 doi:10.20944/preprints202208.0201.v1

Article

Efficient Feature Extraction from High Sparse Binary Genotype Data for Genetic Risk Prediction by Deep Learning Method

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 215123, China
[2] Jiangsu Key Laboratory of Preventive and Translational Medicine for Geriatric Diseases, Medical College of Soochow University, Suzhou 215123, 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
*Corresponding author: sgu8434@sina.com (K.L.); tangzx@suda.edu.cn (Z.X.T.)

Abstract: Genomics involving tens of thousands of genes is a complex system determining phenotype. An interesting and vital issue is that how to integrate highly sparse genetic genomics data with a mass of minor effects into prediction model for improving prediction power. We find that 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 idea may provide benefits in risk prediction based on genome-wide data associated e.g. integrating most of the information in the genotype data. Hence, we developed a multi-stage strategy to extract information from highly sparse binary genotype data and applied it for risk prediction. Specifically, we first reduced the number of biomarkers via a univariable regression model to a moderate size. Then a trainable auto-encoder was used to extract compact representations from the reduced data. Next, we performed a LASSO problem process over a grid of tuning parameter values to select the optimal combination of extracted features. Finally, we applied such feature combination to two prognostic models, and evaluated predictive effect of the models. The results of simulation studies and real data applying indicated that these highly compressed transformation features could better improve predictive performance and did not easily lead to over-fitting.

Keywords: auto-encoder; high sparse binary data; feature extraction; SNV integration

1. Introduction

Present and future are known as "the era of big data" because digital information is growing rapidly and strongly. Health-based big data, especially biological omics data, has now become more widespread used. Thanks to modern omics technologies that can generate powerful large-scale molecular data, e.g. genomic, transcriptomic, proteomic, and metabolomic data, an excellent opportunity is occurred to detect novel biomarkers and build more accurate predictive and prognostic models (Karczewski and Snyder 2018, Manzoni, Kia et al. 2018). For instance, such data have been used in precision medicine to provide tailored healthcare for individuals (Tran, Kondrashova et al. 2021). However, these data also present computational and statistical challenges.

The underlying representation of many real processes is often sparse. From the perspective of data dimension reduction, it can be classified into feature selection and feature extraction. Most of the existing work on sparse learning is based on a variant of $l1$-norm regularization due to its sparsity induced property, convenient convexity, strong theoretical guarantees and great empirical success in kinds of applications (El Ghaoui, Viallon et

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

al. 2012). The paper about the LASSO (full name least absolute shrinkage and selection operator) has had an enormous influence (Tibshirani 1996).

Count data are increasingly ubiquitous in genetic association studies, where it is highly possible to observe excessive zero counts in rare mutation loci. Although the single-variant analysis in standard genome-wide association (GWAS) studies has succeeded in identifying thousands of genetic variants associated with hundreds of various characters (Zhu, Zhang et al. 2016), this approach fails to take into account combining effects of multiple genetic markers on complex traits. In this case, many penalty methods have been adopted in GWAS analyses to select key genetic loci (Ayers and Cordell 2010, Long, Gianola et al. 2011, Prive, Aschard et al. 2019). For example, Yang et al. detected genetic risk factors among millions of single nucleotide polymorphisms (SNPs) in ADNI whole genome sequencing (WGS) data via LASSO regression, along with the EDPP screening rules (Yang, Wang et al. 2015). Another solution lies in reducing the number of markers before employing a shrinkage method in genetic model such as (Tamba, Ni et al. 2017). "Clumping and thresholding" (or called "C+T") is a two-step method that often used to derive polygenic risk score (PRS) from results of GWAS studies (Wray, Goddard et al. 2007).

As a matter of fact, it is well documented that large number of genetic markers and generally small size of their effects make much of the lost heritability hidden, as vast variants with weak effects on disease usually fail to reach prespecified thresholds of significance (Gibson 2012). It is always an interesting issue how to aggregate these small effects. To best utilize big data in reasoning systems, feature extraction method rather than feature selection method should allow for the discovery of new pathways and principles, construct features with amenable distributions (Bengio, Courville et al. 2013). Based on these key factors, we identified auto-encoders as a promising tool (Esteva, Robicquet et al. 2019). The auto-encoder is a derivative of artificial neural networks (ANNs), whose aim is to learn compact and efficient representations from the input data (Hinton and Salakhutdinov 2006). Usually these representations are with a much lower dimension. Departing from supervised ANNs whose performance heavily depends on the quality of gold standards, auto-encoders directly use unlabeled data, i.e. the input data itself is target of reconstruction. Compared to commonly used feature extraction approaches like principle component analysis (PCA) or independent component analysis (ICA) that linearly map input to features, auto-encoders extract features in non-linear space and work much better as a tool to reduce dimensionality of data (Bengio, Courville et al. 2013).

To sum up, we developed a promising process called "SES" that proceeded in multiple stages to extract information from high sparse genotype data and applied it for phenotype prediction. In the first stage (screening), we reduced number of markers via a univariate regression model to a moderate size. In the second stage (extracting), we used a trainable auto-encoder to extract compact and efficient 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 extracted features. Finally, we applied such feature combination to prediction and prognostic models, and evaluated the predictive effect of the models.

2. Materials and methods

2.1. 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) (Tan, Ung et al. 2015). 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's recall the traditional auto-encoder model proposed by Bengio et al. (Bengio 2007). As many machine learning methods do, we first normalize the input data by formula $(x - x_{min}) / (x_{max} - x_{min})$. Thus an auto-encoder with p features takes an input vector **x**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

$\in [0, 1]^p$. The hidden layer representation $\mathbf{y}$ with d dimension is constructed through a deterministic mapping $\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{Wx} + \mathbf{b})$, parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$. $\mathbf{W}$ is a p × d weight matrix and b is a bias vector. Function $s(x)$ is called activation function, which introduces nonlinear properties into our network. Common activation functions include (1) rectified linear unit (ReLU) function and (2) sigmoid function:

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

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

Formula (1) maps a linear set of input values to an interval ranged in $[0, \infty)$ and formula (2) to an interval in [0, 1]. The value contained in the latent representation $\mathbf{y}$ for each neuron node is termed the activity value. Then the resulting hidden layer $\mathbf{y}$ is mapped back to a "reconstructed" vector $\mathbf{z} \in [0, 1]^p$ 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 $\mathbf{W}$ was used for $\mathbf{W'}$. The parameters in this neural network are optimized to minimize the average reconstruction loss between the input layer $\mathbf{x}$ and the reconstructed layer $\mathbf{z}$:

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

where n is the sample size, 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 to an acceptable loss or the given epochs. The specific theory can be referred to the relevant literature (Kriegeskorte and Golan 2019).

