Variational Autoencoder for Anti-Cancer Drug Response Prediction

Hongyuan Dong *
School of Computer Science and Technology
Harbin Institute of Technology at Weihai
170400305@stu.hit.edu.cn

Jiaqing Xie *
School of Engineering
University of Edinburgh
s2001696@ed.ac.uk

Zhi Jing
School of Data and Computer Science
Sun Yat-sen University
jingzh5@mail2.sysu.edu.cn

Dexin Ren
School of Computer Science
University of Arizona
dexinren@email.arizona.edu

Abstract

Cancer has long been a main cause of human death, and the discovery of new drugs and the customization of cancer therapy have puzzled people for a long time. In order to facilitate the discovery of new anti-cancer drugs and the customization of treatment strategy, we seek to predict the response of different anti-cancer drugs with variational autoencoders (VAE) and multi-layer perceptron (MLP). We incorporate two kinds of essential information into our model, which are gene expression data of cancer cell lines and drug molecular data. Our model encode these data with GENEVAE model, which is an ordinary VAE, and rectified junction tree variational autoencoder[1] (JTVAE) model, respectively, and process the encoded features with a Multi-layer Perceptron (MLP) model to produce a final prediction. We reach an average coefficient of determination ($R^2 = 0.83$) in predicting drug response on breast cancer cell lines and an average $R^2 > 0.84$ on pan-cancer cell lines. Additionally, we show that our model can generate unseen effective drug compounds for specific cancer cell lines. Finally, we further explore the latent representations encoded by geneVAE and JTVAE model. Results show that these models are robust and can preserve critical features of original data.

1 Introduction

The discovery of new drugs and the customization of cancer therapy have puzzled people for a long time. Anti-cancer drugs are widely used and of great significance in the therapy of cancer treatment nowadays. However, it is extremely expensive and time-consuming generation of anti-cancer drugs, and because of the diversity of cancer gene features and drug molecular features, it's considerably difficult to customize therapy strategy for patients. In order to facilitate the discovery of new anti-cancer drugs and the selection of drugs to provide personalized treatment strategy, a generative model for accurate prediction of anti-cancer drug response is in an urgent need.

Evidence shows that the responses of anti-cancer drugs are highly relevant with cancer genomic and transcriptomic profile [2]. Some researchers worked on predicting drug response with gene expression data: Chiu et al. built deep neural networks to combine gene expression with mutation profiles to make predictions [3], Geeleher et al. implemented a ridge regression model on before-treatment gene expression data to predict response of chemotherapy [4]. In order to predict the response of

*These authors contributed equally to this work

Preprint. Under review.

different drugs used on different cancer cell lines, we incorporate both gene expression data and drug molecular data into our model.

Many ways are leading to extract representative features from unlabeled data. According to supervised learning methods, random forest can list the importance of each gene, which assists us to filter genes at the very first step. However it will encounter problems when meeting data with no labels, such as the Cancer Cell Line Encyclopedia (CCLE) dataset that we want to explore. Some unsupervised learning methods, such as principal component analysis (PCA) , independent component analysis (ICA) and manifold learning based t-distributed stochastic neighbor embedding (TSNE) are common in analyzing medical data where feature numbers are numerous and features are unlabeled. However, they are primarily for 2D visualization in most cases and might lose many important information when compressing data into high dimensional latent features. If situated in a higher dimension, they do not perform well at all and we cannot judge its performance intuitively by visualizing latent space [5].

To extract features from a huge amount of unlabeled data, we take advantage of variational autoencoder (VAE)[6], which has achieved great success in the field of unsupervised learning of complex probability distribution. VAE could not only capture probabilistic distribution of latent features and enable more complete analysis of cancer gene expression data, but also serve as a generative model to facilitate the development of new . An ordinary VAE model (GeneVAE) whose encoder and decoder are both composed of 2-layer neural network is implemented for gene expression profile of cancer cell lines. As for anti-cancer drugs, we adopt junction tree VAE (JTVAE) [1] model, transforming their molecular graph into junction trees by functional group split to extract their low dimensional features. JTVAE analyzes molecular graphs by functional groups, enabling a more complete understanding of molecules' inborn features. Unlike other deep learning methods such as Graph Convolutional Network (GCN) [7] which could only encode the input data, JTVAE also serves as a generative model, and outperforms many previous work [8, 9, 10] in reconstructing molecules. The excellent ability of JTVAE to always generate valid compounds makes it extremely potential in new anti-cancer drug discovery. In this work, we show that encoded features of drugs could be randomly sampled, and we could select the well-performing features and decode them with JTVAE model, providing a large number of valid compounds which are potentially effective for cancer therapy.

After encoding gene expression data and drug molecular data with geneVAE and JTVAE respectively, we implement a fully connected neural network to combine the extracted features to produce the final result, namely IC_{50} value of the input anti-cancer drug used against the input cancer cell line. Unlike some former works where models are restricted in specific drugs [11, 12, 13], our model could take as input merely any organic compounds to predict their responses. Moreover, combined with JTVAE, it could serve to generate valid organic compounds which are potentially effective in cancer therapy. It is promising to reduce the cost of developing new drugs.

Additional dataset is also incorporated in our work to improve the performance. As researches on cancer are going much deeper than ever before, various kinds of information are available to make a more accurate prediction of anti-cancer drug response. For example, Cancer Genomic Census (CGC) [14] dataset, containing a number of genes highly relevant with cancer, could be used to curate a gene subset from cancer gene data, removing a significant amount of useless information. In this work, we use CGC dataset to filter out a representative gene subset from original gene expression data, and compare the prediction results with those of models where CGC dataset is not incorporated.

Present work Our present work focuses on learning latent embeddings of original data with variational autoencoders (VAE) and using latent vectors to accomplish drug response prediction and effective drug generation. We choose gene expression level as cancer gene data and SMILES representation as drug molecular data. Our model includes an ordinary VAE for cancer gene data input (shown in 1) and a JTVAE for drug molecular sequence data input (shown in 2) . The final drug response model is based on deep neural network, producing $ln(IC_{50})$ value as final prediction (shown in 3) . We choose coefficient of determination metrics(R^2) and root mean squared error($RMSE$) as metrics to evaluate our drug response prediction. The datasets that we adopt are Cancer Cell Line Encyclopedia(CCLE) [15] gene expression dataset, Cancer Gene Census (CGC) [14] dataset, ZINC molecular structure dataset and GDSC drug response dataset [2]. We use our model on breast cancer cell lines at first and then test it on pan-cancer cell lines. We also show that our model could effectively generate chemical compounds which are potentially effective in cancer therapy. Thereafter, we explore the latent representations encoded by geneVAE and JTVAE, to prove the robustness of

our model. While our present work is based on VAE, more ideas can be found in our future work part that we will attempt to realize.

