Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Variational Autoencoder for Anti-Cancer Drug Response Prediction

Hongyuan Dong [1], Jiaqing Xie [2,*], Zhi Jing [3] and Dexin Ren [4]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China
[2] School of Engineering, University of Edinburgh, UK
[3] School of Data and Computer Science, Sun Yat-sen University
[4] School of Computer Science, University of Arizona

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: 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).
Results: Our model takes as input gene expression data of cancer cell lines and anti-cancer drug molecular data, and encode these data with GeneVae model, which is an ordinary VAE, and rectified junction tree variational autoencoder (JtVae) (Jin *et al.* (2018)) model, respectively. Encoded features are processes by 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.
Availability: Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
Contact: s2001696@ed.ac.uk
Supplementary information: Supplementary data are available at *Bioinformatics* online.

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 (Yang *et al.* (2012)). Some researchers work on predicting drug response with gene expression data: Chiu *et al.* (2019) build deep neural networks to combine gene expression with mutation profiles to make predictions, Geeleher *et al.* (2014) implement a ridge regression model on before-treatment gene expression data to predict response of chemotherapy. In order to predict the response of different drugs used on different cancer cell lines, we incorporate both gene expression data and drug molecular data into our model.

There exists many ways to extract representative low-dimensional features from unlabeled data. 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 features are unlabeled and high in dimensionality. However, they are primarily for 2D visualization in most cases and might lose many important information when compressing data into low dimensional latent features. Auto-encoder has been used widely in extracting low dimensional features, but its robustness is doubtful: a slight variance in the encoded vector can lead to huge difference in reconstructed data. Other feature extraction methods such as Graph Convolutional Network (GCN), can be used in unlabeled drug molecular data (Liu *et al.* (2020)). But it only focuses on encoding original data, and cannot function as generative models, which is of great significance in drug discovery.

To extract features from a huge amount of unlabeled data, we take advantage of variational autoencoder (VAE) (Kingma and Welling (2013)), which encodes the distribution of latent features instead of producing certain latent features directly, and 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) (Jin *et al.* (2018) model), transforming their molecular graph into valid substructures to extract their low dimensional features. JTVAE is also a generative model, and outperforms many previous work (Kusner *et al.* (2017); Li *et al.* (2018); Simonovsky and Komodakis (2018)) in reconstructing molecules. Drug compounds generated by JTVAE are always valid, and therefore it is extremely powerful in new anti-cancer drug discovery. In this work, we show that encoded features of drugs can be randomly sampled, and the well-performing features can be decoded by JTVAE, providing a large number of valid compounds which are effective for cancer therapy.

With encoded low dimensional features of gene expression data and drug molecular data, we implement a multi-layer perceptron (MLP) to combine the extracted features to produce the final result, which is $ln(IC_{50})$ value of the input anti-cancer drug used against the input cancer cell line. Different from some former works where models are restricted in specific drugs (Schmainda *et al.* (2014); Yuasa *et al.* (2011); Imamura *et al.* (2013)), our model can take as input merely any organic compounds to predict their responses. Moreover, combined with JTVAE, it can 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) (Lachlan and Hubbard (2004)) dataset, which contains a number of genes highly relevant with cancer, can be used to curate a gene subset from cancer gene data, which removes a significant amount of useless information. In this work, we take advantage of 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 representative low dimensional embeddings of original data with variational autoencoders (VAE) and using these features 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 to encode cancer gene data input and a JTVAE to encode drug molecular data input. An MLP model is implemented based on the extracted features, which produces $ln(IC_{50})$ value as final prediction. We choose coefficient of determination (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) (Barretina *et al.* (2012)) gene expression dataset, Cancer Gene Census (CGC) (Lachlan and Hubbard (2004)) dataset, ZINC molecular structure dataset and GDSC drug response dataset (Yang *et al.* (2012)). 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 can generate chemical compounds which are effective for specific cancer cell lines. Moreover, the latent representations encoded by geneVAE and JTVAE are explored to demonstrate the robustness of our model.

2 Related work

2.1 Dimensionality reduction on features

Encoding features into lower dimensions is commonplace in representation learning tasks. The reduction in feature dimensionality removes a large amount of redundant information and facilitate the analysis. Supervised learning methods can select features which are most relevant with the task: Wenric and Shemirani (2018) use random forest on gene importance in RNA sequence case-control studies, Liu *et al.* (2018b) implement support vector machines (SVM) with double RBF-kernels to filter out irrelevant gene features. Unsupervised learning methods, such as principal component analysis (PCA) and hierarchical learning can help explain the group features of genes while reducing the feature dimensionality (Huang and Kim (2006)). Auto-encoder based on neural networks also learns to encode original data into low dimensional features unsupervisedly, and has been used widely in extracting low dimensional features. Though auto-encoders outperform traditional methods, its robustness is doubtful: a slight variance in the encoded vector can lead to huge difference in reconstructed data. Variational auto-encoder (Kingma and Welling (2013)), which add noise into encoded features to build a more robust auto-encoder model, is proposed to overcome this weakness.

