Deep Generative Models for Cellular Representation Learning and Drug Sensitivity Prediction

by

Yifan Xue

Bachelor of Science, Peking University, 2014

Master of Science, Carnegie Mellon University, 2016

Submitted to the Graduate Faculty of

School of Medicine in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Yifan Xue

It was defended on

February 16, 2021

and approved by

Dr. Gregory Cooper, Professor, Biomedical Informatics

Dr. Xinghua Lu, Professor, Biomedical Informatics

Dr. Songjian Lu, Assistant Professor, Biomedical Informatics

Dr. LiRong Wang, Assistant Professor, Pharmacy

Dissertation Director: Dr. Xinghua Lu, Professor, Biomedical Informatics

Deep Generative Models for Cellular Representation Learning and Drug Sensitivity Prediction

Yifan Xue, PhD

University of Pittsburgh, 2021

The idea of precision oncology with drug sensitivity prediction was first introduced in the 1950s. With the emergence and promotion of in vitro cytotoxicity assays and high-through cell profiling techniques in the past three decades, precision oncology has advanced into an active research topic. The introduction of quantitative and computational methods has further boosted the growth of this area. Some impressive achievements have been made through the advancements, and yet we still have a long way to go towards the goal of precision oncology.

Most previous studies were focused on using traditional statistical models to quantize the correlations between a small set of genetic features and drug responses. The models were often limited to a specific cancer type, and could only predict responses for a small number of drugs. These limitations prevent deploying computational approaches into the standard clinical practice. To further promote precision oncology, we need to embrace the tremendous amount of genomic features and utilize the comprehensive information they provide about the state of cellular signaling systems to build versatile computational tools that can predict sensitivity for various cancer drugs.

In this dissertation project, we explore machine learning techniques, with a special focus on deep generative models for learning cellular state representations from omics data. We hypothesize that such representations can be used to replace traditional clinical and genetic features to significantly improve drug sensitivity prediction accuracy.

Learning latent representations from raw input features and tuning the representations for downstream tasks has been successful in a number of deep learning application areas, including computer vision and natural language processing. Such strategies used to be impractical in systems biology due to the limited amount of data. With large systematic perturbation datasets like TCGA, LINCS, and GDSC that are now available, there is an unprecedented opportunity for introducing representation learning into the study of drug sensitivity prediction. We believe that the integration of deep learning representations presented in this dissertation will help advance the practice of pre-clinical drug-response prediction and contribute to a new age of precision and personalized cancer therapy.

Table of Contents

List of Tables

List of Figures

Preface

We would like to thank Drs. Lujia Chen, Michael Ding, and Jonathan Young for their editorial assistance. We would also like to acknowledge the TCGA group, LINCS group, GDSC group, and CCLE group for making the data used in this dissertation publicly available.

1.0 Overall introduction

The idea of precision oncology with chemosensitivity prediction dates back to the 1950s, and yet little progress in that area was made until the 1990s when large-scale cytotoxicity assays for screening and evaluating bioactive compounds in vitro became available (Bellamy, 1992). The absence of these high-throughput and low-cost cellular profiling techniques limited the scope of early studies. Most of the work was focused on examining the relationships between genomic alterations and chemotherapy resistance through manipulating a single gene on animal models or cultured cell line samples, and then observing the chemosensitivity outcomes (Weller, 1998). The procedure was often time and labor-consuming, and not always effective.

With omic-scale profiling techniques, such as expression profiling and methylation profiling, becoming mature in the past three decades, it has made it possible to take a group of genes and/or gene regulatory elements into consideration and estimate their effects on drug sensitivity in batches (Dry et al., 2010; Shen et al., 2007; Szakács et al., 2004). Around the same time, computational approaches were introduced into the field to leverage the power of statistical modeling to handle an increasing number of predictive features (Frieboes et al., 2009). The study of drug sensitivity prediction was further revolutionized in recent years as large-scale datasets of systematic screens of cellular responses to various drugs, including the NCI-60 (Shoemaker, 2006), the Genomics of Drug Sensitivity in Cancer (GDSC) (Yang et al., 2012), and the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al., 2012), were made available to the research community. With these comprehensive datasets, the current trend is to develop integrative computational models that estimate the correlations between the whole genomic profiles and drug responses. These models vary from simple gene expression methods based on weighted voting classification

models (Staunton et al., 2001), to more complex machine learning-based models such as random forest, Support Vector Machine (SVM), and Elastic net (Dong et al., 2015; Jang, Neto, Guinney, Friend, & Margolin, 2014; Riddick et al., 2010). A review on cytotoxicity assays and the current trend in drug sensitivity prediction methods with data from NCI-60, CCLE, and GDSC is provided by (Cortés-Ciriano, H Mervin, & Bender, 2016).

One branch of machine learning methods, known as deep learning, has become increasingly popular in the academic field in recent years. These models use a layer-wise, directed, graph-based architecture, which are called neural networks. The intuition in designing such models is to mimic the human brain that functions through firing densely connected neurons. A deep learning model, specifically, is constructed by connecting multiple layers of artificial, abstract neurons that each carry out a simple computation. The fact that many layers are included in the neural network model is the sense in which the model is "deep." The deep architecture provides the model with exceptional learning potential, and like other traditional models, a deep learning model can be either supervised, semi-supervised, or unsupervised.