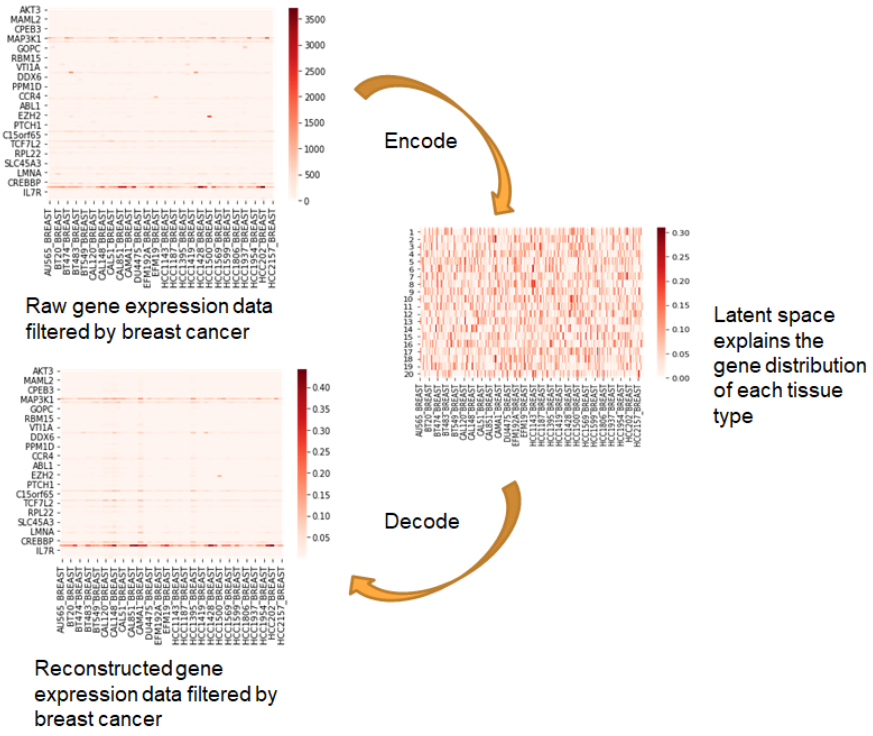

Figure 1: Variational autoencoders on gene expression data. The first 20 genes in each breast cancer cell line is shown.

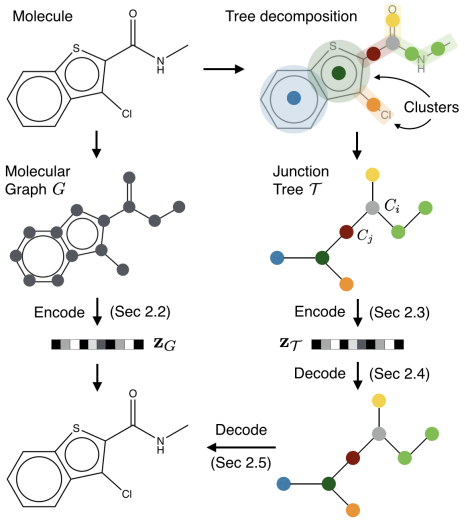

Figure 2: Junction tree VAE on drug molecular data [1]. On the left is graph encoder serves to encode molecular graph while preserving fine-grained connectivity information. On the right is junction tree encoder which splits the molecular by functional groups and encode the molecular junction tree.

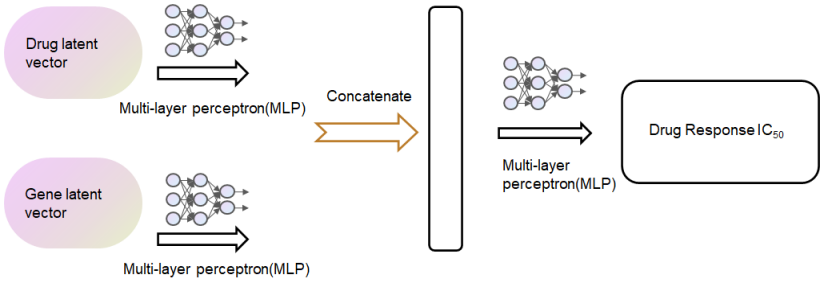

Figure 3: MLP model to produce a final prediction. Two 3-layer MLPs post-process the encoded gene latent vectors and drug latent vector respectively, and then another 3-layer MLP concatenate the output and produce a predicted IC_{50} value.

2 Related work

Dimensionality reduction on features Reducing feature numbers or encoding features into lower dimensions is common in projects which use feature engineering to make predictions or analyze clustering effects. Supervised learning methods can help select gene subsets which are the most related to the research task, such as random forest with feature importance about gene in RNA sequence case-control studies [16] and support vector machines (SVM) with double RBF-kernels to filter irrelevant gene features[17]. Unsupervised learning methods, such as principal component analysis(PCA) and hierarchical learning can help explain the genes' group features and use certain PCs or hierarchical relationship in a lower dimension mapping space [18]. However, due to the weakness of losing important information when compressing original data into latent features, we only adopt these methods for visualization to explain the robustness of our model.

Variational auto-encoders on gene expression profile A plethora of works have been done on encoding important features from gene expression data. The core idea behind feature extraction is how to learn latent vectors effectively from input embedding. Usually, multi-layer perceptrons can avoid the curse of dimensionality and simply encode gene features from the input layer: Chang et al. use convolutional neural network to encode gene mutation and drug molecular data [19], Oskooei and Born et al. implement attention-based neural networks to produce explainable encoded features [20]. An encoder-decoder structure [3] extends a multi-layer perceptron, which considers more about the reconstruction of original data. The bottleneck layer represents latent information from this kind of autoencoder. Recently, variational autoencoder(VAE)[6] has appeared frequently in pre-trained models that encode gene expression data: Gronbech et al. use VAE to estimate expected gene expression level [21], Rampášek et al. implement VAE models to analyze before-treatment and after-treatment gene expression profile [22, 23]. We take advantage of VAE model to process gene expression data. We implement a deep-neural-network based VAE to form a pre-trained encoder, which will be fit into the combined MLP drug prediction network together with JTVAE model.

Representation learning on graph for drug molecular features Graph features can be encoded by deep learning methods, such as convolutional neural network(CNN) [24], recurrent neural network(RNN) [25] and message passing neural network(MPNN) [26]. Besides, variational autoencoder(VAE) is also widely used in graph generation and graph encoders: Kusner et al. propose grammar based methods and use parse trees to produce more valid generated output [8], Simonovsky et al. label the nodes and bonds in molecules to form a graph structure and apply VAE model on it [10], Li et al. also use graph-structured VAE model to generate molecules matching the statistics of the original dataset [9]. In order to avoid generating nodes one by one, which is often of non-sense in drug design, Jin et al. propose a method that combined tree encoder with graph encoder [1]. It treats functional groups as nodes for broadcasting. Also, attention mechanism, which is frequently used in natural language processing and computer vision tasks, could be used with RNN and CNN models [27, 20] to encode drug molecular data, learning attention weights by multihead-attention or self-attention to produce explainable encoded features. Among all these studies, we choose junction tree variational autoencoder (JTVAE) as our pre-trained model to encode drug structures.

Drug response prediction methods Supervised learning methods are useful to predict drug response with encoded information. Support vector regression(SVR) and random forest regressor are basic algorithms to perform regression. Recently, deep neural network methods have been popular in the drug prediction network: Chiu et al. build deep neural networks to analyze gene expression and mutation profiles to make prediction [3], Chang et al. use convolutional neural network based methods on gene mutation profile and drug molecular data [19], Liu et al. also use gene mutation data and drug molecular data, but take advantage of Graph Comvolutional Network (GCN) [7], Oskooei and Born et al. implement attention-based neural networks on gene expression and drug molecular data to make explainable prediction [20]. Our drug response prediction network is also based on deep neural network: we implement an Multi-Layer Perceptron (MLP) model on encoded gene expression and drug molecular data to make predictions.

