


RESEARCH

Open Access



# New neural network classification method for individuals ancestry prediction from SNPs data

H. Soumare<sup>1,2\*</sup> , S. Rezgui<sup>3</sup>, N. Gmati<sup>4</sup> and A. Benkahla<sup>2</sup>

\*Correspondence:

[soumare.harouna@enit.utm.tn](mailto:soumare.harouna@enit.utm.tn)

<sup>1</sup>The Laboratory of Mathematical Modelling and Numeric in Engineering Sciences, National Engineering School of Tunis, Rue Béchir Salem Belkhiria Campus universitaire, B.P. 37, 1002 Tunis Belvédère, University of Tunis El Manar, Tunis, Tunisia

<sup>2</sup>Laboratory of Bioinformatics, bioMathematics, and bioStatistics, 13 place Pasteur, B.P. 74 1002 Tunis, Belvédère, Institut Pasteur de Tunis, University of Tunis El Manar, Tunis, Tunisia

Full list of author information is available at the end of the article

## Abstract

Artificial Neural Network (ANN) algorithms have been widely used to analyse genomic data. Single Nucleotide Polymorphisms (SNPs) represent the genetic variations, the most common in the human genome, it has been shown that they are involved in many genetic diseases, and can be used to predict their development. Developing ANN to handle this type of data can be considered as a great success in the medical world. However, the high dimensionality of genomic data and the availability of a limited number of samples can make the learning task very complicated. In this work, we propose a New Neural Network classification method based on input perturbation. The idea is first to use SVD to reduce the dimensionality of the input data and to train a classification network, which prediction errors are then reduced by perturbing the SVD projection matrix. The proposed method has been evaluated on data from individuals with different ancestral origins, the experimental results have shown the effectiveness of the proposed method. Achieving up to **96.23%** of classification accuracy, this approach surpasses previous Deep learning approaches evaluated on the same dataset.

**Keywords:** Artificial neural network, Dimensionality reduction, Input perturbation, Single nucleotide polymorphism, Singular value decomposition

## Introduction

The human genome contains three billion of base pairs, with only 0.1% difference between individuals [1]. The most common type of genetic variations between individuals is called Single Nucleotide Polymorphism (SNP) [2]. An SNP is a change from one base pair to another, which occurs about once every 1000 bases. Most of these SNPs have no impact on human health. However, many studies have shown that some of these genetic variations have important biological effects and are involved in many human diseases [3, 4]. SNPs are commonly used to detect genes associated with the development of a disease within families [5]. In addition, SNPs can also help to predict a person's response to drugs or their susceptibility to develop one or more particular diseases. In genetics,



© The Author(s). 2021 **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.

Genome-Wide Association Studies (*GWAS*) are observational studies that use high-throughput genotyping technologies to identify a set of genetic variants that are associated to a given trait or disease [6], by comparing variants in a group of cases with variants in a group of controls. However, this approach is only optimal for populations from the same ancestry group, as it is challenging to dissociate the variations associated with a disease from those that characterize the genetic of human populations. In this context, numerous machine learning algorithms have been used to classify individuals according to genetic differences that affect their population. Support Vector Machines (*SVM*) methods have been applied to infer recent genetic ancestry of a subgroup of communities in the USA [7] or coarse ethnicity [8]. However, *SVM* methods are very sensitive to the choice of kernel and its parameters [9]. Deep learning algorithms, such as Neural Networks have been widely used to analyse genomic data as well as gene expression data to classify certain diseases [10–20]. But, the high dimensionality of genomic data (when the number of input features is several times higher than the number of training examples) makes the learning task very difficult. Indeed, when data is composed of a large number of input features  $m$  for a small number of samples  $n$  ( $n \ll m$ ), the problem of overfitting becomes inevitable. In general, overfitting in machine learning occurs when a model fits well with the training data, but not fit the unseen data. The model learns details and noise in the training data, which negatively impact the performance of the model on new data. One way to avoid the problem of overfitting is to reduce the complexity of the problem by removing features that do not contribute or decrease the accuracy of the model [21]. Different techniques are used to deal with the problem of overfitting. The most well-known ones are  $L^1$  and  $L^2$  regularizations [22]. The idea of these techniques is to penalize the higher weights in the model by adding extra terms in the loss function. Another commonly used regularization technique, called "Dropout", introduced by Hinton et al. [23] consists of dropping neurons at random (in hidden layers) in each learning round. However, with such difference between the number of features versus the number of samples, it increases the problem of overfitting. To overcome this problem, dimensionality reduction techniques need to be combined with unsupervised learning methods or other data preprocessing techniques.

There are many ways to transform a high-dimensional data to low-dimensional data, Singular Value Decomposition (*SVD*), Principal Component Analysis (*PCA*) and Autoencoder(*AE*) are the most common dimensional reduction techniques. *SVD* and *PCA* are the most popular linear dimensionality reduction techniques. Both attempt to find  $k$  orthogonal dimensions in an  $n$ -dimensional space, so that  $k < n$ . They are related to each other, but *PCA* uses the covariance matrix of the input data, while *SVD* is performed on the input matrix itself. The Autoencoder is a Neural Network that tries to reconstruct the input data from their compressed form. Indeed, the Autoencoder is used as a method of non-linear dimensionality reduction, it works by mapping an  $n$ -dimensional input data into a  $k$ -dimensional data (with  $k < n$ ).

Recently, *ANNs* have been used in many works to analyse sequencing data and predict complex diseases using *SNPs* data [11, 24–29]. To analyse *SNPs* from sequences [16, 26, 30], many approaches have been proposed to deal with high dimensionality by combining dimensionality reduction techniques, such as unsupervised methods followed by supervised Neural Networks for classification [11, 13, 31–33]. For instance, Zhou et. al. [11] used a three-step Neural Network to characterise the determinants of

Alzheimer's disease. Liu et al. [34] combined Deep Neural Network with an incremental way to select *SNPs* and multiple Dropouts regularization techniques. Kilicarslan et al. [32] used a hybrid model consisting of Relief and stacked Autoencoder as dimensionality reduction technique followed by Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs) for diagnosis and classification of cancer samples. Khan et al. [35] used *PCA* and Neural Network to identify relevant genes and classify cancer samples. Fakoor et al. [14] combined *PCA* with Sparse Autoencoder to improve cancer diagnosis and classification. Romero et al. [33] proposed to reduce the hyperparameters of the classification network by the use of auxiliary networks. Pirmoradi et al. citepirmoradi2020self used Deep Auto-Encoder approach to classify complex diseases from *SNPs* data. Based on our literature review, Romero et al. are the first to use Deep learning algorithms on *SNP* data for genetic ancestry prediction task. They constructed a classification network with an optional reconstruction path and proposed two auxiliary Neural Networks to predict the parameters of the first layer of the classification network and its reconstruction path respectively. They proposed several types of embedding techniques to reduce the number of free parameters in the auxiliary networks, such as *Random projection(RP)*, *Per class histogram*, *SNPtoVec*, *Embedding learnt end-to-end from raw data*.