2.2 Variational auto-encoders on gene profile

A large number of work has been done on encoding gene expression data into representative low dimensional features. Neural networks, such as multi-layer perceptrons (MLP), convolutional neural network (CNN) and etc, can encode gene features effectively: Chang *et al.* (2018) use CNN to encode gene mutation and drug molecular data, Oskooei *et al.* (2018) implement attention-based neural networks to produce explainable encoded features. An encoder-decoder structure (Chiu *et al.* (2019)) extends ordinary multi-layer perceptrons, which is also able to reconstruct the original input. The bottleneck layer represents the latent features encoded by autoencoders. Recently, variational autoencoder (VAE) (Kingma and Welling (2013)), which modifies ordinary auto-encoders to achieve more robustness, is used frequently in pre-trained models for gene expression data: Grønbech *et al.* (2018) use VAE to estimate expected gene expression level, Rampasek *et al.* (2017) implement VAE models to analyze before-treatment and after-treatment gene expression profile. We also take advantage of VAE model to process gene expression data. Latent features of gene expression data provide representative information of cancer cell lines, which enables our model to predict drug response on different cancer tissues.

2.3 Representation learning on graph for drug molecular features

Drug molecular features can be represented in graphs and processed by deep neural networks: Duvenaud *et al.* (2015) build graph convolutional neural networks on circular fingerprints of molecules, Liu *et al.* (2020) implement uniform graph convolutional neural networks (UGCN) to extract representative features from drug molecular data, Gilmer *et al.* (2017) use message passing neural network (MPNN) for molecular property prediction. In addition, attention mechanism can be used with RNN and CNN models (Manica *et al.* (2019); Oskooei *et al.* (2018)) to encode drug molecular data, learning attention weights by multihead-attention or self-attention to produce explainable encoded features. Besides, variational autoencoder (VAE) is widely used in tasks where the models are required to be generative: Kusner *et al.* (2017) propose grammar based VAE and use parse trees to produce more valid generated output, Simonovsky and Komodakis (2018) label the nodes and bonds in molecules to form a graph structure and apply VAE model on it, Li *et al.* (2018) also use graph-structured VAE model to generate molecules matching the statistics of the original dataset. In order to avoid generating atoms one by one, which often leads to invalid output in drug design, Jin *et al.* (2018) propose junction tree VAE which decomposes molecules into valid substructures and generate compounds from a vocabulary of valid components. As a result, molecules generated by JTVAE are always valid. In this work, we take junction tree variational autoencoder (JTVAE) as our pre-trained model to encode drug molecular data and generate effective drugs for cancer cell lines.

2.4 Drug response prediction methods

Drug response prediction is a supervised regression task. Support vector regression(SVR) and random forest regressor are basic algorithms to perform regression. Recently, deep neural network methods have been popular in drug efficacy prediction network: Chiu *et al.* (2019) build deep neural networks to analyze gene expression and mutation profiles to make prediction, Chang *et al.* (2018) use convolutional neural network based methods on gene mutation profile and drug molecular data, Liu *et al.* (2020) also use gene mutation data and drug molecular data, and take advantage of CNN and uniform graph convolutional neural networks (UGCN) to make predictions, Oskooei *et al.* (2018) implement attention-based neural networks on gene expression and drug molecular data to make explainable prediction. In this work, We implement a 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, and produce a drug response prediction in terms of $ln(IC_{50})$. The model consists of geneVAE, which is an ordinary 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

3.1.1 Gene expression data

We obtain gene expression data of 1021 cancer lines with 57820 genes provided by the Cancer Cell Line Encyclopedia (CCLE) (Barretina *et al.* (2012)). 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]`, `[ZR7530_BREAST]` and etc. 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 get access to the Cancer Genomic Census (CGC) dataset (Chang *et al.* (2018)), 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 (Chiu *et al.* (2019)). Eventually, we get gene expression data of 597 genes in 51 breast cancer cell lines.

3.1.2 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 (Jin *et al.* (2018); Kusner *et al.* (2017); Simonovsky and Komodakis (2018); Chang *et al.* (2018); Manica *et al.* (2019); Oskooei *et al.* (2018); Liu *et al.* (2018a); Tsubaki *et al.* (2019)). 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.

3.1.3 Drug response data

Drug response data is obtained from the Genomics of Drug Sensitivity in Cancer (GDSC) project (Yang *et al.* (2012)), 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 Auto-encoder

Variational Auto-encoder (VAE) is a generative model, modeling complicated conditional distribution of latent features with an inference network (encoder) and a generative network (decoder). Let $q(z|x;\phi)$ be the 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 $\mathrm{ELBO}(\phi,\theta)$ (Kingma and Welling (2013)):

$$\begin{aligned}\phi^*, \theta^* &= \arg\max_{\phi,\theta} \mathbb{E}_{q(z|x;\phi)}[\log p(x|z;\theta)] - \mathrm{KL}(q(z|x;\phi)||p(z;\theta)) \\ &= \arg\max_{\phi,\theta} \mathrm{ELBO}(\phi,\theta)\end{aligned} \tag{1}$$

In VAE, the prior distribution of latent variables $p(z;\theta)$ 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. Given the suppositions above, the estimator for this model and datapoint x is (Kingma and Welling (2013)):

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

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 $\mathcal{N}\left(0,1\right)$.