3 Materials and Methods

In this section we show the strategy we adopt to process the datasets and how we implement our model. Our model takes as input the gene expression data of a cancer cell line and SMILES representation of an anti-cancer drug, producing a drug response prediction in terms of $ln(IC_{50})$. It consists of a VAE to extract features from gene expression data, a JTVAE to extract features from drug molecular data and an MLP model to produce a final prediction.

3.1 Data

Gene expression data We obtain gene expression data of 1021 cancer lines with 57820 genes provided by the Cancer Cell Line Encyclopedia (CCLE)[15]. Each cell line belongs to a specific cancer type. Specifically, we choose breast cancer as our research object primarily, and then test our model on pan cancer cell lines. After filtering by key word token `[BREAST]`, we select 51 breast cancer cell lines from this dataset, which are `[AU565_BREAST]`, `[BT20_BREAST]`, `[BT474_BREAST]`, . . . , `[ZR7530_BREAST]`. Gene expression data is given by $G \in R^{g \times c}$, where g is the number of genes and c is the number of cancer cell lines. The elements of matrix G are $log_2(t_{pm}+1)$, where t_{pm} is transcriptome per million (tpm) value of the gene in the corresponding cell line. Moreover, we access the Cancer Genomic Census (CGC) dataset [19], which classifies different genes into two tiers. One tier is for the genes that are closely associated with cancers and have a high probability to mutate in cancers that change the activity of the gene product. The other tier includes genes that might play a strong indicated role in cancer but show little evidence. Genes in both tiers are highly relevant with cancer, and we take all of these genes in our research. We select 51 breast cancer cell lines from CCLE data set and remove expression data of genes which are not in CGC dataset. Each gene expression entrance with a mean of μ which is less than 1 or standard deviation σ which is less than 0.5 is also removed for their little relevance with cancer cell lines [3]. Eventually, we get gene expression data of 597 genes in 51 breast cancer cell lines.

Anti-cancer drug molecular structure data In this research, we prepare ZINC dataset for molecular structure data of organic compounds to train the JTVAE model. Molecular structure data is given in simplified molecular-input line entry system (SMILES) strings. SMILES representation is often used in defining drug structures [1, 8, 10, 19, 27, 20, 28, 29]. They are widely used as inputs in tasks associated with drug structure prediction. Also, SMILES representation makes it easier for us to get embeddings from vocab parsing library that we have generated. From ZINC data set, we select 10, 000 SMILES strings to train our JTVAE model. The number of SMILES strings for pre-training JTVAE is far beyond the actual number of 222 drugs in processed GDSC dataset. The reason is that we would like to make our model more robust on all drugs instead of only anti-cancer drugs.

Drug response data Drug response data is obtained from the Genomics of Drug Sensitivity in Cancer (GDSC) project[2], which contains response data of anti-cancer drugs used against numerous cancer cell lines. Data from GDSC data set is given by a matrix $IC_{CCLE} \in R^{d \times c}$, where d is number of drugs and c is number of cancer lines. The elements in this matrix are $ln(IC_{50})$, where IC_{50} is the half maximal inhibitory concentration value of the drugs used against specific cancer cell lines. We obtain molecular data of anti-cancer drugs from PubChem dataset with their unique PubChem ID

available from GDSC dataset. Eventually, we get 3358 pieces of drug response data in breast cancer cell lines where gene expression data and drug molecular structure are available.

3.2 Variational Inference

In this work, we adopt variational method to conduct approximate inference, producing latent vector for each unlabeled sample. Variational inference turns the problem of inferring a complicated distribution into optimization. In variational inference, we propose a distribution $q\left(z;\phi\right)$, where z is the latent variable and ϕ represents the parameters. We attempt to adjust v to approximate the true distribution $p\left(z|x\right)$ where x is observed data:

$$\phi^* = \arg\min_{\phi} \mathrm{KL}\left(q\left(z;\phi\right)||p\left(z|x\right)\right) \tag{1}$$

where Kullback-Leibler(KL) divergence is used to evaluate the closeness of $q(z;\phi)$ and $p(z|x)$. The lower $\mathrm{KL}(q(z;\phi)||p(z|x))$ is, the more accurate $q(z;\phi)$ is approximating $p(z|x)$.

To avoid dealing with the intractable distribution $p(z|x)$, the minimization of $KL(q(z;\phi)||p(z|x))$ is converted into another form. Given the equation:

$$\log P(x) = \log \frac{P(x,z)}{q(z;\phi)} - \log \frac{P(z|x)}{q(z;\phi)} \tag{2}$$

, and we integrate z above the two sides of the equation:

$$\log P(x) = \mathrm{KL}(q(z;\phi)||p(z|x)) + \mathrm{ELBO}(\phi) \tag{3}$$

, where $ELBO(\phi)$ is Evidence Lower Bound [6], and it stands for:

$$\mathrm{ELBO}(\phi) = \mathbb{E}_{q(z;\phi)}[\log p(x,z) - \log q(z;\phi)] \tag{4}$$

In equation 3, $KL(q(z;\phi)||p(z|x)) > 0$. Since $\log P(x)$ is fixed, it is intuitive that minimising $KL(q(z;\phi)||p(z|x))$ is equivalent with maximising $ELBO(\phi)$. $ELBO(\phi)$ could be converted to:

$$\mathrm{ELBO}(\phi) = \mathbb{E}_{q(z;\phi)}[\log p(x|z)] - \mathrm{KL}(q(z;\phi)||p(z)) \tag{5}$$

When maximizing $\mathrm{ELBO}(\phi)$, $\mathbb{E}_{q(z;v)}[\log p(x|z)]$ is maximized and $\mathrm{KL}(q(z;\phi)||p(z))$ is minimized, and the latter KL term serves to regularize the model and encourage $q(z;\phi)$ to diverse [30].

3.3 Variational Auto-encoder

Variational Auto-encoder (VAE) is a generative model, modeling complicated conditional distribution with an encoder and a decoder based on deep neural networks. Let $q(z;\phi)$ be the prior distribution of approximated latent variables where ϕ is the parameters and $p(x|z;\theta)$ be the conditional probability distribution computed by the generative network (decoder) where θ is the parameters.

The total aim of variational auto-encoder (VAE) is to find parameters ϕ^* and θ^* to maximize $\mathrm{ELBO}(\phi,\theta)$:

$$\phi^*, \theta^* = \arg\min_{\phi,\theta} \mathrm{ELBO}(\phi,\theta) \tag{6}$$

In VAE, the prior distribution of latent variables is approximated as normal Gaussian distribution, and the posterior $q(z|x;\phi)$ is also supposed to obey Gaussian distribution. The conditional probability distribution $p(x|z;\theta)$ is supposed to obey multivariate Gaussian distribution. While there might be other distributions depending on the type of data, we adopt multivariate Gaussian distribution for our continuous input data. Given the suppositions above, the estimator for this model and datapoint x is [6]:

$$\mathrm{ELBO}(\phi,\theta) = \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\right) + \frac{1}{L}\sum_{j=1}^{J}\log p_\theta(\mathrm{x}|z^{(j)}) \tag{7}$$

where J is the dimensionality of latent variable z. In practice, total loss has been processed to be the opposite number of ELBO, which satisfies the requirement of gradient descent. Because $\mathrm{z} \sim \mathcal{N}\left(\mu, \sigma^2\right)$, one valid reparameterization of z to enable back propagation is $\mathrm{z} = \mu + \epsilon\sigma$, where ϵ serves $\mathcal{N}\left(0,1\right)$.