In this work, we propose a New Classification Neural Network based on the perturbation of the input matrix. To address the problem of dimensionality, the training model is constructed in three steps followed by a test phase: (1) use *SVD* to reduce the dimension of the input data, (2) train a Multi-Layer Perceptron (*MLP*) to perform classification tasks, (3) perturb the *SVD* projection matrix in the sense to minimize the training loss function. In the test phase, the test set is multiplied by the perturbed projection matrix to evaluate the performance of the classifier.

The main contribution of this paper, is how the projection matrix is perturbed after the model is trained. This perturbation is inspired by the Targeted Attacks Method, which aims is to change the inputs so that the network classify them into any desired class [36–40]. These inputs are called Adversarial Examples. Previous works on target attacks have been used in image analysis, such as image segmentation [41], face detection [42] or image classification [43]. There are many ways of producing adversarial examples [44–46], the most commonly used one is Fast Gradient Sign Method (*FGSM*) and its variants [40, 47]. The proposed approach uses *FGSM* to perturb the input data iteratively to maximize the probability that each output sample falls into the desired class. Other variants of this method, such as Projected Gradient Descent [45], Basic Iterative Method [47], Boosting *FGSM* with Momentum [48] and many other gradients based methods, could be used [49–51]. For instance, the Projected Gradient Descent is considered as one of the most effective algorithms to generate adversarial samples. However, this method is too time-consuming to be used for training. *FGSM* is a very simple and fast method of generating adversarial examples [40]. The objective is to obtain a good representation of input features in *SVD* projection space, which will be obtained after calculating the perturbed input of the training data.

This work is organized as follows: the proposed method and the dataset used are described in “[Material and methods](#)” section, the obtained results are reported in “[Results](#)” section and the experiments are discussed in “[Discussion](#)” section.

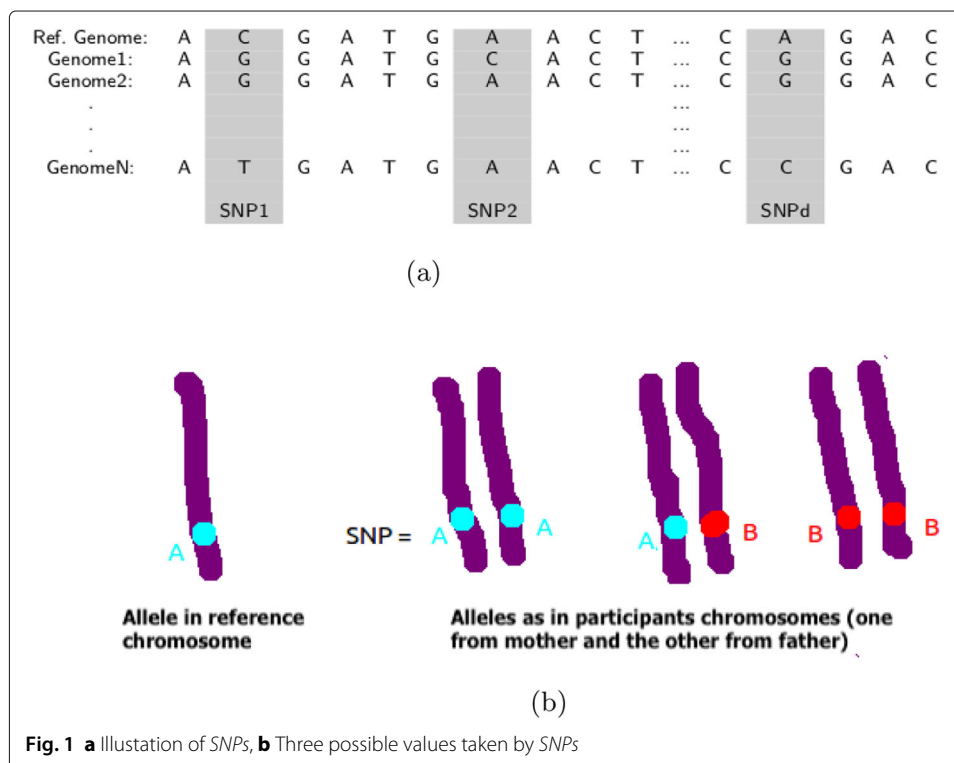
**Material and methods**

The proposed approach uses *SVD* to reduce the number of free parameters of the classification network. However, others dimensionality reduction techniques could be used. For instance, Per class histogram method [33] is a very simple dimensionality reduction technique. The idea of this technique is to represent each feature (*SNP*) in the input data by 3 possible values, corresponding to the proportion of ethnic groups having as genotype 0, 1 or 2 respectively. This produces a projection matrix of size  $m \times 78$ , where  $m$  is number of features. Once the input dimension is reduced, a classification network is trained to find the optimal weight matrix. A perturbed projection matrix is then computed by simply solving a linear system as described in the “Description of the model” section.

**Data description**

1000 Genomes Project set up in 2008 [52], is an international research consortium which aims to produce a detailed catalog of humans genetic variations, from approximately one thousand volunteers from different ethnic groups, with frequencies larger than 1%. It is the first project to sequence the genome of a large number of people from different populations, regions and countries. Data made available to the international community comprises *SNP* profiles of the volunteers (see Fig. 1a), which is a vector where the coordinates are the values taken in a fixed position in the genome sequence (*homozygous reference, heterozygous or homozygous alternate*).

At each locus (fixed position in the genome sequence), an *SNP* is represented by its genotype that takes three possible values for a diploid organism: AA for *homozygous reference*, AB for *heterozygote* and BB for *homozygous alternate* (see Fig. 1b). The *homozygous reference* corresponds to a locus where the two base pairs inherited from the parents are



identical to the one in the reference genome, the *heterozygous* corresponds to a locus where the two base pairs found are different and *homozygous alternate* refers to a locus where the two base pairs found are identical and different from the reference base pair.

Before any further processing, these values were converted into numerical values, e.g., AA=0, AB=1 and BB=2, using the tool Plink [53].

