frontiers | Frontiers in Oncology

TYPE Original Research
PUBLISHED 10 January 2023
DOI 10.3389/fonc.2022.1091767

OPEN ACCESS

EDITED BY
Jinzhuang Dou,
University of Texas MD Anderson Cancer Center, United States

REVIEWED BY
Shaolong Cao,
Biogen Idec, United States
Minglei Yang,
Department of Medical Genetics, Sun Yat-sen University, China

*CORRESPONDENCE
Zaixiang Tang
tangzx@suda.edu.cn
Ke Lu
sgu8434@sina.com

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to Cancer Genetics, a section of the journal Frontiers in Oncology

RECEIVED 07 November 2022
ACCEPTED 19 December 2022
PUBLISHED 10 January 2023

CITATION
Shen J, Li H, Yu X, Bai L, Dong Y, Cao J, Lu K and Tang Z (2023) Efficient feature extraction from highly sparse binary genotype data for cancer prognosis prediction using an auto-encoder. *Front. Oncol.* 12:1091767. doi: 10.3389/fonc.2022.1091767

Efficient feature extraction from highly sparse binary genotype data for cancer prognosis prediction using an auto-encoder

Junjie Shen[1,2], Huijun Li[1,2], Xinghao Yu[2,3], Lu Bai[1,2], Yongfei Dong[1,2], Jianping Cao[4], Ke Lu[5*†] and Zaixiang Tang[1,2*†]

[1]Department of Biostatistics, School of Public Health, Medical College of Soochow University, Suzhou, China, [2]Jiangsu Key Laboratory of Preventive and Translational Medicine for Geriatric Diseases, Medical College of Soochow University, Suzhou, China, [3]Center for Genetic Epidemiology and Genomics, School of Public Health, Medical College of Soochow University, Suzhou, China, [4]School of Radiation Medicine and Protection and Collaborative Innovation Center of Radiation Medicine of Jiangsu Higher Education Institutions, Soochow University, Suzhou, China, [5]Department of Orthopedics, Affiliated Kunshan Hospital of Jiangsu University, Suzhou, China

Genomics involving tens of thousands of genes is a complex system determining phenotype. An interesting and vital issue is how to integrate highly sparse genetic genomics data with a mass of minor effects into a prediction model for improving prediction power. We find that the deep learning method can work well to extract features by transforming highly sparse dichotomous data to lower-dimensional continuous data in a non-linear way. This may provide benefits in risk prediction-associated genotype data. We developed a multi-stage strategy to extract information from highly sparse binary genotype data and applied it for cancer prognosis. Specifically, we first reduced the size of binary biomarkers *via* a univariable regression model to a moderate size. Then, a trainable auto-encoder was used to learn compact features from the reduced data. Next, we performed a LASSO problem process to select the optimal combination of extracted features. Lastly, we applied such feature combination to real cancer prognostic models and evaluated the raw predictive effect of the models. The results indicated that these compressed transformation features could better improve the model's original predictive performance and might avoid an overfitting problem. This idea may be enlightening for everyone involved in cancer research, risk reduction, treatment, and patient care *via* integrating genomics data.

KEYWORDS

auto-encoder, highly sparse binary data, feature extraction, risk prediction, LASSO

1 Introduction

Modern omics technologies can generate large-scale molecular data, such as genomic, transcriptomic, proteomic, and metabolomic data, inducing the opportunity to build more accurate predictive and prognostic models (1, 2). These data have been used to provide tailored healthcare and precision medicine for many individuals (3). However, such data also present computational and statistical challenges because the complexity of the algorithms grows fast with the number of variables.

The underlying representation of many real processes is often sparse. It is of benefit to be able to efficiently eliminate features in a pre-processing step. From the perspective of data dimension reduction, it can be classified into feature selection and feature extraction. Most existing work on feature selection are based on a variant of $l1$-norm penalty due to its sparsity-induced property, strong theoretical guarantees, and great empirical success in kinds of applications (4). The paper about the least absolute shrinkage and selection operator (LASSO) has had an enormous influence (5).

Count data are ubiquitous in genetic risk studies, where it is highly possible to observe excessive zero counts in rare mutation loci. In the face of mass mutation loci, many penalty methods have been adopted in GWAS analyses to select key genetic loci (6–8). For example, Yang et al. detected genetic risk factors among millions of single-nucleotide polymorphisms (SNPs) in ADNI whole genome sequencing data *via* the LASSO method along with the EDPP screening rules (9). Another solution lies in reducing the number of markers before employing a shrinkage method in genetic model such as (10). "Clumping and thresholding" is a two-step method that is often used to derive polygenic risk score (PRS) from results of GWAS studies (11).

Genetic variation is considered associated with cancer prognosis. However, there is little literature on the use of genetic omics data to predict cancer outcomes. As a matter of fact, it is well documented that a large number of genetic markers and generally the small size of their effects make much of the heritability hidden, as a mass of variants with weak effects on disease usually fail to reach the prespecified thresholds of significance (12). It is always an interesting issue how to aggregate these small effects. To better utilize big data in reasoning systems, feature extraction rather than feature selection may allow for discovery of new pathways and principles (13). We identified the auto-encoder as a promising tool. The auto-encoder is a derivative of artificial neural networks (ANNs), with the aim of learning compact and efficient representations from the input data (14). Usually, these representations have a much lower dimension. Departing from supervised ANNs whose performance depends on the quality of gold standards, the auto-encoder directly uses unlabeled data, i.e., the input data itself is the target of reconstruction. Compared to commonly used feature extraction approaches like principal component analysis or independent component analysis that linearly map input to features, the auto-encoder extracts features into non-linear space and work much better as a tool to reduce dimensionality of data (13).

To sum up, we identified that the auto-encoder could learn compact and efficient features from highly sparse binary data and accordingly developed a multiple-stage process to extract information from binary genotype data and applied it for cancer prognosis. In the first stage (screening), we reduced the number of markers *via* a univariable regression model to a moderate size. In the second stage (extracting), we used a trainable auto-encoder to extract representations from the reduced data. In the third stage (selecting), we performed a LASSO process over a grid of tuning parameter values to select the optimal combination of the extracted features. Finally, we applied such feature combination to cancer prognostic models, and evaluated the raw predictive effect of the models.

2 Materials and methods

2.1 The construction of auto-encoders

A simple auto-encoder is much similar to the ANNs, which generally contains three layers: an input layer, a hidden layer, and a reconstructed layer (output layer) (15). The hidden layer corresponds to the constructed features, with each neuron node representing one feature. The reconstructed layer and the input layer had the same dimensions, and the objective optimized function for the algorithm was to minimize the difference between the two layers.