3.4 Gene expression VAE (GeneVAE)

The aim of geneVAE is to extract latent vectors from CCLE gene expression data, and it will be fit into the combined MLP drug prediction network. We use fully connected neural networks for forward propagation with a batch-norm layer before activation.

$$h_1 = \sigma\left(\text{Batchnorm}\left(\mathbf{W}_1^T G + \mathbf{b}_1\right)\right) \tag{8}$$

$\sigma(\cdot)$ is the activation function (ReLU in our model), W_1 is the weight matrix and b_1 is the bias vector at the first dense layer. Batch normalization is used to train our model more efficiently. Activation function help filter out unimportant information. h_1 is the output of the first layer in our MLP model. We connect it to the second layer:

$$\mu_g = \sigma\left(\text{Batchnorm}\left(\mathbf{W}_2^T h_1 + \mathbf{b}_2\right)\right) \tag{9}$$

where $\sigma(\cdot)$ is the activation function , W_2 is the weight matrix and b_2 is the bias vector at the second dense layer. Latent variables $\text{z}_g \sim \mathcal{N}\left(\mu_g, \sigma_g^2\right)$. μ_g is the computed mean value of this Gaussian distribution. Similarly, σ_g is computed by another 2-layer neural network with the same architecture as μ_g. Latent vector z_g is randomly sampled from $\mathcal{N}(\mu_g, \sigma_g)$.

The decoder architecture is constructed by two dense layers with same output dimensions as the input. The decoded gene expression is written as G'.

$$G^{'} = f_4\left(\mathbf{W}_4^T\left(f_3\left(\mathbf{W}_3^T z_g + \mathbf{b}_3\right) + \mathbf{b}_4\right)\right) \tag{10}$$

In our model, both the encoder and the decoder are two-layer fully connected neural network. The sizes of both encoder layers are set as 256, while the sizes of both decoder layers are set the same as input data. We dispose of batch-norm layers in our decoder architecture.

3.5 Junction tree VAE (JTVAE)

Graph encoder We take junction tree variational autoencoder (JTVAE) [1] as one of our encoding model to represent drug's latent space. We use a message passing network[1, 31] as a graph encoder. Suppose there are d $\in \mathbb{R}$ nodes in the graph. Each node *u* has the property of atom type. $b_{uv} \in \mathbb{R}^{d\times d}$ encodes the bond type between node *u* and node *v*. The matrix $M_{ju}^{(t)} \in \mathbb{R}^{d\times d}$ represents the message passing from node j to node u in number of *t* iterations. $M_{ju}^{(0)}$ is set to 0 initially. $\mathbf{W}_{vu}$, $\mathbf{W}_u$, $\mathbf{W}_{uv}^{'}$ are three weight matrices separately. With the knowledge of loopy belief propagation, we can achieve the message passing embedding with a rectified linear unit from node *u* to *v* at time t as:

$$\mathbf{M}_{uv}^{(t)} = \sigma\left(\mathbf{W}_{vu}\sum_{j\in N(v)\setminus u}\mathbf{M}_{ju}^{(t-1)} + \mathbf{W}_u\mathbf{a}_u + \mathbf{W}_{uv}^{'}\mathbf{b}_{uv}\right) \tag{11}$$

Getting the message from the neighborhood of node *u*, we can aggregate those message embedding vectors with its atom type, which can be written in the summation form with a rectified linear unit as following equation(2). Final graph representation is shown as $\mathbf{L}_G^g$.

$$\mathbf{L}_u^g = \sigma\left(\sum_{j\in N(v)}\mathbf{U}_{ju}\mathbf{M}_{ju}^T + \mathbf{U}_u\mathbf{a}_u\right), \qquad \mathbf{L}_G^g = \sum_i \mathbf{L}_i^g/|V| \tag{12}$$

Mean μ_G and variance σ_G can be computed from $\mathbf{L}_G^g$ by an affine layer. The graph latent vector $\mathbf{z}_G$ is sampled from $\mathcal{N}(\mu_G,\sigma_G)$.

Tree encoder The architecture of tree encoder is based on Gated Recurrent Unit(GRU)[1, 32]. The hidden state is $\widetilde{\mathbf{L}_{ij}^{(t)}} \in \mathbb{R}^{d\times d}$ in this tree encoder model. It is used to reserve tree's message passing information from the moment t-1 together with the tree clusters $\{\mathbf{x}_i, \text{i} = 1, 2, \ldots, \text{d}\}$.

$$\widetilde{\mathbf{L}_{ij}^{(t)}} = \tanh\left(\mathbf{W}_i\mathbf{x}_i + \sum_{k\in N(i)\setminus j}\mathbf{r}_{ki}^{(t)} \odot \mathbf{L}_{ki}^{(t-1)}\right) \tag{13}$$

There are two kinds of gates in our tree encoder model, which are reset gate $\mathbf{r}_{ki}^{(t)}$ and update gate $\mathbf{z}_{ij}^{(t)}$. $\mathbf{r}_{ki}^{(t)}$ is used for calculating how much the system is going to reserve while $\mathbf{z}_{ij}^{(t)}$ is used for counting

the probability that how likely the system is going to update the message passing information at the moment t. If the reset gate $\mathbf{r}_{ki}^{(t)}$ is set to 0, the element-wise multiplication in equation 13 will be simplified to tanh($\mathbf{W}_i\mathbf{x}_i$), which means that there's no reserved message at the previous stage.

$$\mathbf{r}_{ki}^{(t)} = \sigma\left(\mathbf{W}_i^{'}\mathbf{x}_i + \mathbf{U}_r^{'}\mathbf{L}_{ki}^{(t-1)}\right), \qquad \mathbf{z}_{ij}^{(t)} = \sigma\left(\mathbf{W}_i^{''}\mathbf{x}_i + \mathbf{U}_r^{''}\mathbf{L}_{ij}^{(t-1)}\right) \tag{14}$$

The total update function, which depends on the previous activation $\mathbf{L}_{ij}^{(t-1)}$ and candidate activation $\widetilde{\mathbf{L}_{ij}^{(t)}}$ can be written into the form of element-wise multiplication.

$$\mathbf{L}_{ij}^{(t)} = (1 - \mathbf{z}_{ij}^{(t)}) \odot \mathbf{L}_{ij}^{(t-1)} + \mathbf{z}_{ij}^{(t)} \odot \widetilde{\mathbf{L}_{ij}^{(t)}} \tag{15}$$

We can get the tree's latent representation of node u by aggregating its updated messages at the t-th iteration(equation 16). z_G^{τ} is calculated in the smiliar way as graph encoder do. Since the graph and tree decoders' structure in JTVAE is also based on GRU method, we will not discuss about it here but instead reference to raw paper of JTVAE [1].

$$\mathbf{L}_u^{\tau} = \sigma\left(W^o x_i + \sum_{k \in N(i)} U^o L_{ij}^{(t)}\right) \tag{16}$$

