A Generative Adversarial Network Model for Disease Gene Prediction With RNA-seq Data

XUE JIANG1, JINGJING ZHAO1, WEI QIAN1, WEICHEN SONG1, AND GUAN NING LIN1,2
1School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai 200030, China
2Shanghai Key Laboratory of Psychotic Disorders, Shanghai Mental Health Center, Shanghai Jiao Tong University School of Medicine, Shanghai 200030, China
Corresponding author: Guan Ning Lin (nickgnlin@sjtu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0909200, in part by the National Natural Science Foundation of China under Grant 81671328 and Grant 81971292, in part by the Program for Professor of Special Appointment (Eastern Scholar) at the Shanghai Institutions of Higher Learning under Grant 1610000043, and in part by the Innovation Research Plan supported by the Shanghai Municipal Education Commission under Grant ZXWF082101.

ABSTRACT Deep learning models often need large amounts of training samples (thousands of training samples) to effectively extract hidden patterns in the data, thus achieving better results. However, in the field of brain-related disease, the omics data obtained by using advanced sequencing technology typically have much fewer patient samples (tens to hundreds of samples). Due to the small sample problem, statistical methods and intelligent machine learning methods have been unable to obtained a convergent gene set when prioritizing biomarkers. Furthermore, mathematical models designed for prioritizing biomarkers perform differently on different datasets. However, the architecture of the generative adversarial network (GAN) can address this bottleneck problem. Through the game between the generator and the discriminator, samples with similar distributions to that of samples in the training set can be generated by the generator, and the prediction accuracy and robustness of the discriminator could be significantly improved. Therefore, in this study, we designed a new generative adversarial network model with a denoising auto-encoder (DAE) as the generator and a multilayer perceptron (MLP) as the discriminator. The prediction residual error was backpropagated to the decoder part of the DAE, modifying the captured probability distribution. Based on this model, we further designed a framework to predict disease genes with RNA-seq data. The deep learning model improves the identification accuracy of disease genes over the-state-of-the-art approaches. An analysis of the experimental results has uncovered new disease-related genes and disease-associated pathways in the brain, which in turn have provided insight into the molecular mechanisms underlying disease phenotypes.

INDEX TERMS Denoising auto-encoder, multilayer perceptron, generative adversarial network, RNA-seq data.

I. INTRODUCTION

With the rapid development and decreasing cost of sequencing technologies, whole-genome sequencing data have become commonly available, providing not only opportunities but also challenges for decoding the pathophysiologic mechanisms of chronic complex diseases from a molecular system level. It is an important research topic in the field of bioinformatics to develop effective and reliable computational tools to screen disease biomarkers and drug targets using omics data.

The associate editor coordinating the review of this manuscript and approving it for publication was Hongbin Chen.
characterization of data in latent space, evaluates the joint probability distribution from the data, and solves the conditional probability as the prediction model [11]. The discriminative model learns the decision function or conditional probability distribution and extracts various abstract features from input data. Compared to the generative model, the discriminative model has been developing rapidly in recent years. Moreover, discriminative models, such as AlexNet [12], VGG-Net [13], ResNet [14], Inception Network [15], etc., often achieve a higher prediction accuracy on large data sets.

Recent advances in deep learning demonstrate promising performance in various fields, including protein structure prediction, natural image analysis, and natural language processing. Deep learning methods often need large amounts of training samples (usually more than thousands) to effectively extract hidden patterns in the data and achieve better results. In the field of biomedicine, biomedical data often have few training samples (tens to hundreds) due to the high-cost, labor-intensive, and time-consuming characteristics of biochemical experiments. However, the problem of small samples usually makes it difficult to screen robust disease biomarkers and results in poor reproducibility of prediction results among different patients. This disadvantage limits the application of deep learning methods in biomedical data. Additionally, it is difficult to design a suitable loss function to learn the potential distribution of the training data. Nevertheless, the architecture of the generative adversarial network (GAN) can address this bottleneck problem [16].

The generator in a GAN can generate samples with similar distributions to that of samples in the training set. Through the game between the unsupervised generator and the supervised discriminator, data that fit the distributions of training samples are generated by the generative model. With the decrease of the loss function, the performance of the generator and discriminator has been significantly improved [17], [18].

Therefore, the architecture of a GAN can be used to solve the bottleneck problem of small samples in the field of bio-medicine, improving the prediction accuracy and robustness of clinically useful biomarkers.