Let us recall the traditional auto-encoder model proposed by Bengio et al. (16). As many machine learning methods do, we first normalize the continuous input data by the formula $(x - x_{min})/(x_{max} - x_{min})$. Thus, an auto-encoder with "p" features takes an input vector $\mathbf{x}$ in $[0, 1]^p$. The hidden layer representation $\mathbf{y}$ with "d" dimension is constructed through a

Abbreviations: LASSO, least absolute shrinkage and selection operator; GWAS, genome-wide association; SNPs, single-nucleotide polymorphisms; ADNI, Alzheimer's Disease Neuroimaging Initiative; EDPP, enhanced dual polytope projections; PRS, polygenic risk score; ANNs, artificial neural networks; ReLU, rectified linear unit; SNR, signal-to-noise ratio; MNIST, Mixed National Institute of Standards and Technology; MSE, mean squared error; MCE, mean cross-entropy; TCGA, The Cancer Genome Atlas; NCI, National Cancer Institute; NHGRI, National Human Genome Research Institute; SNV, simple nucleotide variation; GDC, Genomic Data Commons; BRCA, breast cancer; OV, ovary cancer; OS, overall survival; RSF, random survival forest; ROC, receiver operating characteristic; AUC, area under the curve; CI, credible interval; CNV, copy number variation; COSMIC, catalogue of somatic mutations in cancer; lncRNA, long noncoding RNA; fMRI, functional magnetic resonance imaging.

deterministic mapping $\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{Wx} + \mathbf{b})$, parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$. **W** is a "p × d" weight matrix and **b** is a bias vector. Function $s(x)$ is called activation function, which introduces nonlinear properties into the network. Common activation functions include (1) rectified linear unit (ReLU) function and (2) sigmoid function:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (1)$$

$$g(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

Equation (1) maps a linear set of input values to an interval ranging from $[0, \infty)$ and equation (2) maps a linear set of input values to an interval in [0, 1]. The value contained in the latent representation **y** for each neuron node is termed the activity value. Then, the resulting hidden layer **y** is mapped back to a "reconstructed" vector **z** in $[0, 1]^p$ in a similar manner, by inputting space $\mathbf{z} = g_{\theta'}(\mathbf{y}) = h(\mathbf{W'y} + \mathbf{b'})$ with $\theta' = \{\mathbf{W'}, \mathbf{b'}\}$. The function $h(x)$ is also an activation function, restoring the latent information to the original information. We could use tied weights if the two activation functions are the same, which means that the transpose of **W** was used for **W'**. The parameters in this neural network are optimized to minimize the average reconstruction loss between the input layer **x** and the reconstructed layer **z**:

$$\theta, \ \theta' = \underset{\theta, \theta'}{\arg\min} \ 1/n \sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) \quad (3)$$

where n is the sample size and L is a loss function like squared error loss function $L(\mathbf{x}, \mathbf{z}) = ||\mathbf{x} - \mathbf{z}||^2$. An alternative error loss, cross-entropy loss function, is suggested by the interpretation of **x** and **z** as vectors of bit probabilities:

$$L_H(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{p} [\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{z}_k)] \quad (4)$$

Like other feed-forward ANNs, the auto-encoder takes back propagation algorithm and gradient descent algorithm to compute and update target parameters iteratively until reaching an acceptable loss or the given epochs. The specific theory can be referred to the relevant literature (17).

2.2 The LASSO and its selection rules

Given a linear regression with standardized predictors x_{ij} and centered response values y_i for i = 1, 2,…, N (samples) and j = 1, 2,…, p (features), the LASSO solves the $l1$-penalized regression problem for finding $\beta = \{\beta_j\}$ to minimize

$$\sum_{i=1}^{N} (y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad (5)$$

where $\lambda \geq 0$ is a tuning parameter.

A main reason for using the LASSO is that the $l1$-penalty tends to set some entries of $\tilde{\beta}$ to 0, and therefore, it performs a kind of variable selection. Furthermore, Tibshirani (18) proposed "strong rules" to discard noise signal in the LASSO-type penalty problems. The results indicated that the LASSO performs well in both low signal-to-noise ratio (SNR) and high sparse regimes by incorporating the "strong rules". However, the predictor matrices from their simulated studies were all generated from Gaussian distribution. Subsequent simulation studies that aimed to improve variable selection algorithm using a LASSO-type penalty still concerned continuous predictors mainly (19–21). Guo et al. considered the power of the LASSO for SNP selection in predicting quantitative traits and proved that the LASSO still has good selection ability for high-dimensional and sparse binary predictors (22). However, when the values of these binary predictors become highly sparse (rare mutation) such as 99.9% of zeros and 0.01% of ones, we observed that the power of the LASSO to select non-zero variables declined. This is briefly illustrated in supplementary file part II and Table S1.

2.3 The property of the auto-encoder to feature selection

We explore the feature extraction capability of the auto-encoder using two visualized image datasets from the Mixed National Institute of Standards and Technology database (MNIST) (23) and fashion MNIST. The MNIST is one of the most widely used benchmark dataset for isolated handwritten digit recognition from 0 to 9. Digits are transformed to 28×28 images, and represented as 784×1 vectors. Each component is a number between 0 and 255, which means the gray levels of each pixel. The number of zeros accounts for about 81%. It has a training set of 60,000 examples, and a test set of 10,000 examples. The fashion MNIST is a substitute for the MNIST dataset and is more complex, consisting of 10 types of wearing images. The number of 0 accounts for about 51%. The above datasets are loaded and accessed through the "Keras" module of TensorFlow. The deep learning framework of the auto-encoder is constructed by the TensorFlow library (2.3.0) of Python (3.7) in the Jupyter Notebook platform (6.3.0).

2.3.1 Handwritten digit recognition

We took the first 1,000 examples of training set as training data and the first 1,000 examples of test set as testing data from the MNIST to study the property of our auto-encoder. First, as mentioned above, we reshaped the 28×28 images to 784×1 vectors and normalized the input data from [0, 255] to [0, 1]. Thus, the dimension of input layer as well as reconstructed layer was 784. We set the hidden layer dimension to 100 (this number is optional). See construction of the auto-encoder in Figure S1.

Activation function $s(x)$ was specified to the ReLU function due to its good property and therefore the activity values in the hidden layer **y** ranging from $[0, \infty)$. The activation function $h(x)$ could be either ReLU function or sigmoid function, corresponding to mean squared error (MSE) loss and mean cross-entropy (MCE) loss. We used the two activation functions respectively and compared the fitting effects.

In terms of configuration training method, we used the "Adam" optimizer from the "Keras" module. The size of each update is controlled by learning rate. To speed up the training, samples were randomly grouped into batches, and the number of samples contained in a batch was termed the batch size, with weight and bias being updated after each batch. Training proceeded through epochs, and samples were re-batched at the beginning of each epoch. Training was stopped after a specified number of epochs (termed epoch size) was reached. We performed a full factorial design over all combinations of the following parameters: a learning rate of 0.001, 0.005, and 0.010; a batch size of 32, 64, and 128; and an epoch size of 50, 100, and 150. After a full factorial parameter sweep, the parameters that we selected were as follows: a learning rate of 0.005, a batch size of 128, and an epoch size of 100, which could achieve fast training speed and smooth loss.

