bullet that can address every problem we face in the field (despite what many may say). Neural networks require vast amounts of training data, adequate regularization, proper initialization and optimization, and correct input encoding in order to be successful. For example in our task of SNP genotype to phenotype prediction, the classical 0,1,2 alternate allele encoding would be unlikely to succeed as the network would be unable to assign an effect to homozygous reference alleles (a better approach would be to z-score these values). Other limitations are those inherent to the domain of machine learning as a whole including getting stuck at local minima and identifying optimal hyper-parameter configurations. Despite these limitations, the potential of these networks to capture epistatic interactions not captured by classical methods through non-linear transformations represents an exciting and relatively unexplored avenue of investigation.

Here we built and trained multiple multilayer feedforward neural networks to predict eye color from genotype array data. Eye color was chosen as a phenotype due to the availability of phenotype data in openSNP and the ease of simulating phenotypes for our training set using the IrisPlex model[8]. Utilizing 2504 individuals from the 1000Genomes dataset to train our model, we performed a hyperparameter search to select the best model architecture using 275,563 and 1000 SNP feature sets. Our best models in terms of validation set accuracy were then tested across individuals from openSNP and comparisons were made against gold standard predictions from IrisPlex. In our 275,563 SNP models, we found that we were able to achieve high validation set accuracy, but generalization to the independent openSNP test was poor. Decreasing the feature set to 1000 SNPs correlated to phenotype improved test set accuracy by 13.5%, but remained lower than IrisPlex (81.2%). These results illustrate the difficulty in training models with limited data, simulated phenotypes, and large input dimensions.

2. Methods

2.1 Datasets

2.1.1 1000Genomes

1000Genomes[9] genotypes are publicly available and were downloaded from ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/. A total of 2504 individuals had SNP array data across all 22 autosomes. Phenotypic data is not available for 1000Genomes individuals, so we simulated eye color using the IrisPlex model. The breakdown of predicted phenotype labels was 2230 brown, 268 blue, and 6 other eye colors. The 2504 samples were subject to an 80/20 split into training and validation sets.

2.1.2 OpenSNP

OpenSNP is a repository for user-direct-to-consumer (DTC) genetic testing data[10]. As of April 22nd, 2020, it houses genotype data from 5593 individuals spanning 622 phenotypes from multiple DTC platforms. Of these individuals, 590 were selected for having clearly labeled eye color metadata and parsable genotype information. For simplicity, only brown, blue, and green (other) eye colors were considered. The breakdown of phenotype labels was 312 brown, 194 blue, and 84 other eye colors for this dataset.

2.2 Data preprocessing

Raw uploaded genotype data to openSNP comes from a variety of sources and is not standardized to any one set of SNPs. We therefore found the union of all SNPs across our 590 openSNP samples. Since this SNP set included many spurious SNPs present in only one or a few samples, we filtered based on presence in at least 80% samples. This left us with a set of 276,563 SNPs that could be used as input into our models. All genotypes were assigned numerical values (0-homozygous reference, 1-heterozygous, 2-homozygous alternate) and z-scored before model training and testing.

In an attempt to improve the accuracy of our model, we correlated SNPs to phenotype by assigning numerical labels to each phenotypic class (0-brown, 1-blue, 2-other). We then performed a linear regression on z-scored phenotypes to determine an effect size of each SNP on eye color. Though the classes are not inherently quantitative, this naive approach allowed us to reduce our feature space to SNPs more likely to positively impact classification performance. We used p-values from this analysis to select the top 1000 SNPs most significantly associated with a phenotype. A summary of the datasets and preprocessing can be seen in Figure 1.