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 $\lambda >= 0$ 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 exactly 0, and therefore it performs a kind of variable selection. However, the complexity of the algorithms grows fast with the number of variables. Hence it is of interest to be able to efficiently eliminate features in a pre-processing step. Tibshirani with his colleagues (Tibshirani, Bien et al. 2012) proposed the "strong rules" which were based on the Karush-Kuhn-Tucker (KKT) conditions for discarding predictors in the LASSO and LASSO-type penalties problems. Their results indicated that the LASSO performs quite well in both low signal-to-noise (SNR) and high sparse regimes when incorporate "strong rules". That is the LASSO can efficaciously find most fixed non-zero coefficients from a mass of noise and throw others. However, the predictor matrices from their simulated studies were all generated from the Gaussian distribution, so the predictors were all continuous variables. Subsequent simulation studies that aimed to improving variable selection algorithm using a LASSO-type penalty still concerns continuous predictors mainly (Guo, Zeng et al. 2015, Wang, Wonka et al. 2015, Jiang, He et al. 2016). Guo et al.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

considered the power of the LASSO for SNP selection in predicting quantitative traits, proved that the LASSO still has good selection ability for high dimensional and sparse binary predictors (Guo, Elston et al. 2011). But when the values of these binary predictors become highly sparse (such as rare mutation) e.g. 99.9% of zeros and 0.01% of ones, we observed that power of the LASSO to select non-zero variables may begin to decline due to the extensively discrete information. This will be briefly demonstrated in our following simulation study.

2.3. Simulation study

2.3.1. The LASSO selection for highly sparse binary predictors

We set up five scenarios. For each scenario, we generated n (= 200) observations, each subject i with a survival response, consisting of an observed censored survival time $t^{(i)}$ and a censoring indicator $d^{(i)}$, and a vector of **m** (= 15, 100, 200, 300, 400) binary predictors $\mathbf{x}^{(i)} = (x^{(i)}_1, \ldots, x^{(i)}_m)$. In particular, we used R package *bhGLM* (Yi, Tang et al. 2019) (functions "sim.x()" and "sim.y()" are used to generate different types of high dimension variables and responses, and specify the correlation between covariates within and between groups) to generate the simulated survival responses and genotype predictors. The vector $\mathbf{x}^{(i)}$ was generated with 50 elements in a group, where the intra-group correlation was set to 0.6 and the inter-group correlation was 0. Detailed process referenced to (Tang, Shen et al. 2017). Specially, with a genotype predictor an individual was coded 1 if a rare allele was present and 0 otherwise. Thus the genotype predictors were binary.

We set fifteen coefficients β_1 to β_{15} as non-zero, six of which were negative, and the rest of others to be zero. **Table S1** shows the preset 15 non-zero coefficient values for five simulation scenarios. We set the mutation frequency of these 15 genotype predictors to 0.01, and the rest of others to 0.002. Thus the overall proportion of zero is more than 99%. We analyzed each simulated scenarios using the LASSO Cox model with penalty parameter tuning conducted by 10-fold cross-validation that was implemented in the R package *glmnet* (Engebretsen and Bohlin 2019) for replication with 100 times and recorded average numbers of non-zero predictors that were caught by the LASSO. The result is shown in **Table S2**. As noise variables increase, power of the LASSO to selecting non-zero coefficients plummeted (from 10.83 to 2.96) and it was prone to select more zero coefficient variables. In addition, the possibility that the LASSO would not be able to pick any predictors increases (from 0.02 to 0.23).

2.3.2. The property of auto-encoder to feature selection

We explore the feature extraction capability of auto-encoder using two visualized image data sets from: Mixed National Institute of Standards and Technology database (MNIST) (http://yann.lecun.com/exdb/mnist/) (Lecun, Bottou et al. 1998) and fashion MNIST (https://jobs.zalando.com/en/tech/?gh_src=281f2ef41us). The MNIST is one of the most widely used benchmark data set for isolated handwritten digit recognition from 0 to 9. Digits are transformed to 28×28 images, and represent 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 data set and is more complex, consisting of ten types of wearing images. The number of zeros accounts for about 51%. The above datasets are loaded and accessed through "Keras" module of Python's TensorFlow library. The deep learning framework of the auto-encoder is constructed by TensorFlow library (2.3.0) of Python (3.7) in Jupyter Notebook platform (6.3.0).

2.3.2.1 Handwritten digit recognition

We took the first 1000 examples of training set as training data and the first 1000 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

 doi:10.20944/preprints202208.0201.v1

optional). See construction of the auto-encoder in **Figure S1**. Activation function s(x) was specified to ReLU function because of its good property and therefore the activity values in the hidden layer **y** ranged in [0, ∞). 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: learning rate of 0.001, 0.005, 0.010; batch size of 32, 64, 128; epoch size of 50, 100, 150. After a full factorial parameter sweep, the parameters that we selected were: a learning rate of 0.005, a batch size of 128, an epoch size of 100, which could achieve fast training speed and smooth loss.

When using the sigmoid function as activation function, 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 with 0.8244 in the testing data using the same model. We read the first five images of the training data and testing data, as shown in **Figure 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 S3**).

While using the ReLU function as activation function, 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 with 0.0200 in the testing data using the same model. We also read the first five images of the training data and testing data, as shown in **Figure 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 ranged in [0, 1], sigmoid function could work more suitably.

2.3.2.2 Fashion images recognition

We took the same procedure as section 2.3.2.1 to study the fashion MNIST data. We selected the first 1000 examples of training set as training data. The activation function h(x) was directly specified to sigmoid function. Then we set the same configuration training method except 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 five images of the training data, as shown in **Figure S5A**. We found that the fitting effect was poorer in the fashion MNIST data than 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 the denoising auto-encoders (Vincent, Larochelle et al. 2008), we artificially added some corruption to the training data. Specifically, we set values below 0.21 to zeros in the input data, resulting the proportion of zeros to about 58.5%. Then we retrained the model, the MCE was 0.2440 with a binary accuracy of 0.5924 in the new (corrupted) training data (See **Figure S4B**). The first five images of the new training data are shown in **Figure S5B**. The black icon became a little clearer. Images before and after the corruption are shown in **Figure S5C**. The first and third images were before the corruption, 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

2.3.3. Auto-encoder feature selection for highly sparse binary predictors

We tried to use auto-encoder to extract features from the highly sparse binary predictors data. We randomly selected a simulation data generated from section 2.3.1. The sample size was 200 with 400 binary predictors. Thus in the testing auto-coder, the dimension of the input layer as well as reconstructed layer was 400. We set hidden layer dimension to 100, i.e. extracting 100 important features. We used the "Adam" optimizer, the parameters that we selected were: a learning rate of 0.005, a batch size of 32, an epoch size of 200. The activation function $h(x)$ was set to sigmoid function with MCE loss.

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 square 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 square was 0.379. The result obtained using auto-encoder was much more closed to the performance of scenario1 in section 2.3.1 (average C-index: 0.647, average R square: 0.244, see **Table S2**). Due to selecting much more noise predictors, using the LASSO directly had a virtual height of C-index and R square which would induce to over fit.

3. Real data applying

The Cancer Genome Atlas (TCGA) project was started in 2006 by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) (https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga). Over the past dozen years, 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, clinical data, etc. 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) simple nucleotide variation (SNV) data and phenotype data of GDC TCGA Breast Cancer (BRCA) cohort (female) and GDC TCGA Ovary Cancer (OV) cohort from the given official website "GDC Data Portal" (https://portal.gdc.cancer.gov/repository). 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 the R package *mafTools*. The overview of SNV in BRCA and OV is shown in **Figure S7**. We eliminated data with variants that was nonsense mutation. Next we used the R package *reshape2* to reshape the mutation data by counting how many SNV mutations were present in each gene per patient. Zero means wild-type, and one means mutated (genotype). 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". Thus 952 BRCA patients with 15,124 genotype data were available. After merging survival data, samples with missing survival data were eliminated and 939 subjects were left. Univariate Cox analysis was performed on these 15,124 genotype data as preliminary screening to identify potential contributors, and 1,936 of them with *P*-value 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 data, no variables would be selected by the LASSO (See **Figure S8**). This was reasonable in such scenario because the proportion of zeros reaches

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

for 99.6%. Thus we thought of using 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 random forest method is employed to detect significant SNPs in large-scale GWAS (Bureau, Dupuis et al. 2005).