When using the sigmoid function as activation function $h(x)$, the MCE was 0.0683 with a binary accuracy (calculates how often predictions matches labels) of 0.8156 in the training data (see Figure S2A) and 0.0898 MCE with 0.8244 accuracy in the testing data using the model built in training data. We read the first five images of the training data and testing data, as shown in Figures S3A, B. The first row shows the original images, the second row shows the extracted features, and the third row shows that the images were restored accurately with the extracted features. The results show that the model can be used to extract the key features well. Meanwhile, we used the reconstructed data for handwritten digit prediction and found that the probability of predicting the correct classification was close to 1 (see Table S2).

While using the ReLU function as activation function $h(x)$, the MSE was 0.0067 with an accuracy (calculates how often predictions matches labels) of 0.0150 in the training data (see Figure S2B) and 0.0125 MSE with an accuracy 0.0200 in the testing data using the same model. We also read the first five images of the training data and testing data (Figures S3C, D). It shows that the ReLU function performed quite poorer compared to sigmoid function. Because the labels of corresponding output data are normalized data ranging from [0, 1], sigmoid function could work more suitably.

2.3.2 Fashion image recognition

We took the same procedure as *Section 2.3.1* in fashion MNIST data. We selected the first 1,000 examples of training set as training data. The activation function $h(x)$ was directly specified to sigmoid function. We set the same configuration training method except for an epoch size of 200. The MCE was 0.2667 with a binary accuracy of 0.5166 in the training data (see Figure S4A). We read the first six images of the training data, as shown in Figure S5A. We found that the fitting effect was poorer in the fashion MNIST data than in the MNIST data, because the proportion of zeros is lower in the fashion MNIST data (about 51%) than the MNIST data (about 81%).

Inspired by denoising auto-encoders (24), we artificially added some corruption to training data. Specifically, we set values below 0.21 to zeros in the input data, making the proportion of zeros up to about 58.5%. Then, we retrained the model; the MCE was 0.2440 with an accuracy of 0.5924 in the new (corrupted) training data (see Figure S4B). The first six images of the new training data are shown in Figure S5B. The black icon became a little clearer (e.g., the second on the left, the first on the right). Images before and after the corruption are shown in Figure S5C. The first and third images were before the corruption, and the second and fourth images were after the corruption. Our results show that the higher the proportion of 0 and 1, the better the feature extraction effect of the auto-encoder using the sigmoid function.

2.3.3 Auto-encoder feature selection for highly sparse binary predictors

We used the auto-encoder to extract features from the highly sparse binary data. We randomly used simulation data generated from scenario 5 in Table S1. The sample size was 200 with 400 binary predictors. Thus, in the testing auto-coder, the dimension of the input layer as well as the reconstructed layer was 400. We set the hidden layer dimension to 100, i.e., extracting 100 important features. We used the "Adam" optimizer, and the parameters that we selected were as follows: a learning rate of 0.005, a batch size of 32, and an epoch size of 200. The activation function $h(x)$ was set to sigmoid function.

As a result, the MCE was 0.0001 with a binary accuracy of 1.0000 (see Figure S6A). We read the first five "images" of this simulated data, as shown in Figure S6B. The auto-encoder could recover the scattered genetic signals and when there was no genetic signal in the sample, an identical noise signal was given. The extracted 100 signal features were then used in LASSO Cox regression, and 9 features were selected. We calculated Harrell's concordance index (C-index) with 0.670 (standard error, SE = 0.035) and the R^2 was 0.215. If the LASSO Cox regression were applied directly using 400 binary predictors, a total of 65 predictors were selected (of which 5 were real nonzero predictors). The C-index was 0.721 (SE = 0.030) and the R^2 was 0.379. The result obtained using the auto-encoder was much more close to the performance of scenario 1 in Table S1 (average C-index: 0.647, average R^2: 0.244). Due to the selection of more noise predictors, using the LASSO directly had a virtual height of C-index and R^2 that would induce overfitting.

3 Cancer prognosis application

The Cancer Genome Atlas (TCGA) project was started in 2006 by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI). The database has contained a variety of cancer data from more than 20,000 samples of 33 types of cancer, including transcriptome expression data, genomic variation data, methylation data, and clinical data. As the largest cancer gene database, TCGA has become the first choice for cancer research due to its large sample size, diverse data types and standardized data formats.

We downloaded the latest (in July 2022) single-nucleotide variation (SNV) data and phenotype data of the GDC TCGA Breast Cancer (BRCA) cohort (female) and GDC TCGA Ovary Cancer (OV) cohort from the official website "GDC Data Portal". A total of 977 SNV documents and 1,085 phenotype documents were obtained from BRCA and 480 SNV documents and 597 phenotype documents were obtained from OV. The data type of SNV is masked somatic mutation, read and collated by R package *mafTools*. The overview of SNV in BRCA and OV is shown in Figure S7. We eliminated data with variants that were nonsense mutation. Next, we used the R package *reshape2* to reshape the mutation data to count how many SNV mutations were present in each gene per patient. Zero means wild type, and one means mutated (genotype data with 0/1 values). The interested phenotype in this study was overall survival (OS).

3.1 BRCA data

There were a total of 66,780 SNV items in which 4,910 were nonsense mutation. Many genes had more than one mutation, but we deemed all of them as "mutated" and were labeled "1". A total of 952 BRCA patients with 15,124 genotype data were available. After merging survival data, those with missing survival data were eliminated and 939 subjects were left. Univariable Cox analysis was performed on these 15,124 genotype data as preliminary screening to identify potential contributors, and 1,936 of them with a p-value of less than 0.05 (a rough threshold) were selected for subsequent analysis. We found that if the LASSO Cox regression were applied directly to these 1,936 genotype data, no variables would be selected by LASSO (see Figure S8). This was possible in such a scenario because the proportion of zeros reaches 99.6%. Thus, we thought of using an auto-coder to extract features from these highly sparse binary variables. We also consider random survival forest (RSF) as an alternative to screen the key variables because the random forest method is employed to detect significant SNPs in large-scale GWAS (25).

3.1.1 Feature extraction using an auto-encoder and the development of the prognosis model

Specifically, in our BRCA auto-encoder, the dimension of the input layer as well as the reconstructed layer was 1,936. We set the hidden layer dimension to 100, i.e., extracting 100 important features. Figure 1A shows the construction of the auto-encoder. We used the "Adam" optimizer; the parameters that we selected were as follows: a learning rate of 0.005, a batch size of 32, and an epoch size of 150. The activation function h(x) was set to sigmoid function with MCE loss.