Many studies based on GAN have been used in biomedicine. The generative adversarial network is a deep learning framework, which first puts the generative model and discriminative model into one learning framework [16]. The two models are iteratively and synchronously trained. The training process is accomplished when the two models converge to a Nash equilibrium [19]. During the training process, the discriminative model guides the generative model to learn the probability distribution of training samples, while the false data generated by the generative model help to learn the probability distribution of training samples, while the false data generated by the generative model help to learn the potential distribution of the training data. Shin, et al. designed a method based on a GAN to generate synthetic abnormal MRI images with brain tumors [24]. Experimental results demonstrated the medical images synthesized by the GAN model could improve performance on tumor segmentation.

For gene expression data analysis, the denoising auto-encoder (DAE) is a more suitable choice for the generator. Due to the large amounts of noise, such as missing values, technical variations, sample variations, etc., hidden in the gene expression data, the GAN with a three-layer perceptron as the generator is not suitable for capturing the statistical characteristics of the input data. Nevertheless, the DAE can be seen as a multilayer perceptron that is trained for denoising to some extent. The incomplete learning of the DAE can well learn the intrinsic features of the data and capture the most significant statistical characteristics. Through the incomplete learning representation, the DAE can filter out the noise and capture probability distributions of input data, then reconstruct that original data as high-quality output [25]. On the other hand, the pattern is often lost in the learning process of a GAN, which could lead to the degeneracy of the generator and the generation of data with the same probability distributions, consequently stopping the learning process. However, the incomplete representation of learning of the DAE, which is a means for dimension reduction, can prevent the divergence or overfitting of the generator in the traditional GAN model.

Based on the above discussion, to solve the problem of small sample sizes in the field of biomedicine, we designed a new generative adversarial network model with the DAE as the generator and the MLP as the discriminator (GAN-DAEMLP). First, an MLP with more than three layers can fit arbitrarily complex functions and learn the mapping relationship well between the input data and output labels. The prediction residual error of the discriminator was backpropagated to the decoder part of the DAE, modifying the captured probability distribution. Through the game between the generator and discriminator, the prediction performance of the MLP could be greatly improved. Based on the GAN-DAEMLP model, we further designed an analytical pipeline to predict disease candidate genes with RNA-seq data.

The main contribution of the GAN-DAEMLP can be summarized in three points. First, the DAE is introduced in the role of the generator, which can prevent the overfitting and divergence of the model. Second, the original data instead of randomly generated noise were used as the input of the generator to reduce the training epochs and accelerate the convergence of the model. Third, a framework is designed for disease gene prediction.

Finally, to verify the disease gene prediction performance of GAN-DAEMLP, we conducted various comparative experiments with different hyperparameters and network structures to select the best performing model structure. Then, the best performing GAN-DAEMLP model was further compared with DESeq2 [26], edgeR [27], limma [28], t-test [29], FC [29], MLP, and GAN [16] approaches. The comparison
results showed that GAN-DAEMLP achieves the best performance.

Moreover, disease-related genes prioritized by this model were further analyzed. Nine top-ranked genes were selected to explore the molecular mechanisms under the complex disease phenotypes. Functional annotation and enrichment analysis of the nine genes showed that biological processes, such as calcineurin-NFAT signaling cascade, mitotic cell cycle phase transition, and protein dephosphorylation, were severely affected in the striatum tissue of Huntington’s disease mice.

II. MATERIALS AND METHODS

A. GENE EXPRESSION DATA

We extracted RNA-seq data of the striatum tissue of a Huntington’s disease mouse model from http://www.hdinhd.org. The model included 6-month-old and 10-month-old experimental mice. There were six genotypes for the experimental mice at each stage, including poly Q20, poly Q80, poly Q92, poly Q110, poly Q140, and poly Q175. The detailed description of the samples used in this study is illustrated in Table 1. There were eight samples for each genotype. The modifier genes used in this study were extracted from literature [30], [31], including 89 disease-related genes, and 431 non-disease genes.

| Species | Tissue  | Age                    | Genotype | Sample number |
|---------|---------|------------------------|----------|---------------|
| mouse   | striatum| 6-month-old            | poly Q20 | 8             |
|         |         |                        | poly Q80 | 8             |
|         |         |                        | poly Q92 | 8             |
|         |         |                        | poly Q110| 8             |
|         |         |                        | poly Q140| 8             |
|         |         |                        | poly Q175| 8             |

B. DISCRIMINATOR

The perceptron is a linear model for dichotomy, which cannot classify nonlinear data. Theoretically, a three-layer network can fit arbitrarily complex functions. Therefore, we deepen the hierarchy of the network and use a three-layer perceptron as the discriminator. Forward propagation is used to transfer information from the input layer to the output layer. Binary cross-entropy is used as the loss function, and a ReLU [32] is used as the activation function for the first two layers, and the sigmoid function is used for the output layer. To minimize the loss function, the residual error backpropagation algorithm and stochastic gradient descent are used to update parameters in the network.