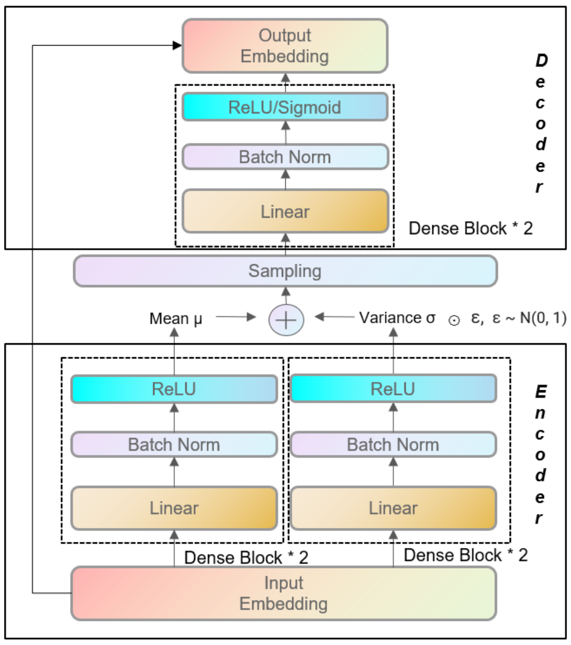


Fig. 1. The architecture of geneVAE. The encoder computes parameters μ and σ of Gaussian distribution $\mathcal{N}(0, 1)$ with separate dense blocks. Sampled latent vectors are processed by the decoder where the first layer adopt ReLU activation and the second layer use Sigmoid activation to reconstruct the input.

3.3 Gene expression VAE (GeneVAE)

The aim of geneVAE is to extract latent vectors from CCLE gene expression data, and the extracted latent vectors will be used for drug response prediction. In this work, geneVAE is an ordinary VAE based on fully connected neural networks. For the encoder, we use 2-layer fully connected neural networks for forward propagation with a batch-norm layer before activation.

$$h_1 = \tau\left(\mathrm{BN}\left(\mathbf{W}_1^T G + \mathbf{b}_1\right)\right) \tag{3}$$

$\tau(\cdot)$ 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, which is represented as BN, is used to train our model more efficiently. h_1 represents the output of the first layer. It is connected to the second layer:

$$\mu_g = \tau\left(\mathrm{BN}\left(\mathbf{W}_2^T h_1 + \mathbf{b}_2\right)\right) \tag{4}$$

Latent variables $\mathrm{z}_g \sim \mathcal{N}\left(\mu_g, \sigma_g^2\right)$. 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 also a 2-layer fully connected neural network. The decoded gene expression data is written as G':

$$G^{'} = \sigma\left(\mathrm{BN}\left(\mathbf{W}_4^T\left(\tau\left(\mathrm{BN}(\mathbf{W}_3^T z_g + \mathbf{b}_3)\right) + \mathbf{b}_4\right)\right)\right) \tag{5}$$

where σ represents sigmoid activation. In our model, both the encoder and the decoder are 2-layer fully connected neural network, and the architecture is shown in **Figure** 1. The sizes of both encoder layers are set as 256, while the sizes of both decoder layers are set the same as input data. When encoding gene expression data into latent vectors, we take μ_g as encoded features instead of sampling these vectors from a Gaussian distribution.

3.4 Junction tree VAE (JTVAE) (Jin *et al.* (2018))

JTVAE consists of a graph VAE and a tree VAE. Molecules are decomposed as junction trees where nodes are valid molecular substructures. The decomposed junction tree is encoded by tree VAE while the original molecular graph is encoded by graph VAE. When generating molecules, the decoder of tree VAE reconstructs the junction tree of the molecule, and the decoder of graph VAE provides complementary connectivity information to reproduce the full molecular graph.

Graph encoder. The encoder of graph VAE, which takes atoms as nodes in the graph, is implemented with a message passing network (Gilmer *et al.* (2017)). Messages are passing from node to node for t iterations. Final representation of each node is computed by aggregating its relevant message in the message passing network, and these representations are used to produce the final graph representation $\mathbf{h}_G$. The graph latent vector $\mathbf{z}_G$ is sampled from $\mathcal{N}(\mu_G, \sigma_G)$, where μ_G and σ_G are computed by 2 separate affine layers from the graph representation.

Tree encoder. The encoder of tree VAE, on the other hand, takes valid substructures of the molecule graph as nodes, and implement a message passing network based on Gated Recurrent Unit (GRU) (Chung *et al.* (2014)). The message $\mathbf{m}_{ij}$ passing from node i to j is updated as:

$$\mathbf{m}_{ij} = \mathbf{GRU}(\mathbf{x}_i, \{\mathbf{m}_{ki}\}_{k\in N(i)\setminus j}) \tag{6}$$

where $\mathbf{x}_i$ represents the type of substructure i, and $N(i)$ is the neighbor of i. Messages are passed from leaves to a randomly selected root and then from the root to leaves. After message passing, tree representation $\mathbf{h}_T$ is produced by aggregating messages relevant with the root node. The tree latent vector $\mathbf{z}_T$ is sampled in a similar way with $\mathbf{z}_G$.