As a result, the MCE was 0.0006 with a binary accuracy of 1.0000 (Figure 1B). We read the first five "images" of these data, as shown in Figure 1C. The auto-encoder could recover the scattered genetic signals well as expected. The extracted 100 signal features were continuous variables (see Table S3 for example) and then thrown into the LASSO Cox regression. Finally, 25 features were selected (see Figure 2). We build a prognosis signature called SNV signature based on these 25 features using the R functions "predict()", "cph()", and "coxph ()" among BRCA patients. The mean C-index of this signature was 0.830 (SE = 0.069), and the mean R^2 was 0.245, which was performed with a fivefold cross-validation process and stepAIC to avoid overfitting.

We used this signature to divide the population into two groups. The optimal cutoff value of the signature was determined using the R package *survminer*. The R package *survival* was used to perform survival analysis between these two groups. The Kaplan–Meier (K-M) curve was used to show difference of survival curves between groups (discrimination). Log-rank test was used to evaluate statistical differences of the survival. The receiver operating characteristic (ROC) curve and its area under the curve (AUC) values were utilized to evaluate the specificity and sensitivity of the signature in a time-dependent manner using the R package *timeROC*. We drew observed survival curves and predicted survival curves to compare the agreement (calibration), by calculating baseline hazard using the R function "basehaz()". We also assessed calibration with calibration plots. A 45°C diagonal line represents perfect calibration, while deviation below or above this line implies overestimation or underestimation of survival.

SNV signature ranging from (−3.564, 6.445) with a mean of 0. Patients were divided into a low-risk group (n = 820) and a high-risk group (n = 119); optimal cutoff value was 1.243 (see Figure 3A). The low-risk group had a much higher survival rate compared to the high-risk group (p < 0.0001). The 8-year survival rate of the low-risk group was over 0.75, whereas that of the high-risk group was almost 0. The time-dependent AUC curve was approximately 0.9 during 8 years (Figure 3B). The 2-, 5-, and 8-year AUC of the signature were 0.928 (95% CI: 0.870–

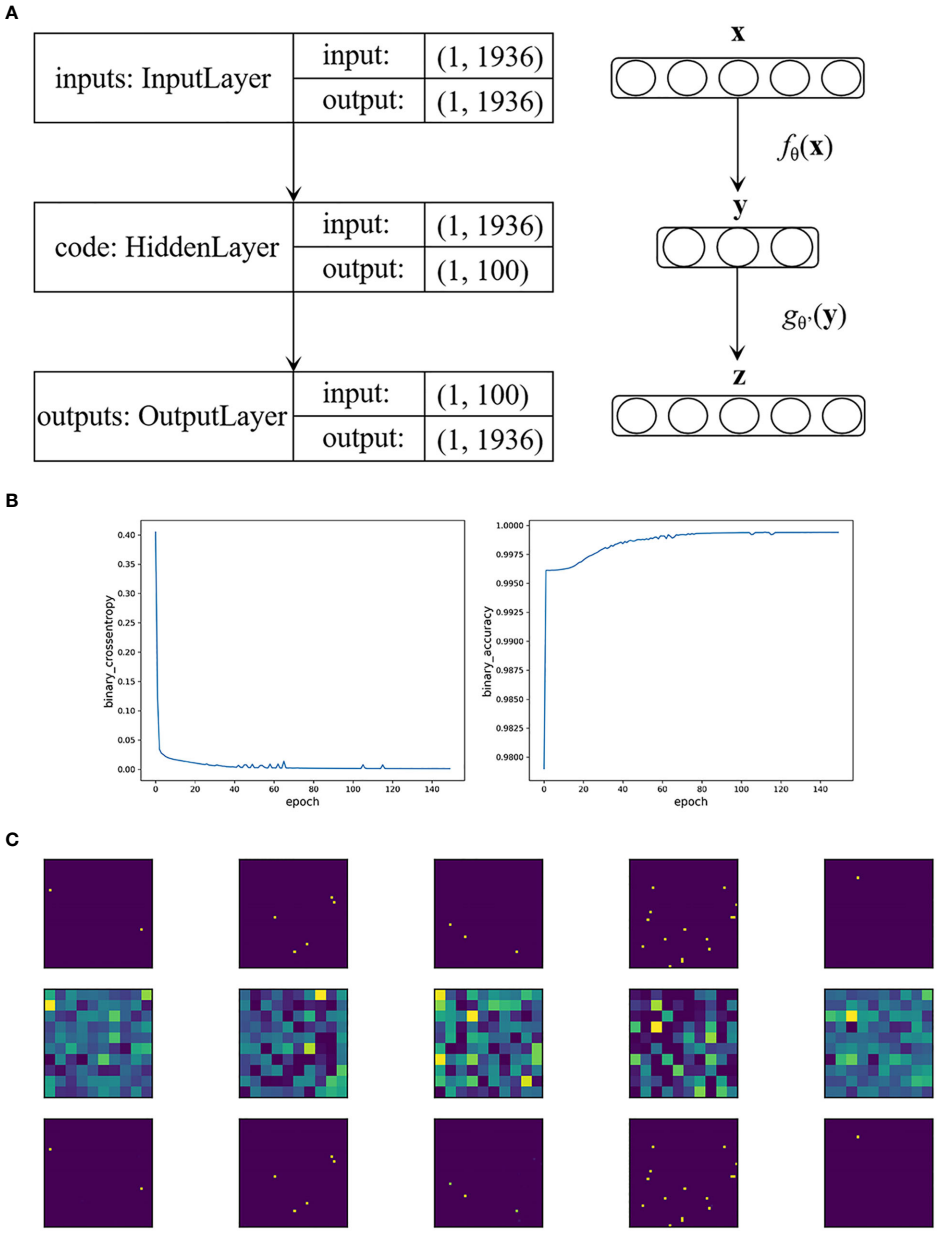

FIGURE 1
(A) The construction of the auto-encoder in BRCA data. **(B)** Loss function value and accuracy of the auto-encoder in BRCA data by the epoch times. **(C)** The first five visualized genetic signal of BRCA data. The first row shows the original images, the second row shows the extracted features, and the third row shows that the images were restored accurately with the extracted features.

0.987), 0.894 (95% CI: 0.840–0.949), and 0.879 (95% CI: 0.821-0.937), respectively. (Figure 3C). The observed survival curves (solid line) and predicted survival curves (dotted line) are shown in Figure 3D. The predicted survival curves were in the credible interval. The signature overestimated survival probability for the low-risk group and underestimated survival probability for the high-risk group. The calibration plot of these two groups shows the same result at 2, 5, and 8 years (Figure 3E).