C. GENERATOR

To accurately capture the intrinsic features of gene expression data and reconstruct the data with high quality, we use DAEs as the generator [33]–[35]. Let \( X = (x_1, \ldots, x_n) \) represent the gene expression of a sample and \( H = (h_1, \ldots, h_m) \) represent hidden variables. Let \( \tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_n) \) represent the sample with noise data. Let \( f \) represent the function of the encoder and \( g \) represent the function of the decoder. The training objective function of the auto-encoder can be written as

\[
L(X, g(f(X)))
\]

Meanwhile, \( L \) is a loss function, which constrains the distance between \( g(f(X)) \) and \( X \). The training objective function of the denoising auto-encoder can be written as

\[
L(X, g(f(\tilde{X})))
\]

The denoising learning process forces the encoder function \( f \) and decoder function \( g \) to learn the structure of the probability distribution of the input data. Denoising auto-encoders are trained to reconstruct clean data from noise data \( \tilde{X} \). This can be achieved by minimizing the following loss function

\[
L(X, \tilde{X}) = - \log p_{\text{decoder}}(X | H = f(\tilde{X}))
\]

Through the minimization of the reconstruction error, the denoising auto-encoder learns the useful features and reconstructs clean data.

D. THE GENERATIVE ADVERSARIAL NETWORK MODEL

In this study, to improve the prediction accuracy and solve the problem of the small sample size in biomedicine, which limits the application of deep learning methods in biomedicine, we designed a generative adversarial network with an MLP as the discriminator and a DAE as the generator.

The loss function of the MLP is denoted as \( D \), and that of the DAE is denoted as \( G \). Let \( G(X) \) represent the sample generated by the generator. The training process of the novel generative adversarial network is shown below.

Step 1. The noise expression data are taken as the input of the generator \( G \).

Step 2. Through the generator, we obtain \( X_g = G(X) \).

Step 3. Sample the gene expression data and the generated gene expression data as the input of the discriminator.

Step 4. The generated gene expression data are labeled as 0, and the original gene expression data are labeled as 1.

Step 5. Compute the loss of the MLP, and then the loss is backpropagated to the decoder part of the generator.

During the training process, the objective of the discriminator is to distinguish the original gene expression and the generated gene expression, namely, \( D(G(X)) = 0 \), \( D(X) = 1 \). Nevertheless, the objective of the generator is to generate data that more accurately describe the distributions of the real data, namely, \( D(G(X)) = 1 \). In this way, the generator can well learn the statistical characteristics of the real data and filter out noise in the original data.

E. TRAINING

We train the discriminator to maximize the accuracy and train the generator to minimize \( \log(1 - D(G(X))) \).
Therefore, the objective function can be written as
\[
\min_G \max_D V(D, G) = \mathbb{E}_{X \sim p_{data}}[\log D(X)] + \mathbb{E}_{X \sim p_{data}}[\log (1-D(G(X)))]
\] (4)

Since the freedom degree of the GAN is too high, it should be well synchronized between the discriminator and generator, which is hard to balance in the actual training process. We use an alternate iterative strategy to train the DAE and MLP. Moreover, to avoid pattern loss and accelerate the convergence speed, we employ the samples in the training set as the input of both the DAE and MLP and utilize batch normalization [36] during the training process of DAE. The prediction residual error of the MLP is backpropagated to the decoder part of the DAE, modifying the captured probability distribution of input gene expression data for the DAE. Mini-batch random gradient descent training is used to train the networks in both the generator and discriminator. Adam is used as the optimizer with a learning rate of 0.0002 [37]. Until the DAE learns the intrinsic features of the training data, the performance of both the generator and the discriminator are all optimized. Then, the trained discriminator is used to predict the labels of unlabeled samples with gene expression data.

Next, we used disease gene expression data to train GAN-DAEMLP for disease gene prediction and use non-disease gene expression data to train GAN-DAEMLP for the non-disease gene prediction. Let \( s_d(g) \) represent the disease gene prediction score, and \( s_{nd}(g) \) represent the non-disease gene prediction score for gene \( g \). The disease risk score of gene \( g \) can be computed by
\[
s(g) = \frac{s_d(g)}{(s_d(g) + s_{nd}(g))}
\] (5)

In summary, the detailed training process of the model is shown below.

