Variational Autoencoder for Anti-Cancer Drug Response Prediction

Hongyuan Dong,[†] Jiaqing Xie,[*,‡] Zhi Jing,[¶] and Dexin Ren[§]

[†]*School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China*

[‡]*School of Engineering, University of Edinburgh, UK*

[¶]*School of Data and Computer Science, Sun Yat-sen Universit*

[§]*School of Computer Science, University of Arizona*

E-mail: s2001696@ed.ac.uk

Phone: +123 (0)123 4445556. Fax: +123 (0)123 4445557

Abstract

Cancer is a primary cause of human death, but discovering drugs and tailoring cancer therapies are expensive and time-consuming. We seek to facilitate the discovery of new drugs and treatment strategies for cancer using variational autoencoders (VAEs) and multi-layer perceptrons (MLPs) to predict anti-cancer drug responses. Our model takes as input gene expression data of cancer cell lines and anti-cancer drug molecular data and encodes these data with our GeneVae model, which is an ordinary VAE model, and a rectified junction tree variational autoencoder (JTVae) model, respectively. A multi-layer perceptron processes these encoded features to produce a final prediction. Our tests show our system attains a high average coefficient of determination ($R^2 = 0.83$) in predicting drug responses for breast cancer cell lines and an average $R^2 > 0.84$ for pan-cancer cell lines. Additionally, we show that our model can generates effective drug compounds not previously used for specific cancer cell lines.

Introduction

The discovery of new drugs and the customization of cancer therapy remain difficult problems. Cancer drugs are a widely used primary treatment. However, development of these drugs is expensive and time-consuming, and it is difficult to tailor therapy to individual patients. We propose a generative model for accurate prediction of anti-cancer drug response to help with this critical need.

The effectiveness of cancer drugs is highly dependent on the genomic and transcriptomic profile of the specific cancers.[1] Some researchers have predicted drug response using gene expression data. Chiu et al.[2] build deep neural networks to combine gene expression with mutation profiles to make predictions, and Geeleher et al.[3] implement a ridge regression model on before-treatment gene expression data to predict response of chemotherapy. Our strategy incorporates both gene expression and anti-cancer drug molecular data to predict responses of different drugs on various cancer cell lines.

There are many ways to extract representative low-dimensional features from unlabeled data. Among these, principal component analysis (PCA), independent component analysis (ICA), and manifold learning based t-distributed stochastic neighbor embedding (TSNE) are commonly used to analyze medical data with unlabeled, high-dimensional features. However, these methods are used primarily for 2D visualization and often lose important information when compressing the data into low dimensional latent features. Auto-encoders have been used widely to extract low dimensional features, but they are not very robust, with slight variances in the encoded vector sometimes leading to huge differences in the reconstructed data. Other feature extraction methods such as graph convolutional networks (GCNs), can be used in unlabeled molecular drug data.[4] But GCNs only focus on encoding original data and do not function as the generative models of great significance in drug discovery.

To extract features from huge amounts of unlabeled data, we employ a variational autoencoder (VAE),[5] which encodes the distribution of latent features instead of producing specific latent features directly. An ordinary VAE model (GeneVAE) with its encoder and

decoder both composed of 2-layer neural networks is implemented for gene expression profile of cancer cell lines. For analyzing anti-cancer drugs, we adopt a junction tree VAE (JTVAE)[6] model to transform the molecular graphs into valid substructures to extract their low dimensional features. JTVAE is also a generative model and outperforms many previous approaches[7–9] in reconstructing molecules. Drug compounds generated by JTVAE are always valid, making it extremely powerful for discovering new anti-cancer drugs. Our research shows that encoded features of drugs can be randomly sampled, with the well-performing features decoded by JTVAE to reveal a large number of valid compounds effective for cancer therapy.

Using the encoded low-dimensional features of the gene expression and drug molecular data, we implement a multi-layer perceptron (MLP) to combine the extracted features and produce the final result, which is the $ln(IC_{50})$ value of the target anti-cancer drug used against a specific cancer cell line. Differing from previous works with models restricted to specific drugs,[10–12] our model can take any organic compound as input to predict its usefulness in treatment. Moreover, combined with JTVAE, our method can propose a number of organic compounds that are potentially effective in cancer therapy. Thus, our method offers promise in reducing the development cost of new drugs.

We also incorporate additional datasets in our work to improve performance. Using continually advancing cancer research, additional information is available to make more accurate predictions of anti-cancer drug response. For example, the Cancer Genomic Census (CGC)[13] dataset, which contains a number of genes highly relevant to cancer, can be used to curate a gene subset from the cancer gene data, removing a significant amount of useless information. We take advantage of the CGC dataset to filter out a representative gene subset from original gene expression data and compare the prediction results with those of models not using the CGC dataset.

Present work

Our present work focuses on learning representative low-dimensional embeddings of original data with variational autoencoders (VAEs) and using these features to predict drug response and generate effective drugs. We use the gene expression level as cancer gene data and the 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. We implement an MLP model using the extracted features, producing the $ln(IC_{50})$ value, where IC_{50} is the half maximal inhibitory concentration value of the drug used against a specific cancer cell line, as its final prediction. We choose the coefficient of determination (R^2) and the root mean squared error ($RMSE$) as metrics to evaluate our drug response prediction. For datasets, we adopt the Cancer Cell Line Encyclopedia (CCLE) gene expression dataset,[14] the CGC dataset,[13] the ZINC molecular structure dataset, and the GDSC drug response dataset.[1] We use our model on breast cancer cell lines at first and then test it on pan-cancer cell lines. We demonstrate that our model can generate chemical compounds that are effective for specific cancer cell lines. We also explore the latent representations encoded by geneVAE and JTVAE to demonstrate the robustness of our model.