For a summary of SNVs in both the low-risk group (Figure 4A) and the high-risk group, see Figure 4B. The median of variants per sample in the low-risk group was 30 but 74 in the high-risk group. The rank and distribution of the top 10 mutated genes in the low-risk group was similar to the whole population (Figure S7A). Peculiarly, we plotted the detailed distribution of the top 10 mutated genes in the high-risk group (Figure 4C). Fifty-seven percent of the samples had TP53

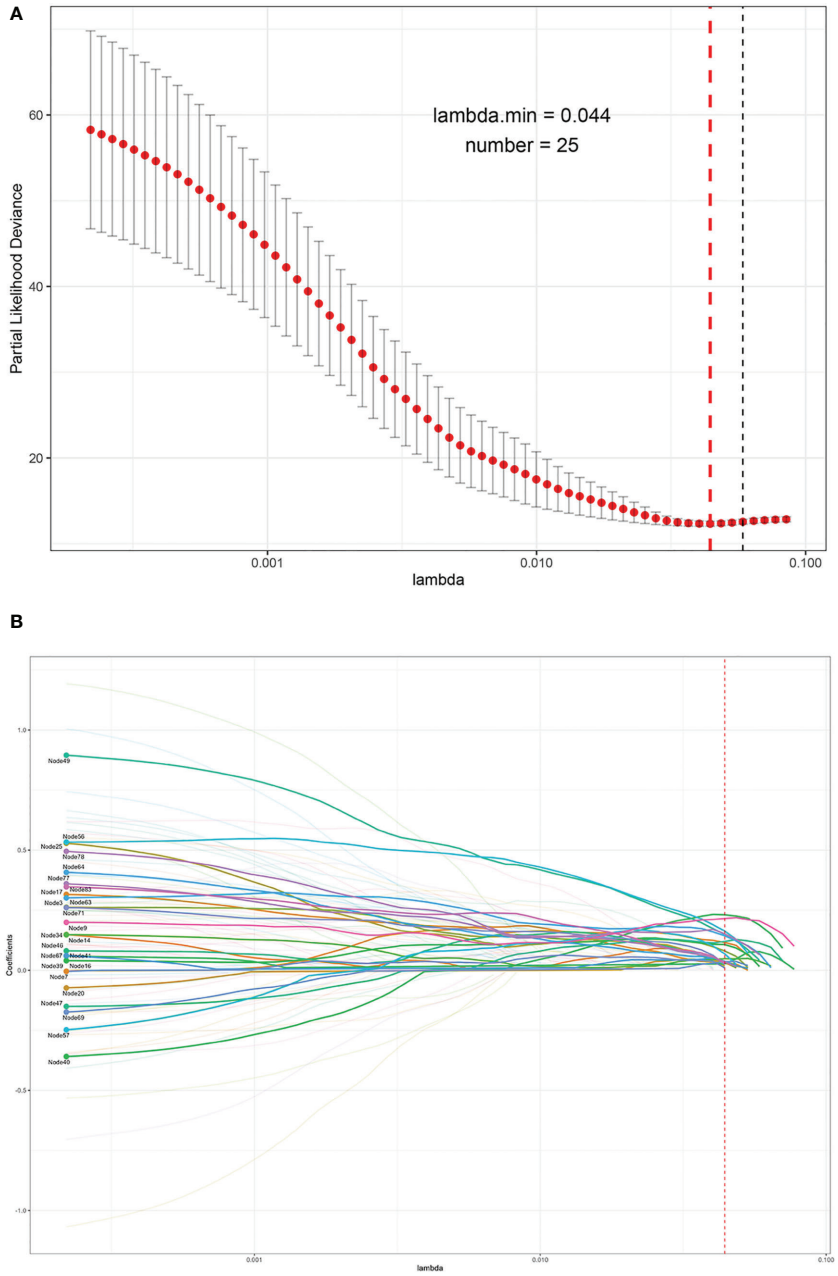

FIGURE 2
The process of the LASSO to select optimal predictors in BRCA data. **(A)** Penalty parameter tuning conducted by 10-fold cross-validation. **(B)** The solution pathway of the 25 features.

mutation in the high-risk group compared to 31% in the low-risk group; 38% of the samples had TTN mutation in the high-risk group compared to 14% in the low-risk group.

3.1.2 RSF for variable screening

RSF is used for prediction and variable selection for right-censored survival and competing risk data (26). A random forest of survival trees is used for ensemble estimation of cumulative hazard function in right-censored settings. Different survival tree splitting rules are used to grow trees. An estimate of C-index is provided for assessing prediction accuracy. Variable importance for single or grouped variables can be used to filter variables and to assess variable predictiveness.

We used the R package *randomSurvivalForest* to build an RSF model and ranked the importance of variables. Number of trees to grow was set to 10,000 in order to ensure that every input row got predicted at least a few times. The result of the model is shown in the Figure S9. Prediction error is measured by the 1 –

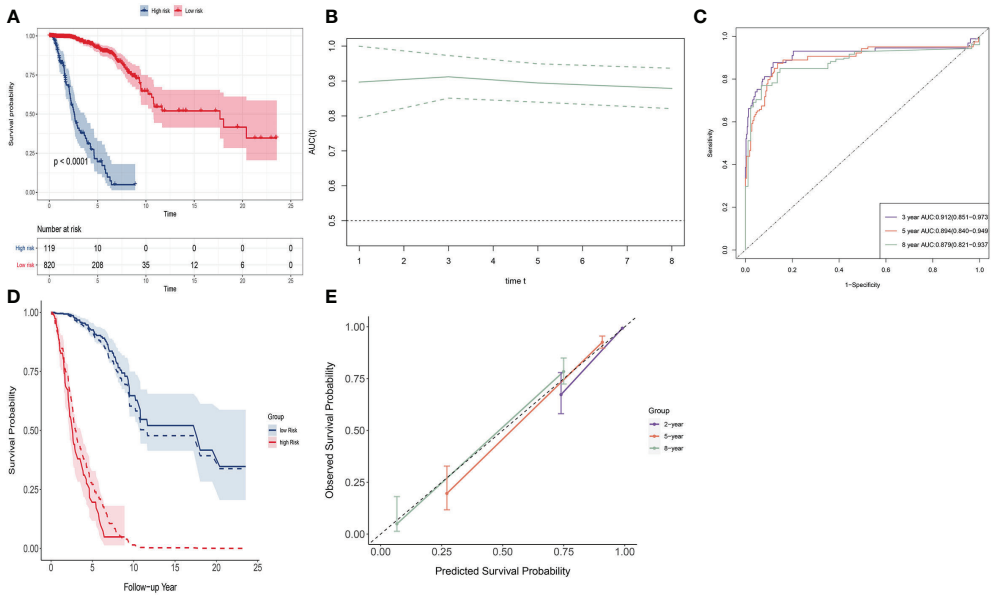

FIGURE 3
Discrimination and calibration of SNV signature in BRCA data. **(A)** The K-M curve of the low-risk group and the high-risk group. **(B)** Time-dependent AUC of SNV signature. **(C)** The 2-, 5-, and 8-year AUC of SNV signature. **(D)** The observed survival curves (solid line) and predicted survival curves (dotted line). **(E)** Calibration plot for 2-, 5-, and 8-year AUC of SNV signature.