**Algorithm 1 Training for GAN-DAEMLP**

1: For \( i = 1, 2, \ldots, n \) do
2:     For DAE do
3:       Sample mini-batch of samples from original data prior
4:       Update the DAE by descending stochastic gradient of a loss function
5:     End for
6:     For MLP do
7:       Sample mini-batch of samples from original data prior
8:       Sample mini-batch of samples from data generated by DAE
9:       Update the MLP by descending stochastic gradient of a loss function
10:      Update the decoder part of DAE using gradients introduced from MLP
11:     End for
12:     End for

**F. DISEASE GENE PREDICTION**

The framework of GAN-DAEMLP for disease gene prediction is shown in Fig. 1. Based on the GAN-DAEMLP, we designed a pipeline to prioritize disease genes. The details of the pipeline are illustrated in Fig. 2. Finally, we ranked genes in descending order according to their disease risk scores. Top ranking genes are considered to be the most likely disease genes.
may develop the disease with mild symptoms. The more CAG repeats, the earlier the disease occurs, and the more severe the symptoms are. The impacted genes are confirmed by gene knockdown technology. If the symptoms become more serious when one gene is knocked-down, the gene is considered to be disease-related.

We used gene expression of the Huntington mouse model from brain samples of 6-month-old and 10-month-old mice with genotypes of poly Q80, poly Q92, poly Q110, poly Q140, and poly Q175 to train GAN-DAEMLP for disease gene prediction, and used non-disease gene expression of samples of 6-month-old and 10-month-old mice with those genotypes to train GAN-DAEMLP for non-disease gene prediction.

A. PERFORMANCE OF GAN-DAEMLP
To verify the performance of GAN-DAEMLP, we conducted various experiments with different hyperparameters and different hidden layers of the generator. The GAN-DAEMLP with two hidden layers in the decoder part of the DAE was denoted as mode 1, that with four hidden layers in the decoder part of the DAE decoder was denoted as mode 2, and that with eight hidden layers in the decoder part of the DAE was denoted as mode 3. We also tried different batch sizes and training epochs. In total, three model structures, three different batch sizes, and eight different epochs have been used to test the performance of GAN-DAEMLP. The AUCs and AUPRs of GAN-DAEMLP for disease gene prediction with different parameters are shown in Table 2 and Table 3, respectively. We observed that the prediction accuracy of GAN-DAEMLP was increased slightly with the increase of epochs. Comparatively, batch size had little effect on the prediction accuracy. We observed that GAN-DAEMLP, with four hidden layers in the DAE decoder, 30 batch sizes, and 1200 epochs performed best. The best training parameters for the MLP and DAE in this GAN model are shown in Table 4 and Table 5, respectively.

| TABLE 2. The AUCs of ROC curves using different parameters. |
|---|
| Epoch | 50 | 100 | 200 | 400 | 600 | 800 | 1000 | 1200 |
| Batch size | |
| Mode 1 | 10 | 0.44 | 0.45 | 0.45 | 0.50 | **0.54** | 0.52 | 0.54 |
| | 30 | 0.45 | 0.45 | 0.45 | 0.49 | 0.53 | 0.53 | 0.54 |
| | 60 | 0.45 | 0.44 | 0.45 | 0.50 | 0.52 | 0.53 | 0.54 |
| | 10 | 0.45 | 0.46 | 0.44 | 0.50 | 0.52 | **0.55** | 0.53 |
| Mode 2 | 30 | 0.45 | 0.45 | 0.44 | 0.51 | 0.54 | 0.53 | **0.58** |
| | 60 | 0.45 | 0.45 | 0.45 | 0.54 | **0.55** | 0.54 | 0.53 |
| | 10 | 0.45 | 0.43 | 0.46 | 0.50 | **0.55** | 0.54 | 0.48 |
| Mode 3 | 30 | 0.45 | 0.46 | 0.47 | 0.50 | 0.54 | **0.55** | 0.53 |
| | 60 | 0.45 | 0.44 | 0.45 | 0.49 | 0.53 | **0.54** | 0.52 |

To test the prediction accuracy of GAN-DAEMLP, we selected the best performing model parameters to conduct a further comparison. The receiver operating characteristic (ROC) curves and precision-recall curves are shown in Fig. 3 and Fig. 4, respectively. We further used the prediction results of GAN-DAEMLP with the best parameter setting, i.e., epochs = 1200, batch size = 30, to conduct a comparison with the state-of-the-art approaches.