Related work

Feature dimensionality reduction

Encoding features into lower dimensions is commonly used for representation learning tasks. The reduction in feature dimensionality removes a large amount of redundant information to facilitate the analysis. Supervised learning methods can select features which are most relevant with the task. As examples, Wenric and Shemirani[15] use random forests to determine gene importance in RNA sequence case-control studies, and Liu et al.[16] implement support vector machines (SVM) with double RBF-kernels to filter out irrelevant gene features. Un-

supervised learning methods, such as PCA and hierarchical learning are useful in explaining the group features of genes while reducing the feature dimensionality.[17] Auto-encoders based on neural networks also learn to encode original data into low-dimensional features without supervision and have been widely used to extract low-dimensional features. Though auto-encoders outperform traditional methods, their robustness is unpredictable; slight variances in the encoded vector can lead to huge differences in the reconstructed data. Variational auto-encoders,[5] which add noise to the encoded features to build a more robust auto-encoder model, have been proposed to overcome this weakness.

Variational auto-encoders with gene profiles

Much work has been done on encoding gene expression data into representative low-dimensional features. Neural networks, such as MLPs and convolutional neural networks (CNNs), can encode gene features effectively. Chang et al.[18] use a CNN to encode gene mutation and drug molecular data, and Oskooei et al.[19] implement attention-based neural networks to produce explainable encoded features. An encoder-decoder structure[2] extends ordinary MLPs that are also able to reconstruct the original input. The bottleneck layer represents the latent features encoded by autoencoders. Recently, the VAE,[5] which modifies ordinary auto-encoders to improve robustness, has been used frequently in pre-trained models for gene expression data. Grønbech et al.[20] use a VAE to estimate expected gene expression level, and Rampasek et al.[21] implement VAE models to analyze pre- and post-treatment gene expression profiles. We also incorporate the VAE model to process gene expression data. Latent features of gene expression data provide representative information about cancer cell lines that enables our model to predict drug responses for different cancer tissues.

Representation learning on graphs for drug molecular features

Drug molecular features can be represented as graphs and processed by deep neural networks. Duvenaud et al.[22] build graph CNNs on circular fingerprints of molecules. Liu et al.[4]

implement uniform graph convolutional neural networks (UGCNs) to extract representative features from drug molecular data. Gilmer et al.[23] use a message passing neural network (MPNN) for molecular property prediction.

In addition, attention mechanisms can be used with RNN and CNN models[19,24] to encode drug molecular data, learning attention weights by multihead-attention or self-attention to produce explainable encoded features. The VAE is also widely used in tasks that require the models to be generative. Kusner et al.[7] propose a grammar-based VAE and use parse trees to produce more valid generated output, and Simonovsky and Komodakis[9] label the nodes and bonds in molecules to form a graph structure and apply the VAE model on it. Li et al.[8] use a 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.[6] propose a JTVAE that decomposes molecules into valid substructures and generates compounds from a vocabulary of valid components. As a result, the molecules generated by JTVAE are always valid. For this reason, we make use of the JTVAE as our pre-trained model to encode molecular drug data and generate effective drugs for cancer cell lines.

Drug response prediction methods

Drug response prediction is a supervised regression task. Support vector regression (SVR) and random forest regressors are basic algorithms to perform regression. Recently, deep neural network methods have become popular in drug efficacy prediction. Chiu et al.[2] build deep neural networks to analyze gene expression and mutation profiles to make predictions, and Chang et al.[18] use CNN-based methods on gene mutation profiles and drug molecular data. Liu et al.[4] also use gene mutation data and drug molecular data and apply CNNs and UGCNs to make predictions, while Oskooei et al.[19] implement attention-based neural networks for gene expression and molecular drug data to make explainable predictions. In

our approach, we implement a MLP model for encoded gene expression and drug molecular data to make predictions.

Materials and methods

In this section we present our strategy for processing the datasets along with our model implementation. Our model takes as input the gene expression data of a cancer cell line and the SMILES representation of an anti-cancer drug, and produce a drug response prediction in terms of $ln(IC_{50})$. The model consists of geneVAE, an ordinary VAE, to extract features from the gene expression data, a JTVAE to extract features from the molecular drug data, and an MLP model to produce a final prediction.

Data

Gene expression data

We use gene expression data of 1021 cancer lines with 57820 genes provided by the CCLE.[14] Each cell line belongs to a specific cancer type. Specifically, we choose breast cancer as our primary research object, and later test our model on pan cancer cell lines. After filtering by the key word token `[BREAST]`, we select 51 breast cancer cell lines from this dataset: `[AU565_BREAST]`,`[BT20_BREAST]`, `[ZR7530_BREAST]`, and so on. 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 the transcriptome per million (tpm) value of the gene in the corresponding cell line.

We also use the CGC dataset,[18] 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 into cancers that change the activity of the gene product. The other tier includes genes that possibly play a strong role in cancer but lack evidence. Genes in both tiers are highly relevant with cancer, which is why we incorporate both tiers. We select 51 breast cancer cell

lines from the CCLE data set and remove expression data of genes which are not in the 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 due to their low relevance to cancer cell lines.[2] Our final set contains gene expression data of 597 genes in 51 breast cancer cell lines.

Anti-cancer drug molecular structure data

In our research, we prepare the 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. The SMILES representation is often used to define drug structures[6,7,9,18,19,24–26] and are widely used as inputs for drug structure prediction. The SMILES representation simplifies obtaining the embeddings from the vocabulary parsing library we have generated. From the ZINC data set, we select 10000 SMILES strings to train our JTVAE model. The number of SMILES strings used for pre-training is far larger than the actual number of 222 drugs in the processed GDSC dataset. The reason is that we would like to improve our model's robustness with all drugs, not just anti-cancer drugs.