3.6 Drug response prediction network

Since gene VAE and molecular VAE have been trained at this stage, we implement two multi-layer perceptron (MLP) models to post-process the output of the two VAE models respectively, and then build another MLP model to concatenate them and produce the final drug response prediction. The input to the final MLP model is $\mathbf{a}_{all} = [\mathbf{a}_{gene}, \mathbf{a}_{drug}]$, where $\mathbf{a}_{gene}$ and $\mathbf{a}_{drug}$ are outputs of the two post-processing MLP models. Suppose $\mathbf{a}_{gene} \in \mathbb{R}^{d_1}$ and $\mathbf{a}_{drug} \in \mathbb{R}^{d_2}$, then $\mathbf{a}_{all} \in \mathbb{R}^{d_1+d_2}$, where d_1 is the dimensionality of $\mathbf{a}_{gene}$ and d_2 is the dimensionality of $\mathbf{a}_{drug}$. Values of perceptrons in the i^{th} layer in the final MLP model is computed according to:

$$a_{all}^{i+1} = f^{'}(\mathbf{W}^{(i+1)T} a_{all}^{i} + \mathbf{b}^{(i+1)}) \tag{17}$$

where $W^{(i+1)}$ is the weight matrix of the i-th layer in the final MLP model and $f^{'}$ is a non-linear activation function for which we choose Parametric Rectified Linear Unit (PReLU) in our model. The predicted IC_{50} is computed at the last layer of the final MLP model:

$$ln(IC_{50}) = f^{'}(\mathbf{W}^{(n)T} a_{all}^{n-1} + \mathbf{b}^{(n)}) \tag{18}$$

where n is the number of layers in the final MLP model.

In our model, both of the two post-processing MLP consist of 3-layer fully connected neural network. Since the outputs of geneVAE and JTVAE are 256-dimension vectors and 56-dimension outputs respectively, we set sizes of two post-processing MLP as (256, 256, 64) and (128, 128, 64). The final combiner MLP is a 4-layer fully connected neural network with 128, 128, 64 units in its hidden layers.

Baseline model We use Support Vector Regression (SVR) in substitute for MLP as our baseline model, showing a convenient way to take advantage of machine learning methods to make drug response prediction. We choose poly kernel in our SVR model and set the parameter C as 10.

4 Experiments and Results

Experiment set-up Firstly, we train our geneVAE model and JTVAE model unsupervisedly. We use geneVAE to encode gene expression data either filtered by CGC data set or not respectively on breast cancer cell lines, and use Junction Tree VAE(JTVAE) to encode anti-cancer drugs. With these encoded features, we train our SVR model which is the baseline model, and MLP model on breast cancer cell lines. Thereafter, our model is tested on pan-cancer cell lines. Besides drug response prediction, we also show that our model could be used to generate effective drugs for cancer cell line. Training set and test set are split by 9:1 for SVR models, while training set, validation set and test set are split by 18:1:1 for MLP models.

4.1 Pre-training geneVAE

We aim to minimize the sum of reconstruction loss and KL loss when training geneVAE model. The reconstruction loss is $\mathcal{L}(G, G^{'})$, where G represents initial input gene expression data and $G^{'}$ represents reconstructed data. It can be mean squared loss `[MSEloss]` or cross entropy loss `[CrossEntropyloss]`. We choose cross entropy loss as our reconstruction loss in our experiments since we normalize the input data and insert a sigmoid activation at the last layer to make sure that inputs and outputs are values between 0 and 1.

Filtering out a representative gene subset using CGC data set also matters in the training of our gene expression VAE model. We have mentioned in the 3.1 part that for breast cancer cell lines, the selected gene number from CGC is 597. We test our model either filtering out a gene subset or not on breast cancer cell lines, and the result indicates that curating such a gene subset could help improve the accuracy of the predicted $ln(IC_{50})$ value. A warm-up strategy is adopted in the training of geneVAE. The total VAE loss is set as

$$\mathbf{VAE_{Loss}} = \mathcal{L}(G, G^{'}) + \beta KL \tag{19}$$

where KL is KL loss and β is a parameter gradually increase from 0 to 1 during training. According to the evaluation of total loss, our tests show that at the beginning of the training loop, validation VAEloss (-ELBO) is much higher than training VAEloss, and it starts to convergent after 100 epochs. Model on CGC-selected gene expression data has an average VAEloss of 27.3. Model without CGC selected gene expression data has an average VAEloss of 68 after validation loss becomes stable. We set our default weight decay as 0.01. Learning rate will be around 1*10e-5 after epochs of weight decay. Tuning weight decay is important since it alleviates the effect of stepping into local minimum.

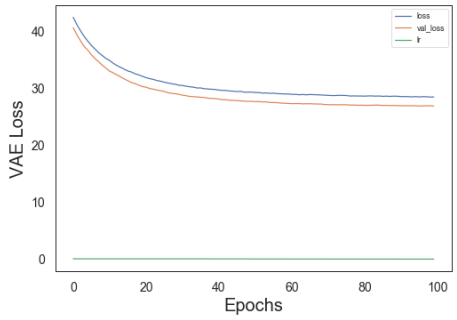

Figure 4: VAEloss and lr with CGC

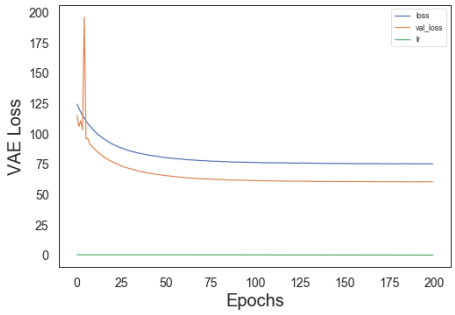

Figure 5: VAEloss and lr without CGC

4.2 Results on breast cancer

We propose several models and test them on breast cancer cell lines, and the results show that VAE and CGC datasets contribute to more accurate predictions. We select 2 metrics: Coefficient of Determination (R2 score) and Root Mean Square Error (RMSE) to evaluate the discrepancy between our predicted drug response and true drug response. We propose 6 models and the results have been listed in Table 1. Among these models, the first 5 models are targeted on breast cancer, and the last one is tested on pan cancer cell lines: $\mathbf{1)CGC + SVR}$: Support Vector Regression model trained on drug molecular structure data encoded by VAE model and gene expression data filtered by CGC dataset. $\mathbf{2)CGC + VAE + SVR}$: Support Vector Regression model trained on gene expression data filtered by CGC dataset and drug molecular structure data which are both encoded by VAE model. $\mathbf{3)CGC + MLP}$: Multi-Layer Perceptron model trained on drug molecular structure data encoded by VAE model and gene expression data filtered by CGC dataset. $\mathbf{4)RAW + VAE + MLP}$: Multi-Layer Perceptron model trained on raw gene expression data (not filtered by CGC dataset) and drug molecular structure data which are both encoded by VAE model. $\mathbf{5)CGC + VAE + MLP}$: Multi-Layer Perceptron model trained on gene expression data filtered by CGC dataset and drug molecular structure data which are both encoded by VAE model. $\mathbf{6)CGC + VAE + MLP}$: Multi-Layer Perceptron model trained on gene expression data filtered by CGC dataset and drug molecular structure data which are both encoded by VAE model. This model is trained on pan cancer dataset.

The test results of these models are shown in Table 1. We can see that the MLP model and VAE model bring about huge improvement in the performance of our models: $\mathbf{CGC + MLP}$: model outperforms$\mathbf{CGC + SVR}$: model by 0.143 R2 score, and $\mathbf{CGC + VAE + MLP}$: model performs even better than $\mathbf{CGC + MLP}$: model with a 0.008 higher R2 score. Moreover, the selection of representative gene subset is essential to the performance of our models. For example, $\mathbf{CGC + VAE + MLP}$: model on breast cancer cell lines reaches 0.830 R2 score, much better than that of $\mathbf{RAW + VAE + MLP}$: model with a 0.025 higher R2 score.