| TABLE 3. The AUPRs of precision-recall curves using different parameters. |
|---|
| Epoch | 50 | 100 | 200 | 400 | 600 | 800 | 1000 | 1200 |
| Batch size | |
| Mode 1 | 10 | 0.14 | 0.14 | 0.14 | 0.14 | 0.17 | **0.18** | 0.17 | 0.18 |
| | 30 | 0.14 | 0.14 | 0.14 | 0.14 | 0.18 | **0.18** | 0.15 | 0.17 | 0.18 |
| | 60 | 0.14 | 0.14 | 0.14 | 0.18 | 0.18 | 0.18 | 0.17 | 0.18 | 0.18 |
| | 10 | 0.14 | 0.14 | 0.14 | 0.17 | 0.17 | **0.19** | 0.17 | 0.17 | 0.17 |
| Mode 2 | 30 | 0.14 | 0.14 | 0.14 | 0.17 | 0.18 | 0.17 | 0.18 | 0.18 | **0.21** |
| | 60 | 0.14 | 0.14 | 0.14 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 |
| | 10 | 0.14 | 0.14 | 0.14 | 0.17 | 0.17 | **0.19** | 0.18 | 0.18 | 0.18 |
| Mode 3 | 30 | 0.14 | 0.14 | 0.14 | 0.16 | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 |
| | 60 | 0.14 | 0.14 | 0.14 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | **0.19** |

| TABLE 4. Model training parameters for MLP. |
|---|
| Item | Value |
| Input layer | |
| Activation function for hidden layer | ReLU |
| Hidden layer | |
| Activation function for hidden layer | ReLU |
| Output layer | |
| Activation function for the Output layer | Sigmoid |
| Training batch size | 30 |
| Training epochs | 1200 |
| Optimizer | Adam |
| Learning rate | 0.0002 |
| Decay | 0.9 |
| Loss function | Binary cross-entropy |
TABLE 5. Model training parameters for DAE.

| Item                        | Value          |
|-----------------------------|----------------|
| Encoder                     |                |
| Input layer                 |                |
| Activation function for the hidden layer | ReLU           |
| Hidden layer                |                |
| Activation function for the hidden layer | ReLU           |
| Decoder                     |                |
| Input layer                 |                |
| Activation function for the hidden layer | ReLU           |
| Hidden layer                |                |
| Activation function for the hidden layer | ReLU           |
| Hidden layer                |                |
| Activation function for the hidden layer | softmax        |
| Output layer                |                |
| Training batch size         | 30             |
| Training epochs             | 1200           |
| Optimizer                   | Adam           |
| Learning rate               | 0.0002         |
| Decay                       | 0.9            |
| Loss function               | Mean squared error |

**FIGURE 3.** The ROC curves of GAN-DAEMLP with different parameters.

**FIGURE 4.** The precision-recall curves of GAN-DAEMLP with different parameters.

**FIGURE 5.** The ROC curves of t-test, FC, DESeq2, edgeR, limma, MLP, GAN, and GAN-DAEMLP.

**FIGURE 6.** The precision-recall curves of t-test, FC, DESeq2, edgeR, limma, MLP, GAN, and GAN-DAEMLP.

B. COMPARISON OF GAN-DAEMLP WITH THE STATE-OF-THE-ART APPROACHES

We conducted large amounts of experiments on gene expression data of the Huntington’s disease mice. To investigate the effectiveness of GAN-DAEMLP proposed in this study, we also conducted experiments with the t-test, FC, DESeq2, edgeR, limma, MLP, and GAN. The comparison results of the eight methods are shown in Fig. 5 and Fig. 6. It is clearly shown that GAN-DAEMLP performed best compared with other approaches. Moreover, GAN-DAEMLP and MLP had a high prediction accuracy for top ranking genes. Although the training accuracy of MLP was high, it was hard to prioritize disease genes (Fig. 6).

To make the effectiveness of the proposed method more convincing, we compared the results of GAN-DAEMLP with that of the state-of-the-art methods by using a t-test. The p-value of AUCs is 4.08e-06 and that of AUPRs is 1.18e-06, indicating the prominent advantage of the proposed method.