Reconstruct molecules from latent vectors. With given latent vector z_G and z_T, the decoder of tree VAE generates a junction tree from z_T at first, and then the decoder of graph VAE combines the substructures into the junction tree to produce the final reconstructed molecule.

The tree decoder starts from the root, and traverse the junction tree in depth-first order recurrently. It predicts the probability whether the current node has children. Every time a child node is generated, the label of the child node is predicted. Nodes in junction tree are labeled with the valid substructure that it is most likely be. The decoder of graph VAE follows the order where the junction tree is reconstructed, and only assembles one node at a time. While there might exist many ways to assemble substructures, assembling strategy which scores the highest is adopted (Jin *et al.* (2018)).

In this research, we take advantage of JTVAE model pre-trained on ZINC dataset. Similar with geneVAE, we take the predicted mean value of latent vectors as encoded features instead of sampling these vectors from a Gaussian distribution.

3.5 Drug response prediction network

As illustrated in Figure 2, we implement two multi-layer perceptron (MLP) models to post-process the latent features encoded by the two VAE models, respectively. Another MLP is implemented to concatenate the processed output 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:

$$a_{all}^{i+1} = f^{'}(\mathbf{W}^{(i+1)T} a_{all}^{i} + \mathbf{b}^{i+1}) \tag{7}$$

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

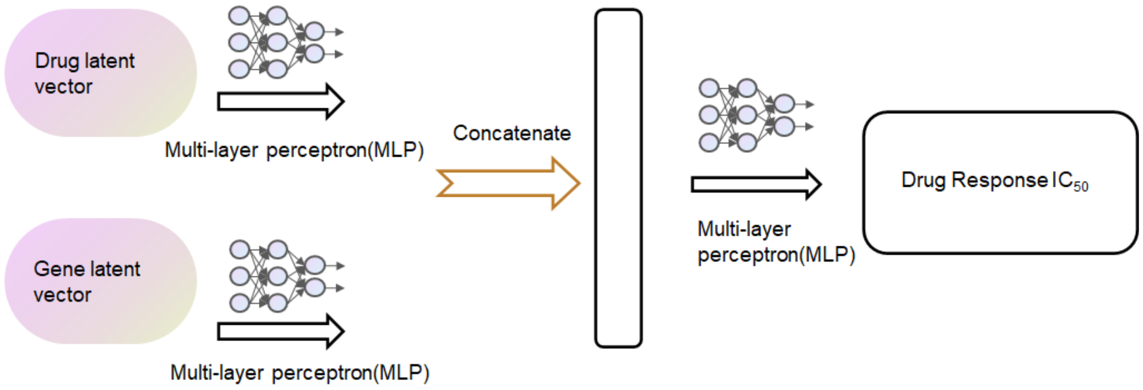

Drug latent vector
Multi-layer perceptron(MLP)
Concatenate
Gene latent vector
Multi-layer perceptron(MLP)
Multi-layer perceptron(MLP)
Drug Response IC_{50}

Fig. 2. The architecture of drug response network to produce a final prediction. Two 3-layer MLP models post-process the encoded gene latent vectors and drug latent vector, respectively, and then another 4-layer MLP concatenate the output and produce a predicted $ln(IC_{50})$ value.

IC_{50} is computed at the last layer of the final MLP model:

$$ln(IC_{50}) = f^{'}(\mathbf{W}^{(n)T}a_{all}^{n-1} + \mathbf{b}^{n}) \tag{8}$$

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, respectively, we set sizes of the 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.

3.6 Baseline model

We use Support Vector Regression (SVR) in substitution for MLP as our baseline model, which shows 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. GeneVAE and JTVAE are trained unsupervisedly at the first stage. Pre-trained geneVAE is used to encode gene expression data either filtered by CGC data set or not on breast cancer cell lines, respectively, and JTVAE is used to encode anti-cancer drug molecular data. With these encoded features, we train the baseline SVR model and MLP model for drug response prediction. Our model is tested breast cancer cell lines at first, and then tested on pan-cancer cell lines. Besides drug response prediction, we also demonstrate that our model can generate effective drugs for given cancer cell lines. 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 reconstruction loss in our experiments, since we normalize the input data and use sigmoid as activation at the last layer to make sure that inputs and outputs are values between 0 and 1.

We pre-train geneVAE model on cancer cell line gene expression data either filtered by CGC dataset or not. During training, a warm-up strategy is adopted. The total VAE loss is set as:

$$\mathbf{VAE\_Loss} = \mathcal{L}(G, G^{'}) + \beta KL \tag{9}$$

where KL is KL loss and β is a parameter which gradually increases from 0 to 1 during training. Because batchnorm layers are incorporated in geneVAE, the learning rate is set as 0.1 initially for a faster learning. A learning rate decay strategy is also adopted in the training process where the learning rate is multiplied by 0.8 when validation loss is fluctuating in a range of 0.5 for over 10 epochs. The minimum learning rate is set as 0.01. In our tests, the total VAEloss (-ELBO) start to convergent after approximately 100 epochs. As shown in Figure 3 and Figure 4, our model on CGC-selected gene expression data has an average VAEloss of 27.3, and the model without CGC selected gene expression data has an average VAEloss of 68 after validation loss becomes stable.