3.1.1. Feature extraction using auto-encoder and construction of prognosis

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

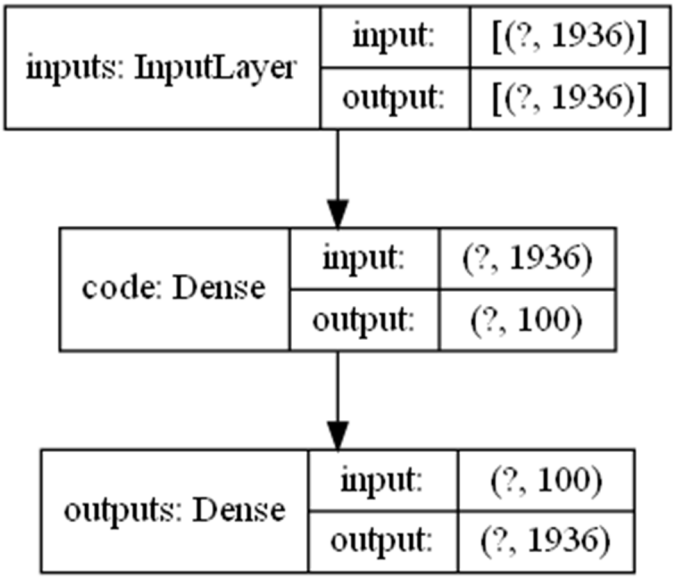


Figure 1. The construction of the auto-encoder in BRCA data.

As a result, the MCE was 0.0006 with a binary accuracy of 1.0000 (**Figure 2**). We read the first five "images" of this data, as shown in **Figure 3**. The auto-encoder could recover the scattered genetic signals well as expected. The extracted 100 signal features were continuous variables (see **Table S4** for example) and then thrown into the LASSO Cox regression. Finally, 25 features were selected (see **Figure 4**). We then build a prognosis signature called SNV-signature based on these 25 features using R function "predict()" among BRCA patients. The C-index of this prognosis signature was 0.877 (SE = 0.023) and the R square was 0.329.

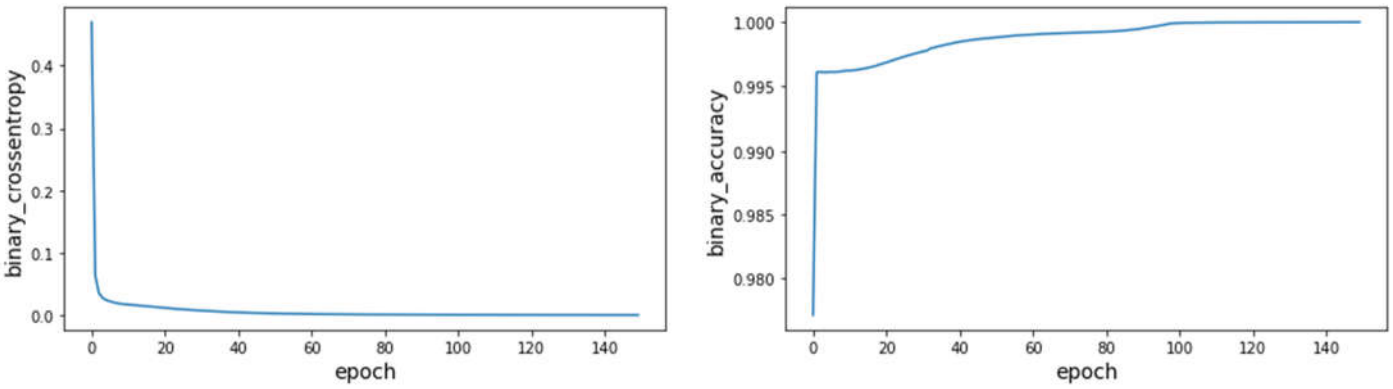


Figure 2. Loss function value and accuracy of the auto-encoder in BRCA data by the epoch times.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

