Modeling Post-treatment Gene Expression Change with a Deep Generative Model

Ladislav Rampasek[1,2,*], **Daniel Hidru**[1,2], **Peter Smirnov**[3], **Benjamin Haibe-Kains**[3,4], **and Anna Goldenberg**[1,2,*]

[1]University of Toronto, Department of Computer Science, Toronto, ON, Canada
[2]The Hospital for Sick Children, Toronto, postcode, ON, Canada
[3]Princess Margaret Cancer Centre, University Health Network, Toronto, ON Canada
[4]Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada

1 Introduction

Despite tremendous advances in the pharmaceutical industry, many patients worldwide do not respond to the first medication they are prescribed. Personalized medicine, an approach that uses patient's own genomic data, promises to tailor the treatment program to increase the probability of positive response. Most recent computational research in drug response prediction was motivated by the public release of high throughput drug screening in cell lines. The greatest advantage of cell lines is that it is relatively inexpensive to test them with thousands of drugs providing a rich basis for learning predictive models. This screening task was undertaken by several large consortia and pharmaceutical companies resulting in publicly available datasets of various sizes, most notably: NCI-60 dataset with 60,000 compounds tested on 60 cell lines[1], Genomics of Drug Sensitivity in Cancer (GDSC) with 138 drugs[2] across 700 cancer cell lines, and the Cancer Cell Line Encyclopedia (CCLE)[3] with 24 drugs tested on a panel of >1000 cell lines. Gene expression and drug response data for these cell lines are now publicly available.

The problem with drug sensitivity data is that it does not help to understand what happened to a cell line mechanistically (biologically) in response to a drug. To combat this problem, a database of perturbations was generated[4]. This database contains over 16,000 experiments showing how the expression of 1000 (landmark) genes changes in response to a drug (gene expression is recorded before and after drug application for many drugs). This information allows to assess the biological change in the cell line but does not directly translate into response/non-response. Combining response and perturbation data is expected to ultimately yield a better and more biologically relevant model of drug response, though likely more experiments will be needed, since all drugs are tested only on several different cell lines.

In this paper we present a new Perturbation Variational Autoencoder (PertVAE), that learns latent representation of the underlying gene states before and after a drug application. PertVAE is a deep generative model based on Variational Autoencoder (VAE)[5,6]. To fit generative and approximate inference distributions for our model, we use a combination of Stochastic Gradient Variational Bayes[5] and Inverse Autoregressive Flow[7]. We tested PertVAE on 19 drugs, predicting post treatment gene expression. The highest number of cell lines tested across the drugs was 56, which is a very small sample size for training complex models. Nevertheless, PertVAE can at least partially predict drug perturbations for 5 out of 8 drugs for which there is the most data available. Furthermore we found that the correlation of the reconstruction data is better when the size of the latent space is relatively small.

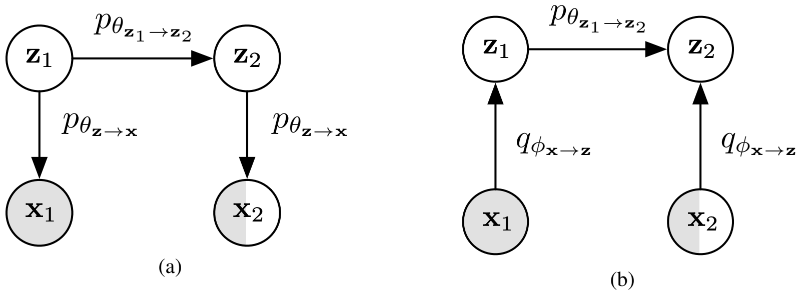
We believe that this is a promising result showing that even with a small sample size, deep models are able to learn some level of reconstruction of post-treatment gene expression. The next step would be to combine pre- and predicted post-treatment data together with drug sensitivity to predict drug response. While in this work we focused on analyzing reconstruction accuracy of post treatment data, our framework is easily extendible to that integration scenario.

2 Methods

We propose a Variational Autoencoder approach for modeling drug perturbation effects, i.e. given gene expression of a cell line before the drug is applied (pre-treatment gene expression), we are aiming to predict gene expression after the drug is applied (post-treatment state). To this end we propose a deep generative model, Perturbation VAE (PertVAE).