The dataset taken as input for the model is a matrix  $X \in \mathbb{R}^{3450 \times 315345}$ . The rows of the matrix correspond to individuals (1000Genome's volunteers), the columns correspond to *SNPs* positions, and the elements are 0, 1 or 2 (corresponding to the three possible values taken by an *SNP*). 3450 is the number of individuals sampled worldwide from 26 population groups from the 5 continents (see [Appendices](#)) and 315345 is the number of included features (*SNPs* positions).

We use a classification Neural Network composed of an input layer, an output layer and two hidden layers with 100 neurons. This neural network is constructed using Keras and Tensorflow open source libraries. Given the input matrix  $X$ , the output of the model is a vector  $Y \in \mathbb{R}^c$  whose components correspond to the population groups (26 classes in the used example). A relu activation function is used in the two hidden layers followed by a softmax layer to perform ancestry prediction.

### Singular value decomposition

Before applying *SVD*, input data set is divided into two sets, the training set and the test set. *SVD* takes as input the training set matrix transpose denoted by  $X^T \in \mathbb{R}^{m \times n}$  ( $m > n$ ) with  $\text{rank}(X) = r$  and decomposes it into a product of three matrices [54]; two orthogonal matrices  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$  and a matrix  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) \in \mathbb{R}^{m \times n}$ ,  $\sigma_i > 0$  for  $1 \leq i \leq r$ ,  $\sigma_i = 0$  for  $i \geq r + 1$ , such that

$$X^T = U \Sigma V^T = \sum_{i=1}^r U_i \Sigma_i V_i^T.$$

The first  $r$  columns of the orthogonal matrices  $U$  and  $V$  are, respectively, the right and the left eigenvectors associated with the  $r$  nonzero eigenvalues of  $X^T X$ .  $U_i$ ,  $V_i$  and  $\Sigma_i$  are, respectively, the  $i$ th column of  $U$ ,  $V$  and  $\Sigma$ . The diagonal elements of  $\Sigma$  are the nonnegative square roots of the  $n$  eigenvalues of  $X^T X$ .

The dimension of the input matrix  $X$  is then reduced by projecting it onto a space spanned by  $\{U_1, U_2, \dots, U_k\}$ , the top  $k$  ( $k \leq r$ ) singular vectors of  $X$ . Given a set of samples  $x_1, x_2, \dots, x_N$  of dimension  $m$ , the projection matrix  $U^k$  whose columns are formed by the  $k$  first singular vectors of  $X$  must minimize

$$\sum_{i=1}^N \|P(x_i) - x_i\|_2^2 = \sum_{i=1}^N \|x_i U^k - x_i\|_2^2 = \|X U^k - X\|_2^2,$$

where  $P$  is the projection defined by :

$$P : \mathbb{R}^m \longrightarrow \mathbb{R}^k \\ x \longrightarrow x' = x U^k$$

The input data in reduced dimension is denoted by  $X' = X U^k$ .

**Description of the model**

Let's consider a  $L$  hidden layers of a Multi-Layer Perceptron (*MLP*), in which  $n$  input training samples  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  are labeled, i.e., for each input  $\mathbf{x}_i$ , the corresponding output by the model is known and denoted  $y_i$  or  $Y(\mathbf{x}_i)$ .  $Y$  is a vector that contains all the labels. A *MLP* can be described as follows:

$$a_j^{(l)} = \phi(z_j^l), \tag{1}$$

$$z_j^l = \sum_i w_{ij}^l a_i^{(l-1)} + b_j^l = \mathbf{a}^{(l-1)} \cdot \mathbf{w}_j^l + b_j^l, \tag{2}$$

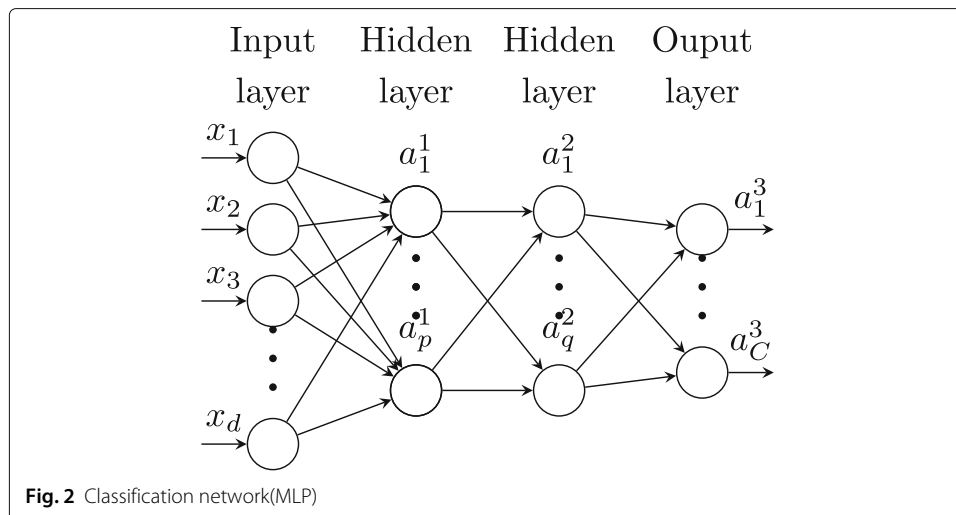
where  $z_j^l$ ,  $b_j^l$  and  $a_j^l$  ( $a_j^0 = x_j$ , for an input  $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_d)^T$ ) are the  $j$ th hidden unit, bias term and activation function of layer  $l$ , respectively.  $w_{ij}^l$  is the weight that links the  $i$ th unit of the  $(l - 1)$ th layer to the  $j$ th unit of the  $l$ th layer.  $\mathbf{w}_j^l$  and  $\mathbf{a}^{(l-1)}$  are, respectively, the incoming weight vector to the  $j$ th neuron of layer  $l$  and the output vector of  $(l-1)$ th layer,  $\phi$  is any activation function. Learning the model consists in finding all the parameters  $\mathbf{w}_j$  and  $b_j$  so that the output  $\mathbf{a}^L$  from the model approximates the true output vector  $\mathbf{y}(\mathbf{x})$ , for all training inputs  $\mathbf{x}$ . For simplification, we consider that there are no bias terms  $b_j^l$  or simply we consider it as an additional component of  $\mathbf{w}_j^l$  and denote by  $W^l$  the matrix whose columns are the vectors  $\mathbf{w}_j^l$  (Fig. 2).