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: $1)\mathbf{CGC + SVR}$: Support Vector Regression model trained on drug molecular structure data encoded by JTVAE and gene expression data filtered by CGC dataset. $\mathbf{2})\mathbf{CGC + VAE + SVR}$: Support Vector Regression model trained on drug molecular data encoded by JTVAE, along with gene expression data filtered by CGC dataset and encoded by geneVAE. $\mathbf{3})\mathbf{CGC + MLP}$: Multi-Layer Perceptron model trained on drug molecular structure data encoded by JTVAE and gene expression data filtered by CGC dataset. $\mathbf{4})\mathbf{RAW + VAE + MLP}$: Multi-Layer Perceptron model trained on drug molecular data encoded by JTVAE and raw gene expression data (not filtered by CGC dataset) encoded by geneVAE. $\mathbf{5})\mathbf{CGC + VAE + MLP}$: Multi-Layer Perceptron model trained on drug molecular structure data encoded by JTVAE, along with gene expression data filtered by CGC and encoded by geneVAE. $\mathbf{6})\mathbf{CGC + VAE + MLP}$: Multi-Layer Perceptron model trained on drug molecular structure data encoded by JTVAE, along with gene expression data filtered by CGC and encoded by geneVAE. This model is trained on pan cancer dataset.

A comparison of performance of our proposed models are listed in Table 1, and scatter plots, which illustrates the relation between true

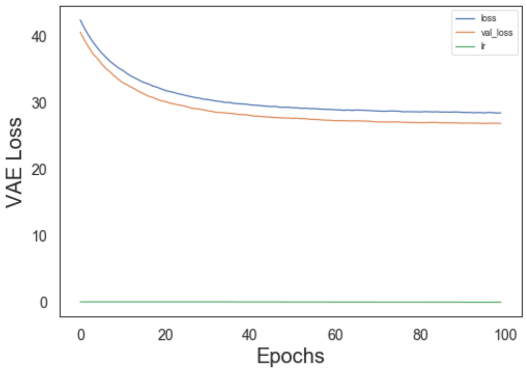


Fig. 3. VAEloss and lr with CGC

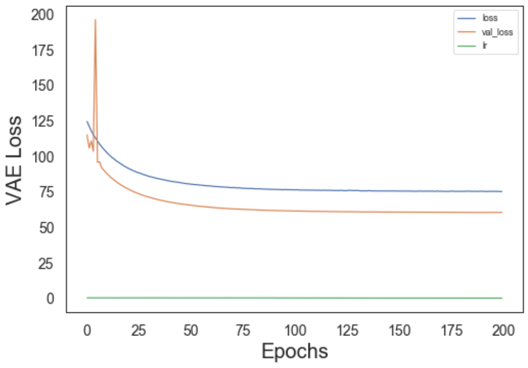


Fig. 4. VAEloss and lr without CGC

Table 1. Performance of 6 proposed models on breast and pan cancer dataset

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

values and predicted values from the models on test sets, are shown in Figure 5 to Figure 10. Results indicate that MLP and VAE improve the performance of our models a lot: $\mathbf{CGC + MLP}$ 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, filtering out an important gene subset with CGC dataset is also essential to the performance of our models. For example, $\mathbf{CGC + VAE + MLP}$ model on breast cancer cell lines reaches 0.830 R2 score, which outperforms $\mathbf{RAW + VAE + MLP}$ model with a 0.025 higher R2 score.

4.3 Test on pan-cancer

We further test our model on the pan-cancer cell lines from CCLE dataset. The total number of pan-cancer cell lines is 1021, and 13605 pieces of anti-cancer drug response data are used to train and test our model. The $\mathbf{CGC + VAE + MLP}$ model, which performs best on breast cancer cell lines, achieves an even higher $R^2 score$ 0.845 on pan-cancer cell lines.

4.4 Effective drug compound generation

Compared with other representation learning methods on molecules, JTVAE has the advantage of reconstructing 100% valid drugs, which makes it considerably powerful in generating effective drugs for specific cancer cell lines. In our experiments, HCC1187, which is a breast cancer cell line, is taken as an example to demonstrate how our model is used to generate customized effective drug compounds for a given cancer cell line. Firstly, we sample several 56 dimension vectors, which are the same in dimensionality as latent vectors encoded by JTVAE, from Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 7$. The randomly sampled drug vectors are concatenated with the encoded latent vector of gene expression profile of HCC1187. The MLP model ingest the concatenated vectors and 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, which produces more effective generated drugs. We select 10 generated drug latent vectors whose $ln(IC_{50})$ values on HCC1187 are below -1.0 and decode them with JTVAE model. Results are shown in 11. As JTVAE always decode drug latent vectors into valid compounds, these decoded drug compounds, which might have not been used as anti-cancer drugs ever before, indicate promising ways for anti-cancer drug discovery.