To address this difficulty, it is better to devise a robust differentially expressed gene set by intersecting the top ranking 1000 genes in the ranking lists obtained by the eight approaches, which could be considered as the most related to the disease. Finally, TRAIP, Bsgnt2, Ugt8a, PPP3CA, PMEPA1, RGS4, PPP3R1, CHN1, and ST8SIA3 were selected. We drew a clustered heatmap of the nine feature
genes using the expression under all samples, see Fig. 7. We observed that samples of poly Q20 were clustered using the nine feature genes, and the nine feature genes were in turn clustered into a single category.

To computationally verify the effectiveness of the nine biomarkers, we conducted genotype classification based on a support vector machine. The classification accuracy of the control genotype (poly Q < 80) from the case genotype (poly Q ≥ 80) by using all the disease genes in the training set was 0.759, while the classification accuracy was 0.827 by using the nine feature genes. It indicated that the expression of the nine genes was changed obviously with the repeat elongation in the huntingtin gene.

To acquire a deep insight into the dynamic molecular mechanism and the pathological mechanism underlying complicated clinical disease phenotypes, we performed gene ontology analysis and enrichment analysis.

C. GENE ONTOLOGY ANALYSIS OF DISEASE RISK GENES

We used the annotations from PsyMuKB (http://www.psymukb.net/) to annotate the nine genes. Seven of them are protein-coding genes. Meanwhile, TRAIP (TRAF Interacting Protein) locates at 3p21.31. It has been reported that Seckel syndrome 9 and Seckel syndrome are associated with TRAIP. Gene ontology annotations related to this gene include ligase activity and obsolete signal transducer activity, downstream of the receptor. PPP3CA (protein phosphatase 3 catalytic subunit alpha) locates at 4q24. Epileptic encephalopathy, infantile or early childhood 1, and arthrogryposis, cleft palate, craniosynostosis, and impaired intellectual development are associated with the gene. Gene ontology annotations for the gene are calcium ion binding and enzyme binding. PMEPA1 (prostate transmembrane protein) locates at 20q13.

31, which functions as WW domain binding and R-SMAD binding. RGS4 (regulator of G protein signaling 4) locate at 1q23.3. Schizophrenia and psychotic disorder are associated with RGS4 [39]. Gene ontology annotations related to this gene include GTPase activity and G protein alpha-subunit binding. PPP3R1 (protein phosphatase 3 regulatory subunit B, alpha) locates at 2p14. Diseases associated with PPP3R1 include extracranial neuroblastoma and cervical neuroblastoma. Gene ontology annotations related to this gene include calcium ion binding and calmodulin binding. It has been reported that PPP3R1 is related to a higher cerebrospinal fluid tau level. Moreover, single-nucleotide polymorphisms (SNPs) located in the gene regulatory subunit of PPP3R1 (rs1868402). Our study showed PPP3R1 is also a critical differentially expressed gene, consistent with previous research. CHN1 (chimerin 1) locates at 2q31.1. Duane Retraction Syndrome 2 and Duane Syndrome Type 2 are associated with CHN1 [40]. Gene ontology annotations related to this gene include GTPase activator and ephrin receptor binding. ST8SIA3 (ST8 alpha-N-acetyl-neuraminide alpha-2,8-sialyltransferase 3) locates at 18q21.31. Gene ontology annotations for this gene are sialyltransferase activity and alpha-N-acetylneuraminate alpha-2,8-sialyltransferase activity.

Collectively, the descriptions of these genes indicated they are indeed disease risk genes and potential candidates for Huntington disease.

D. PATHOMECHANISM ANALYSIS

Due to the clinical heterogeneity of the chronic complex neurodegenerative disease, it is imperative and necessary to conduct pathway analysis to identify key pathways associated with the disease.
We further annotated the seven genes using GeneAnalytics (https://ga.genecards.org/). The seven genes mainly expressed in globus pallidus, caudate nucleus, cerebral cortex, subthalamic nucleus, pons, HyStem +TGFBeta3+GDF5-induced SK11 cells, and medulla oblongata in the brain. Moreover, it has been reported that Seckel syndrome 9, epileptic encephalopathy, infantile or early childhood 1, arthrogryposis, cleft palate, craniosynostosis, and impaired intellectual development, Duane retraction syndrome 2, Duane syndrome type 2, Duane retraction syndrome, Seckel syndrome, undetermined early-onset epileptic encephalopathy, and autosomal dominant non-syndromic intellectual disability are associated with these seven genes.

GO terms for the seven genes include calcium-dependent protein serine/threonine phosphatase activity, calcineurin complex, calcineurin-NFAT signaling cascade, cyclosporine a binding, calcmodulin binding, response to amphetamine, mnt signaling pathway, and calcium modulating pathway.