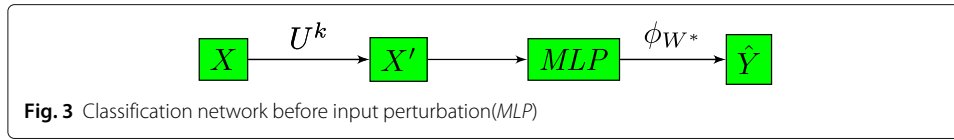
Due to the high dimension of the input data, the proposed approach consists to first project the original data onto a lower dimensional space using *SVD*. Once the dimension of the input data is reduced, a multilayer perceptron (*MLP*) classification network is constructed in three steps:

**Step 1 Learning the weight matrix  $W$**  : First, a classification network(see Fig. 3) is trained to find  $W^*$ , the optimal weight by solving:

$$W^* = \underset{W}{\operatorname{arg\,min}} C_W(X', Y). \tag{3}$$

Where  $C_W(X', Y) = \|\phi_W(X') - Y\|_2^2$  and  $\hat{Y} = \phi_{W^*}(X')$ .  $\phi_W$  is the output activation function for the weight matrix  $W$ .  $Y$  represents the true classification labels.





**Fig. 3** Classification network before input perturbation(MLP)

**Step 2 Input matrix perturbation  $X'$ :** Once the classification network is sufficiently trained, its weight matrix  $W^*$  is fixed and the training input matrix  $X'$  is perturbed to find  $X'^*$  solution of the following problem :

$$X'^* = \underset{Z}{\operatorname{arg\,min}} C_{W^*}(Z, Y), \tag{4}$$

To perturb the input data, we use an iterative version of *FGSM*(see [Appendices](#): Fast gradient sign method) that adds a non random noise whose direction is opposed to the gradient of the loss function.

**Step 3 Projection matrix perturbation  $U^k$ :** After finding the optimal perturbation  $X'^*$ , we look for a perturbed projection matrix  $U^{k*}$  by solving the following linear system :

$$U^{k*} = \underset{V}{\operatorname{arg\,min}} \|XV - X'^*\|_2^2. \tag{5}$$

Where  $X$  is the original training matrix and  $V$  any matrix, with the same size as  $U^k$ . After the three construction steps , the output of the *MLP*, is  $\hat{Y} = \phi_{W^*}^*(X'^*)$ . Once  $U^{k*}$  is calculated, we project the original test set on the latter to evaluate the performance of the classification network.

It is worth noting that, after recovery of the perturbed inputs, the classification network (see Fig. 4) can be re-trained or tested with the fixed weight matrix  $W^*$ (in **Step 2**). From Step 1 and after having solved the system (4), the input matrix  $X$  can be perturbed by solving :

$$U^{k*} = \underset{V}{\operatorname{arg\,min}} \|\phi_{W^*}(XV) - Y\|_2^2. \tag{6}$$

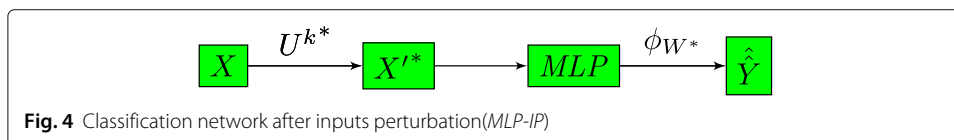
But the high dimensionality of input data makes the non-linear optimization problem difficult to solve and the results less accurate.

### Results

In this section, the obtained results using the proposed method are reported and its performance is compared to that of the once recommended in [33] (the **Per class histogram**, see [Appendices](#): Thin parameters for fat genomics, Table 2).

### Proposed method

In the table below, we summarize the accuracy of the classification with respect to the number of modes (principal components)  $k$  chosen between 20 and 1000.



**Fig. 4** Classification network after inputs perturbation(MLP-IP)

**Table 1** Results obtained by the classification network, before and after inputs perturbation

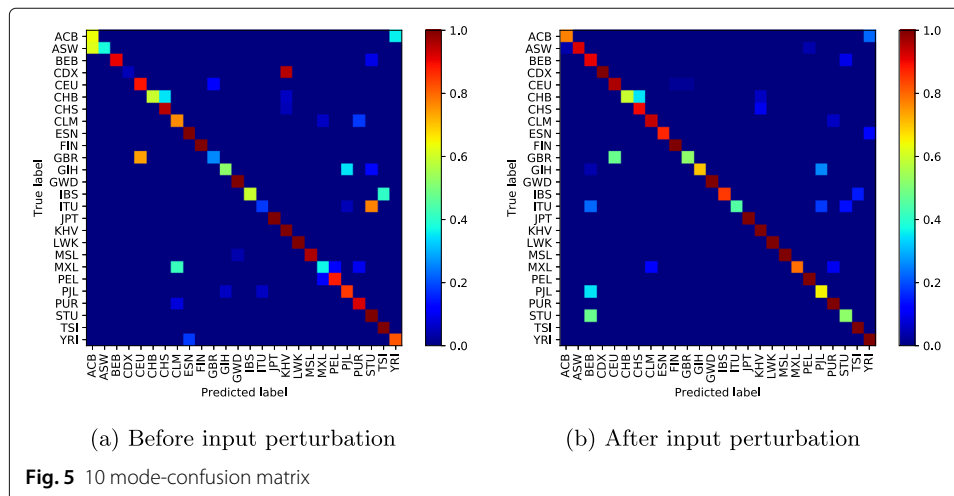
<i>k</i>	<i>MLP</i>	<i>MLP-IP</i>	<i>MLP-IP-R</i>
10	76.46	84.63	87.82
20	84.84	92.02	92.29
50	91.88	96.23	95.71
100	92.75	95.21	94.55
200	92.89	95.65	95.68
500	93.93	94.92	95.44
1000	94.05	94.34	94.02

Table 1 represents in the second column (resp. third column) the results obtained by the classification network before (resp. after) input perturbation. After input perturbation, the training model can be evaluated using the fixed weight matrix (in the third column) as well as re-trained (in the last column). It is clear from the above results that input perturbation has significantly reduced misclassification.