Drug response data

We use drug response data from the Genomics of Drug Sensitivity in Cancer (GDSC) project,[1] which contains response data for cancer drugs against numerous cancer cell lines. Data from the GDSC data set is given by a matrix $IC_{CCLE} \in R^{d \times c}$, where d is number of drugs, and c is the number of cancer lines. The elements in this matrix are $ln(IC_{50})$ values, where IC_{50} is the half maximal inhibitory concentration value of the drugs used against specific cancer cell lines. We obtain molecular data for the drugs from the PubChem dataset with their unique PubChem ID available from the GDSC dataset. In total, we have 3358 pieces of drug response data for breast cancer cell lines where gene expression data and drug molecular structure are available.

Variational Auto-encoder

The variational auto-encoder is a generative model, modeling the complicated conditional distribution of latent features with an inference network (encoder) and a generative network (decoder). In describing the VAE, $q(z|x;\phi)$ is the distribution of approximated latent variables with parameter set ϕ, where $p(x|z;\theta)$ is the conditional probability distribution computed by the generative network (decoder) with parameter set θ. The aim of VAE is to find the parameters ϕ^* and θ^* to maximize ELBO(ϕ,θ):[5]

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

In a VAE, the prior distribution of latent variables $p(z;\theta)$ is approximated as a normal Gaussian distribution, and the posterior $q(z|x;\phi)$ is also expected to follow a Gaussian distribution. The conditional probability distribution $p(x|z;\theta)$ should follow a multivariate Gaussian distribution. Given these suppositions, the estimator for this model and datapoint x is[5]):

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

where J is the dimensionality of latent variable z, and L is number of samples used to compute $\mathbb{E}_{q(z|x;\phi)}[\log p(x|z;\theta)]$ approximately. In practice, the total loss of the VAE model is set to be the opposite number of ELBO, which satisfies the gradient descent requirement. Because $\mathrm{z} \sim \mathcal{N}(\mu,\sigma^2)$, one valid reparameterization of z to enable back propagation is $\mathrm{z} = \mu + \epsilon\sigma$, where $\epsilon \sim \mathcal{N}(0,1)$.

Gene expression VAE (geneVAE)

GeneVAE extracts latent vectors from CCLE gene expression data, with the extracted latent vectors used for drug response prediction. 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:

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

where $\tau(\cdot)$ 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 (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 using

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

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. The 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 networks, with the architecture shown in Figure 1. The sizes of both encoder layers are set as 256, while the sizes of both decoder layers are set to match 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.

JTVAE[6]

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 with a tree VAE while the original molecular graph is encoded with a graph VAE. When generating molecules, the decoder of the tree VAE reconstructs the junction tree of the molecule, and the decoder of the graph VAE provides complementary connectivity

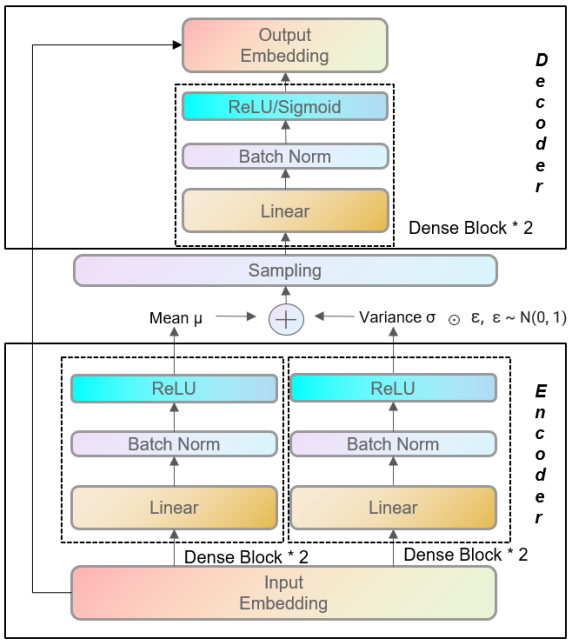

Figure 1: The architecture of geneVAE. The encoder computes parameters μ and σ of the Gaussian distribution $\mathcal{N}(0, 1)$ with separate dense blocks. Sampled latent vectors are processed by the decoder where the first layer uses ReLU activation and the second layer uses sigmoid activation to reconstruct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

information to reproduce the full molecular graph.

Graph encoder

The encoder of the graph VAE, which takes atoms as nodes in the graph, is implemented with a message passing network.[23] Messages pass from node to node for t iterations. The final representation of each node is computed by aggregating its relevant messages from the message passing network, with these representations 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 the tree VAE, in contrast, uses valid substructures of the molecule graph as nodes, and implements a message passing network based on a gated recurrent unit (GRU).[27]

The message $\mathbf{m}_{ij}$ passed 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, the tree representation $\mathbf{h}_T$ is produced by aggregating messages relevant to 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

Using the given latent vectors z_G and z_T, the tree VAE decoder generates a junction tree from z_T first, and then the graph VAE decoder combines the substructures into the junction tree to produce the final reconstructed molecule.

The tree decoder starts from the root and traverses the junction tree in depth-first order recursively. It predicts the probability of the current node having children. Every time a child node is generated, the label of the child node is predicted. Nodes in the junction tree are labeled with the most likely valid substructure. The graph VAE decoder follows the order when the junction tree is reconstructed and only assembles one node at a time. While there may be multiple ways to assemble the substructures, JTVAE uses the highest scoring strategy.[6] We make use of a JTVAE model pre-trained with the ZINC dataset. Similar to geneVAE, we use the predicted mean value of latent vectors as encoded features instead of sampling these vectors from a Gaussian distribution.