Perturbation Variational Autoencoder (PertVAE) is an unsupervised model for drug-induced gene expression perturbations, that embeds the data space (gene expression) in a lower dimensional latent space. In the latent space we model the drug-induced effect as a linear function, which is trained jointly with the encoder and decoder of the embedding.

We fit PertVAE on "perturbation pairs" $[\mathbf{x}_1, \mathbf{x}_2]$ of pre-treatment and post-treatment gene expression with shared stochastic embedding encoder $q_{\phi_{\mathbf{x}\to\mathbf{z}}}$ and decoder $p_{\theta_{\mathbf{z}\to\mathbf{x}}}$. The original dimension of each vector $\mathbf{x}$ is 903 genes. Additionally we use unpaired pre-treatment data (with no know post-treatment state) to improve learning of the latent representation. The graphical representation of PertVAE model is shown in Figure 1.


$p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}$
$\mathbf{z}_1$
$\mathbf{z}_2$
$p_{\theta_{\mathbf{z}\to\mathbf{x}}}$
$p_{\theta_{\mathbf{z}\to\mathbf{x}}}$
$\mathbf{x}_1$
$\mathbf{x}_2$
$p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}$
$\mathbf{z}_1$
$\mathbf{z}_2$
$q_{\phi_{\mathbf{x}\to\mathbf{z}}}$
$q_{\phi_{\mathbf{x}\to\mathbf{z}}}$
$\mathbf{x}_1$
$\mathbf{x}_2$

(a)

(b)

Figure 1. Perturbation VAE: (a) Factorization of the generative distribution p, (b) Factorization of the approximate posterior distribution q. Note, we use the generative $p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}$ in case $\mathbf{x}_2$ is not observed.

Joint distribution. Our Perturbation VAE models joint $p(\mathbf{x}_1,\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2)$, which we assume to factorize as:

$$p(\mathbf{x}_1,\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2) = p(\mathbf{x}_1|\mathbf{z}_1)\cdot p(\mathbf{x}_2|\mathbf{z}_2)\cdot p(\mathbf{z}_2|\mathbf{z}_1)\cdot p(\mathbf{z}_1) \tag{1}$$

Generative distributions p. Perturbation VAE's generative process factorization consists of the following distributions:

$$p(\mathbf{z}_1) = N(\mathbf{0},\mathbf{I}) \tag{2}$$

$$p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}(\mathbf{z}_2|\mathbf{z}_1) = N\left(\mathbf{z}_2|\mu_{\mathbf{z}_2} = f_\theta(\mathbf{z}_1), \sigma_{\mathbf{z}_2} = \exp^{f_\theta(\mathbf{z}_1)}\right) \tag{3}$$

$$k\in\{1,2\} : p_{\theta_{\mathbf{z}\to\mathbf{x}}}(\mathbf{x}_k|\mathbf{z}_k) = N\left(\mathbf{x}_k|\mu_{\mathbf{x}_k} = f_\theta(\mathbf{z}_k), \sigma_{\mathbf{x}_k} = \exp^{f_\theta(\mathbf{z}_k)}\right) \tag{4}$$

The parameters of these distributions are computed by functions f_θ, which are neural networks with a total set of parameters θ. For brevity we refer to these parameters as θ instead of more specific subsets $\theta_{\mathbf{z}\to\mathbf{x}}$ or $\theta_{\mathbf{z}_1\to\mathbf{z}_2}$ when such level of detail unnecessarily clutters the notation.

We constrain the mean function in $p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}$ to be a linear function $f_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}(\mathbf{z}_1)$ of the following form:

$$f_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}(\mathbf{z}_1) \equiv \mathbf{z}_1 + \mathbf{W}\mathbf{z}_1 + \mathbf{b} \tag{5}$$

with $\mathbf{W}$ and $\mathbf{b}$ initialized close to zero such that $f_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}(\mathbf{z}_1)$ starts as an identity function. We found that together with L2 penalization this formulation improves stability and generalization of the model.