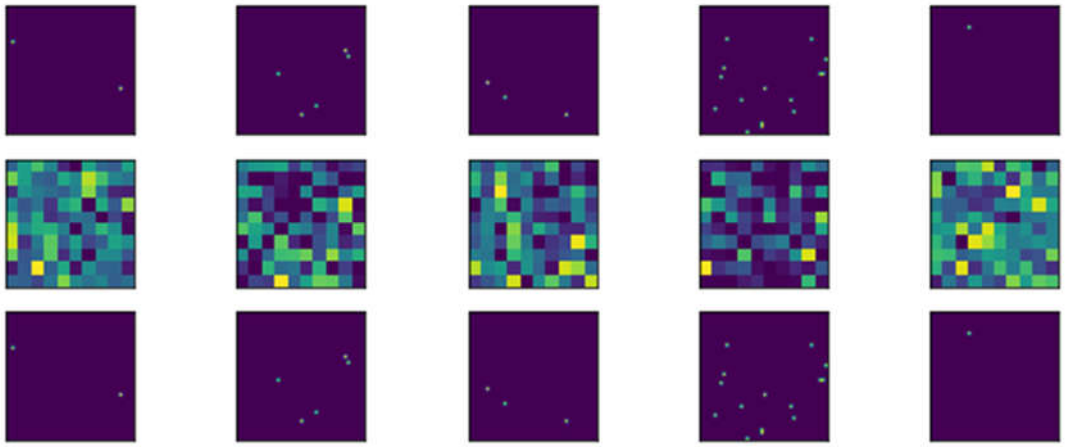
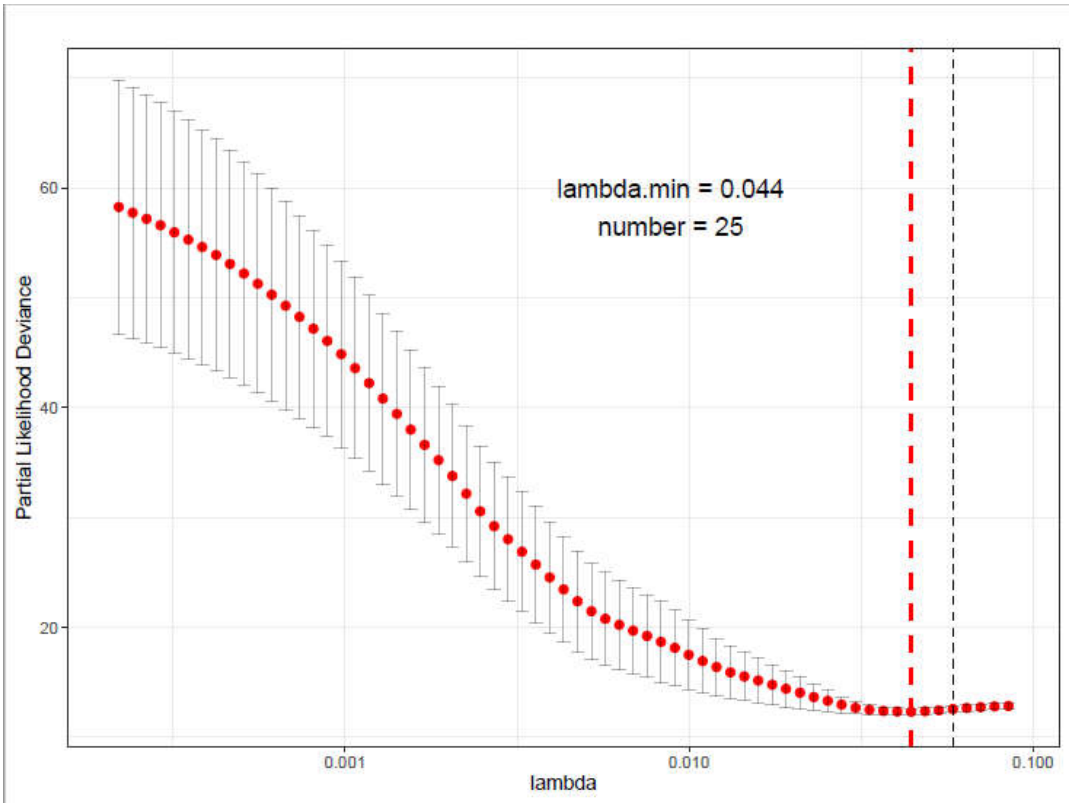
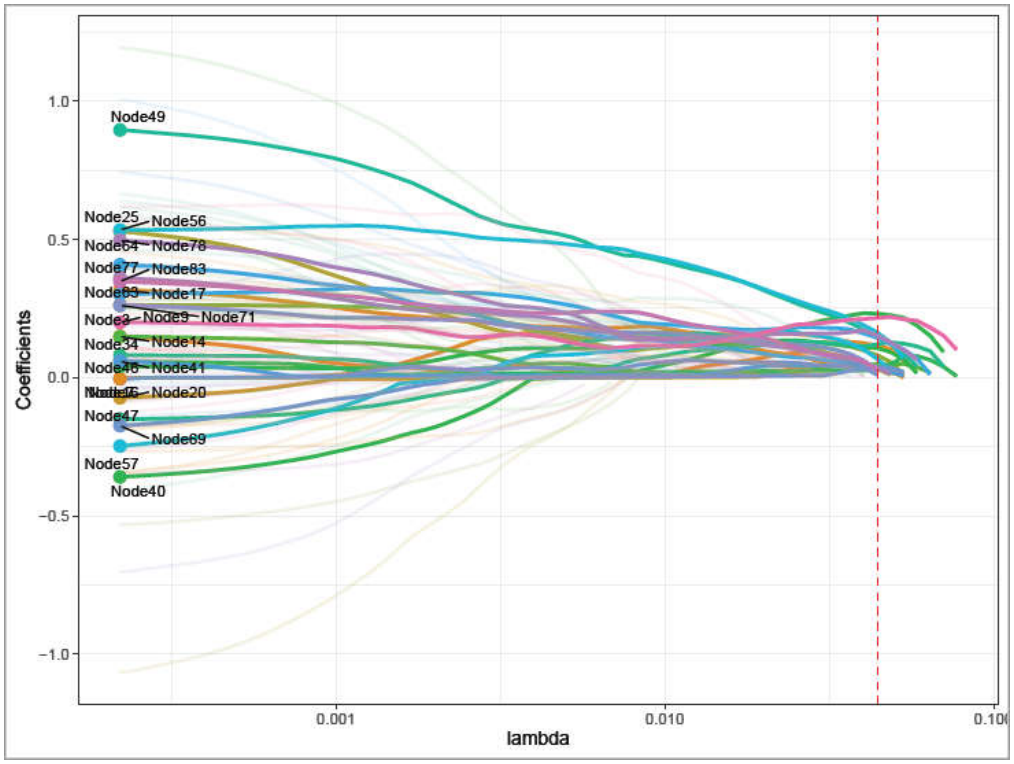


Figure 3. The first five visualized genetic signal of BRCA data. The first row shows the original images, the second row shows the extracted features, and the third row shows that the images were restored accurately with the extracted features.



(A)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1



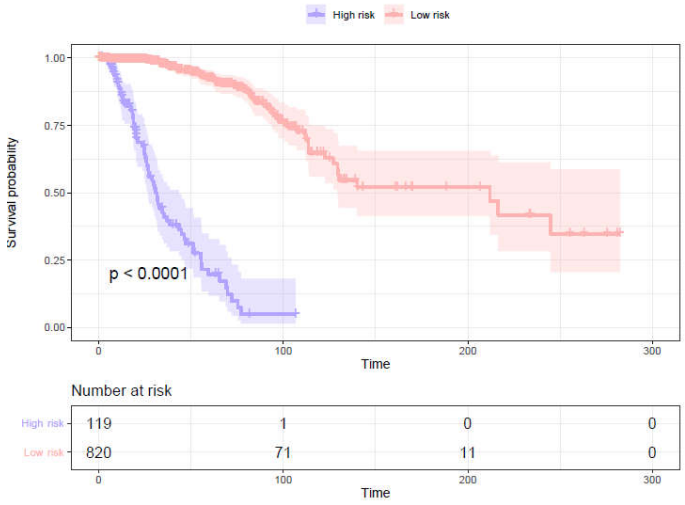
(B)

Figure 4. 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.

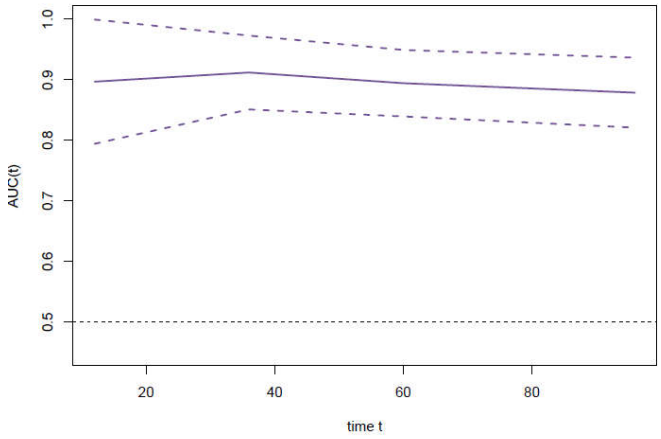
We used this signature to divide the population. The optimal cutoff value of signature was determined using R package *survminer*. R package *survival* was used to perform survival analysis between this two groups. Kaplan-Meier (K-M) curve was used to show difference of survival curves between groups (discrimination). Log-rank test evaluated statistically differences of survival. Receiver operating characteristic (ROC) curves 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 package *timeROC*. The agreement between the expected and observed outcome rates was using calibration curve.

Patients were divided into low risk group (n = 820) and high risk group (n = 119) (See **Figure 5A**). Low risk group had a much higher survival rate compared to high risk group ($P < 0.0001$). The 8-year survival rate of low risk group was more than 0.75 whereas high risk group was lower than 0.10. Time-dependent AUC curve was around 0.9 (**Figure 5B**). The 3-year, 5-year and 8-year AUC of the signature were 0.912 (CI95%: 0.851 - 0.973), 0.894 (CI95%: 0.840 - 0.949), 0.879 (CI95%: 0.821 - 0.937), respectively. (**Figure 5C**). Calibration plot was shown in **Figure 5D** (the agreement was not very high). We looked at the summary of SNVs in both low risk group (**Figure 6A**) and high risk group (**Figure 6B**). The median of variants per sample in low risk group was 30 but 74 in high risk group. The rank and distribution of top 10 mutated genes in low risk group was similar to the whole population (**Figure S7A**). Peculiarly, we plotted the detailed distribution of top 10 mutated genes in high risk group (**Figure 6C**). 57% samples had TP53 mutation in high risk group compared to 31% in low risk group. 38% samples had TTN mutation in high risk group compared to 14% in low risk group.