4.5 Exploring latent vectors from geneVAE

In this section, we disclose that latent vectors encoded by geneVAE retains critical features of pan-cancer gene expression data. We adopt t-SNE method to reduce the dimensionality of both original gene expression data (filtered by CGC dataset) and latent vectors of gene expression data encoded by geneVAE, and visualize them to reveal their similarity. We begin with giving each cell line its tissue type, from "CERVIX" to "OVARY". Especially, "HAEMATOPOIETIC_AND_LYMPHOID_TISSUE" is renamed as "HALT" for brevity. Parameters are perplexity and the number of iterations for the single t-SNE model. We set perplexity to n/120, where n is the number of cell lines and set the number of iterations as 3000. We further eliminate cancer types where the number of cancer tissues is below 30 for a better visualization result. Twelve main cancer types are reserved, which are `[BREAST, CENTRAL_NERVOUS_SYSTEM, FIBROBLAST, HALT, KIDNEY, SKIN, STOMACH]` and etc. After tissues of rare cancer types are removed, the data is visualized Figure 12 and Figure 13. The results of encoded latent vectors and those of original data remain similar, where primary cancer tissue types, which have been marked with black boxes, are separated clearly. Therefore, latent vectors encoded by geneVAE model retain the essential features of original data. With geneVAE, our models are able to focus on the low-dimensional critical features of original data and produce more accurate predictions.

4.6 Exploring latent vectors from JTVAE

Many drugs which shares similar latent vectors encoded by JTVAE are also similar in their molecular structures. We measure the similarity of latent vectors of different drugs in terms of Euclidean distance. Shorter distance indicates a higher similarity between two drug latent vectors. For example, MG132 (inhibitor) and Proteasome (inhibitor) share a short Euclidean distance between their latent vectors which is about 23.73. We obtain their molecular structures from Pubchem database and find that they share a majority of functional groups, which is shown in Figure 14. Small differences lie in a carboxyl and an amide at the endings of the molecule.

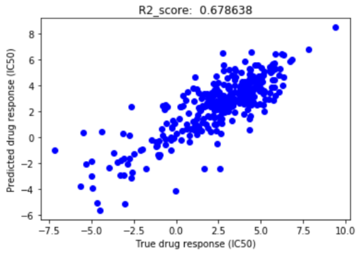


Fig. 5. CGC+SVR

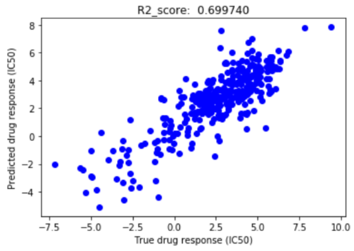


Fig. 6. CGC+VAE+SVR

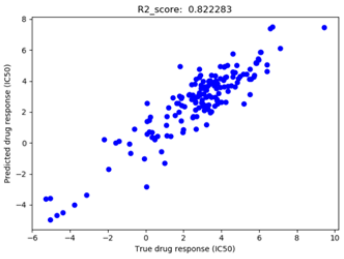


Fig. 7. CGC+MLP

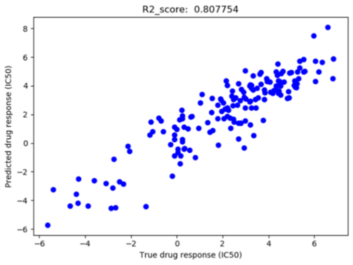


Fig. 8. Raw+VAE+MLP

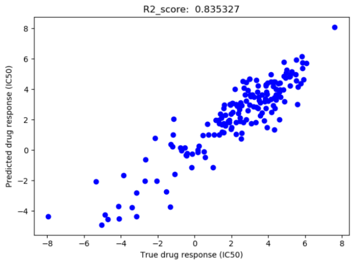


Fig. 9. CGC+VAE+MLP

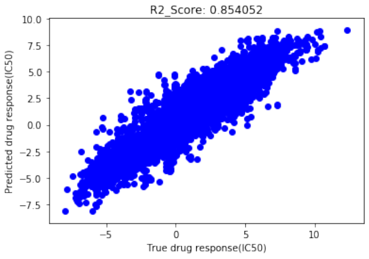


Fig. 10. CGC+VAE+MLP (tested on pan cancer cell lines)