Table 1 Metrics evaluation on different gene subsets in breast and pan cancer dataset (average).

Models	Cancer type	R^2_{test}	$RMSE_{test}$
CGC + SVR	Breast	0.658	1.582
CGC + VAE + SVR	Breast	0.692	1.491
CGC + MLP	Breast	0.822	1.133
RAW + VAE + MLP	Breast	0.805	1.163
CGC + VAE + MLP	Breast	0.830	1.130
CGC + VAE + MLP	Pan-cancer	0.845	1.080

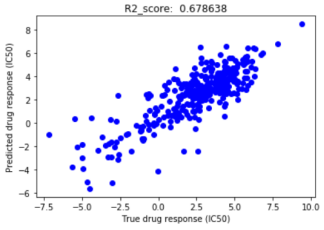

Figure 6: CGC+SVR

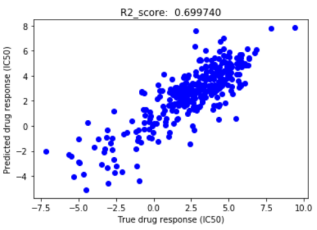

Figure 7: CGC+VAE+SVR

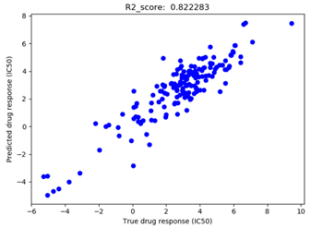

Figure 8: CGC+MLP

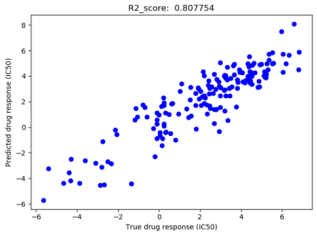

Figure 9: Raw+VAE+MLP

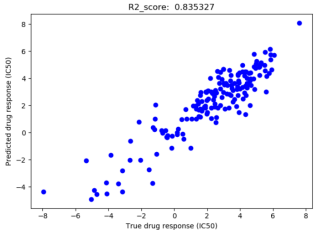

Figure 10: CGC+VAE+MLP

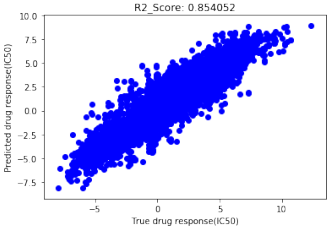

Figure 11: CGC+VAE+MLP(pan)

4.3 Test on pan-cancer

We test our model on the pan-cancer cell lines based on CCLE dataset. The only difference in pan-cancer gene expression data from that of breast cancer is that the total number of pan-cancer cell lines is 1021. Our $\mathbf{CGC + VAE + MLP}$ model achieves an even higher $R^2 score$ 0.845 on pan-cancer cell lines. To make our model more robust, we will incorporate more data into our model like TCGA dataset.

4.4 Effective drug compound generation

Compared with other representation learning methods on molecules, JTVAE has the advantage of reconstructing 100% valid drugs, making it possible to generate new drugs for specific cancer cell lines. To customize effective compounds for a given cancer cell line, we firstly sample a 56 dimension vector, which is the same as latent vectors encoded by JTVAE, from Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 7$. Thereafter, the randomly sampled drug vector is concatenated with the latent vector of gene expression profile of breast cancer cell line HCC1187, and they are computed by the MLP model to produce a prediction. We set the threshold of effective drugs as $-1.0\ ln(IC_{50})$ value.

If the $ln(IC_{50})$ value of a randomly generated drug latent vector is below -1.0, it is considered to be effective on HCC1187. Also, the threshold could be set as -1.5, -2.0 etc, producing more effective generated drugs, but it will take a longer time to generate effective drugs. We select 10 generated drug latent vectors to decode with JTVAE model, and the results are shown in 12. As JTVAE could always decode drug latent vectors into valid compounds, these generated and decoded drug compounds, though might have not been used as anti-cancer drugs, provide a promising way for anti-cancer drug discovery. The amazing abilities of VAE to learn effectively from unlabeled data and reconstruct input data remain to developed for more powerful applications.

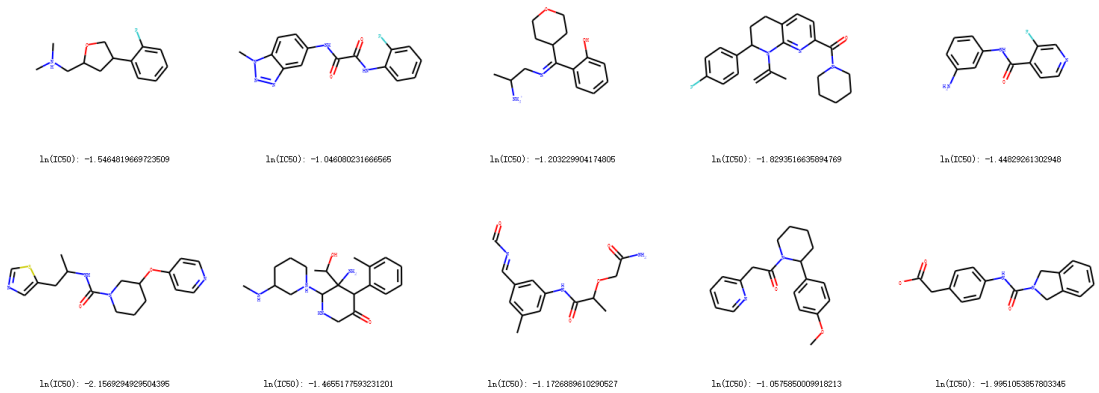

Figure 12: 10 effective drugs whose $ln(IC_{50})$ values on cancer cell line HCC1187 are below -1.0.

4.5 Exploring latent vectors from geneVAE

Taking advantage of the diversity of cancer types in pan-cancer dataset, we discover that latent vectors encoded by geneVAE retains the features of original data. We visualize latent vectors of gene expression data into two dimension Euclidean space. Effects of dimensionality reduction are evaluated by a single t-SNE model compared with another t-SNE mdodel combined with our pretrained VAE encoder. Generally t-SNE is just used for visualization on a two dimensional plane since t-SNE model performs worse at a higher dimension space due to curse of dimensionality. We begin with giving each cell line its tissue type, from "CERVIX" to "OVARY". We encode them by extracting the pattern after their first underscore in CCLE dataset. Especially we rename "HAEMATOPOIETIC_AND_LYMPHOID_TISSUE" to "HALT" since it's too long to present in a picture. Parameters are perplexity and iterations for the single t-SNE model. We set perplexity to n/120, where n are the numbers of cell lines and we set iterarions(n_iter in python) to 3000. The same settings are applied to the combined model. The result shows that many clusters are apparent both in a single t-SNE model and a combined model. Therefore, the latent vectors encoded by geneVAE model retains the unique features of input data.

In the single model, tissue type labels with `[HALT]`, `[AUTONOMIC_GANGLIA]` ,`[BREAST]` and `[SKIN]` etc are separate obviously, while some other type of tissues are clustered together with other similar tissue types. For example, genes do not have a great difference on their expression according to "STOMACH" and "LARGE INTESTINE". Several tissue types are so rare in cancer cell lines that they might be confused with another tissue, because t-SNE doesn't exactly explain the real distance between cancer types.