Pathways associated with the seven genes include DARPP-32 phosphorylation, G-AlphaQ signaling, NNOS signaling at neuronal synapses, Nur77 signaling in T cells, initiation of transcription and translation elongation at the HIV-1 LTR, MAPK-Erk pathway and tacrolimus/cyclosporine pathway.

IV. DISCUSSION AND CONCLUSION

Currently, applications of machine learning and deep learning methods are becoming ubiquitous in the field of biomedicine to screen disease-related genes. Furthermore, the accumulation of the omics data and single-cell sequencing data have greatly promoted the development of statistical machine learning methods, evolutionary methods, and deep learning methods. Our understanding of the molecular mechanisms of complicated disease can be further deepened with the accumulation of large amounts of omics data. Additionally, the prediction accuracy can be significantly improved with the advanced algorithms.

However, in the field of biomedicine and bioinformatics, samples with labels are rare, which significantly limits the application of deep learning methods. To address this problem, we developed a generative adversarial network model. Through the game between the generator and discriminator, the prediction performance of the discriminator is greatly improved. Moreover, the denoising auto-encoder was capable of filtering out noise in gene expression data and deepened our insights into the distribution of gene expression data in the latent space.

Finally, nine genes were selected and considered to be associated with Huntington’s disease. Gene functional analysis and pathway analysis demonstrated that neurodegenerative disorders are comorbidities associated with many other neuropsychiatric disorders.

ACKNOWLEDGMENT
(Xue Jiang and Jingjing Zhao contributed equally to this work.)

REFERENCES
[1] P. D. McNicholas and T. B. Murphy, “Model-based clustering of microarray expression data via latent Gaussian mixture models,” Bioinformatics, vol. 26, no. 21, pp. 2705–2712, Aug. 2010.
[2] A. Krogh, B. Larsson, G. von Heijne, and E. L. L. Sonnhammer, “Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes,” J. Mol. Biol., vol. 305, no. 3, pp. 567–580, 2001.
[3] X. Jiang, H. Zhang, F. Duan, and X. Qian, “Identify Huntington’s disease associated genes based on restricted Boltzmann machine with RNA-seq data,” BMC Bioinf., vol. 18, no. 1, pp. 439–447, Oct. 2017.
[4] J. Ngiam, A. Khosla, M. Kim, J. Nam, and A. Y. Ng, “Multimodal deep learning,” in Proc. 28th Int. Conf. Mach. Learn. (ICML), Bellevue, WA, USA, Jun./Jul. 2011, pp. 1–8.
[5] Y. Zhang and Z. Lu, “Exploring semi-supervised variational autoencoders for biomedical relation extraction,” Methods, vol. 166, pp. 112–119, Aug. 2019.
[6] O. O. Aalen, “Further results on the non-parametric linear regression model in survival analysis,” Statist. Med., vol. 12, no. 17, pp. 1599–1588, Sep. 1993.
[7] E. I. Altman, G. Marco, and F. Varetto, “Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience),” J. Banking Finance, vol. 18, no. 3, pp. 505–529, May 1994.
[8] A. Guo, T. Hastie, and R. Tibshirani, “Regularized linear discriminant analysis and its application in microarrays,” Biostatistics, vol. 8, no. 1, pp. 86–100, Apr. 2006.
[9] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, “Support vector machine classification and validation of cancer tissue samples using microarray expression data,” Bioinformatics, vol. 16, no. 10, pp. 906–914, Oct. 2000.
[10] M. L. Vaughn, S. J. Taylor, M. A. Foy, A. J. B. Fogg, and S. J. Cavill, “Direct knowledge discovery and interpretation from a multilayer perception network which performs low-back-pain classification,” in Knowledge Discovery and Data Mining, 1999, pp. 160–179, doi: 10.1049/PBPC001E.
[11] M. Liang, Z. Li, T. Chen, and J. Zeng, “Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 12, no. 4, pp. 928–937, Jul. 2015.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017.
[13] K. Simdynan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015.
[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
[16] I. J. Goodfellow, J. Pouget-Abadie, and M. Mirza, “Generative adversarial nets,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[17] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-rolling using a conditional generative adversarial network,” IEEE Trans. Circuits Syst. Video Technol., to be published.
[18] M. Ozery-Flato, P. Thodoroff, M. Ninio, M. Rosen-Zvi, and T. El-Hay, “Adversarial balancing for causal inference,” Oct. 2018, arXiv:1810.07406. [Online]. Available: https://arxiv.org/abs/1810.07406
[19] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in Proc. Neural Inf. Process. Syst., 2017, pp. 6626–6637.
[20] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?” Jan. 2018, arXiv:1801.04406. [Online]. Available: https://arxiv.org/abs/1801.04406
[21] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” IEEE Trans. Pattern Anal. Mach. Intell., to be published.
[22] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing training of generative adversarial networks through regularization,” May 201, arXiv:1705.09367. [Online]. Available: https://arxiv.org/abs/1705.09367
[23] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Synthetic data augmentation using GAN for improved liver lesion classification,” in Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI), Apr. 2018, pp. 289–293.