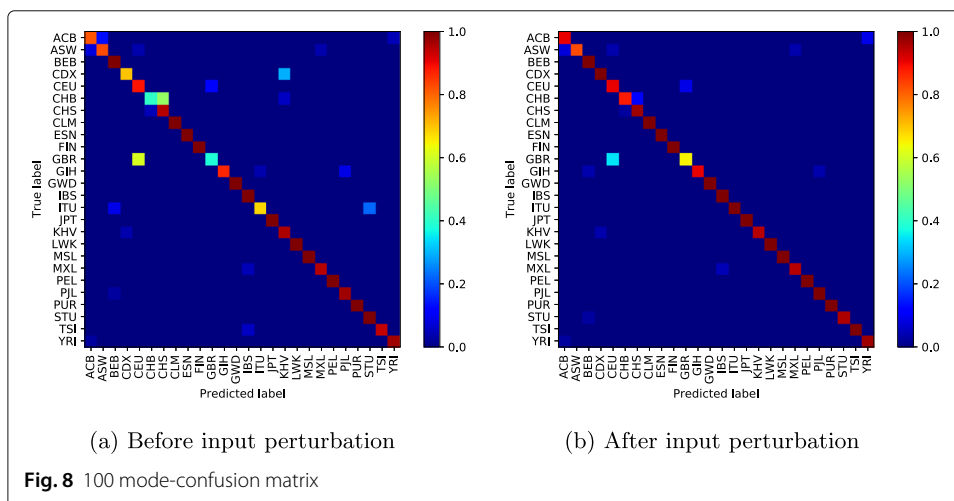
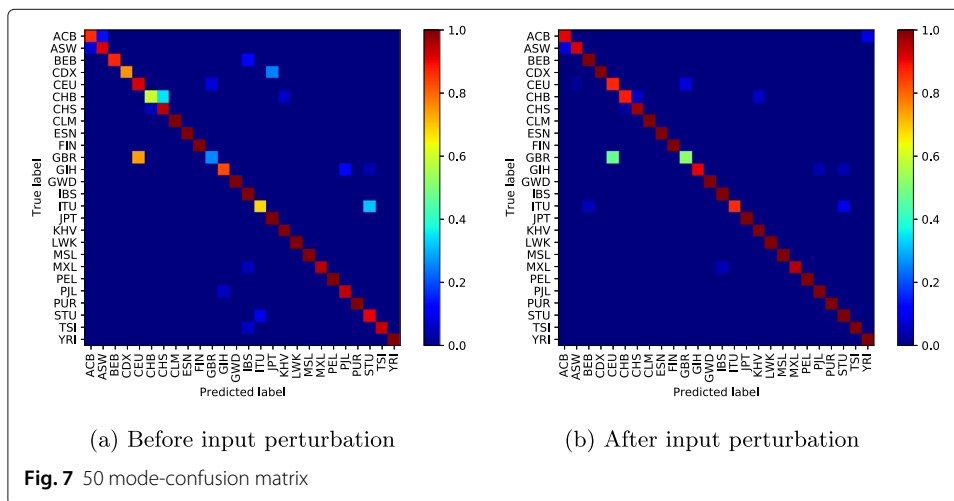
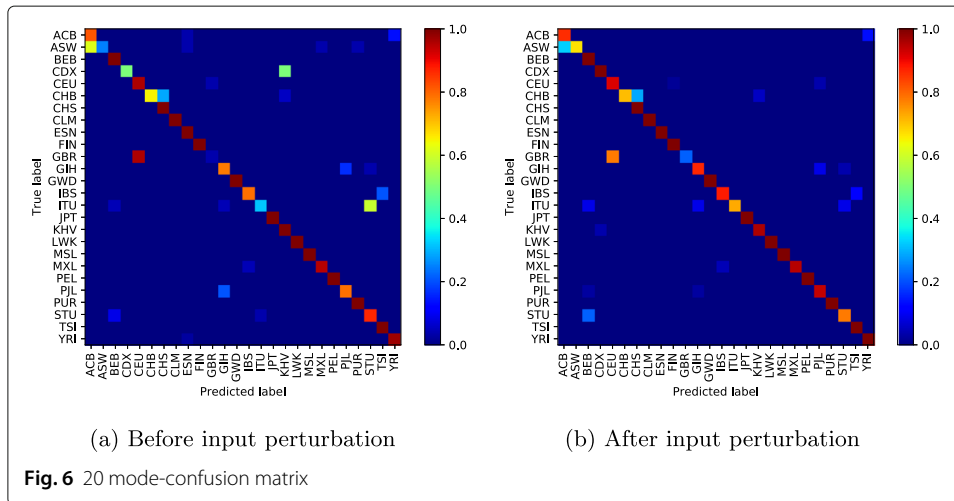
To illustrate the effectiveness of the proposed method, we display the confusion matrix of our classification network to see the effect of input perturbation.

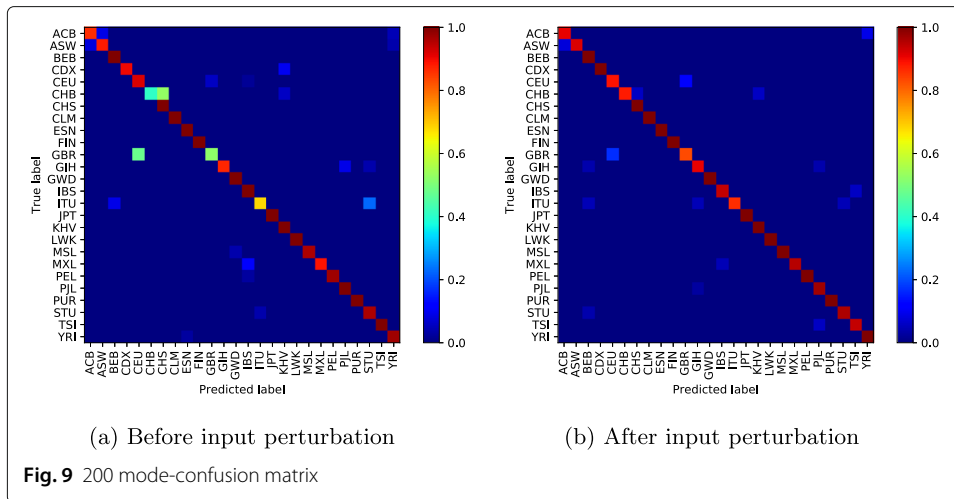
In Fig. 4a (before input perturbation), we observe high classification errors between some population groups such as Chinese Dai in Xishuangbanna and the Kinh in Ho Chi Minh City; Indian Telugu in the UK and Sri Lankan Tamil in the UK; or British in England and Scotland and Utah Residents (CEPH) with Northern and Western Ancestry. Figure 4b shows how our approach has reduced these misclassifications, particularly the classification error between the CDX and KHV classes from **0.95%** to **0.05%**.

However, as the number of modes increases and the classification errors decrease, one can notice throughout our experiences a weak classification error between the British ethnic groups in England, Scotland and Utah Residents (CEPH) with Northern and Western Ancestry, who appear to be genetically very similar (Figs. 5, 6, 7, 8 and 9).