Approximate posterior q. Depending on the type of the data, we assume the approximate posterior q with a set of parameters ϕ to factorize as:

$$\text{perturbation pairs:}\quad q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}_1,\mathbf{x}_2) = q_{\phi_{\mathbf{x}\to\mathbf{z}}}(\mathbf{z}_1|\mathbf{x}_1)\cdot q_{\phi_{\mathbf{x}\to\mathbf{z}}}(\mathbf{z}_2|\mathbf{x}_2) \tag{6}$$

$$\text{pre-treatment singleton:}\quad q_\phi(\mathbf{z}_1,\mathbf{z}_2,\mathbf{x}_2|\mathbf{x}_1) = q_{\phi_{\mathbf{x}\to\mathbf{z}}}(\mathbf{z}_1|\mathbf{x}_1)\cdot p_{\theta_{\mathbf{z}_1\to\mathbf{z}_2}}(\mathbf{z}_2|\mathbf{z}_1)\cdot p_{\theta_{\mathbf{z}\to\mathbf{x}}}(\mathbf{x}_2|\mathbf{z}_2) \tag{7}$$

Analogously to the shared generative $p_{\theta_{\mathbf{z}\to\mathbf{x}}}$ distribution, also $q_{\phi_{\mathbf{x}\to\mathbf{z}}}(\mathbf{z}_k|\mathbf{x}_k)$ is shared for both $k\in\{1,2\}$. Here, instead of directly using a standard diagonal Gaussian as the approximate posterior

$$k\in\{1,2\} : q_{\phi_{\mathbf{x}\to\mathbf{z}}}(\mathbf{z}_k|\mathbf{x}_k) = N\left(\mathbf{z}_k|\mu_{\mathbf{z}_k} = f_\phi(\mathbf{x}_k), \sigma_{\mathbf{z}_k} = \exp^{f_\phi(\mathbf{x}_k)}\right) \tag{8}$$

we apply two steps of "LSTM-type" Inverse Autoregressive Flow (IAF)[7] updates to facilitate a richer family of approximate distributions.

Fitting θ and ϕ parameters. We jointly optimize the generative model θ and variational ϕ parameters to maximize Evidence Lower Bound (ELBO) of the data:

$$\sum^{N_P}\log p(\mathbf{x}_1,\mathbf{x}_2) + \sum^{N_S}\log p(\mathbf{x}_1) \geq \text{ELBO}_{\text{PertVAE}} \tag{9}$$

$$\text{ELBO}_{\text{PertVAE}} = \sum^{N_P}\mathscr{L}_P(\mathbf{x}_1,\mathbf{x}_2;\theta,\phi) + \sum^{N_S}\mathscr{L}_S(\mathbf{x}_1;\theta,\phi) \tag{10}$$

which is a sum of the evidence lower bound of N_P perturbation pairs and the lower bound of N_S unpaired “singleton” examples that we leverage to train the latent space Variational Autoencoder as well. The individual per-example lower bounds $\mathscr{L}_P$ and $\mathscr{L}_S$ take the following form:

$$\mathscr{L}_P(\mathbf{x}_1,\mathbf{x}_2;\ \theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}_1,\mathbf{x}_2)}\left[\log p_\theta(\mathbf{x}_1,\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2) - \log q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}_1,\mathbf{x}_2)\right] \tag{11}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x}_1)}\left[\log p_\theta(\mathbf{x}_1|\mathbf{z}_1)\right] - D_{KL}\left[q_\phi(\mathbf{z}_1|\mathbf{x}_1)||p(\mathbf{z}_1)\right] + \tag{12}$$

$$+\mathbb{E}_{q_\phi(\mathbf{z}_2|\mathbf{x}_2)}\left[\log p_\theta(\mathbf{x}_2|\mathbf{z}_2)\right] - D_{KL}\left[q_\phi(\mathbf{z}_2|\mathbf{x}_2)||p_\theta(\mathbf{z}_2|\mathbf{z}_1)\right]$$

$$\mathscr{L}_S(\mathbf{x}_1;\ \theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x}_1)}\left[\log p_\theta(\mathbf{x}_1|\mathbf{z}_1)\right] - D_{KL}\left[q_\phi(\mathbf{z}_1|\mathbf{x}_1)||p(\mathbf{z}_1)\right]$$

Using Stochastic Gradient Variational Bayes (SGVB)[5,6] it is possible to backpropagate through $\text{ELBO}_{\text{PertVAE}}$ and we use Adam[8] to compute gradient updates for both θ and ϕ parameters. As we use IAF to model $q_\phi(\mathbf{z}_k|\mathbf{x}_k)$, the Kullback–Leibler divergence D_{KL} cannot be computed numerically and therefore we use a Monte Carlo estimate. Additionally we allow “free bits” in D_{KL} to mitigate the problem of overly strong prior causing the optimization to get stuck in bad local optima[7].