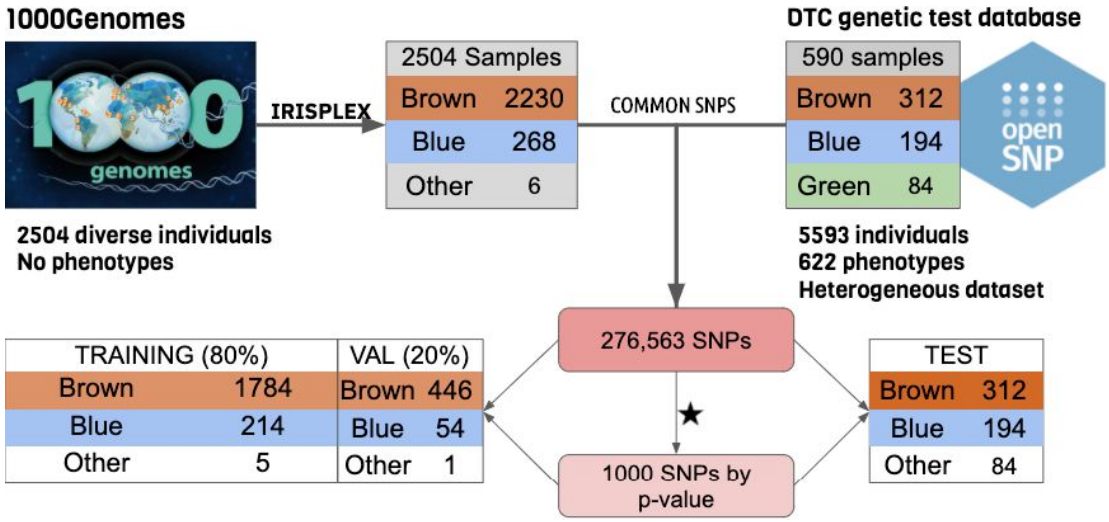


Figure 1: Dataset descriptions and preprocessing pipeline

2.3 Model development and architecture

We started with the hypothesis that neural networks could function as heritability augmenters, capturing elusive genetic variability associated with phenotypes. Given the nature of the dataset

(i.e., SNPs opposed to the entire genomic sequence), we believed a deep feed-forward network, composed of a series of linear hidden layers, could capture epistatic SNP interactions. To develop these models, we used the open source machine learning library Pytorch.

The model inputs, which were the post-processed z-scored SNPs as mentioned in 2.2, were fed through a series of hidden linear layers (described in more detail in 2.4) in the network before outputting a probability associated with eye-color class, similar to IrisPlex. This probability was calculated by taking the output of our final layer of depth 3 (corresponding to the 3 eye-color classes) and applying a softmax function. The rectified linear unit, or ReLU, function was used as our activation between each hidden layer. The loss function, which is the metric by which the model quantified its error, utilized categorical cross-entropy, a standard approach when dealing with multiclass classification. We weighted our loss function by the observed frequency of eye-color class in the training set to address potential problems of class imbalance. The "learning" (i.e., weight adaptations) occurred through adaptive momentum using the Adam optimizer with a seed learning rate of 5×10^{-3}. Regularization was handled through a technique known as dropout at a fraction of 0.5. All models were trained on the datahub GPU (through our access to our CSE 253 container) using mini-batches of 32 to increase the training rate (we could not increase the batch size beyond this due to memory constraints). All models were trained for 100 epochs (or complete passes through the entire training set), but employed early stopping with a patience of 4 epochs to prevent overfitting to the training set. For clarity we have provided an outline of our model architecture in Figure 2 below.

To handle missing SNPs in our test set (i.e., OpenSNP) missing SNPs were assigned a value of 0 in the input layer to shut off contributions of these SNPs in phenotypic prediction. As the original reported SNPs were z-scored this prevented false contributions of unreported SNPs.