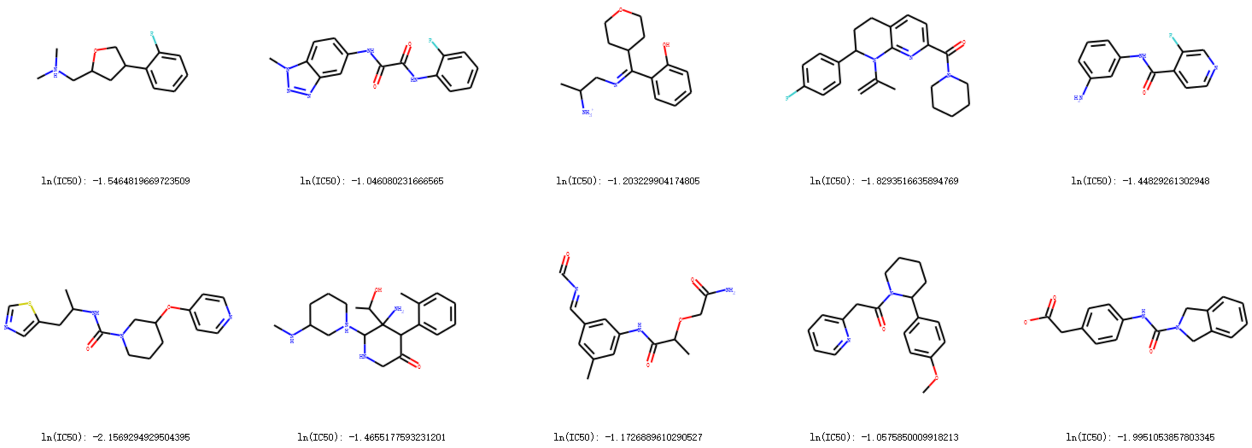

ln(IC50): -1.5464819669723509
ln(IC50): -1.046080231666565
ln(IC50): -1.203229904174805
ln(IC50): -1.8293516635894769
ln(IC50): -1.44829261302948
ln(IC50): -2.1569294929504395
ln(IC50): -1.4655177593231201
ln(IC50): -1.1726889610290527
ln(IC50): -1.0575850009918213
ln(IC50): -1.9951053857803345

Fig. 11. Ten effective drugs whose $ln(IC_{50})$ values on cancer cell line HCC1187 are below -1.0.

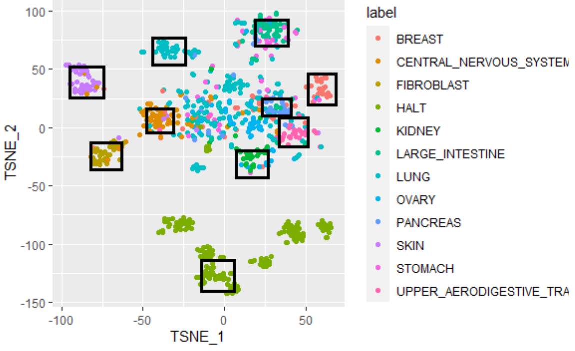


Fig. 12. T-SNE results of original gene expression data

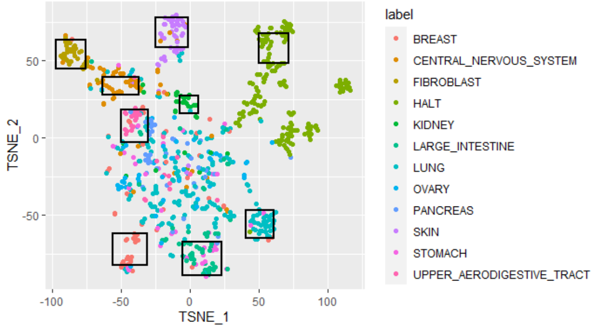
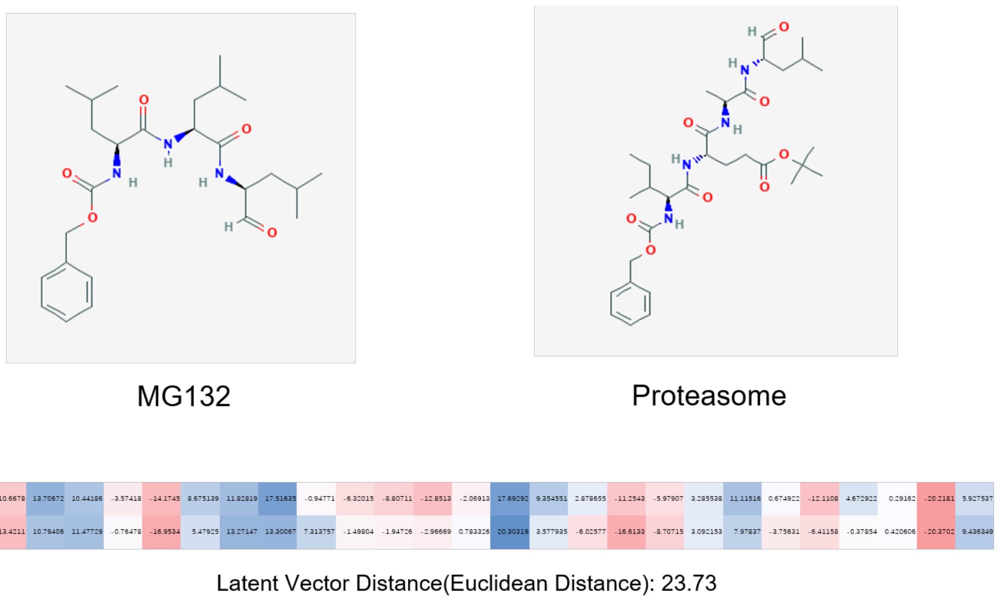


Fig. 13. T-SNE results of latent vectors encoded by geneVAE



MG132	15.59756	-30.9778	-10.6678	13.70672	10.44186	-3.57418	-14.1745	8.675139	11.82819	17.51635	-0.94771	-6.32015	-8.80711	-12.8513
Proteasome	15.48884	-31.3183	-13.4211	10.79406	11.47729	-0.76478	-16.9534	5.47925	13.27147	13.30067	7.313757	-1.49804	-1.94726	-2.96669

MG132	-2.06913	17.69292	9.354551	2.878655	-11.2543	-5.97907	3.285538	11.11516	0.674922	-12.1108	4.672922	0.29162	-20.2181	5.927537
Proteasome	0.783326	20.30319	3.577935	-6.02577	-16.6133	-8.70715	3.092153	7.97837	-3.75631	-6.41158	-0.37854	0.420606	-20.3702	9.436349

Latent Vector Distance(Euclidean Distance): 23.73

Fig. 14. MG132 and Proteasome, which are close in terms of the Euclidean distance between their latent vectors, share a majority of common functional groups.