Eliminating rare cancer types could help improve the t-SNE results. We set the threshold of 30 to filter the tissue types, where 12 tissues are hold. They are `[BREAST, CENTRAL_NERVOUS_SYSTEM, FIBROBLAST, HALT, KIDNEY, LARGE_INTESTINE, LUNG, OVARY, PANCREAS, SKIN, STOMACH, UPPER_AERODIGESTIVE_TRACT]`. We remain the gene subset that we have filtered and eliminated from raw data. We visualize it again and find that more clusters are apparent in the picture, where we use black frames to represent. The clustering results of latent vectors and original data still remain similar in Figure 14, where primary cancer tissue types are separated clearly. Therefore, latent vectors encoded by geneVAE model retain the essential features of original data robustly. With

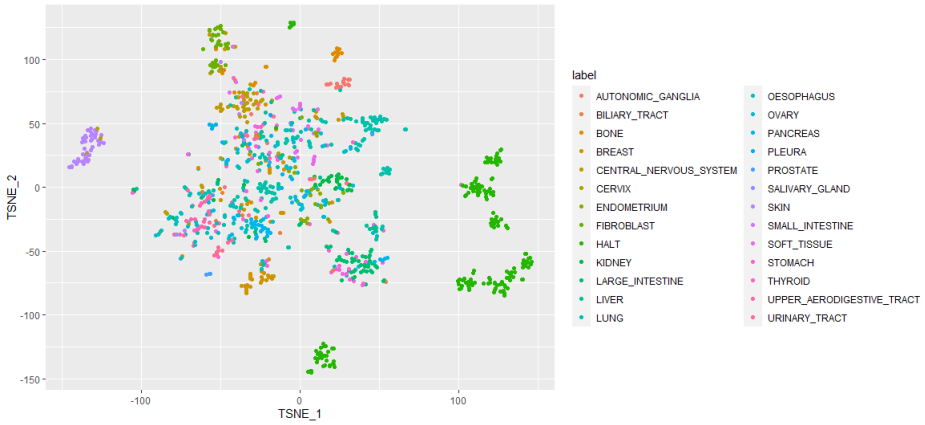

Figure 13: t-SNE on pancancer dateset before applying VAE

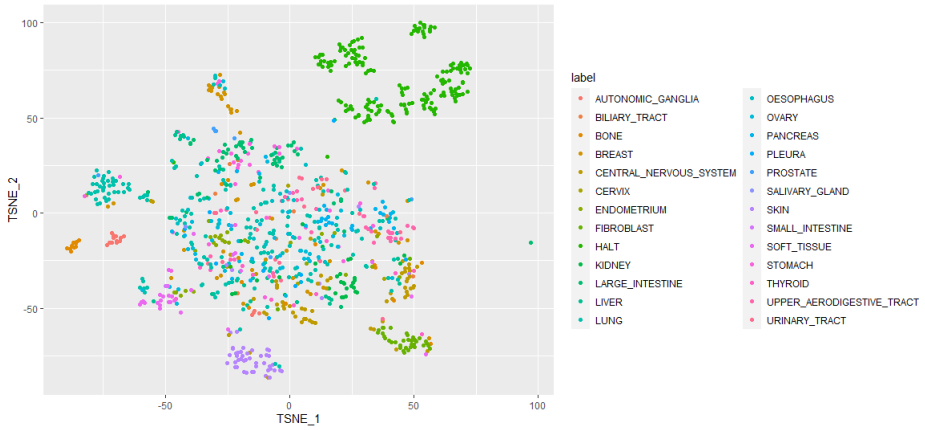

Figure 14: t-SNE on latent vectors after applying VAE

geneVAE, our models are able to focus on the low-dimensional critical features of original data and produce a more accurate prediction.

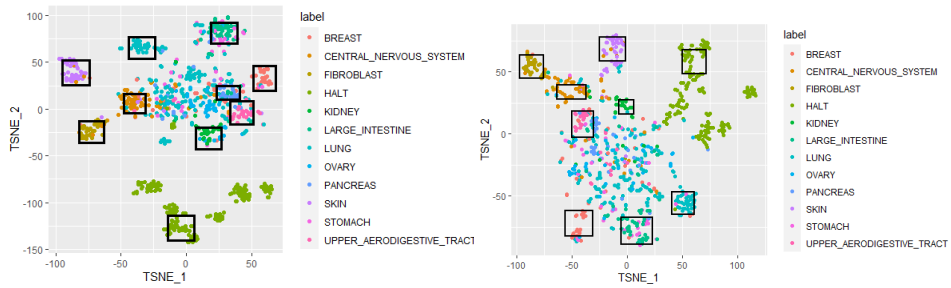

Figure 15: t-SNE with threshold 30 before and after VAE

4.6 Exploring latent vectors from JTVAE

Drugs sharing similar molecular structure are also similar in latent vectors. We get access to latent vectors encoded by JTVAE. It reveals that drugs which share similar molecular structure are also similar in latent vectors. We measure the similarity of latent vectors of different drugs in terms of Euclidean Distance. Shorter distance indicates a higher similarity between two drugs. For example, MG132(inhibitor) and Proteasome (inhibitor) share a shortest distance which is about 23.73. We

obtain their molecular structures in Pubchem database and find that a majority of functional groups are similar between these two drugs. Small differences lie in a carboxyl and an amide at the endings of the molecule. However, not all related drugs have such a great similarity. According to drug Imatinib and Linifanib, their are even closer in terms of Euclidean distance between their latent vectors but they have only middle part of the functional groups exactly the same. JTVAE model might discover underlying similarity among functional groups that are not exactly the same. Also, the message passing network in JTVAE is based on GRU, and it might forget some functional groups during propagation by neighbours.

Though similar drugs share close latent vectors, our MLP model is still able to capture subtle differences and produce an accurate prediction. We focus on the example of MG132 and Proteasome used against HCC1187 cancer cell line. We remove these two pieces of data from training set, and test our trained model on them. The predicted IC_{50} of MG132 and Proteasome in cell line HCC1187 are 0.84 and -0.866 in our best model. True value of these two drugs are 1.589 and -0.181 respectively. Although two values are not too close to the expected values, they do not step into the range of the other's confidence interval. Therefore, despite the considerably high similarity between similar drugs, our MLP model is still able to differentiate each of them and produce a reasonable result.

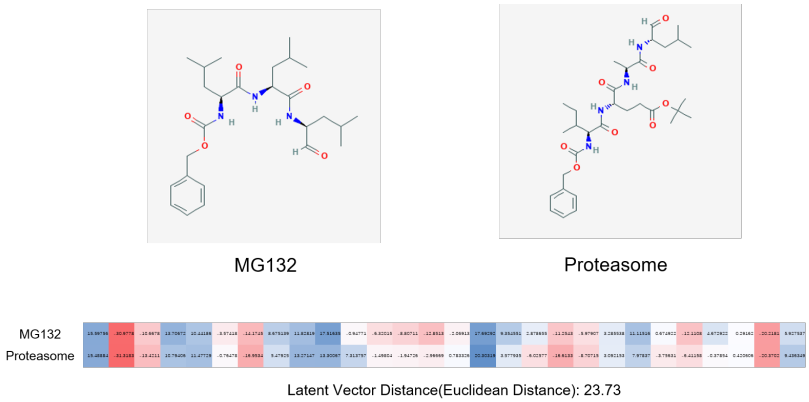

Figure 16: Similar latent vectors on similar drug structures.