C-index. The estimate of prediction error rate of this model was 0.449 (Figure S9A). We selected variables with an importance index greater than 0.3 (21 mutant genes) and plotted them in Figure S9B. However, we selected the 100 most important variables (see Table S4) and threw them into the LASSO Cox regression model. Twenty-three predictors were left (Figure S10). They offered 0.624 (SD = 0.048) of mean C-index and 0.081 of mean R^2 performed with a fivefold cross-validation process and stepAIC. It was not surprising that the C-index and R^2 were much lower using the RSF model when compared to using the auto-encoder (they used a similar number of variables: 25 versus 23) because the RSF model only selected the 100 most important variables and the auto-encoder used whole information.

3.1.3 Genotype and gene expression

We also performed univariable Cox analysis with gene expression data of BRCA. Data category is transcriptome profiling, data type is gene expression quantification, and workflow type is "STAR-Counts". We also selected 1,936 of them with the lowest p-value in univariable Cox analysis. Then, the multivariable LASSO Cox was used to select final predictors. A total of 60 predictors were left (Figure S11). They offered 0.831 (SD = 0.059) of mean C-index and 0.239 of mean R^2 performed with a fivefold cross-validation process and stepAIC. We drew a Venn plot of approximately 1,936 genotypes, 1,936 genes, and 60 predictors (see Figure S12), and found many common genes. Based on an explicit assumption of temporal ordering from genotype, gene expression, and survival outcome, survival mediation analysis of gene expression with multiple genotype exposures is feasible, referring to (27).

3.2 OV data

There were a total of 30,210 SNV items in which 1,650 were nonsense mutation. A total of 406 OV patients with 11,322 genotype data were available. After merging survival data, those with missing survival data were eliminated and 359 subjects were left. Univariable Cox analysis was performed on these 11,322 genotype data, and 1,089 of them with a p-value of less than 0.05 were selected for subsequent analysis. Then, the LASSO Cox regression was applied directly to these data, and a total of 95 predictors were selected by LASSO (see Figure S13A). The mean C-index was 0.707 (SD = 0.032) and the mean R^2 was 0.091 performed with a fivefold cross-validation process and stepAIC. We also used the auto-coder to extract features from the 1,089 binary variables. A total of 19 features were selected from 100 extracted features using the LASSO process (see Figure S13B). The mean C-index of the 19 features was 0.734 (SD = 0.025) and mean R^2 was 0.297 performed with a fivefold cross-validation process and stepAIC.

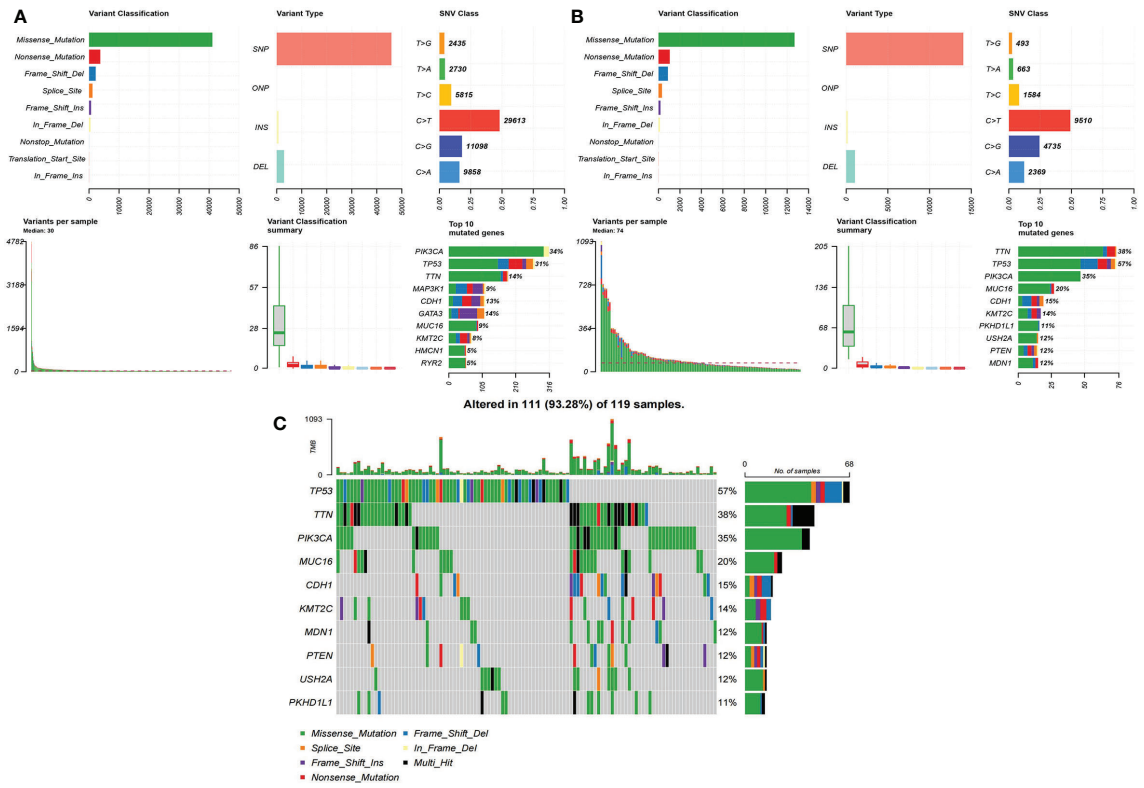

FIGURE 4
The summary of SNVs in two groups in BRCA data. **(A)** Low-risk group. **(B)** High-risk group. **(C)** The detailed distribution of the top 10 mutated genes in the high-risk group.

4 Discussion

The use of transcriptome data to construct cancer prognostic models has become very popular, and its performance in the internal verification is often satisfactory. However, due to different sequencing platforms and sequencing methods, instability of transcriptome data expression, and data standardization problems, extrapolation is still questionable. Trying to get the same desirable results from a random external data is always going to be less than expected.