Drug response prediction network

As illustrated in Figure 2, we implement two MLP models to post-process the latent features encoded by the two VAE models. We implement another MLP model 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 the outputs of the two post-processing MLP models. If $\mathbf{a}_{gene} \in \mathbb{R}^{d_1}$ 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

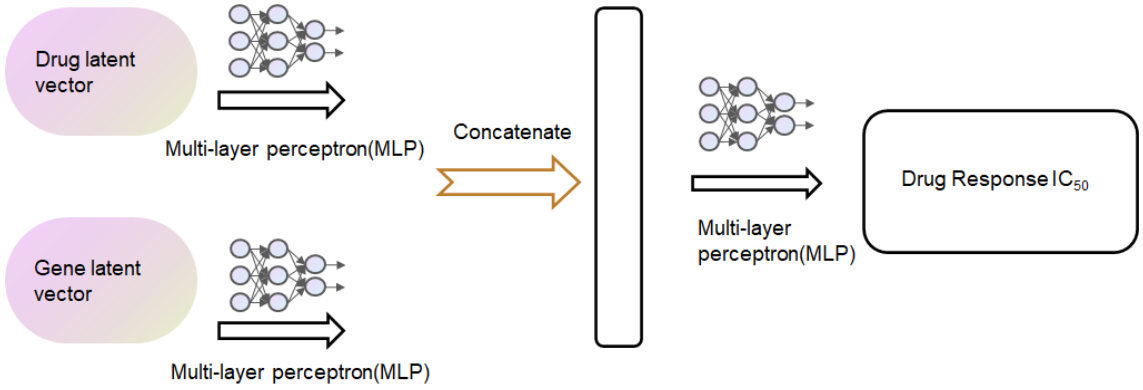

Figure 2: The architecture of the drug response network to produce a final prediction. Two 3-layer MLP models post-process the encoded gene latent vectors and drug latent vector, and then another 4-layer MLP concatenates the output and produces a predicted $ln(IC_{50})$ value

dimensionality of $\mathbf{a}_{gene}$, and d_2 is the dimensionality of $\mathbf{a}_{drug}$. We compute the values of the perceptrons in the i^{th} layer in the final MLP model 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 the latter, we use the parametric rectified linear unit (PReLU) in our model. We complete the predicted $ln(IC_{50})$ in the last layer of the final MLP model according to

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

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

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

Baseline model

We substitute a support vector regression (SVR) network for MLP in our baseline model, showing a convenient way of using machine learning methods to make drug response predictions. We choose a poly kernel in our SVR model and set the parameter C as 10.

Experiments and results

Experiment set-up

For our experiments, we trained geneVAE and JTVAE without supervision at the first stage. We used the pre-trained geneVAE to encode gene expression data filtered by the CGC data set or not on the breast cancer cell lines. We used JTVAE to encode anti-cancer drug molecular data. With these encoded features, we trained the baseline SVR model and MLP model for drug response prediction. We first tested our model on breast cancer cell lines followed by pan-cancer cell lines. Besides drug response prediction, we also demonstrated that our model generated effective drugs for given cancer cell lines. We split the training and test sets in a 9:1 ratio for the SVR models, and the training, validation, and test sets in an 18:1:1 ratio for the MLP models. We implemented and debugged our models using PyCharm running under Microsoft Windows. We trained the model using an Nvidia GeForce RTX 2070 Super GPU.

Pre-training geneVAE

We aimed to minimize the sum of reconstruction and KL losses when training the geneVAE model. The reconstruction loss is $\mathcal{L}(G, G^{'})$, where G represents initial input gene expression data and $G^{'}$ represents reconstructed data. The loss function could be either the mean squared loss `[MSEloss]` or the cross entropy loss `[CrossEntropyloss]`. We chose the cross entropy loss as the reconstruction loss in our experiments, since we normalized the input

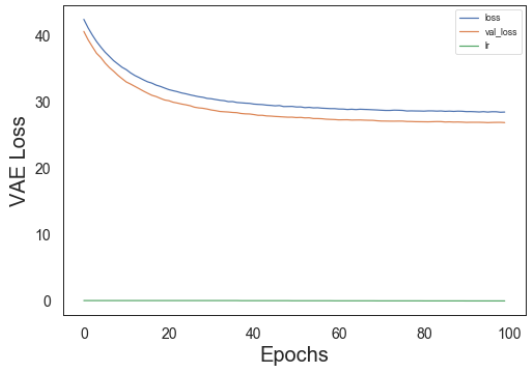

Figure 3: VAEloss and lr with CGC

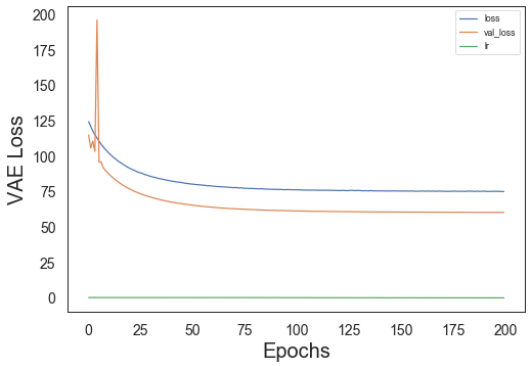

Figure 4: VAEloss and lr without CGC

data and used sigmoid activation in the last layer to ensure that inputs and outputs were values between 0 and 1.