3 Datasets

We test our methods on a panel of 19 drugs for which there are perturbation experiments available. These 19 drugs were also used in recent AstraZeneca-Sanger DREAM Challenge and therefore we use it as a representative sample of anti-cancer drugs.

The Library of Network-Based Cellular Signatures (LINCS) consortium screened perturbation effects that drugs have on gene expression of L1000 landmark genes in cancer cell lines[4]. The L1000 perturbation dataset is relatively sparse, for the 19 drugs, only up to 56 different cell lines were screened, albeit at various concentrations and with many biological replicates (as high as 15). In our results we use measurements at the highest drug concentration and all the biological replicates of such experiments. In cross-validation of our models we use cell-line-wise splitting so that the biological replicates for a particular cell line are in the same data fold.

Additionally, we use the union of ~1300 cell lines in Genomics of Drug Sensitivity in Cancer (GDSC)[2] and the Cancer Cell Line Encyclopedia (CCLE)[3] for which there are no know post-treatment states in an unsupervised fashion, to improve training of our gene expression latent embedding. We obtained these datasets using PharmacoGx R package[9].

4 Results

Architecture. The encoder’s input size corresponds to the number of landmark genes (> 1000) on the input, and is followed by two hidden layers with 500 and 300 units, respectively. From the last hidden layer the parameters of initial Gaussian distribution $\mu_{\mathbf{z}_k}$ and $\sigma_{\mathbf{z}_k}$ are computed together with 200 hidden units on which the subsequent Inverse Autoregressive Flow is conditioned. We use 2 steps of IAF, each with one hidden layer of 300 units. Architecture of data decoder mirrors that of data encoder, but without IAF. We use ELU activation function[10] and Weight Normalization[11].

The presented experiments are evaluated in 10-times randomized 5-fold cross-validation and we report the average metric across these 50 data splits. The models were fitted independently for each of the 19 drugs, but with the same hyperparameters.

4.1 Modeling gene expression

Variational Autoencoder[5] is an expressive non-linear model, while PCA has the best reconstruction loss among linear models. To evaluate how well a VAE with our architecture can model gene expression, we fitted a VAE with various number of stochastic latent variables and compared its reconstruction to reconstructions by PCA with equivalent number of principal components. As the measure of reconstruction quality we used Spearman’s ρ between reconstruction mean and the observed gene expression. We plot the results in Figure 2. A Variational Autoencoder with the above described encoder/decoder architecture does better for small latent spaces (< 20) after which it seems to overfit compared to PCA.

We chose the encoder/decoder architecture and latent space of 100 stochastic units for our PertVAE. We expect that PertVAE then has enough expressive power and capacity to not just model gene expression, but also find such a latent space in which a drug perturbation effect can be modeled as a stochastic linear function.

4.2 Predicting drug-induced change

We trained a PertVAE for each drug independently with the goal of predicting drug perturbation effects. That is, we optimized the $\text{ELBO}_{\text{PertVAE}}$ and stopped training when perturbation prediction loss started to increase on the validation set.