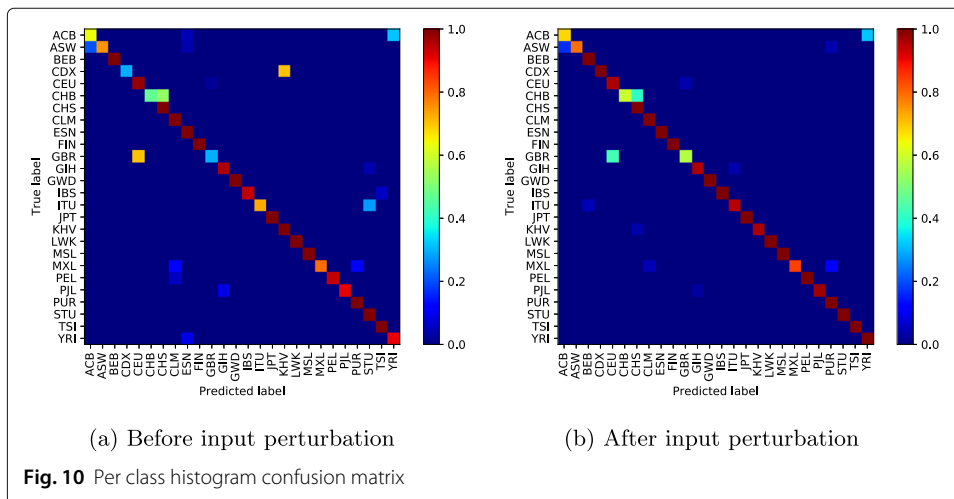


**Per class histogram**

In Fig. 10, we present confusion matrices obtained by per histogram embedding methods and Per class histogram embedding input perturbation. Perturbing per class embedding input reduced misclassification errors and allowed the classifier to reach **94,49%** of accuracy.

**Discussion**

Deep learning application to high-dimensional genomic data, such as *SNPs* is more challenging. In order to deal with problems of high dimensionality, many efforts have been made. In [11], the authors proposed to learn the feature presentations using a Neural Network followed by another classification network. Unsupervised clustering or Deep Autoencoder is jointly trained with a classification network [13, 32, 33, 55]. However, these methods are generally applied to datasets with relatively small features where, the computational cost increases linearly with the number of features and they require more training samples to converge. When Autoencoder network was trained jointly with the classification network on the used dataset, the best accuracy obtained was 85.36%. In



addition to the high dimensionality of the data, there is another challenge related to the high genetic similarity between certain population groups. To mitigate these difficulties, the proposed method reduces the dimension of the input data using SVD algorithm. However, the SVD algorithm extracts linear combinations of features from the input data and fails to take into account the genetic similarity between some population groups as shown in Figs. 4a-10a. To improve these results, the SVD projection matrix is modified to minimize the training loss function of the classification network using FGSM algorithm. The FGSM algorithm allowed us to find the best representation of the input features in SVD projection space. This new representation makes the classification network more robust to small variations in the input and takes into account the genetic similarity between different populations, as shown in the last two columns of Table 1 and Figs. 4b-10b. We are not limited to the SVD algorithm, when Per class histogram is used to reduce the dimension of the input data, the proposed perturbation has significantly reduced classification errors.

The proposed method has achieved its best results when the input features were reduced from 300M to 50, which means that the number of free parameters of the classification network has reduced by a factor of 6000. This method outperforms previous work (see [Appendices](#): Thin parameters for fat genomics) in term of accuracy and the number of free parameters required by the model. For future work, we expect to improve this method by using different targeted attacks algorithms with other dimensionality reduction techniques.

## Conclusion

In this work, we proposed a New Neural Network method for the prediction of individual ancestry from *SNPs* data. To deal with the high dimensionality of the *SNPs* data, our approach first uses *SVD* to reduce the dimensionality of its inputs, then train a classification network and then reduce prediction errors by perturbing the input data set.

The obtained results showed how input perturbation reduced classification errors despite genetic similarities between some ethnic groups. With such accuracy in the task of predicting genetic ancestry, this method will make it possible to deal with more complex problems in the healthcare field. We therefore, intend to apply our method to gene expression profiles as well as *SNPs* data in order first to predict and then prevent the development of patients genetic diseases.

## Appendices

### Fast gradient sign method

*FGSM* ([40]) : uses the gradient of the loss function to determine in which direction the input data features should be changed to minimize the loss function :

$$x' = x - \epsilon \text{sign}(\nabla_x C_W(x, y)),$$

$\epsilon$  is a tunable parameter. Iterative Fast Gradient Sign Method (*IFGSM*) consists in adding the perturbation iteratively [47]. In our context, given any input training sample  $z_i$  ( a row of the training input matrix  $X$ ) and its corresponding one-hot label  $y_c$ , we perturb it in the direction of the input space which yields to the highest decrease of the loss function  $C_{W^*}$ , using the Targeted Iterative Fast Gradient Sign Method (*IFGSM*) given by the formula :

**Table 2** Obtained results by [33]

Model & Embedding	Mean Misclassif. Error. (%)	# of free param.
Basic	8.31 ± 1.83	31.5M
Raw end2end	8.88 ± 1.41	21.27K
Random Projection	9.03 ± 1.20	10.1K
SNP2Vec	7.60 ± 1.28	10.1K
Per class histograms	7.88 ± 1.40	7.9K
Basic with reconstruction	7.76 ± 1.38	63M
Raw end2end with reconstruction	8.28 ± 1.92	227.3K
Random Projection with reconstruction	8.03 ± 1.03	20.2K
SNP2Vec with reconstruction	7.88 ± 0.72	20.2K
Per class histograms with reconstruction	7.44 ± 0.45	15.8K

$$z_i^{(m)} = z_i^{(m-1)} - \epsilon \text{sign} \left( \nabla_{z_i} C_{W^*} \left( z_i^{(m-1)}, y_c \right) \right),$$

where  $m = 1, \dots, M$ ,  $z_i^{(0)} = z_i$ ,  $M$  is the number of iterations and  $z_i^* = z_i^{(M)}$  the perturbed version of  $z_i$ . After perturbation, the rows of the matrix  $X^*$  are composed of  $z_i^*$  for  $i = 1, \dots, n$ . Where  $n$  is the number of training samples.

