Trait-Predicting Algorithms: Proposed Algorithm for Predicting Phenotypes from Genetic Data

CS4991 Capstone Report, 2023

Paul Vann
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
pjv4yj@virginia.edu

ABSTRACT
Trait-predicting algorithms in genetic science and computer science are capable of predicting physical traits from genetic information, allowing for much more advanced criminal profiling where DNA samples are present. I propose a neural network-based solution to this problem that takes genetic information in as input, and outputs predicted facial features. Trained with pairs of genetic information and corresponding facial features, the proposed model would be structured so that once it has been trained on a significant processor or GPU, it would be able to identify where in the genetic code certain phenotypes are defined. The anticipated outcome of such a model with the correct training data is the ability to accurately predict five key facial features: eye color, nose structure, jaw structure, hair color, and skin color. The key factors to increase the accuracy of the algorithm's predictions are the data the model is trained with, and the processor the model is trained on. In the future, actual experimentation with genetic data would be helpful to fine tune the model to be more accurate overall.

1. INTRODUCTION

Over the last decade genetic technology and research has expanded extensively, ranging from nationality tracing to individual specific medicines and medical care. One innovation that has come to the forefront more recently is trait prediction, which is the ability to predict phenotypes (physical features) from genetic information. In recent years, this has been done manually allowing for simple phenotypes such as eye color and hair color to be predicted. However, with the rise of machine learning and techniques such as reinforcement learning and neural networks, researchers are expanding further into more advanced predictions.

Neural networks have been used extensively in the past few years for extremely complex machine learning tasks including advanced classification and regression tasks. For this reason, Neural networks provide a very strong framework for a trait predicting algorithm. In a neural network-based trait-prediction algorithm, the input is a set of genetic information and the output is a set of predicted facial features including eye color, nose structure, jaw structure, hair color, and skin color. In order to train such a model, pairings of genetic information with associated phenotype data are necessary.

2. RELATED WORKS

The main idea for this project originated from reading Pośpiech, et al. (2022) which highlighted the current state of the art in the field of trait prediction. They noted that researchers have developed predictive tools

for some simpler genetic markers and physical features, but have not yet been able to move to those that are more advanced or harder to predict. They also discussed the need for a significant amount of genetic data paired with an advanced machine learning based approach in order to detect some of these more advanced genetic markers within genetic code. While they made very good points about what was necessary for a trait predicting algorithm, more research is necessary to determine what an ideal model structure would look like.

Based on course content from Kaggle course, Introduction to Deep Learning (Holbrook, 2021), I decided a strong approach to this problem would be using a neural network-based approach, a novel and advanced form of model in the machine learning field. This introductory course covered basics on neural networks, how to implement a neural network classification algorithm in Python, and methods for optimizing such an algorithm.

Finally, Yoo, et al. (2022) report on a machine learning algorithm that was directly applied to trait prediction to assist in individual skin care treatment. While their project is a different use-case than my proposal they presented an experimental design and structure for carrying out trait prediction via genetic information. Their work is similar to my proposal in that I propose a similar approach with machine learning to predict facial features from genetic data. However, I present a neural network-based proposal for more general and accurate facial feature prediction.

3. PROJECT DESIGN

The following subsections provide a breakdown of the proposed algorithm and training process.

3.1 Training Data

Aside from the design of the neural network model, the most important factor in the accuracy of this trait predicting algorithm is the training data that is used. A neural network merely acts as the base structure of the algorithm, while the training data actually fills that structure in, giving it meaning and providing context for it to classify traits correctly. There are two important factors when it comes to training data: the types of data, and the amount of data.

The following two figures highlight the data types of the training data and how they should be measured:

Data	Data Type
Genetic Code	String
Eye Color	*Int
Hair Color	*Int
Skin Color	*Int
Jaw Width	Float
Nose Height	Float
Nose Width	Float

Figure 1: Training Data Types
(* corresponds to color ID)

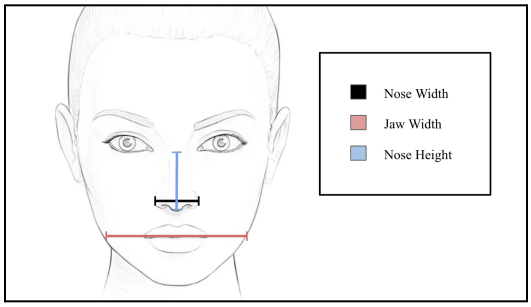

Figure 2: Training Data Proportions on Human Face

In regards to hair, eye, and skin color, each unique color that is included in the training dataset should be assigned a unique integer ID and be tagged accordingly in the dataset.
In order for the neural network to accurately identify components of genetic data that correspond to these physical traits, it is necessary to have a minimum number of samples for each phenotype permutation relative to other permutations of that phenotype. For example, if you have 20 training samples with blue eye color, it is crucial to have at least 40 other samples with other eye colors. I propose such a model so that the algorithm has enough unique genetic data for each eye color in this example to be able to classify it outside of training data. Therefore, I propose the following minimum requirements for training data: each phenotype should have at least three unique permutations of that phenotype; and each of those permutations should have at least 20 samples.