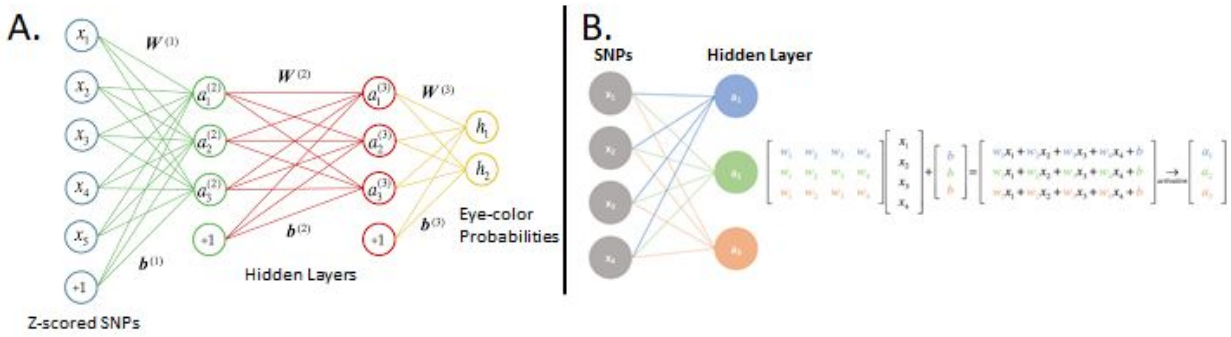


Figure 2: **A.** Sample model architecture (note our final output has 3 outputs for blue, green, and other as opposed to the 2 in the figure; **B.** Inner-working of the operations between layers

2.4 Hyperparameter search

Identifying optimal network parameters is a difficult problem as the combinatorics of searchable network depth, layer width, and weight initializations is infinite. We resorted to some best practices to identify optimal configurable parameters. Specifically we constrained that for any given layer in the network i, the width of layer i+1 $\leq$ layer i $\leq$ layer i-1. This approach helps to prevent overfitting. Due to memory constraints of the datahub GPU we capped our maximum network hidden layer depth to 5 layers and our maximum hidden layer width to be 1024. Network weights were initialized with Xavier initialization as opposed to varying weight

initialization as another hyperparameter. This was done as Xavier initialization has shown to outperform standard uniform initializations by maintaining the variance of activations and back-propagated gradients through all the network layers.[11] Moreover, the aforementioned reference emphasized improved performance for networks of one to five hidden layers, the same number of layers to which we constrained our search space, supporting our justification of initialization methodology. For each subset of SNPs trained on we determined the best model configuration from the maximal validation set accuracy and in the case of a tie, we selected the model with the lowest validation loss. In total we trained and evaluated 33 models for our entire SNP set and 28 models for our 1000 SNP subset (as obtained through methodology described in 2.2).

2.5. Metrics of Evaluation

To evaluate our neural network methodology, we focused on three performance metrics relative to the gold-standard methodology for eye-color prediction, IrisPlex: 1) global accuracy, 2) class-wise accuracy and, 3) area under the curve (AUC) of the receiver operating characteristic (ROC) curve (weighted across classes). To assess prediction robustness in a real-world setting, we decided to use the OpenSNP dataset (described above) as our test set, where variability in reported SNPs is an important consideration. As described in 2.3, missing SNPs for an individual were assigned a value of zero to prevent their contribution in eye-color prediction.

Results

Table 2. Results of hyperparameter search with both all-SNP and 1000-SNP feature sets

Model	# Hidden	Layer Widths	Train Accuracy	Validation Accuracy
Cinco De Mayo 1000	5	512; 256;128; 64; 32	96.4%	96.4%
Trisomy 1000	3	512; 256; 128	98.2%	95.8%
George 1000	1	128	99.5%	95.8%
Paul 1000	1	512	99.8%	95.8%
Glaucoma 1000	5	512; 512; 512; 512; 512	97.8%	95.6%
Let It Be	3	512; 128; 32	98.5%	93.4%
Double Trouble	2	1024; 512	97.9%	93.4%
Dos Equis	2	512; 256	98.7%	93.4%
Winter Is Coming	3	512; 256; 256	98.6%	93.4%
Cinco De Mayo	5	512; 256; 128; 64; 32	97.8%	93.4%

3.1 Hyperparameter search

In total we trained 61 different models in an attempt to identify the optimal network configurations across both SNP sets. Across the 33 models for the entire SNP set, validation set accuracies ranged from 81.03% to 93.41%, confirming the importance of network configuration in prediction performance. As expected due to the high number of SNPs, the models began overfitting to the training set with all but 2 models triggering the early stopping criterion before the full 100 epochs. As shown in Table 2 above, our top 5 models for the full SNP set all reported the same validation accuracy of 93.41%. In accordance with 2.4, we used validation

loss as our tie-breaker with model Let It Be reporting the lowest validation loss across the top 5 models and was therefore used as our best model for test set evaluation.