SNV is a widely studied type of gene mutation (SNP is the most common type), which exists stably in somatic cells and plays a key role in regulating transcriptome expression. Aggregating small effects of SNV is a convincing attempt with promising applications. Our research shows that auto-encoders can extract effective information from dichotomous data well, even in the case of highly sparse variable values. It maps the linear combination of input dichotomous variables to a continuous value space with a lower dimension by neural networks and activation function. These features can retain most of the original information without worrying about overfitting issues, because our goal is to get the original information as possible. In addition, compared to highly sparse binary variables, low-dimensional continuous variables are better utilized. Therefore, we thought of using the auto-encoder to integrate such highly sparse binary SNV data.

Studies have shown that inherited genetic variation is associated with cancer prognosis (28–30). However, few studies have used SNV information to predict cancer prognosis in female patients. A study using multi-omics data [including gene expression data, copy number variation (CNV) data, and SNP] to predict the prognosis of BRCA patients had a 5-year survival AUC of 0.65 through their six-gene signature (31). By contrast, our study shows the power of feature extraction using the deep learning method. Based on the aggregated SNV information, we can greatly improve the ability to predict cancer patients' outcome.

In our study, BRCA patients were stratified into a low-risk group and a high-risk group based on the SNV signature. The high-risk group had higher TP53 and TTN mutation. TP53 is a well-known mutated gene and is a mutant in 30% of all breast cancers. It is clear that the role of TP53 in the management of breast cancer matters (32). Moreover, we searched the existing mutational signatures of BRCA in COSMIC (the catalogue of somatic mutations in cancer, https://cancer.sanger.ac.uk/signatures/) and found that TP53 mutation is validated to be concordant with transcriptome expression (33). TTN-AS1 is a long noncoding RNA (lncRNA) that binds to titin mRNA (TTN). Many studies

have shown that overexpression of TTN-AS1 correlates with poor prognosis in breast cancer and with more advanced pathology (34).

Furthermore, we searched for studies on SNP analysis with the auto-encoder in PubMed (8, 35–37). The most cutting-edge methods take auto-encoders to extract features from SNP data too (35). Specifically, the authors applied a deep canonically correlated sparse auto-encoder to extract key features from SNP data and functional magnetic resonance imaging (fMRI) data and then stacked these features together for classification. Their approach is very interesting and engaging because they addressed the nonlinear dimension reduction and considered the correlation between the above two types of data. The AUC score of their proposed model for the SNP data was 0.984 and that for fMRI data was 0.953, which were the highest AUC scores among all models. The difference of our study is that we have made an interesting experiment on the feature extraction property of auto-encoders. We compared the selection of activation functions in the output layer and found that the sigmoid function was more suitable for feature extraction than the ReLU function. The effect of dichotomous data was better than continuous data. In addition, the data involved in our study were from publicly available databases; thus, all results are reliable and reproducible.

Our study has its limitations. First, a person's entire sequencing genome data are not easy to come by, which makes it difficult to verify the performance of the prediction model externally, but it is hoped to be achieved in the future. Second, although we considered the correlation between covariates within and between groups in our simulation study in supplementary files, we did not incorporate genetic elements such as linkage disequilibrium. Third, due to the randomness of parameter initialization, results of deep neural network training are also random. Therefore, the characteristics obtained from each training time are always different. For example, in the BRCA dataset, each time the auto-encoder was retrained, the obtained features used for the LASSO analysis were different, as well as the C-index. However, the difference was not apparent, only causing the raw C-index to move around an interval, say 0.865 to 0.915 (see Table S5). Therefore, any training result is feasible in a single test. Furthermore, there may be many other scenarios where deep neural networks can be used to extract features and make use of them. This remains to be discovered by the scholars.

5 Conclusion

Integrating minor effects from highly sparse genetic genome data could improve prediction power. We studied the feature extraction property of the auto-encoder and found that it can work well to extract features by transforming highly sparse binary data (e.g., rare mutation) to lower-dimensional continuous data in a non-linear way. We applied this method to two cancer prognosis studies that had genotype data and achieved good predictive performance. This idea may provide something for everyone involved in cancer research, risk reduction, treatment, and patient care.

Data availability statement

We obtained image data information from MNIST (http://yann.lecun.com/exdb/mnist/) and fashion MNIST (https://jobs.zalando.com/en/tech/?gh_src=281f2ef41us). We obtained data information of BRCA and OV from official website "GDC Data Portal" (https://portal.gdc.cancer.gov/repository).

Author contributions

Study conception and design: JS, KL, and ZT. Data collection and cleaning: JS and HL. Real data analysis and interpretation: JS, XY, and JC. Drafting of the manuscript: JS, LB, and YD. All authors reviewed the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by the National Natural Science Foundation of China (81773541), funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions at Soochow University and the State Key Laboratory of Radiation Medicine and Protection (GZK1201919) to ZT, the National Natural Science Foundation of China (81872552 and U1967220) to JC, and the National Natural Science Foundation of China (82172441) and Suzhou Key Clinical Diagnosis and Treatment Technology Project (LCZX201925) to KL. The funding body did not play any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Acknowledgments

We acknowledge the contributions of the TCGA cohort study and MNIST team.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fonc.2022.1091767/full#supplementary-material

References

1. Karczewski KJ, Snyder MP. Integrative omics for health and disease. *Nat Rev Genet* (2018) 19(5):299–310. doi: 10.1038/nrg.2018.4

2. Manzoni C, Kia DA, Vandrovcova J, Hardy J, Wood NW, Lewis PA, et al. Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. *Briefings Bioinf* (2018) 19(2):286–302. doi: 10.1093/bib/bbw114

3. Tran KA, Kondrashova O, Bradley A, Williams ED, Pearson JV, Waddell N. Deep learning in cancer diagnosis, prognosis and treatment selection. *Genome Med* (2021) 13(1):152. doi: 10.1186/s13073-021-00968-x

4. El Ghaoui L, Viallon V, Rabbani T. Safe feature elimination in sparse supervised learning. *Pacific J Optimization.* (2012) 8(4):667–98.

5. Tibshirani R. Regression shrinkage and selection *via* the lasso. *J R Stat Soc Ser B-Methodological* (1996) 58(1):267–88. doi: 10.1111/j.2517-6161.1996.tb02080.x

6. Ayers KL, Cordell HJ. SNP selection in genome-wide and candidate gene studies *via* penalized logistic regression. *Genet Epidemiol* (2010) 34(8):879–91. doi: 10.1002/gepi.20543

7. Long N, Gianola D, Rosa GJ, Weigel KA. Dimension reduction and variable selection for genomic selection: application to predicting milk yield in holsteins. *J Anim Breed Genet* (2011) 128(4):247–57. doi: 10.1111/j.1439-0388.2011.00917.x

8. Prive F, Aschard H, Blum MGB. Efficient implementation of penalized regression for genetic risk prediction. *Genetics* (2019) 212(1):65–74. doi: 10.1534/genetics.119.302019