[24] H.-C. Shin, N. A. Tenenholz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andrieo, and M. Michalski, “Medical image synthesis for data augmentation and anonymization using generative adversarial networks,” in Proc. Int. Workshop Simul. Synth. Med. Imag., 2018, pp. 1–11.

[25] J. Pan, C. Canton Ferrer, K. McGuinness, N. E. O’Connor, J. Torres, E. Sayrol, and X. Giro-i-Nieto, “SalGAN: Visual saliency prediction with generative adversarial networks,” 2017, arXiv:1701.01081.[Online]. Available: http://arxiv.org/abs/1701.01081

[26] M. D. Robinson and G. K. Smyth, “Moderated statistical tests for assessing differences in tag abundance,” Bioinformatics, vol. 23, no. 21, pp. 2881–2887, Sep. 2007.

[27] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, “EdgeR: A biocomputor package for differential expression analysis of digital gene expression data,” Bioinformatics, vol. 26, no. 1, pp. 139–140, Nov. 2009.

[28] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, “LIMMA powers differential expression analyses for RNA-sequencing and microarray studies,” Nucleic Acids Res., vol. 43, no. 7, p. e47, Jan. 2015.

[29] F. Hong and R. Breitling, “Auto-encoding variational bayes,” Bioinformatics, vol. 24, no. 3, pp. 374–382, 2008.

[30] P. Langfelder et al., “Integrated genomics and proteomics define huntingtin CAG length–dependent networks in mice,” Nature Neurosci., vol. 19, no. 4, pp. 623–633, Feb. 2016.

[31] G. Alain and Y. Bengio, “What regularized auto-encoders learn from the data generating distribution,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 3563–3593, 2012.

[32] Y. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in Proc. Int. Conf. Mach. Learn., 2010, pp. 807–814.

[33] Y. Bengio, L. Yao, G. Alain, and P. Vincent, “Generalized denoising auto-encoders as generative models,” 2013, arXiv:1305.6663.[Online]. Available: http://arxiv.org/abs/1305.6663

[34] A. Deblan, L. Jamone, A. R. Kanepff, and J. Santos-Vctor, “Denoising auto-encoders for learning of objects and tools affordances in continuous space,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2016, pp. 4866–4871.

[35] D. P. Kingma, “Fast gradient-based inference with generative adversarial networks,” 2017, arXiv:1701.01081.[Online]. Available: http://arxiv.org/abs/1701.01081

[36] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, arXiv:1502.03167.[Online]. Available: http://arxiv.org/abs/1502.03167

[37] N. Lodatko and R. Maiboroda, “An adaptive moment estimator of a parameter of a distribution constructed from observations with admixture,” Theory Probab. Math. Statist., vol. 75, pp. 71–83, Jan. 2008.

[38] C. A. Ross, E. H. Aylward, E. J. Wild, D. R. Langbehn, J. D. Long, J. H. Warner, R. I. Scabill, B. R. Leavitt, J. C. Stout, J. S. Paulsen, R. Reilmann, P. G. Unschuld, A. Wexler, R. L. Margolis, and S. J. Tabrizi, “Huntington disease: Natural history, biomarkers and prospects for therapy,” Nature Rev. Neurol., vol. 10, no. 4, pp. 204–216, Mar. 2014.

[39] D. W. Morris, A. Rodgers, and K. A. Mcghee, “Confirming RGS4 as a susceptibility gene for schizophrenia,” Amer. J. Med. Genet., vol. 125, no. 1, pp. 50–53, 2004.

[40] N. Miyake, J. K. Chilton, and M. Psatha, “Human CHRNA mutations hyper-activate α2-chimaerin and cause Duane’s retraction syndrome,” Science, vol. 321, no. 5890, pp. 839–843, 2008.