5 Discussion and conclusions

In this research we build gene expression VAE (geneVAE) model, junction tree VAE (JTVAE)model, Support Vector Regression (SVR) model (baseline) and several multi-layer perceptron (MLP) models to predict anti-cancer drug response. We extract latent vectors with geneVAE and JTVAE model to fit them into our drug prediction network. Our combined models are compared with baseline SVR models to see their improvement in performance. Generally speaking, we have achieved a great coefficient of determination value with our model (0.845 R2 score on pan-cancer and 0.830 on breast cancer). Besides, we show the amazing ability of our model to generate potentially effective anti-cancer drugs. We also discuss the effectiveness of geneVAE and JTVAE model from the perspective of visualization and drug similarity, further proving the validity of our pipeline.

There are still some interesting aspects that we have met during our research. Hyper-parameter tuning and layer setting are superior aspects when speaking of accuracy improvement. Different hyper-parameters lead to different consequences. For example, adding Batch-Norm (BN) layer in MLP model results in a worse performance. Batch normalization is a widely used technique to avoid gradient exploding and vanishing. However, its effectiveness is doubtful when it is used in shallow networks with Rectified Linear Unit, where gradient exploding and vanishing seldom occur. Moreover, the inconsistency among mini-batches could influence the performance of the batch-norm layers badly. Besides BN layer, proportion of train-valid-test split is essential to the final result, as well as the proportions of batch size and learning rate. We set the default batch-size of 8 and learning rate 0.001 in the training loop. Larger values of these two hyper-parameters convergent faster, however might meet with problem of falling into local minimum. A proper proportion of

train and validation and test sets are 10:1:1 and for each epoch we choose them randomly from total dataset. K-fold cross validation is also a good choice.

We suggest dimensional reduction should be done without PCA, as well as t-SNE and some other powerful clustering methods. PCA is not suitable when reducing dimension numbers to 56 or even higher in our research. t-SNE is better than PCA at 2-dimensional representation. However we also find that there are something mess up when applying t-SNE at 2-dimensional space. Besides, we will showcase more predictions in our future works with similar structure to further generalize our idea that although similar, their latent vectors could not be changed when making predictions. Unless we find the prediction value is not in each other's confidence interval.

6 Future work

We have witnessed a great improvement of prediction accuracy by filtering out a gene subset with CGC dataset. More promising methods in selecting a representative gene subset like network propagation based on STRING could be used to further improve our model. Moreover, attention mechanism based models are included in our future works. Using attention mechanism is not only popular in transformer, BERT model which belong to natural language processing field, but also widely used in drug structure translation field[27, 20]. Apart from attention based models, there are other sequence generation models like GMM and graph neural network(GNN). Moreover, we'd like to build a toolkit for drug response prediction if given one cancer cell line data and corresponding drug's response. Last but not least, we will select better gene subset of each drug since drug response may have different gene contributions.

Acknowledgement

Thanks to professor Manolis Kellis from MIT, CSAIL Lab for reviewing this article.

References

[1] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.

[2] Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A Smith, I Richard Thompson, et al. Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, 41(D1):D955–D961, 2012.

[3] Yu-Chiao Chiu, Hung-I Harry Chen, Tinghe Zhang, Songyao Zhang, Aparna Gorthi, Li-Ju Wang, Yufei Huang, and Yidong Chen. Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC medical genomics*, 12(1):18, 2019.

[4] Paul Geeleher, Nancy J Cox, and R Stephanie Huang. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome biology*, 15(3):1–12, 2014.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Qiao Liu, Zhiqiang Hu, Rui Jiang, and Mu Zhou. Deepcdr: a hybrid graph convolutional network for predicting cancer drug response. *bioRxiv*, 2020.

[8] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

[9] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[10] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer, 2018.

[11] Kathleen M. Schmainda, Melissa Prah, Jennifer Connelly, Scott D. Rand, Raymond G. Hoffman, Wade Mueller, and Mark G. Malkin. Dynamic-susceptibility contrast agent MRI measures of relative cerebral blood volume predict response to bevacizumab in recurrent high-grade glioma. *Neuro-Oncology*, 16(6):880–888, 01 2014.

[12] Takeshi Yuasa, Shunji Takahashi, Kiyohiko Hatake, Junji Yonese, and Iwao Fukui. Biomarkers to predict response to sunitinib therapy and prognosis in metastatic renal cell cancer. *Cancer science*, 102(11):1949–1957, 2011.

[13] Teruhiko Imamura, Koichiro Kinugawa, Shun Minatsuki, Hironori Muraoka, Naoko Kato, Toshiro Inaba, Hisataka Maki, Taro Shiga, Masaru Hatano, Atsushi Yao, et al. Urine osmolality estimated using urine urea nitrogen, sodium and creatinine can effectively predict response to tolvaptan in decompensated heart failure patients. *Circulation Journal*, 77(5):1208–1213, 2013.

[14] J M Lachlan and Tim J Hubbard. A census of human cancer genes. *Nat Rev Cancer*, 4(3):177–183, 2004.

[15] Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

[16] Stephane Wenric and Ruhollah Shemirani. Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, 9:297, 2018.

[17] Shenghui Liu, Chunrui Xu, Yusen Zhang, Jiaguo Liu, Bin Yu, Xiaoping Liu, and Matthias Dehmer. Feature selection of gene expression data for cancer classification using double rbf-kernels. *BMC bioinformatics*, 19(1):1–14, 2018.

[18] Haiyan Huang and Kyungpil Kim. Unsupervised clustering analysis of gene expression. *Chance*, 19(3):49–51, 2006.

[19] Yoosup Chang, Hyejin Park, Hyun-Jin Yang, Seungju Lee, Kwee-Yum Lee, Tae Soon Kim, Jongsun Jung, and Jae-Min Shin. Cancer drug response profile scan (cdrscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Scientific reports*, 8(1):1–11, 2018.

[20] Ali Oskooei, Jannis Born, Matteo Manica, Vigneshwari Subramanian, Julio Sáez-Rodríguez, and María Rodríguez Martínez. Paccmann: Prediction of anticancer compound sensitivity with multi-modal attention-based neural networks. *arXiv preprint arXiv:1811.06802*, 2018.

[21] Christopher H Grønbech, Maximillian F Vording, Pascal N Timshel, Capser K Sønderby, Tune H Pers, and Ole Winther. scvae: Variational auto-encoders for single-cell gene expression datas. *bioRxiv*, page 318295, 2018.

[22] Ladislav Rampasek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr. vae: Drug response variational autoencoder. *arXiv preprint arXiv:1706.08203*, 2017.

[23] Ladislav Rampášek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr. vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, 35(19):3743–3751, 2019.

[24] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[25] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. *arXiv preprint arXiv:1602.07776*, 2016.

[26] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[27] Matteo Manica, Ali Oskooei, Jannis Born, Vigneshwari Subramanian, Julio Sáez-Rodríguez, and Mariéa Rodriéguez Martiénez. Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Molecular Pharmaceutics*, 2019.

[28] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, pages 7795–7804, 2018.

[29] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, 2019.

[30] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[31] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016.

[32] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.