We pre-trained the geneVAE model on cancer cell line gene expression data both filtered by CGC dataset and without filtered. During training, we employed a warm-up strategy. The total VAE loss was set to

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

where KL is the KL loss and β is a parameter that gradually increases from 0 to 1 during training. Because batchnorm layers were incorporated in geneVAE, we set the learning rate as 0.1 initially for a faster learning. We also adopted a learning rate decay strategy in the training process, where the learning rate was multiplied by 0.8 when validation loss fluctuated in a range of 0.5 for over 10 epochs. The minimum learning rate was set as 0.01.

In our tests, the total VAEloss (-ELBO) began to converge after approximately 100 epochs. As shown in Figures 3 and 4, our model on CGC-selected gene expression data had an average VAEloss of 27.3, and the model without CGC selected gene expression data had an average VAEloss of 68 after the validation loss became stable.

Results on breast cancer

We prepared several models and tested them on breast cancer cell lines, with the results showing that the VAE and CGC datasets contributed to more accurate predictions. We selected

2 metrics to evaluate performance. The coefficient of determination (R^2 score) and RMSE evaluated the discrepancy between our predicted drug response and true drug response. We prepared 6 models, with results shown in Table 1. Among these models, the first 5 models targeted breast cancer, and the last one is tested on pan cancer cell lines. The models used were: **1**) $\mathbf{CGC + SVR}$: An SVR model trained on drug molecular structure data encoded by JTVAE and gene expression data filtered by the CGC dataset. **2**) $\mathbf{CGC + VAE + SVR}$: An SVR model trained on drug molecular data encoded by JTVAE, along with gene expression data filtered by the CGC dataset and encoded by geneVAE. **3**) $\mathbf{CGC + MLP}$: An MLP model trained on drug molecular structure data encoded by JTVAE and gene expression data filtered by the CGC dataset. **4**) $\mathbf{RAW + VAE + MLP}$: An MLP model trained on drug molecular data encoded by JTVAE and raw gene expression data (not filtered by CGC dataset) encoded by geneVAE. **5**) $\mathbf{CGC + VAE + MLP}$: An MLP model trained on drug molecular structure data encoded by JTVAE along with gene expression data filtered by CGC and encoded by geneVAE. **6**) $\mathbf{CGC + VAE + MLP}$: An MLP model trained on drug molecular structure data encoded by JTVAE along with gene expression data filtered by CGC and encoded by geneVAE. This model was trained on pan cancer dataset.

Table 1 presents the performance comparison between our proposed models. Scatter plots illustrating the relationship between true and predicted values from the models on the test sets are shown in Figures 5 to 10. Results indicate MLP and VAE improved the performance of our models significantly. The $\mathbf{CGC + MLP}$ model outperformed the $\mathbf{CGC + SVR}$ model by 0.164 on the R^2 score, and the $\mathbf{CGC + VAE + MLP}$ model performed even better than the $\mathbf{CGC + MLP}$ model with a 0.008 higher R^2 score. Filtering out an important gene subset with the CGC dataset was also essential to the performance of our models. For example, the $\mathbf{CGC + VAE + MLP}$ model on breast cancer cell lines reached an R^2 score of 0.830, outperforming the $\mathbf{RAW + VAE + MLP}$ model by 0.025.

Table 1: Performance of the 6 proposed models on breast and pan cancer datasets

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

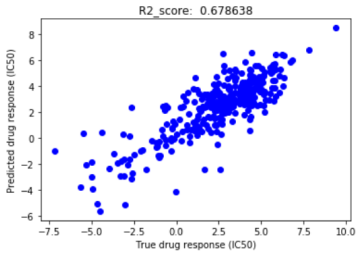

Figure 5: CGC+SVR

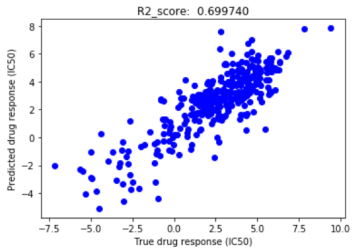

Figure 6: CGC+VAE+SVR

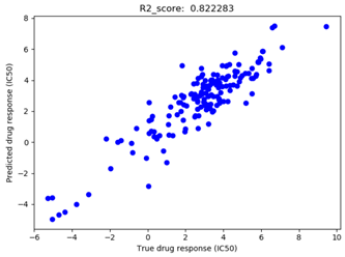

Figure 7: CGC+MLP

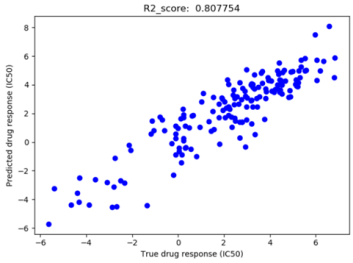

Figure 8: Raw+VAE+MLP

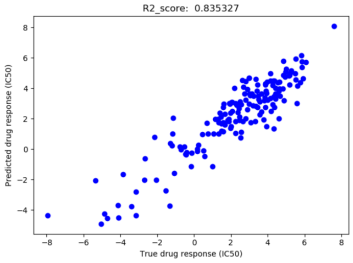

Figure 9: CGC+VAE+MLP

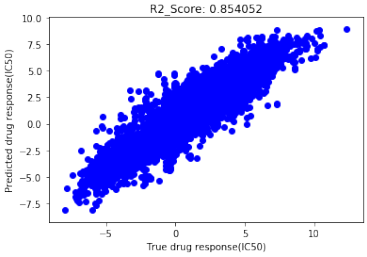

Figure 10: CGC+VAE+MLP (tested on pan cancer cell lines)