To mitigate the strong propensity of our models to overfit, we narrowed the dimensionality of our input feature set of SNPs. Applying a quantitative 'label' to phenotypes (brown=0, blue=1, other=2), we performed a linear association test for each SNP to assess a correlation between phenotype and genotype. From this we took the top 1000 SNPs by p-value from these tests. We chose 1000 SNPs as a cutoff as we believed that this was a reasonable feature set size to complement our training set size and would offer a stark enough contrast to our much larger feature set of 276,563 SNPs. Table 2 shows the results of a hyperparameter search, conducted as with the larger feature set. Our Cinco De Mayo 1000 model with 5 hidden layers showed the best validation set accuracy at 96.4%. Loss and accuracy curves (Figure 3A and 3B) also indicate a significant decrease in model overfitting. Together these observations suggest that with current training dataset size, a smaller feature set is desirable.

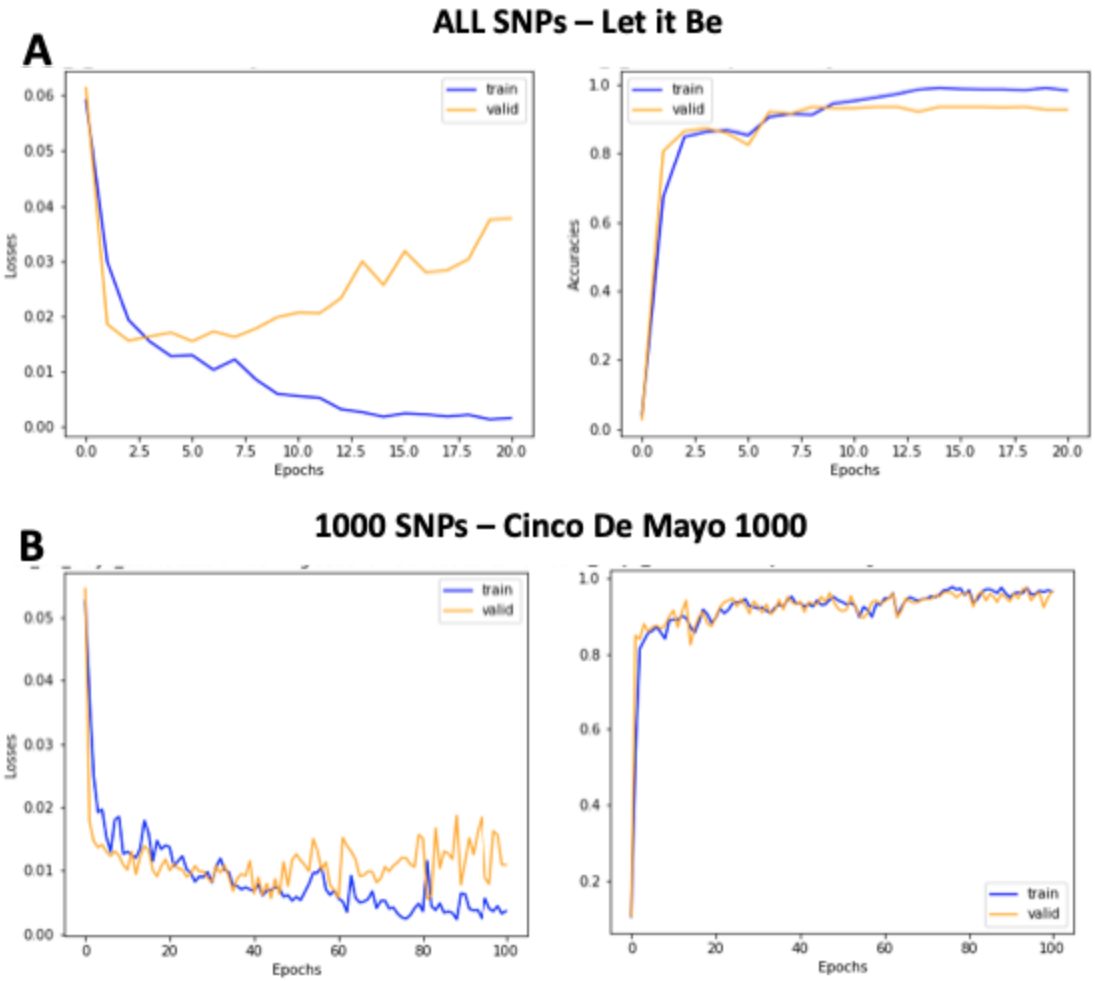
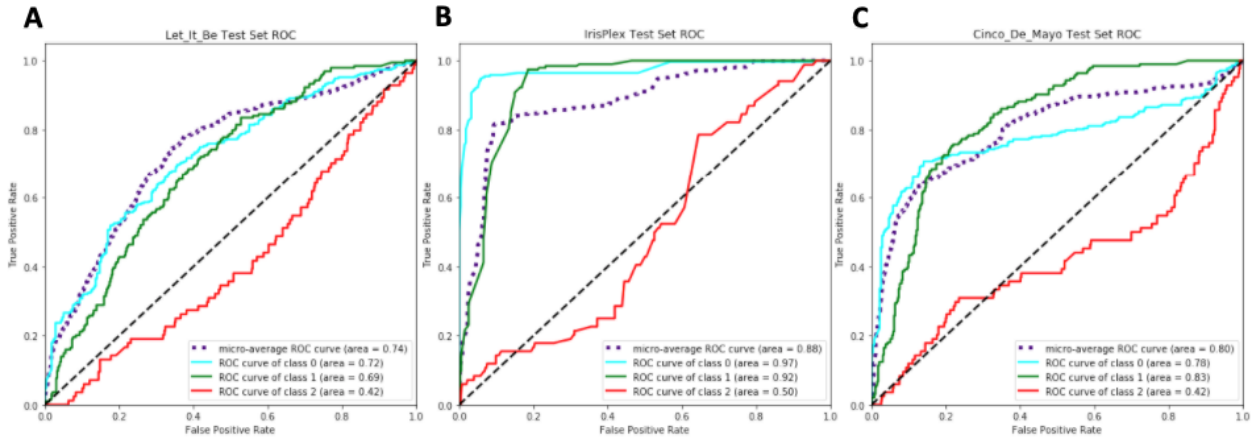