### 1000 genome project legends

#### Population ethnicity legend

**ACB:** African Caribbeans in Barbados; **ASW:** Americans of African Ancestry in SW USA; **BEB:** Bengali from Bangladesh; **CDX:** Chinese Dai in Xishuangbanna; **CEU:** Utah Residents (CEPH) with Northern and Western Ancestry; **CHB:** Han Chinese in Beijing; **CHS:** Southern Han Chinese; **CLM:** Colombians from Medellin; **ESN:** Esan in Nigeria; **FIN:** Finnish in Finland; **GBR:** British in England and Scotland; **GIH:** Gujarati Indian from Houston; **GWD:** Gambian in Western Divisions in the Gambia; **IBS:** Iberian Population in Spain; **ITU:** Indian Telugu from the UK; **JPT:** Japanese in Tokyo; **KHV:** Kinh in Ho Chi Minh City; **LWK:** Luhya in Webuye; **MSL:** Mende in Sierra Leone; **MXL:** Mexican Ancestry from Los Angeles; **PEL:** Peruvians from Lima; **PJL:** Punjabi from Lahore; **PUR:** Puerto Ricans; **STU:** Sri Lankan Tamil from the UK; **TSI:** Toscani in Italia and **YRI:** Yoruba in Ibadan.

#### Geographical region legend

**AFR:** African; **AMR:** Ad Mixed American; **EAS:** East Asian; **EUR:** European and **SAS:** South Asian.

### Thin parameters for fat genomics

We represent in Table 2, different results from [33].

#### Authors' contributions

The author(s) read and approved the final manuscript.

#### Funding

This project was partly funded by H3ABioNet, which is supported by the National Institutes of Health Common Fund under grant number U41HG006941. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

#### Availability of data and materials

The dataset used in this work is freely available ([http://ftp.1000genomes.ebi.ac.uk:21/vol1/ftp/release/20130502/supporting/hd\\_genotype\\_chip/](http://ftp.1000genomes.ebi.ac.uk:21/vol1/ftp/release/20130502/supporting/hd_genotype_chip/)) and the open source libraries used be can found here (<https://www.tensorflow.org/guide/keras/overview>)

## Declarations

### Consent for publication

Not applicable. This manuscript does not contain data from any individual person.

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>The Laboratory of Mathematical Modelling and Numeric in Engineering Sciences, National Engineering School of Tunis, Rue Béchir Salem Belkhiria Campus universitaire, B.P. 37, 1002 Tunis Belvédère, University of Tunis El Manar, Tunis, Tunisia. <sup>2</sup>Laboratory of Bioinformatics, bioMathematics, and bioStatistics, 13 place Pasteur, B.P. 74 1002 Tunis, Belvédère, Institut Pasteur de Tunis, University of Tunis El Manar, Tunis, Tunisia. <sup>3</sup>ADAGOS. Le Belvédère centre, 61 rue El Khartoum, El Menzah, Tunis, Tunisia. <sup>4</sup>College of sciences & Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, 31441, Dammam, Kingdom of Saudi Arabia, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

Received: 1 December 2020 Accepted: 29 March 2021

Published online: 28 June 2021

## References

1. Ku CS, Loy EY, Salim A, Pawitan Y, Chia KS. The discovery of human genetic variations and their use as disease markers: past, present and future. *J Hum Genet.* 2010;55(7):403. <https://doi.org/10.1038/jhg.2010.55>.
2. Collins FS, Brooks LD, Chakravarti A. A dna polymorphism discovery resource for research on human genetic variation. *Geno Res.* 1998;8(12):1229–31. <https://doi.org/10.1101/gr.8.12.1229>.
3. Group ISMW, et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature.* 2001;409(6822):928. <https://doi.org/10.1038/35057149>.
4. Meyer-Lindenberg A, Weinberger DR. Intermediate phenotypes and genetic mechanisms of psychiatric disorders. *Nat Rev Neurosci.* 2006;7(10):818. <https://doi.org/10.1038/nrn1993>.
5. Risch NJ. Searching for genetic determinants in the new millennium. *Nature.* 2000;405(6788):847. <https://doi.org/10.1038/35015718>.
6. Welter D, MacArthur J, Morales J, Burdett T, Hall P, Junkins H, Klemm A, Flicek P, Manolio T, Hindorf L, et al. The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic Acids Res.* 2013;42(D1):D1001–6. <https://doi.org/10.1093/nar/gkt1229>. Oxford University Press.
7. Haas RJ, McCarty CA, Payseur BA. Genetic ancestry inference using support vector machines, and the active emergence of a unique american population. *EJHG.* 2013;21(5):554. <https://doi.org/10.1038/ejhg.2012.258>.
8. Lee C, Mândoiu II, Nelson CE. Inferring ethnicity from mitochondrial dna sequence. In: *BMC proceedings*, vol. 5. Springer; 2011. p. 1–9.
9. Cawley GC, Talbot NLC. On over-fitting in model selection and subsequent selection bias in performance evaluation. *JMLR.* 2010;11:2079–107. <https://doi.org/10.1016/j.patcog.2006.12.015>.
10. Wen J, Thibeau-Sutre E, Diaz-Melo M, Samper-González J, Routier A, Bottani S, Dormont D, Durrleman S, Burgos N, Colliot O, et al. Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation. *Med Image Anal.* 2020;63:101694.
11. Zhou T, Thung K-H, Zhu X, Shen D. Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis. *Hum Brain Mapp.* 2019;40(3):1001–16.
12. Maldonado C, Mora F, Contreras-Soto R, Ahmar S, Chen J-T, do Amaral Júnior AT, Scapim CA. Genome-wide prediction of complex traits in two outcrossing plant species through deep learning and bayesian regularized neural network. *Front Plant Sci.* 2020;11:1734.
13. Pirmoradi S, Teshnehlab M, Zarghami N, Sharifi A. A self-organizing deep auto-encoder approach for classification of complex diseases using snp genomics data. *Appl Soft Comput.* 2020;97:106718.
14. Fakoor F, Ladhak R, Nazi Z, Huber M. Using deep learning to enhance cancer diagnosis and classification. In: *Proceed. of the Inter. Conf. on ML.* New York: ACM; 2013. <https://doi.org/10.1109/ICSCAN.2018.8541142>.
15. Fergus P, Montanez CC, Abdulaimma B, Lisboa P, Chalmers C. Utilising deep learning and genome wide association studies for epistatic-driven preterm birth classification in African-American women. *IEEE/ACM Trans Comput Biol Bioinform.* 2018;17(2):668–78. <https://doi.org/10.1109/TCBB.2018.2868667>.
16. Friedman S, Gauthier L, Farjoun Y, Banks E. Lean and deep models for more accurate filtering of snp and indel variant calls. *Bioinformatics.* 2020;36(7):2060–7.
17. Dorj OU, Lee KK, Choi JY, Lee M. The skin cancer classification using deep convolutional neural network. *Mult Tools App.* 2018;77(8):9909–24. <https://doi.org/10.2196/11936>.
18. Montesinos-López OA, Montesinos-López JC, Singh P, Lozano-Ramírez N, Barrón-López A, Montesinos-López A, Crossa J. A multivariate poisson deep learning model for genomic prediction of count data. *G3 Genes Genomes Genet.* 2020;10(11):4177–90.
19. Danaee P, Ghaeini R, Hendrix DA. A deep learning approach for cancer detection and relevant gene identification. In: *Pacific Symposium on Biocomputing 2017.* World Scientific 5 Toh Tuck Link Singapore, 596224, Singapore; 2017. p. 219–29.
20. Singh R, Lanchantin J, Robins G, Qi Y. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics.* 2016;32(17):639–48. <https://doi.org/10.1093/bioinformatics/btw427>.
21. Dash M, Liu H. Feature selection for classification. *Intel Data Anal.* 1997;1(3):131–56. [https://doi.org/10.1016/S1088-467X\(97\)00008-5](https://doi.org/10.1016/S1088-467X(97)00008-5).
22. Owen AB. A robust hybrid of lasso and ridge regression. *Contemp Maths.* 2007;443(7):59–72. <https://doi.org/10.1090/conm/443/08555>.

23. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580. 2012.
24. Uppu S, Krishna A, Gopalan RP. A deep learning approach to detect snp interactions. *JSW*. 2016;11(10):965–75.
25. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. *Nat Methods*. 2015;12(10):931–4.
26. Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, et al. A universal snp and small-indel variant caller using deep neural networks. *Nat Biotechnol*. 2018;36(10):983–7.
27. Heinrich F, Wutke M, Das PP, Kamp M, Gültas M, Link W, Schmitt AO. Identification of regulatory snps associated with vicine and convicine content of vicia faba based on genotyping by sequencing data using deep learning. *Genes*. 2020;11(6):614.
28. Lenz S, Hess M, Binder H. Unsupervised deep learning on biomedical data with boltzmannmachines. *jl. bioRxiv*. 2019578252.
29. Hess M, Lenz S, Blätte TJ, Bullinger L, Binder H. Partitioned learning of deep boltzmann machines for snp data. *Bioinformatics*. 2017;33(20):3173–80.
30. Poplin R, Newburger D, Dijamco J, Nguyen N, Loy D, Gross S, McLean CY, DePristo MA. Creating a universal SNP and small indel variant caller with deep neural networks. 2016. <https://doi.org/10.1101/092890>.
31. Baliarsingh SK, Vipsita S, Gandomi AH, Panda A, Bakshi S, Ramasubbareddy S. Analysis of high-dimensional genomic data using mapreduce based probabilistic neural network. *Comput Methods Prog Biomed*. 2020;195:105625.
32. Kilicarslan S, Adem K, Celik M. Diagnosis and classification of cancer using hybrid model based on relief and convolutional neural network. *Med Hypotheses*. 2020;137:109577.
33. Romero A, Carrier PL, Erraqabi A, Sylvain T, Auvoilat A, Dejoie E, Legault MA, Dubé MP, Hussin JG, Bengio Y. Diet networks: thin parameters for fat genomics. arXiv preprint arXiv:1611.09340. 2016. <https://doi.org/10.1038/ejhg.2012.258>.
34. Liu B, Wei Y, Zhang Y, Yang Q. Deep neural networks for high dimension, low sample size data. In: International Joint Conference on Artificial Intelligence, California, USA; 2017. p. 2287–93.
35. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med*. 2001;7(6):673. <https://doi.org/10.1038/89044>.
36. Metzzen JH, Genewein T, Fischer V, Bischoff B. On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267. 2017.
37. Kos J, Fischer I, Song D. Adversarial examples for generative models. In: 2018 IEEE Security and Privacy Workshops (SPW). IEEE, New York City, 3 Park Ave, USA; 2018. p. 36–42. <https://doi.org/10.1109/SPW.2018.00014>.
38. Carlini N, Wagner D. Audio adversarial examples: Targeted attacks on speech-to-text. In: 2018 IEEE SPW. IEEE, New York City, 3 Park Ave, USA; 2018. p. 1–7. <https://doi.org/10.1109/SPW.2018.00009>.
39. Zheng S, Song Y, Leung T, Goodfellow I. Improving the robustness of deep neural networks via stability training. In: Proceed. of the IEEE conference on computer vision and pattern recognition. IEEE, New York, US; 2016. p. 4480–8.
40. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. 2014.
41. Arnab A, Miksik O, Torr PHS. On the robustness of semantic segmentation models to adversarial attacks. In: The IEEE Conf. on CVPR. IEEE, New York, US; 2018. <https://doi.org/10.1109/CVPR.2018.00099>.
42. Sharif M, Bhagavatula S, Bauer L, Reiter MK. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceed. of the 2016 ACM SIGSAC Conf. on Comp. and Communications Security. ACM, 1601 Broadway, 10th Floor New York, NY, 10019-7434; 2016. p. 1528–40. <https://doi.org/10.1145/2976749.2978392>.
43. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199. 2013.
44. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: 2017 IEEE SP. IEEE, New York City, 3 Park Ave, USA; 2017. p. 39–57. <https://doi.org/10.1109/SP.2017.49>.
45. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083. 2017.
46. Xie C, Wu Y, Maaten Lvd, Yuille AL, He K. Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, New York, US; 2019. p. 501–9.
47. Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236. 2016.
48. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York, US; 2018. p. 9185–93.
49. Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204. 2017.
50. Tramer F, Boneh D. Adversarial training and robustness for multiple perturbations. arXiv preprint arXiv:1904.13000. 2019.
51. Maini P, Wong E, Kolter Z. Adversarial robustness against the union of multiple perturbation models. In: International Conference on Machine Learning. PMLR; 2020. p. 6640–50.
52. Consortium GP, et al. A map of human genome variation from population-scale sequencing. *Nature*. 2010;467(7319):1061. <https://doi.org/10.1038/nature09534>.
53. Purcell S. Plink. 2009. <https://zzz.bwh.harvard.edu/plink/gvar.shtml>. Accessed 03 Feb 2021.
54. Berry MW. Large-scale sparse singular value computations. *Int J Supercomp Appl*. 1992;6(1):13–49. <https://doi.org/10.1177/109434209200600103>.
55. Chen R, Yang L, Goodison S, Sun Y. Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data. *Bioinformatics*. 2020;36(5):1476–83.

## Publisher's Note

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