Test on pan cancer

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

Effective drug compound generation

Compared with other representation learning methods on molecules, JTVAE has the advantage of reconstructing 100% valid drugs, making it powerful in generating effective drugs for specific cancer cell lines. In our experiments, we used the breast cancer cell line HCC1187 as an example to demonstrate how our model generated customized and effective drug compounds for a given cancer cell line. First, we sampled several 56-dimension vectors to match the dimensionality of the latent vectors encoded by JTVAE from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 7$. The randomly sampled drug vectors were concatenated with the encoded latent vector of gene expression profile of HCC1187. The MLP model ingested the concatenated vectors and produced 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 was below -1.0, it was considered to be effective on HCC1187. Also, the threshold could be set as -1.5, -2.0 etc. to produce more effective generated drugs. We selected 10 generated drug latent vectors whose $ln(IC_{50})$ values on HCC1187 were below -1.0 and decoded them with the JTVAE model. Results are shown in Figure 11. As JTVAE always decode drug latent vectors into valid compounds, these decoded drug compounds, which might have not been used as cancer drugs previously, showed promise for cancer treatment.

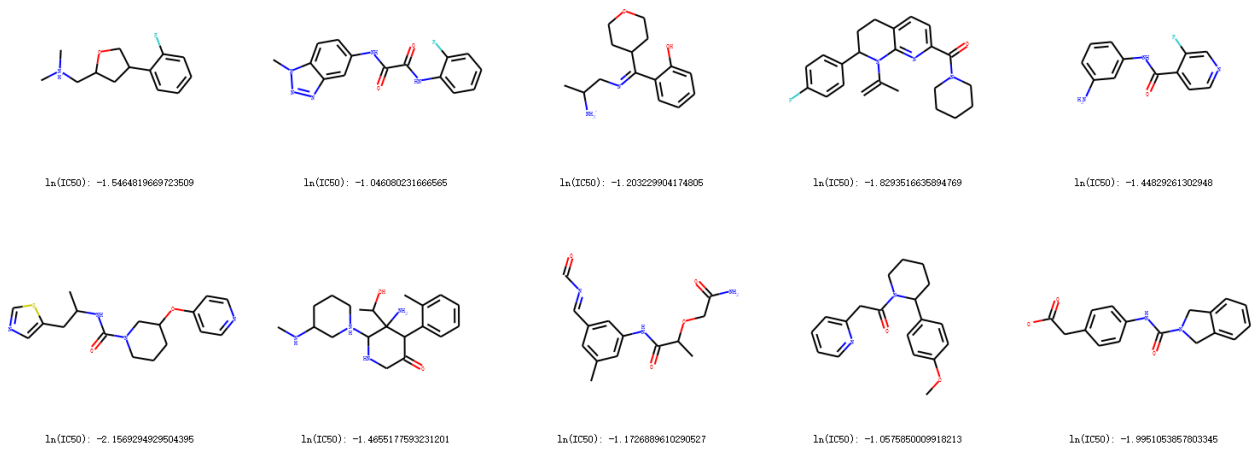

Figure 11: Ten effective drugs whose $ln(IC_{50})$ values on cancer cell line HCC1187 are below -1.0

Exploring latent vectors from geneVAE

In this section, we demonstrate that the latent vectors encoded by geneVAE retained critical features of pan cancer gene expression data. We adopted the t-SNE method to reduce the dimensionality of both the original gene expression data (filtered by CGC dataset) and the latent vectors of gene expression data encoded by geneVAE, and we visualized them to reveal their similarity. We began by labelling the tissue type of each cell line, e.g., "CERVIX" or "OVARY." We renamed "HAEMATOPOIETIC_AND_LYMPHOID_TISSUE" as "HALT for brevity. The parameters consisted of 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 the number of iterations to 3000. We further eliminated cancer types where the number of cancer tissues was below 30 for a better visualization result. Twelve main cancer types remained: `[BREAST, CENTRAL_NERVOUS_SYSTEM, FIBROBLAST, HALT, KIDNEY, SKIN, STOMACH]`. After removing tissues of rare cancer types, we visualize the data as in Figures 12 and 13. The results of the encoded latent vectors and those of the original data remained similar, where primary cancer tissue types (marked with black boxes) are separated clearly. Therefore, latent vectors encoded by geneVAE model retained the essential features of the original data. With

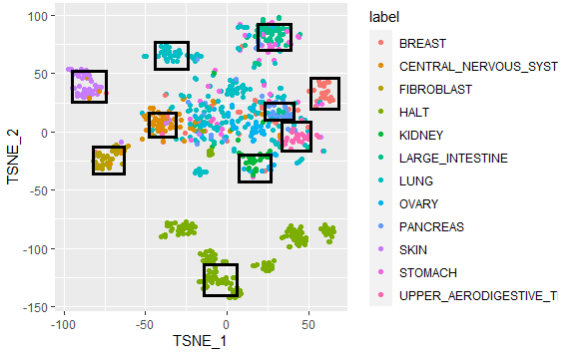

Figure 12: T-SNE results of original gene expression data

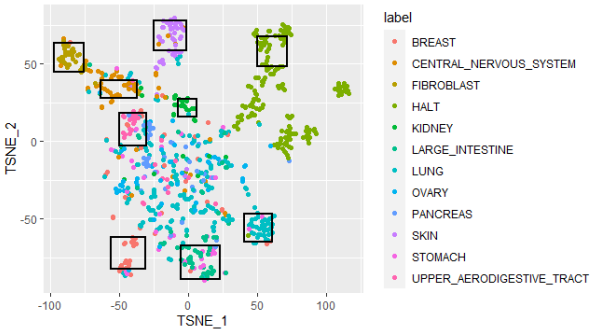

Figure 13: T-SNE results of latent vectors encoded by geneVAE

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

Exploring latent vectors from JTVAE

Many drugs having similar latent vectors encoded by JTVAE are also similar in their molecular structures. We measured the similarity of latent vectors of different drugs in terms of Euclidean distance. Shorter distances indicate a higher similarity between two drug latent vectors. For example, MG132 (inhibitor) and Proteasome (inhibitor) share a short Euclidean distance between their latent vectors of about 23.73. We obtained their molecular structures from the Pubchem database and found that they share a majority of functional groups, as shown in Figure 14. Small differences were found in a carboxyl group and an amide at the ends of the molecules.