Figure 3: Loss (left) and accuracy (right) curves over epochs. **A.** All SNPs Let it Be model. **B.** 1000 SNPs Cinco De Mayo 1000 model.

3.2 Performance on OpenSNP

3.2.1 All SNPs vs. Irisplex

We applied our best entire SNP model (Let It Be) on the openSNP test set, and noticed a marked reduction in performance relative to our validation set accuracy from 93.4% to 55.1%. The micro-average AUC of the ROC curve, which accounts for class-imbalance between the eye colors, was 0.74 as depicted below (Figure 4A). Relative to IrisPlex, the performance was significantly worse with IrisPlex resulting in an 81.2% test accuracy and 0.88 micro-average AUC of the ROC curve (Figure 4B). Interestingly both IrisPlex and Let It Be struggled significantly to correctly classify individuals with an eye-color of “other” with accuracies of 6% and 0% respectively.



All SNPs Test Accuracy: 55.1%

Brown 54.2%; **Blue** 80.4%; **Other** 0%

IrisPlex Test Accuracy: 81.2%

Brown 90.7%; **Blue** 98.5%; **Other** 6%

1000 SNPs Test Accuracy: 68.6%

Brown 76.6%; **Blue** 85.6%; **Other** 0%

Figure 4. ROC curves, accuracy and by class accuracy on openSNP test set

3.2.2 1000 SNPs vs Irisplex

We also tested our best 1000 SNP model (Cinco De Mayo) on the openSNP test set (Figure 4C). Subsetting to a feature set of 1000 SNPs proved to significantly improve micro-average AUC (by 0.06) and test set accuracy (by 13.5%). We saw the most significant increase in discriminating blue eyed individuals, as the 1000SNP model increased by an AUC of 0.14 on that class. As previously noted with IrisPlex and the Let It Be model, the “other” eye color class proved especially difficult to classify, with performance worse than random chance on that category. Even with the improved performance, Cinco De Mayo still performed significantly worse than IrisPlex in all classes (Figure 4B, C), suggesting that even the reduced feature set could not overcome the overfitting problem.