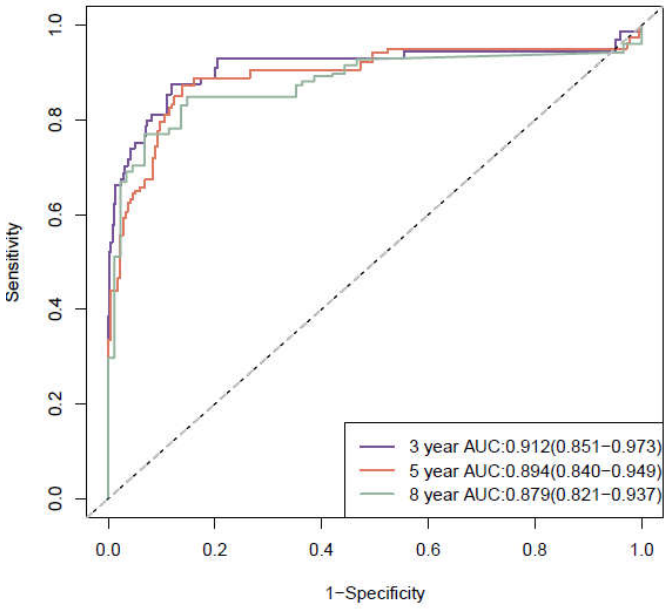
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1



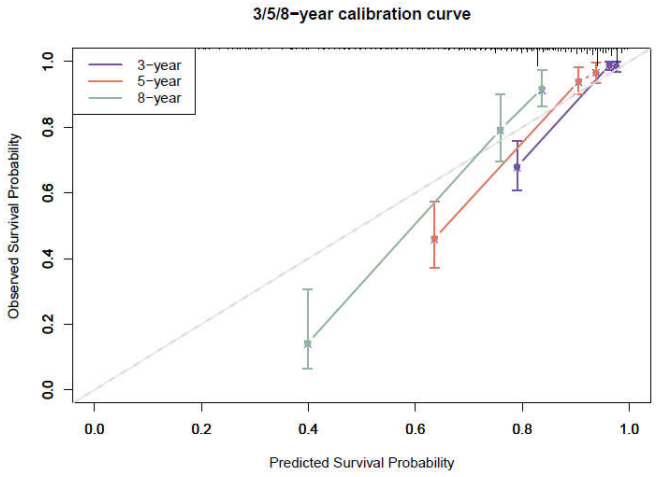
(A)



(B)



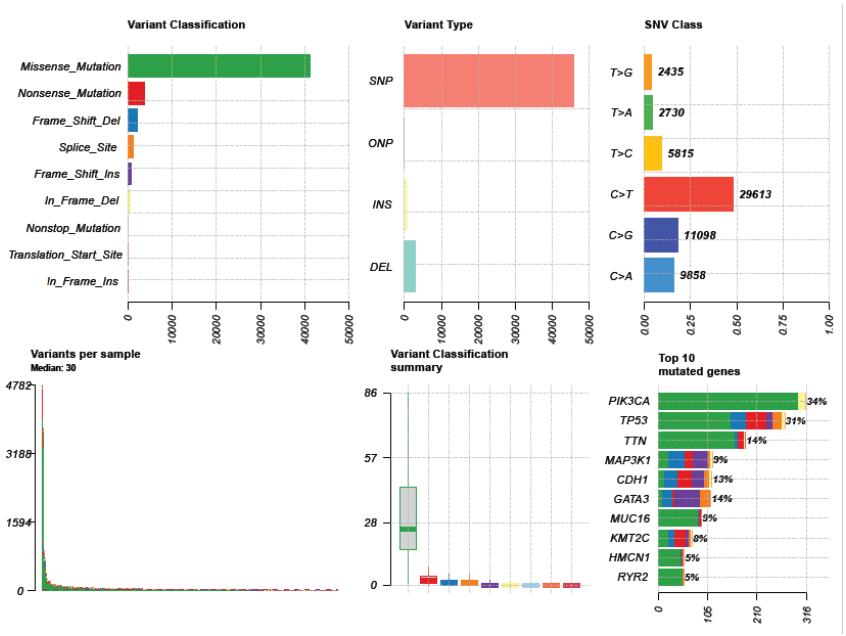
(C)



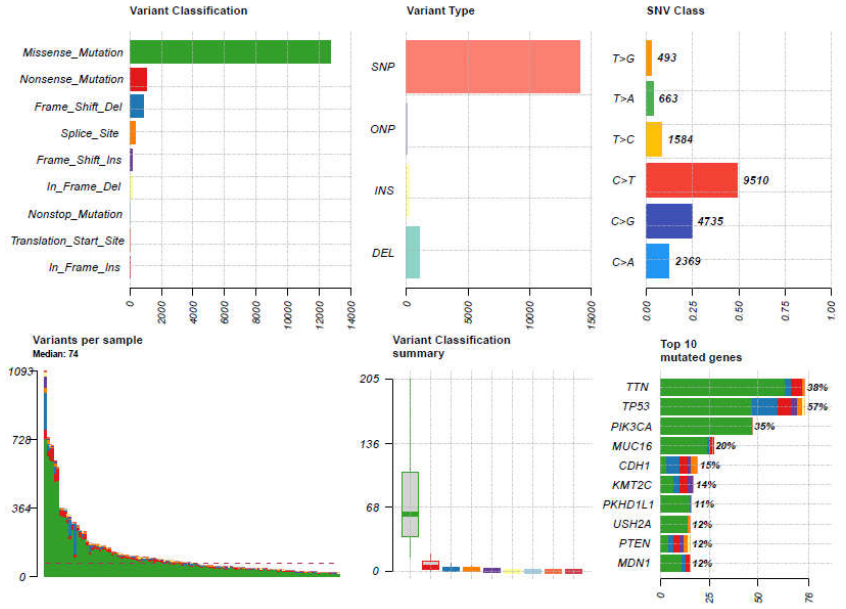
(D)

Figure 5. Discrimination and calibration of SNV-signature in BRCA data. (A) The K-M curve of low risk group and high risk group. (B) Time-dependent AUC of SNV-signature. (C) The 3-, 5- and 8-year AUC of SNV-signature. (D) Calibration plot for 3-, 5- and 8-year of SNV-signature.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