Although many drugs are similar in their latent vectors, their performance varies when used against different cancer cell lines. However, our drug prediction network captured these subtle differences and produced accurate predictions. We focused on the example of MG132 and Proteasome used against the HCC1187 cancer cell line. We removed these two pieces of data from the training set, and tested our trained model on them. The predicted $ln(IC_{50})$ of MG132 and Proteasome in cell line HCC1187 were 0.84 and -0.866 in our best model, while the actual $ln(IC_{50})$ values of these two drugs are 1.589 and -0.181, respectively. Although

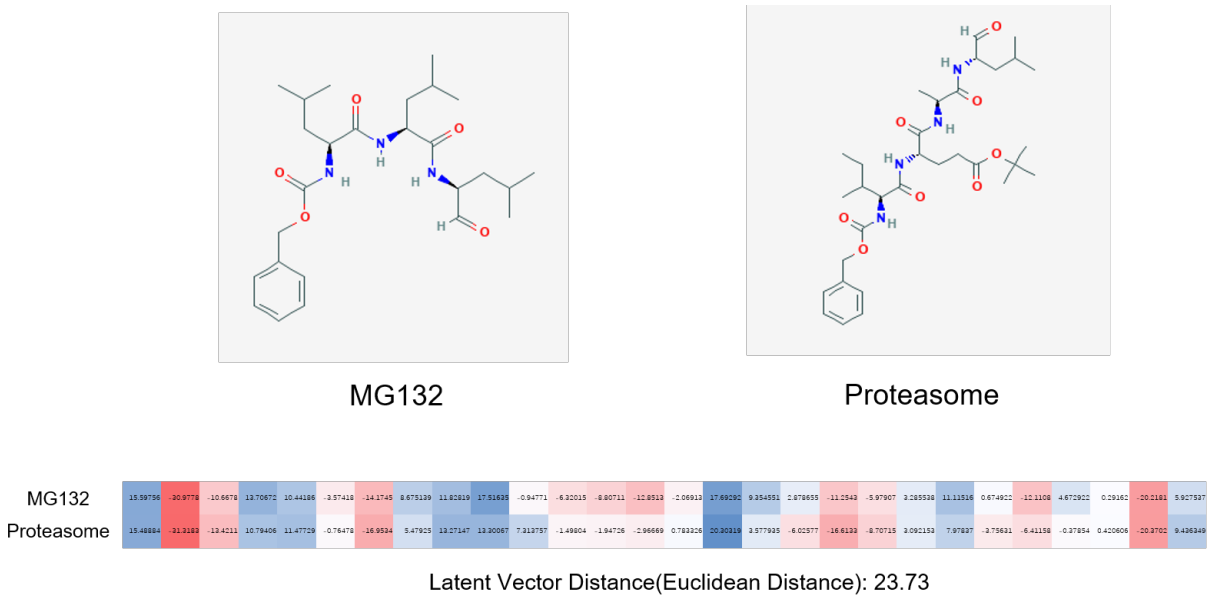

Figure 14: MG132 and Proteasome, which are close in terms of the Euclidean distance between their latent vectors, share a majority of common functional groups

the predicted values were not very close to the expected ones, our models did not confuse these two samples. Therefore, despite strong drug similarities, our drug prediction network still differentiated each of them and produced reasonable results.

Conclusions

Since it is extremely expensive and time-consuming to develop new cancer drugs and propose personalized cancer treatment therapies, we seek to use VAE and MLP models to produce accurate predictions of drug efficacy and to generate effective drugs for given cancer cell lines. We use the JTVAE[6]) model and construct the geneVAE model to process SMILES drug data and gene expression profiles of cancer cell lines, respectively. JTVAE and geneVAE encode these data into representative low dimensional features, which the MLP model uses to make drug efficacy predictions. Our comparison of models using both breast cancer and multiple cancer cell lines show that encoding data using VAE and curating a gene subset with the CGC dataset contribute to better performance. Our best $\mathbf{CGC + VAE + MLP}$

model achieves an encouraging coefficient of determination value (0.845 R^2 score on pan cancer and 0.830 on breast cancer). In addition, we demonstrate that our model works as a generative model to generate effective cancer drugs for given cancer cell lines. We have also explored the latent vectors encoded by geneVAE and JTVAE to demonstrate the validity of our pipeline.

Future work

Since the usefulness of filtering out a gene subset with the CGC dataset is demonstrated by our experiments, we think it worth exploring the importance of selecting representative gene subsets. More promising methods such as network propagation based on the STRING protein-protein interaction database could be used to improve our model further.[19] We also wish to explore incorporating attention mechanism-based models to improve performance.[24] Recent work using graph neural networks (GNN) shows the potential of GNNs in dealing with drug molecular data. A combination of VAE and GNN (VGAE)[28] could be adopted for this problem. VGAE would take advantage of VAE model to be generative and incorporate the GNN to process graph data efficiently. We believe a modified version of the originally proposed VGAE[28] is a promising way to predict drug response and generate new drugs. Finally, since our model performs well on drug response prediction with good potential for drug discovery, we would like to build a toolkit based our model.