Discussion

“The first matrix I designed was quite naturally perfect. It was a work of art. Flawless. Sublime. A triumph only equaled by its monumental failure.” - The architect, The Matrix Reloaded

Here we constructed a flexible neural network architecture for multi-class prediction that can be generalized and adapted to any task. We applied this architecture to the prediction of eye color

on the 1000Genomes dataset, and compared model performances on an independent test set from the DTC database openSNP. We tested two feature sets as input to our model (276,563 and 1000 SNPs), and used a hyperparameter search to identify the best architecture for each feature set. We found that the 1000 SNP feature set models performed better across the board, with no clear correlation between architecture and performance. Test set classification metrics from the top models trained on each feature set lagged significantly behind IrisPlex in both AUC and test set accuracy, and loss curves consistent with overfitting were observed in all model training.

Despite our attempts at regularization through dropout and early-stopping, we were unable to mitigate overfitting in all our models. We can think of three potential reasons for this. The first is the imbalance between the number of SNPs and the number of training samples. Specifically, the number of SNPs in our entire SNP set exceeded the number of training samples by a factor greater than 1000. The second is the inherent noise in not having ground-truth labels for the 1000 genomes data set. Eye color “ground-truth” labels were obtained via IrisPlex, which is not infallible and likely altered weight contributions between SNPs. The third reason is that this overfitting trend is an artifact of the “other” eye-color class. Figure 3 shows relatively stable validation accuracies despite increasing loss. This indicates that the network was increasing its output prediction confidence in the wrong direction for hard to classify samples. The stability in accuracy though means that these hard-to-classify individuals made up a small fraction of the overall population suggesting these individuals had a self-reported eye color of neither blue nor brown.

The poor performance on the test set is likely attributable to our methodology for handling the inconsistency in reported SNPs across openSNP individuals. As described in 2.3, we assigned missing SNPs a value of 0 to shut off contributions of these SNPs in phenotypic prediction. However, the number of these missing SNPs was significant with 11.2% missing on average for the full 276,563 SNP set. As the model was not trained on corrupted data, this high fraction of missing SNPs, combined with potential overfitting compounded to result in poor overall generalization on the test set.

Due to the poor performance of the all-SNP model, we decided to test if a reduced feature set would improve AUC and test set accuracy. We noted that the stark overfitting observed in the all-SNP models was significantly reduced when using a 1000 SNP feature set. Moreover, validation set accuracy increased by an average of about 3% across all models and test set performance improved dramatically. However, this linear correlation methodology for selecting the top 1000 SNPs based on p-value, likely moves the network closer to a simple additive model and negates the model’s ability to capture more heritable variation and broad epistatic interactions. The overfitting still observed in this reduced feature set is likely caused by the aforementioned issues with our datasets, and further illustrates the necessity of much larger training sets for deep learning on complex inputs.

The original IrisPlex model[8] was designed to predict eye color from a set of 6 SNPs shown to be strongly associated with this phenotype. The model was based on multinomial logistic regression formulae presented by Liu et al in 2009[3], and assigns probabilities of eye color to the categories blue, brown, and other. As noted by Liu, et al., this model’s performance was lowest in the “other” category and this was postulated to be due to poor phenotypic characterization and missing heritability. Coming from DTC genetic testing uploads, the eye color metadata

associated with openSNP samples was extremely heterogeneous, and it is likely these phenotypes are even more poorly characterized. Due to this, we chose a “green” eye color from openSNP to represent the “other” category from the IrisPlex predictions, and it is likely that this contributed to all three models performing worse than or similar to random classification of that category.

So where do we go from here? A natural place to begin would be to develop a robust architecture that can handle missing SNPs and corrupted data rather than shutting off missing SNP contributions as we mentioned previously. In particular we believe that stacking a denoising autoencoder on top of our feed-forward network would strongly correct for this problem. Succinctly, denoising autoencoders attempt to recreate the true data set from corrupted data and essentially could serve as a SNP imputation method, preventing these lost SNP contributions. We could also attempt to impute SNPs in a classical way with one of the numerous methods currently available. Other interesting ideas we had would be to use techniques such as attention or reinforcement learning to identify significant SNPs associated with eye-color and overlap these found SNPs with those discovered by IrisPlex.

Author contributions

James (written by Adam) The training master implemented the network and the training code. He also implemented the code for hyper-parameter searching and test set evaluation. In the presentation, James took care of building the slides for and presenting all things model and model results. He co-wrote the introduction, methods and discussion of the report, focusing on model architecture and implementation related background, results and future directions.