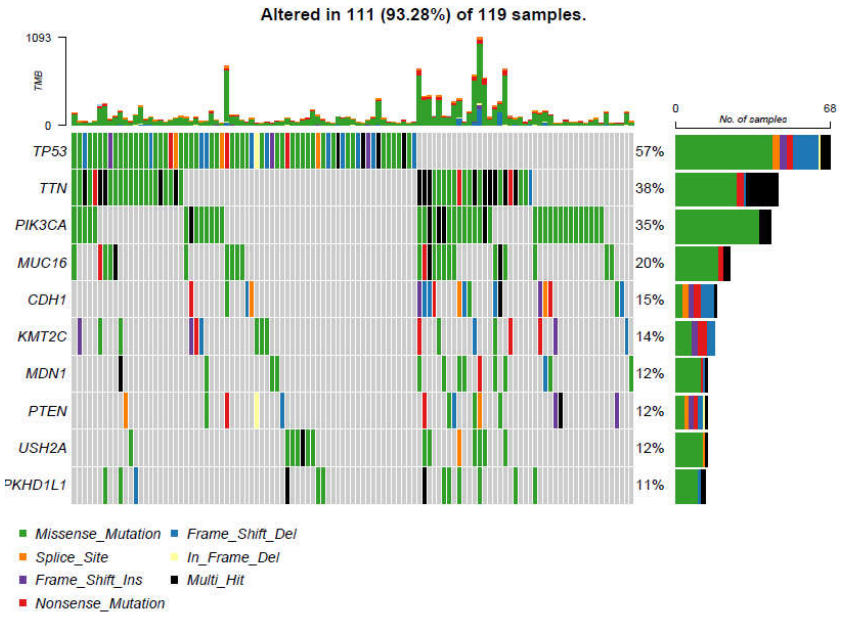


(A)



(B)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1



(C)

Figure 6. The summary of SNVs in two groups in BRCA data. (A) Low risk group. (B) High risk group. (C) The detailed distribution of top 10 mutated genes in high risk group.

3.1.2. RSF for variables screening

RSF is using for prediction and variable selection for right censored survival and competing risk data (Ishwaran, Gerds et al. 2014). 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, as well as grouped variables, can be used to filter variables and to assess variable predictiveness.

We used R package *randomSurvivalForest* to build 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 was shown in the **Figure S9**. Prediction error is measured by 1 - C-index. The estimate of prediction error rate of this model was 0.449 (**Figure S9A**). We selected variables with importance index greater than 0.3 (21 mutant genes) and plotted them in **Figure S9B**. However, we selected the most 100 important variables (See **Table S5**) throwing into the LASSO Cox regression. 23 predictors were left (**Figure S10**). They offered 0.694 (SE = 0.029) of C-index and 0.168 of R square. It was not surprising the C-index and R square were much lower using RSF model when compared to using auto-encoder (where they used similar number of variables: 25 versus 23) because RSF model only selected the most 100 important variables and auto-encoder used the whole information.

3.1.3. Genotype and gene expression

We also performed univariate 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 lowest *P*-value. Then we used the LASSO Cox to select predictors. A total of 60 predictors were left (**Figure S11**). They offered 0.903 (SE = 0.014) of C-index and 0.417 of R square. We drew a Venn plot about 1,936 genotype, 1,936 genes and 60 predictors (**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 (Shao, Wang et al. 2021).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 August 2022 doi:10.20944/preprints202208.0201.v1

3.2. OV data

There were a total of 30,210 SNV items in which 1,650 were nonsense mutation. 406 OV patients with 11,322 genotype data were available. After merging survival data, samples with missing survival data were eliminated and 359 subjects were left. Univariate Cox analysis was performed on these 11,322 genotype data, and 1,089 of them with *P*-value less than 0.05 were selected for subsequent analysis. Then the LASSO Cox regression were applied directly to these data, a total of 95 predictors were selected by the LASSO (See **Figure S13A**). This was also common that the LASSO did select predictors in this scenario that the proportion of zeros takes account for 99.4%. The C-index of the 95 predictors was 0.857 (SE = 0.017) and the R square was 0.791. We also used the auto-coder to extract features from 1,089 binary variables. 19 features were selected from 100 extracted features using the LASSO process (See **Figure S13B**). C-index of the 19 features was 0.777 (SE = 0.019) and R square was 0.443. We thought that although the C-index and R square obtained directly by the LASSO were much higher, the reason was that the predictors were much more, and noise variables selected might be also more, thus there was possibly over-fitting.

4. Discussion

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

SNV is a widely studied type of gene mutation (of which SNP is the most common type, see **Figure S7**), which exists stably in somatic cells and plays a key role in regulating transcriptome expression. However, the giant number of SNVs and the generally tiny size of their effects make it very hard for researchers to detect important genetic factors with a desired statistical significance in a small sample study. What's more, in standard GWAS, the contribution rate of positive SNPs obtained through rigorous variable screening process is often limited. Therefore, aggregating these small effects is a more convincing method and has more promising applications. To best utilize such data in reasoning systems, the feature extraction method may play to more advantages than feature selection method.

Based on these key factors, we identified auto-encoders as a promising approach. Our simulated research shows that the auto-encoder can extract effective information from dichotomous data very well, even in the case of highly sparse variable values. It maps the linear combination of input dichotomous variables to a continuous value space that is lower dimensional by neural networks and activation function. These features can retain most of the original information without the need for worrying about over-fitting issue, because our goal is to get the original information as possible. The use and analysis of these extracted feature information may achieve unexpected results, as compared to highly sparse binary variables, low-dimensional continuous variables are better used. In our proposed process called "SES", considered that the underlying representation is often sparse, we start by sifting through a huge number of variables (screening) to find the ones that might work. Then we do efficient feature extraction by deep learning method (extracting) to make full use of most of the information, while obtaining data types that are easier to analyze. Finally, we use the classical $l1$-norm penalty method to select (selecting)the extracted features and build predictive models. The first step of this process is probably the most time-consuming because the training process of the latter two steps usually only take a few seconds.

Studies have shown that inherited genetic variation is associated with cancer prognosis (Lu, Katsaros et al. 2012, Rafiq, Tapper et al. 2013, Barrdahl, Canzian et al. 2015). However, few studies have used SNV information to predict cancer prognosis in female

 doi:10.20944/preprints202208.0201.v1

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 gained the five-year survival AUC for 0.65 through their 6-gene signature (Mo, Ding et al. 2020). By contrast, our study shows the power of feature extraction using deep learning method. Based on the aggregated SNV information, we can greatly improve the ability to predict patient outcome. In our study, BRCA patients were stratified into low risk group and high risk group based on the SNV-signature. The high risk group had higher TP53 and TTN mutation. TP53 is well known mutated gene and is mutant in 30% of all breast cancers. It is clear that the role of TP53 in the management of breast cancer maters (Shahbandi, Nguyen et al. 2020). TTN-AS1 is a long noncoding RNAs (lncRNA) that binds to titin mRNA (TTN). Many studies have shown that over-expression of TTN-AS1 correlates with poor prognosis in breast cancer and with more advanced pathology (Zheng, Wang et al. 2021). It is not difficult to think of the poor prognosis of breast cancer may cause by TTN mutation.

Furthermore, we searched for studies on SNPs analysis with the auto-encoder in PubMed (Prive, Aschard et al. 2019, Fergus, Montanez et al. 2020, Li, Han et al. 2020, Massi, Gasperoni et al. 2020). The most cutting-edge methods take auto-encoders to extract features from SNP data too (Li, Han et al. 2020). Specifically, the authors applied deep canonically correlated sparse auto-encoder to extract key features from SNPs data and functional magnetic resonance imaging (fMRI) data, and then stacked these features together for classification. Their approach is very interesting and engaging for they addressed the nonlinear dimension reduction and considered the correlation between the above two types of data. The AUC scores of their proposed model for the SNP data were 0.984 and for fMRI data were 0.953, which were the highest AUC scores than other 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 find that sigmoid function was more suitable for feature extraction than ReLU function. And the effect of dichotomous data was better than continuous data. In addition, the data involved in our study were from publicly available databases, so all results are reliable and reproducible (We will provide the R scripts and Python codes on Github).