To evaluate the prediction performance we computed Spearman’s correlation $\rho_{\text{pred,pert}}$ between the mean of predicted gene expression distribution $\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}_1)}[p_\theta(\mathbf{x}_2|\mathbf{z}_1)]$ and the true post-treatment gene expression in the test set. We compare this correlation to the correlation $\rho_{\text{rec,pert}}$ between the mean of pre-treatment reconstruction distribution $\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x}_1)}[p_\theta(\mathbf{x}_1|\mathbf{z}_1)]$ and

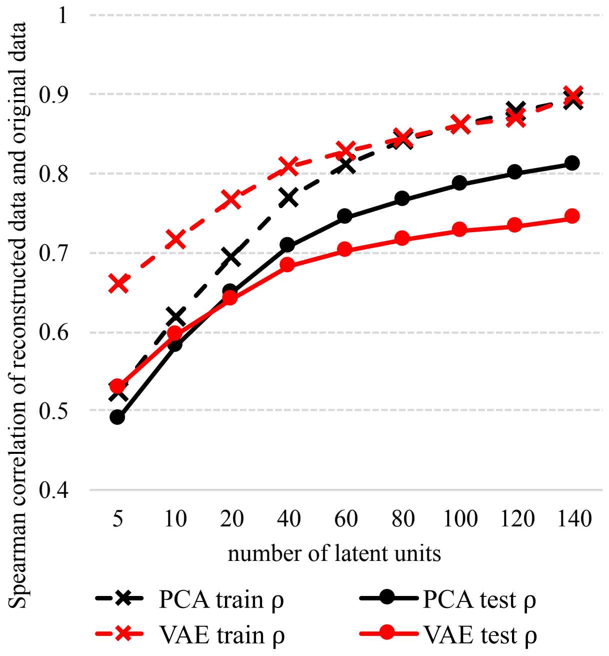


Figure 2. PCA and VAE reconstruction quality comparison for varying latent space size.

Table 1. Perturbation VAE prediction results with latent space size 100.

drug	# cell lines	ρ rec,pert	ρ pred,pert	p-value
olaparib	56	0.529	0.517	1
selumetinib	56	0.457	0.466	**0.004**
vorinostat	56	0.475	0.584	**7.9E-08**
bortezomib	51	0.444	0.508	**1.1E-11**
navitoclax	51	0.505	0.485	1
SN-38	51	0.433	0.509	**3.8E-14**
temsirolimus	51	0.488	0.504	**0.001**
tipifarnib	51	0.538	0.536	0.713
GDC-0941	19	0.488	0.494	0.361
gefitinib	19	0.545	0.541	0.795
NU-7441	19	0.502	0.502	0.548
saracatinib	19	0.517	0.514	0.682
vinorelbine	14	0.51	0.504	0.659
docetaxel	13	0.524	0.509	0.981
paclitaxel	13	0.465	0.443	1
afatinib	12	0.481	0.472	0.562
etoposide	12	0.49	0.481	0.745
doxorubicin	8	0.254	0.311	**0.016**
linsitinib	6	0.502	0.5	0.628

the true post-treatment gene expression. Note, that in training the "drug effect" mean function is initialized close to identity. If PertVAE would either underfit or overfit on the training set, we would expect $\rho_{\text{pred,pert}}$ to be no larger than $\rho_{\text{rec,pert}}$. Therefore we calculate Mann-Whitney single-sided test with the alternative hypothesis $\text{H}_1 = \rho_{\text{rec,pert}} < \rho_{\text{pred,pert}}$ on the results of our 10-times randomized 5-fold CV. The average correlation values and p-values of the statistical test are in Table 1, showing that PertVAE can at least partially predict drug perturbations for 5 out of 8 drugs (p-value ≤ 0.001) for which the data set consists of perturbation experiments in at least 50 unique cell lines.

References

1. Shankavaram, U. T. *et al.* Cellminer: a relational database and query tool for the nci-60 cancer cell lines. *BMC genomics* **10**, 1 (2009).
2. Yang, W. *et al.* Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research* **41**, D955–D961 (2013).
3. Barretina, J. *et al.* The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nat.* **483**, 603–607 (2012).
4. Duan, Q. *et al.* Lincs canvas browser: interactive web app to query, browse and interrogate lincs l1000 gene expression signatures. *Nucleic acids research* gku476 (2014).
5. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
6. Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).
7. Kingma, D. P., Salimans, T. & Welling, M. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934* (2016).
8. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
9. Smirnov, P. *et al.* Pharmacogx: an r package for analysis of large pharmacogenomic datasets. *Bioinforma.* btv723 (2015).
10. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015).
11. Salimans, T. & Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, 901–901 (2016).