Adam (written by James) Have you ever wanted to fight a panda? The data wrangler fought tooth and nail against the 29 foot bear that was the OpenSNP dataset. He downloaded, extracted, and processed all the SNPs used for this project, integrated them into a Pytorch dataloader class, and performed the linear association to identify our 1000 SNP subset. He channeled his inner Rugrat, speeding up the processing and data loading pipelines converting between slow csv files to efficient pickles. For the presentation he organized the slides relating to project motivation, the dataset, and processing. For the report he wrote the abstract and co-wrote the remaining sections focusing on the dataset, processing, and the 1000 SNP subset. By the end of this project he had only one thing to say: “Spumoni!”

Winner (aka who has the best paragraph): ?

Reflection

Simply put we launched ourselves off a cliff without knowing how deep the water was below, but it was an absolutely glorious flight. We expected to struggle to achieve high levels of classification performance with such a large feature set and relatively few training examples, but we thoroughly enjoyed trying. We wish we had more time to fine tune our feature selection, and potentially clean up our test set a little more, but feel we made headway in understanding how the problems presented here could be addressed. We faced major challenges in finding and preprocessing our dataset, and openSNP was a major, major headache. We felt that guidance for this project was adequate, but realized we were in somewhat uncharted territory at the same time. We started working on this project around week 4 and probably spent around 75 total

hours on it throughout the quarter. In terms of team dynamics, James knows I can't stand him, but I guess I'll have to put up with him for the next 3+ years in the Carter lab. To that James retorts, "I do not take your sarcasm with affront, but as vindication of your ability to blend in."

References

1. Buckler, E. S., Holland, J. B., Bradbury, P. J., Acharya, C. B., Brown, P. J., et al. (2009). The genetic architecture of maize flowering time. Science, 325(5941), 714–718.
2. Schaeffer, L. (2006). Strategy for applying genome-wide selection in dairy cattle. J. Anim. Breed. Genet. 123 (4), 218–223. doi: 10.1111/j.1439-0388.2006.00595.x
3. Liu, Fan, et al. "Eye Color and the Prediction of Complex Phenotypes from Genotypes." Current Biology, vol. 19, no. 5, Mar. 2009, pp. R192–93. DOI.org (Crossref), doi:10.1016/j.cub.2009.01.027.
4. Maher, Brendan. "Personal Genomes: The Case of the Missing Heritability." Nature, vol. 456, no. 7218, Nov. 2008, pp. 18–21. DOI.org (Crossref), doi:10.1038/456018a.
5. Visscher, Peter M et al. "Five years of GWAS discovery." American journal of human genetics vol. 90,1 (2012): 7-24. doi:10.1016/j.ajhg.2011.11.029
6. Poplin, Ryan, et al. "A Universal SNP and Small-Indel Variant Caller Using Deep Neural Networks." Nature Biotechnology, vol. 36, no. 10, Nov. 2018, pp. 983–87. DOI.org (Crossref), doi:10.1038/nbt.4235.
7. Zhou, Jian, and Olga G. Troyanskaya. "Predicting Effects of Noncoding Variants with Deep Learning–Based Sequence Model." Nature Methods, vol. 12, no. 10, Oct. 2015, pp. 931–34. DOI.org (Crossref), doi:10.1038/nmeth.3547.
8. Walsh, Susan et al. "IrisPlex: a sensitive DNA tool for accurate prediction of blue and brown eye colour in the absence of ancestry information." Forensic science international. Genetics vol. 5,3 (2011): 170-80. doi:10.1016/j.fsigen.2010.02.004
9. The 1000 Genomes Project Consortium. "A Global Reference for Human Genetic Variation." Nature, vol. 526, no. 7571, Oct. 2015, pp. 68–74. DOI.org (Crossref), doi:10.1038/nature15393.
10. Greshake, Bastian, et al. "OpenSNP–A Crowdsourced Web Resource for Personal Genomics." PLoS ONE, edited by Tricia A. Thornton-Wells, vol. 9, no. 3, Mar. 2014, p. e89204. DOI.org (Crossref), doi:10.1371/journal.pone.0089204.
11. Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (2010).