There is still some limitation. First, although the deep neural network can almost fully extract information of the SNV data, a person's entire sequencing genome is not easy to come by. This may be easy to achieve in the future. Second, it's important to note, however, due to the randomness of parameter initialization, the results of deep neural network training are also random. Therefore, the characteristics obtained from each training are always different, or, random. For example, in the BRCA dataset, each time the auto-encoder was retrained, the features obtained that were used for the LASSO analysis were different, and so the C-index. However, the difference was not apparent, only causing the C-index to move around an interval, say 0.85 to 0.91 (see **Table S6**). Therefore, any training result is feasible in a single test. Of course, 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 deep learning method can work well to extract features by transforming highly sparse dichotomous data (which is a special data type) to lower dimensional continuous data in a non-linear way. This idea may provide benefits in analyzing genome-wide data associated risk prediction issues. We applied this method to cancer prognosis studies which had genome-wide data, and achieved good results.

Author Contributions: Study conception and design: SJJ, LK and ZTT; Data collection and clean: SJJ and LHJ; Real data analysis and interpretation: SJJ, YXH and CJP; Drafting of the manuscript: SJJ, BL and DYF; All authors reviewed the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (81773541), funded from the Priority Academic Program Development of Jiangsu Higher Education Institutions at Soochow University, the State Key Laboratory of Radiation Medicine and Protection (GZK1201919) to ZTT, National Natural Science Foundation of China (81872552, U1967220) to JPC. National Natural Science Foundation of China (82172441), 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.

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).

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

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations:

LASSO: least absolute shrinkage and selection operator

GWAS: genome-wide association

SNPs: single nucleotide polymorphisms

ADNI: Alzheimer's Disease Neuroimaging Initiative

WGS: whole genome sequencing

EDPP: enhanced Dual Polytope Projections

PRS: polygenic risk score

ANNs: artificial neural networks

ReLU: rectified linear unit

KKT: Karush-Kuhn-Tucker

SNR: signal-to-noise

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

CNV: copy number variation

lncRNA: long noncoding RNA

fMRI: functional magnetic resonance imaging

References

Ayers, K. L. and H. J. Cordell (2010). "SNP Selection in Genome-Wide and Candidate Gene Studies via Penalized Logistic Regression." Genetic Epidemiology **34**(8): 879-891.

Barrdahl, M., F. Canzian, S. Lindstrom, I. Shui, A. Black, R. N. Hoover, R. G. Ziegler, J. E. Buring, S. J. Chanock, W. R. Diver, S. M. Gapstur, M. M. Gaudet, G. G. Giles, C. Haiman, B. E. Henderson, S. Hankinson, D. J. Hunter, A. D. Joshi, P. Kraft, I. M. Lee, L. Le Marchand, R. L. Milne, M. C. Southey, W. Willett, M. Gunter, S. Panico, M. Sund, E. Weiderpass, M. J. Sanchez, K. Overvad, L. Dossus, P. H. Peeters, K. T. Khaw, D. Trichopoulos, R. Kaaks and D. Campa (2015). "Association of breast cancer risk loci with breast cancer survival." International Journal of Cancer **137**(12): 2837-2845.

Bengio, Y., A. Courville and P. Vincent (2013). "Representation Learning: A Review and New Perspectives." Ieee Transactions on Pattern Analysis and Machine Intelligence **35**(8): 1798-1828.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). "Greedy layerwise training of deep networks." Advances in Neural Information Processing

Systems 19: PP. 153–160.

Bureau, A., J. Dupuis, K. Falls, K. L. Lunetta, B. Hayward, T. P. Keith and P. Van Eerdewegh (2005). "Identifying SNPs predictive of phenotype using random forests." Genetic Epidemiology **28**(2): 171-182.

El Ghaoui, L., V. Viallon and T. Rabbani (2012). "Safe Feature Elimination in Sparse Supervised Learning." Pacific Journal of Optimization **8**(4): 667-698.

Engebretsen, S. and J. Bohlin (2019). "Statistical predictions with glmnet." Clin Epigenetics **11**(1): 123.

Esteva, A., A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun and J. Dean (2019). "A guide to deep learning in healthcare." Nature Medicine **25**(1): 24-29.

Fergus, P., C. C. Montanez, B. Abdulaimma, P. Lisboa, C. Chalmers and B. Pineles (2020). "Utilizing Deep Learning and Genome Wide Association Studies for Epistatic-Driven Preterm Birth Classification in African-American Women." Ieee-Acm Transactions on Computational Biology and Bioinformatics **17**(2): 668-678.

Gibson, G. (2012). "Rare and common variants: twenty arguments." Nat Rev Genet **13**(2): 135-145.

Guo, P., F. F. Zeng, X. M. Hu, D. M. Zhang, S. M. Zhu, Y. Deng and Y. T. Hao (2015). "Improved Variable Selection Algorithm Using a LASSO-Type Penalty, with an Application to Assessing Hepatitis B Infection Relevant Factors in Community Residents." Plos One **10**(7).

Guo, W., R. C. Elston and X. Zhu (2011). "Evaluation of a LASSO regression approach on the unrelated samples of Genetic Analysis Workshop 17." BMC Proc **5 Suppl 9**: S12.

Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks." Science **313**(5786): 504-507.

Ishwaran, H., T. A. Gerds, U. B. Kogalur, R. D. Moore, S. J. Gange and B. M. Lau (2014). "Random survival forests for competing risks." Biostatistics **15**(4): 757-773.

Jiang, Y., Y. X. He and H. P. Zhang (2016). "Variable Selection With Prior Information for Generalized Linear Models via the Prior LASSO Method." Journal of the American Statistical Association **111**(513): 355-376.

Karczewski, K. J. and M. P. Snyder (2018). "Integrative omics for health and disease." Nat Rev Genet **19**(5): 299-310.

Kriegeskorte, N. and T. Golan (2019). "Neural network models and deep learning." Curr Biol **29**(7): R231-R236.

Lecun, Y., L. Bottou, Y. Bengio and P. Haffner (1998). "Gradient-based learning applied to document recognition." Proceedings of the Ieee **86**(11): 2278-2324.

Li, G., D. P. Han, C. Wang, W. X. Hu, V. D. Calhoun and Y. P. Wang (2020). "Application of deep canonically correlated sparse autoencoder for the classification of schizophrenia." Computer Methods and Programs in Biomedicine **183**.

Long, N., D. Gianola, G. J. Rosa and K. A. Weigel (2011). "Dimension reduction and variable selection for genomic selection: application to predicting milk yield in Holsteins." J Anim Breed Genet **128**(4): 247-257.

Lu, L. G., D. Katsaros, S. T. Mayne, H. A. Risch, C. Benedetto, E. M. Canuto and H. Yu (2012). "Functional study of risk loci of stem cell-associated gene lin-28B and associations with disease survival outcomes in epithelial ovarian cancer." Carcinogenesis **33**(11): 2119-2125.

Manzoni, C., D. A. Kia, J. Vandrovcova, J. Hardy, N. W. Wood, P. A. Lewis and R. Ferrari (2018). "Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences." Briefings in Bioinformatics **19**(2): 286-302.