9. Yang T, Wang J, Sun Q, Hibar DP, Jahanshad N, Liu L, et al. Detecting genetic risk factors for alzheimer's disease in whole genome sequence data *via* lasso screening. *Proc IEEE Int Symp BioMed Imaging* (2015) 2015:985–9. doi: 10.1109/ISBI.2015.7164036

10. Tamba CL, Ni YL, Zhang YM. Iterative sure independence screening EM-Bayesian LASSO algorithm for multi-locus genome-wide association studies. *PLoS Comput Biol* (2017) 13(1):e1005357. doi: 10.1371/journal.pcbi.1005357

11. Wray NR, Goddard ME, Visscher PM. Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Res* (2007) 17(10):1520–8. doi: 10.1101/gr.6665407

12. Gibson G. Rare and common variants: twenty arguments. *Nat Rev Genet* (2012) 13(2):135–45. doi: 10.1038/nrg3118

13. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intelligence* (2013) 35(8):1798–828. doi: 10.1109/TPAMI.2013.50

14. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* (2006) 313(5786):504–7. doi: 10.1126/science.1127647

15. Tan J, Ung M, Cheng C, Greene CS. Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. *Pac Symp Biocomput* (2015), 20:132–43. doi: 10.1142/9789814644730_0014

16. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layerwise training of deep networks. *Adv Neural Inf Process Syst 19* (2007), PP. 153–60.

17. Kriegeskorte N, Golan T. Neural network models and deep learning. *Curr Biol* (2019) 29(7):R231–R6. doi: 10.1016/j.cub.2019.02.034

18. Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, Taylor J, et al. Strong rules for discarding predictors in lasso-type problems. *J R Stat Soc Ser B Stat Methodol* (2012) 74(2):245–66. doi: 10.1111/j.1467-9868.2011.01004.x

19. Guo P, Zeng FF, Hu XM, Zhang DM, Zhu SM, Deng Y, et al. Improved variable selection algorithm using a LASSO-type penalty, with an application to assessing hepatitis b infection relevant factors in community residents. *PloS One* (2015) 10(7):e0134151. doi: 10.1371/journal.pone.0134151

20. Wang J, Wonka P, Ye JP. Lasso screening rules *via* dual polytope projection. *J Mach Learn Res* (2015) 16:1063–101. doi: 10.48550/arXiv.1211.3966

21. Jiang Y, He YX, Zhang HP. Variable selection with prior information for generalized linear models *via* the prior LASSO method. *J Am Stat Assoc* (2016) 111(513):355–76. doi: 10.1080/01621459.2015.1008363

22. Guo W, Elston RC, Zhu X. Evaluation of a LASSO regression approach on the unrelated samples of genetic analysis workshop 17. *BMC Proc* (2011) 5 Suppl 9:S12. doi: 10.1186/1753-6561-5-S9-S12

23. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc Ieee* (1998) 86(11):2278–324. doi: 10.1109/5.726791

24. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. ICML '08: Proceedings of the 25th international conference on Machine learning (2008), 1096–103 p. doi: 10.1145/1390156.1390294

25. Bureau A, Dupuis J, Falls K, Lunetta KL, Hayward B, Keith TP, et al. Identifying SNPs predictive of phenotype using random forests. *Genet Epidemiol* (2005) 28(2):171–82. doi: 10.1002/gepi.20041

26. Ishwaran H, Gerds TA, Kogalur UB, Moore RD, Gange SJ, Lau BM. Random survival forests for competing risks. *Biostatistics* (2014) 15(4):757–73. doi: 10.1093/biostatistics/kxu010

27. Shao ZH, Wang T, Zhang M, Jiang Z, Huang SP, Zeng P. IUSMMT: Survival mediation analysis of gene expression with multiple DNA methylation exposures and its application to cancers of TCGA. *PloS Comput Biol* (2021) 17(8):e1009250. doi: 10.1371/journal.pcbi.1009250

28. Barrdahl M, Canzian F, Lindstrom S, Shui I, Black A, Hoover RN, et al. Association of breast cancer risk loci with breast cancer survival. *Int J Cancer* (2015) 137(12):2837–45. doi: 10.1002/ijc.29446

29. Rafiq S, Tapper W, Collins A, Khan S, Politopoulos I, Gerty S, et al. Identification of inherited genetic variations influencing prognosis in early-onset breast cancer. *Cancer Res* (2013) 73(6):1883–91. doi: 10.1158/0008-5472.CAN-12-3377

30. Lu LG, Katsaros D, Mayne ST, Risch HA, Benedetto C, Canuto EM, et al. Functional study of risk loci of stem cell-associated gene lin-28B and associations with disease survival outcomes in epithelial ovarian cancer. *Carcinogenesis* (2012) 33(11):2119–25. doi: 10.1093/carcin/bgs243

31. Mo WJ, Ding YQ, Zhao S, Zou DH, Ding XW. Identification of a 6-gene signature for the survival prediction of breast cancer patients based on integrated multi-omics data analysis. *PLoS One* (2020) 15(11):e0241924. doi: 10.1371/journal.pone.0241924

32. Shahbandi A, Nguyen HD, Jackson JG. TP53 mutations and outcomes in breast cancer: Reading beyond the headlines. *Trends Cancer* (2020) 6(2):98–110. doi: 10.1016/j.trecan.2020.01.007

33. Smid M, Rodriguez-Gonzalez FG, Sieuwerts AM, Salgado R, Prager-Van der Smissen WJ, Vlugt-Daane MV, et al. Breast cancer genome and transcriptome integration implicates specific mutational signatures with immune cell infiltration. *Nat Commun* (2016) 7:12910. doi: 10.1038/ncomms12910

34. Zheng QX, Wang J, Gu XY, Huang CH, Chen C, Hong M, et al. TTN-AS1 as a potential diagnostic and prognostic biomarker for multiple cancers. *Biomedicine Pharmacother* (2021) 135:111169. doi: 10.1016/j.biopha.2020.111169

35. Li G, Han DP, Wang C, Hu WX, Calhoun VD, Wang YP. Application of deep canonically correlated sparse autoencoder for the classification of schizophrenia. *Comput Methods Programs Biomed* (2020) 183:105073. doi: 10.1016/j.cmpb.2019.105073

36. Massi MC, Gasperoni F, Ieva F, Paganoni AM, Zunino P, Manzoni A, et al. A deep learning approach validates genetic risk factors for late toxicity after prostate cancer radiotherapy in a REQUITE multi-national cohort. *Front Oncol* (2020) 10. doi: 10.3389/fonc.2020.541281

37. Fergus P, Montanez CC, Abdulaimma B, Lisboa P, Chalmers C, Pineles B. Utilizing deep learning and genome wide association studies for epistatic-driven preterm birth classification in African-American women. *Ieee-Acm Trans Comput Biol Bioinf* (2020) 17(2):668–78. doi: 10.1109/TCBB.2018.2868667