3.2 System Architecture

In order to quickly and accurately train this neural network, it is essential that it runs on a high-powered computer or GPU. This is important for both storage of the training data, and the speed and ease of actually training the neural network. For this algorithm, I propose using the Google Colab Pro+ NVIDIA P100 or T4 GPU.

Google Colab is a Google cloud service that provides a Jupyter notebook-like interface for machine learning and data science purposes. These GPUs can be accessed via the cloud using a Google account and are relatively cheap and easy to use. More importantly, these two GPUs can easily be used to train this model as they are very powerful when working with the amount of data that this model will be trained on.

3.3 Designing and Training the Model

The first step in constructing the model is pre-processing the training data and splitting it into test and training data. When designing the structure of the base training data, I made sure to make it relatively easy to do this. The first step is converting the genetic data, which is currently a string, into a numerical representation. This can be done using one-hot encoding to represent each nucleotide as a binary value. For example, this could be done where T is encoded as (1,0,0,0), A as (0,0,0,1), C as (0,1,0,0), and G as (0,0,1,0).

Hair, eye, and skin color are already processed into corresponding integer IDs so do not require preprocessing. While nose width, jaw width, and nose height could be kept the same since they are in numerical format, it is good practice for them to be scaled to a range such as from 0 to 10 or 0 to 1 to create a standard range across the dataset. The training data can then be split into training and testing data. Because genetic data is difficult to come by due to privacy regulation, there may be limited data to work with so I propose a 80% training to 20% testing ratio.

With the data pre-processed and split into training and testing data, the neural network model can be built. The first component of this neural network should be an input layer which includes the genetic information and phenotype data highlighted in the training data. This input layer is what brings in the data for deeper layers to work with. The next component of the neural network is a series of dense hidden layers with ReLU activation functions. The number of hidden layers should be determined by the complexity and amount of training data. The last component of the neural network should be an output layer with a softmax activation function. This layer should have a neuron for each phenotype being predicted. The model structure highlighted above can be built using

the PyTorch Python library, with many of the features discussed built-in to the library (Pytorch Contributors, 2023).

4. ANTICIPATED RESULTS

While this model design does not provide tangible code or training data to build a trait predicting algorithm, the implications of the design highlighted here are extensive and could have many benefits going forward. For one, I anticipate that the design of the training data and how it is pre-processed could be used extensively in future projects where researchers may have more access to genetic data. The way the training data is structured and how it is processed really streamlines the overall training process and makes it much easier for a neural network to interpret and understand the data. Furthermore, the training data that is specifically looked at such as nose width and jaw width is unique from other research, so I anticipate this will become a point of future research. Additionally, I anticipate that with this neural network design and an adequate number of epochs this model could be very accurate when predicting phenotypes from genetic data.

5. CONCLUSION

There are many significant benefits and use cases for a highly accurate trait predicting algorithm such as the one highlighted above. For one, a trait-predicting algorithm would assist forensic scientists and law enforcement agencies in profiling and identifying suspects in criminal investigations much more quickly, even if the suspect's DNA is not in their database. Furthermore, trait-predicting algorithms can span past human trait prediction and be used on historical samples to determine what creatures who roamed before us may have looked like. Overall, the benefits of such an algorithm are highly significant and with the correct training, data, and configurations, such a model could have a significant impact in many fields.

6. FUTURE WORK

Going forward, there is much work to be done in both the field of genetics and trait-predicting algorithms. While I did not have the resources or genetic data to be able to build a tangible model following the design highlighted above, the next step is for a researcher to actually put this neural network into code and test it against real genetic data.

Furthermore, it would be interesting to look at other phenotypes and features aside from facial features. This could include adult height or medical inclinations, which extend into use cases other than forensic science. As a whole, though, the next big milestone in regards to this project would be building a tangible model and testing it against real-world data.

REFERENCES

Pośpiech E, Teisseyre P, Mielniczuk J, Branicki W. (2022). Predicting physical appearance from DNA data—Towards genomic solutions. Genes. 13(1):121. Retrieved on March,7,2023? https://doi.org/10.3390/genes13010121

Pytorch Contributors. (2023). Pytorch. PyTorch. Retrieved April 3, 2023, from https://pytorch.org/docs/stable/index.html

Holbrook, R. (2021). Learn Introduction to Deep Learning. Kaggle Learn. Retrieved April, 3, 2023, from https://www.kaggle.com/learn/intro-to-deep-learning

Yoo, H. Y., Lee, K. C., Woo, J. E., Park, S. H., Lee, S., Joo, J., Bae, J. S., Kwon, H. J., & Park, B. J. (2022). A genome-wide association study and machine-learning algorithm analysis on the prediction of facial phenotypes by genotypes in Korean women. Clinical, cosmetic and investigational dermatology, 15, 433–445. Retrieved on

March,7,2023
https://doi.org/10.2147/CCID.S339547