Massi, M. C., F. Gasperoni, F. Ieva, A. M. Paganoni, P. Zunino, A. Manzoni, N. R. Franco, L. Veldeman, P. Ost, V. Fonteyne, C. J. Talbot, T. Rattay, A. Webb, P. R. Symonds, K. Johnson, M. Lambrecht, K. Haustermans, G. De Meerleer, D. de Ruysscher, B. Vanneste, E. Van Limbergen, A. Choudhury, R. M. Elliott, E. Sperk, C. Herskind, M. R. Veldwijk, B. Avuzzi, T. Giandini, R. Valdagni, A. Cicchetti, D. Azria, M. P. F. Jacquet, B. S. Rosenstein, R. G. Stock, K. Collado, A. Vega, M. E. Aguado-Barrera, P. Calvo, A. M. Dunning, L. Fachal, S. L. Kerns, D. Payne, J. Chang-Claude, P. Seibold, C. M. L. West, T. Rancati and R. Consortium (2020). "A Deep Learning Approach Validates Genetic Risk Factors for Late Toxicity After Prostate Cancer Radiotherapy in a REQUITE Multi-National Cohort." Frontiers in Oncology **10**.

Mo, W. J., Y. Q. Ding, S. Zhao, D. H. Zou and X. W. Ding (2020). "Identification of a 6-gene signature for the survival prediction of breast cancer patients based on integrated multi-omics data analysis." Plos One **15**(11).

Prive, F., H. Aschard and M. G. B. Blum (2019). "Efficient Implementation of Penalized Regression for Genetic Risk Prediction." Genetics **212**(1): 65-74.

Rafiq, S., W. Tapper, A. Collins, S. Khan, I. Politopoulos, S. Gerty, C. Blomqvist, F. J. Couch, H. Nevanlinna, J. J. Liu and D. Eccles (2013). "Identification of Inherited Genetic Variations Influencing Prognosis in Early-Onset Breast Cancer." Cancer Research **73**(6): 1883-1891.

Shahbandi, A., H. D. Nguyen and J. G. Jackson (2020). "TP53 Mutations and Outcomes in Breast Cancer: Reading beyond the Headlines." Trends in Cancer **6**(2): 98-110.

Shao, Z. H., T. Wang, M. Zhang, Z. Jiang, S. P. Huang and P. Zeng (2021). "IUSMMT: Survival mediation analysis of gene expression with multiple DNA methylation exposures and its application to cancers of TCGA." Plos Computational Biology **17**(8).

Tamba, C. L., Y. L. Ni and Y. M. Zhang (2017). "Iterative sure independence screening EM-Bayesian LASSO algorithm for multi-locus genome-wide association studies." PLoS Comput Biol **13**(1): e1005357.

Tan, J., M. Ung, C. Cheng and C. S. Greene (2015). "Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders." Pac Symp Biocomput: 132-143.

Tang, Z., Y. Shen, X. Zhang and N. Yi (2017). "The spike-and-slab lasso Cox model for survival prediction and associated genes detection." Bioinformatics **33**(18): 2799-2807.

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso." Journal of the Royal Statistical Society Series B-Methodological **58**(1): 267-288.

Tibshirani, R., J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor and R. J. Tibshirani (2012). "Strong rules for discarding predictors in lasso-type problems." J R Stat Soc Series B Stat Methodol **74**(2): 245-266.

Tran, K. A., O. Kondrashova, A. Bradley, E. D. Williams, J. V. Pearson and N. Waddell (2021). "Deep learning in cancer diagnosis, prognosis and treatment selection." Genome Medicine **13**(1).

Vincent, P., H. Larochelle, Y. Bengio and P.-A. Manzagol (2008). Extracting and composing robust features with denoising autoencoders.

Wang, J., P. Wonka and J. P. Ye (2015). "Lasso Screening Rules via Dual Polytope Projection." Journal of Machine Learning Research **16**: 1063-1101.

Wray, N. R., M. E. Goddard and P. M. Visscher (2007). "Prediction of individual genetic risk to disease from genome-wide association studies." Genome Res **17**(10): 1520-1528.

Yang, T., J. Wang, Q. Sun, D. P. Hibar, N. Jahanshad, L. Liu, Y. Wang, L. Zhan, P. M. Thompson and J. Ye (2015). "Detecting Genetic Risk Factors for Alzheimer's Disease in Whole Genome Sequence Data via Lasso Screening." Proc IEEE Int Symp Biomed Imaging **2015**: 985-989.

Yi, N. J., Z. X. Tang, X. Y. Zhang and B. Y. Guo (2019). "BhGLM: Bayesian hierarchical GLMs and survival models, with applications to genomics and epidemiology." Bioinformatics **35**(8): 1419-1421.

Zheng, Q. X., J. Wang, X. Y. Gu, C. H. Huang, C. Chen, M. Hong and Z. Chen (2021). "TTN-AS1 as a potential diagnostic and prognostic biomarker for multiple cancers." Biomedicine & Pharmacotherapy **135**.

Zhu, Z. H., F. T. Zhang, H. Hu, A. Bakshi, M. R. Robinson, J. E. Powell, G. W. Montgomery, M. E. Goddard, N. R. Wray, P. M. Visscher and J. Yang (2016). "Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets." Nature Genetics **48**(5): 481-+.

 doi:10.20944/preprints202208.0201.v1

Supplementary Materials

TableS1: The preset 15 non-zero coefficient values for simulation study.

TableS2: The performance of the LASSO in different scenarios.

TableS3: The predicted probability distribution of the true label using the extracted features.

TableS4: The original input data and the extracted features.

TableS5: The ranking of the 100 most important variables using RSF.

TableS6: The influence of randomness of neural network training on results.

FigureS1: The construction of the auto-encoder in MNIST data.

FigureS2: Loss function value and accuracy of the auto-encoder in MNIST training data by the epoch times. (A) Using sigmoid function. (B) Using ReLU function.

FigureS3: The first five image of MNIST training data and testing data. (A) Training data using sigmoid function. (B) Testing data using sigmoid function. (C) Training data using ReLU function. (D) Testing data using ReLU function.

FigureS4: Loss function value and accuracy of the auto-encoder in fashion MNIST training data by the epoch times. (A) Original data using sigmoid function. (B) Corrupted data using sigmoid function.

FigureS5: The first five image of fashion MNIST training data. (A) Original data using sigmoid function. (B) Corrupted data using sigmoid function. (C) Images of original data V.S. corrupted data. The first and third images were original data, the second and fourth images were corrupted data.

FigureS6: Auto-encoder feature selection for highly sparse binary predictors. (A) Loss function value and accuracy of the auto-encoder in simulated data by the epoch times. (B) The first five visualized genetic signal of simulated 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. 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.

FigureS7: The summary of SNVs in BRCA data and OV data. (A) BRCA data. (B) OV data.

FigureS8: The process of the LASSO to directly select predictors using 1,936 genotype data in BRCA.

FigureS9: The process of variables selection using RSF. (A) Error rate by number of trees. (B) 21 variables with importance index greater than 0.3.

FigureS10: The process of the LASSO to select predictors using 100 most important variables selected using RSF in BRCA.

 doi:10.20944/preprints202208.0201.v1

FigureS11: The process of the LASSO to directly select predictors using 1,936 gene expression data in BRCA.

FigureS12: The Venn plot about 1,936 genotype, 1,936 genes and 60 predictors.

FigureS13: The process of the LASSO to select predictors using genotype data in OV. (A) Directly select predictors using 1,089 genotype data. (B) 19 features were selected from 100 extracted features using the LASSO process