Acknowledgement

The authors thank professor Manolis Kellis from the MIT CSAIL Lab for advice for this article.

References

(1) Yang, W.; Soares, J.; Greninger, P.; Edelman, E. J.; Lightfoot, H.; Forbes, S.; Bindal, N.; Beare, D.; Smith, J. A.; Thompson, I. R., et al. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res.* **2012**, *41*, D955–D961.

(2) Chiu, Y.-C.; Chen, H.-I. H.; Zhang, T.; Zhang, S.; Gorthi, A.; Wang, L.-J.; Huang, Y.; Chen, Y. Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC Med. Genomics* **2019**, *12*, 119.

(3) Geeleher, P.; Cox, N. J.; Huang, R. S. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome Biol.* **2014**, *15*, 1–12.

(4) Liu, Q.; Hu, Z.; Jiang, R.; Zhou, M. DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *bioRxiv* **2020**,

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

(6) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint* **2018**, arXiv:1802.04364.

(7) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar variational autoencoder. *arXiv preprint* **2017**, arXiv:1703.01925.

(8) Li, Y.; Vinyals, O.; Dyer, C.; Pascanu, R.; Battaglia, P. Learning deep generative models of graphs. *arXiv preprint* **2018**, arXiv:1803.03324.

(9) Simonovsky, M.; Komodakis, N. Graphvae: towards generation of small graphs using variational autoencoders. Artificial Neural Networks and Machine Learning – ICANN

2018. ICANN 2018. Lecture Notes in Computer Science, vol 11139. Cham, 2018; pp 412–422.

(10) Schmainda, K. M.; Prah, M.; Connelly, J.; Rand, S. D.; Hoffman, R. G.; Mueller, W.; Malkin, M. G. Dynamic-susceptibility contrast agent MRI measures of relative cerebral blood volume predict response to bevacizumab in recurrent high-grade glioma. *Neuro-Oncology* **2014**, *16*, 880–888.

(11) Yuasa, T.; Takahashi, S.; Hatake, K.; Yonese, J.; Fukui, I. Biomarkers to predict response to sunitinib therapy and prognosis in metastatic renal cell cancer. *Cancer Sci.* **2011**, *102*, 1949–1957.

(12) Imamura, T.; Kinugawa, K.; Minatsuki, S.; Muraoka, H.; Kato, N.; Inaba, T.; Maki, H.; Shiga, T.; Hatano, M.; Yao, A., et al. Urine osmolality estimated using urine urea nitrogen, sodium and creatinine can effectively predict response to tolvaptan in decompensated heart failure patients. *Circ. J.* **2013**, *77*, 1208–1213.

(13) Lachlan, J. M.; Hubbard, T. J. A Census of Human Cancer Genes. *Nat. Rev. Cancer* **2004**, *4*, 177–183.

(14) Barretina, J. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **2012**, *483*, 603–607.

(15) Wenric, S.; Shemirani, R. Using supervised learning methods for gene selection in RNA-Seq case-control studies. *Front. Genet.* **2018**, *9*, 297.

(16) Liu, S.; Xu, C.; Zhang, Y.; Liu, J.; Yu, B.; Liu, X.; Dehmer, M. Feature selection of gene expression data for Cancer classification using double RBF-kernels. *BMC Bioinf.* **2018**, *19*, 396.

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

(18) Chang, Y.; Park, H.; Yang, H.-J.; Lee, S.; Lee, K.-Y.; Kim, T. S.; Jung, J.; Shin, J.-M. Cancer drug response profile scan (CDRscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Sci. Rep.* **2018**, *8*, 8857.

(19) Oskooei, A.; Born, J.; Manica, M.; Subramanian, V.; Sáez-Rodríguez, J.; Martínez, M. R. PaccMann: prediction of anticancer compound sensitivity with multi-modal attention-based neural networks. *arXiv preprint* **2018**, arXiv:1811.06802.

(20) Grønbech, C. H.; Vording, M. F.; Timshel, P. N.; Sønderby, C. K.; Pers, T. H.; Winther, O. scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **2020**, *36*, 4415–4422.

(21) Rampasek, L.; Hidru, D.; Smirnov, P.; Haibe-Kains, B.; Goldenberg, A. Dr.VAE: drug response variational autoencoder. *arXiv preprint* **2017**, arXiv:1706.08203.

(22) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. Advances in Neural Information Processing Systems 28 (NIPS 2015). Montreal, Quebec, Canada, 2015; pp 2224–2232.

(23) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. *arXiv preprint* **2017**, arXiv:1704.01212.

(24) Manica, M.; Oskooei, A.; Born, J.; Subramanian, V.; Sáez-Rodríguez, J.; Rodríguez Martínez, M. Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Mol. Pharmaceutics* **2019**, *16*, 4797–4806.

(25) Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. Constrained graph variational autoencoders for molecule design. Advances in Neural Information Processing Systems 31 (NIPS 2018). Montreal, Canada, 2015; pp 7795–7804.

(26) Tsubaki, M.; Tomii, K.; Sese, J. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* **2019**, *35*, 309–318.

(27) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint* **2014**, arXiv:1412.3555.

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

Graphical TOC Entry

Some journals require a graphical entry for the Table of Contents. This should be laid out "print ready" so that the sizing of the text is correct.
Inside the `tocentry` environment, the font used is Helvetica 8 pt, as required by *Journal of the American Chemical Society*.
The surrounding frame is 9 cm by 3.5 cm, which is the maximum permitted for *Journal of the American Chemical Society* graphical table of content entries. The box will not resize if the content is too big: instead it will overflow the edge of the box.
This box and the associated title will always be printed on a separate page at the end of the document.