Although many drugs are similar in their latent vectors, their performance are distinct when used against different cancer cell lines. However, our drug prediction network can capture these subtle differences and produce accurate predictions. 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 $ln(IC_{50})$ of MG132 and Proteasome in cell line HCC1187 are 0.84 and -0.866 in our best model, while true $ln(IC_{50})$ value of these two drugs are 1.589 and -0.181, respectively. Although the predicted values are not so close to the expected ones, our model do not mistake these two samples. Therefore, despite the considerably high similarity between similar drugs, our drug prediction network is still able to differentiate each of them and produce results.

5 Conclusions

Since it is extremely expensive and time-consuming to develop new anti-cancer drugs and make personalized cancer treatment therapy, we seek to use VAE and MLP models to produce accurate predictions of anti-cancer drug efficacy, and generate effective drugs for given cancer cell lines. We take advantage of junction tree VAE (JTVAE) (Jin *et al.* (2018)) model and build gene expression VAE (geneVAE) model, which take as input SMILES representation of drugs and gene expression profile of cancer cell lines, respectively. JTVAE and geneVAE encode these data into representative low dimensional features, which are ingested by MLP model to make drug efficacy prediction. We propose several models and compare their performance on breast cancer and pan cancer cell lines. Results indicate that both encoding original data with VAE models and curating a gene subset with CGC dataset contribute to a better performance. Our best $\mathbf{CGC + VAE + MLP}$ model achieve a great coefficient of determination value (0.845 R2 score on pan-cancer and 0.830 on breast cancer). In addition, we demonstrate that our model can function as a generative model to produce effective anti-cancer drugs for given cancer cell lines. The latent vectors encoded by geneVAE and JTVAE are also explored to demonstrate the validity of our pipeline.

6 Future work

Since the validity of filtering out a gene subset with CGC dataset is demonstrated in our experiments, the importance of selecting representative gene subsets is worth discussing. More promising methods such as network propagation based on STRING protein-protein interaction database can be used to further improve our model (Oskooei *et al.* (2018)). Moreover, attention mechanism based models which improves the interpretability of our model can be incorporated (Manica *et al.* (2019)). Besides, recent work based on graph neural networks (GNN) shows the potential of GNN in dealing with drug molecular data. A combination of VAE and GNN (VGAE) (Kipf and Welling (2016)) can be adopted for this problem. VGAE takes advantage of VAE model to be generative, and adopt GNN to process graph data efficiently. We believe a modified version of the originally proposed VGAE (Kipf and Welling (2016)) is a promising way to predict drug response and generate new drugs. Finally, since our model performs well on drug response prediction and is very potential for drug discovery, we would like to build a toolkit based our model.

Acknowledgement

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

References

Barretina, J. *et al.* (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483**(7391), 603–607.

Chang, Y. *et al.* (2018). Cancer drug response profile scan (cdrscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Scientific reports*, **8**(1), 1–11.

Chiu, Y.-C. *et al.* (2019). Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC medical genomics*, **12**(1), 18.

Chung, J. *et al.* (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Duvenaud, D. K. *et al.* (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232.

Geeleher, P. *et al.* (2014). 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.

Gilmer, J. *et al.* (2017). Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*.

Grønbech, C. H. *et al.* (2018). scvae: Variational auto-encoders for single-cell gene expression datas. *bioRxiv*, page 318295.

Huang, H. and Kim, K. (2006). Unsupervised clustering analysis of gene expression. *Chance*, **19**(3), 49–51.

Imamura, T. *et al.* (2013). 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.

Jin, W. *et al.* (2018). Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kusner, M. J. *et al.* (2017). Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*.

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

Li, Y. *et al.* (2018). Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*.

Liu, Q. *et al.* (2018a). Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, pages 7795–7804.

Liu, Q. *et al.* (2020). Deepcdr: a hybrid graph convolutional network for predicting cancer drug response. *bioRxiv*.

Liu, S. *et al.* (2018b). Feature selection of gene expression data for cancer classification using double rbf-kernels. *BMC bioinformatics*, **19**(1), 1–14.

Manica, M. *et al.* (2019). Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Molecular Pharmaceutics*.

Oskooei, A. *et al.* (2018). Paccmann: Prediction of anticancer compound sensitivity with multi-modal attention-based neural networks. *arXiv preprint arXiv:1811.06802*.

Rampasek, L. *et al.* (2017). Dr. vae: Drug response variational autoencoder. *arXiv preprint arXiv:1706.08203*.

Schmainda, K. M. *et al.* (2014). 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.

Simonovsky, M. and Komodakis, N. (2018). Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer.

Tsubaki, M. *et al.* (2019). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**(2), 309–318.

Wenric, S. and Shemirani, R. (2018). Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, **9**, 297.

Yang, W. *et al.* (2012). Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, **41**(D1), D955–D961.

Yuasa, T. *et al.* (2011). Biomarkers to predict response to sunitinib therapy and prognosis in metastatic renal cell cancer. *Cancer science*, **102**(11), 1949–1957.