One special class of unsupervised deep learning models are the deep generative models (DGMs), which are autoencoder-based neural networks that are intended to simulate the data generation process given some statistical assumptions (e.g., conditional independence assumptions). These models are primarily used to learn reduced representations from high dimensional raw input data, which can be further tuned for subsequent tasks in a supervised manner. The-state-of-art models in a variety of computer application areas, such as Transformers in language processing (NLP) (Vaswani et al., 2017), are just variations of DGMs.

DGMs are of particular interest in the domain of systems biology. If a model learns to accurately regenerate biological data produced under different cellular perturbations (e.g., gene

expression or proteomics of cells under different stresses like drugs or diseases), the model plausibly includes some form of representation of the cellular signaling system responding to the perturbation. Such representations shed light on the mechanisms of how distinct perturbations impact different cellular processes. Despite their great success in computer vision and NLP, however, representation learning methods such as DGMs, have not been widely applied for systems biology tasks, including drug sensitivity prediction. This situation was due in part to the limited amount of large-scale genomic data, which makes training deep models impractical because they usually require large datasets to perform well. In addition, the lack of interpretability of conventional DGMs also presents an obstacle to their popularization. Deep models typically behave like a "black-box" in that it is difficult to explicitly understand how the neurons are associated with the biological entities in the cellular system. With the rapid growth of high-throughput sequencing data and drug response data, however, designing DGMs that are more adaptive to genomic data has become more feasible.

In the research reported here, we describe our development and examination of different DGMs for learning interpretable representations of cellular signaling systems, with the goal of improving drug sensitivity prediction practice and realizing the promise of precision oncology.

1.1 Hypothesis

In this dissertation, we hypothesize that latent representations learned by deep generative models from genomic data capture well the state of the cellular signaling system of a cancer sample. Such latent representations can be used as features to significantly enhance the capability of drug sensitivity prediction compared to using the original gene expression data as features. We further

hypothesize that a combination of gene expression data with genomic alteration data can learn more interpretable representations of cellular systems, which will further improve the drug sensitivity prediction performance.

2.0 Background

This section provides literature reviews from four aspects that lay the background for this dissertation project: a review of the recent progress in drug sensitivity prediction studies; an introduction of the datasets that are commonly used in computational genomics and drug sensitivity prediction; a review of the history of DGMs, where most representative models and their training algorithms are introduced; and an introduction of four categories of supervised learning algorithms that can be applied for drug sensitivity prediction.

2.1 Recent progress in drug sensitivity prediction

Precision oncology is defined as molecular profiling of tumors to identify prognostic biomarkers for predicting patients' responses to specific cancer therapies. The area is rapidly developing and has entered the mainstream of clinical practice in recent years (Fojo, 2016; Garraway, Verweij, & Ballman, 2013; Prasad, Fojo, & Brada, 2016). The biomarkers that are currently used for guiding drug selection, however, are only available for a small number of drugs and can only benefit a small proportion of patients (Prasad, 2016; Tannock & Hickman, 2016). For other nonspecific drugs, no biomarker has been found and yet a difference in their effects on different subgroups of patients is observed (Rubio-Perez et al., 2015). In order to stratify patients for more specific treatments, the current trend of precision oncology is to utilize computational tools to analyze molecular profiling data in batches and systematically identify biomarkers for predicting drug response.

The emergence of large-scale datasets of systematic screens of cellular responses to bioactive compounds like NCI-60, GDSC, and CCLE has further advanced the studies of computational drug sensitivity prediction. For example, (Staunton et al., 2001) developed a weighted voting classification algorithm for classifying chemosensitivity of a cell line based on gene expression data from the NCI-60 dataset. They modeled the drug response prediction as a binary classification problem, where a cell line with $\log_{10} GI50 \geq \text{mean} + 0.8\text{std}$ was labeled as resistant and $\log_{10} GI50 \leq mean - 0.8std$ was labeled as sensitive; cell lines with an in-between value were labeled as intermediate and were excluded from the training and testing datasets. For each compound of interest, marker genes that are eligible to vote for the sensitivity class were selected based on the expression diversity of the gene across all training cell lines. The marker genes were then weighted according to the correlation between their expression level and the sensitivity status of the training cell lines. The performance of this weighted voting classification algorithm was measured as the average accuracy of classifying sensitive cell lines vs. resistant cell lines. The classifiers were compared to random guess predictions based on a Kolmogorov-Smirnov test for distribution differences. For 232 drugs, 88 received a significant classifier, with a median accuracy of 75%, ranged from 64% to 92%.

Instead of selecting marker genes via a frequent-based or correlation-based approach, (Riddick et al., 2010) applied a random forest-based method on the NCI-60 dataset for identifying gene signatures for predicting cell line response to drugs. In this case, the sensitivity prediction task was modeled as a regression problem, where the value of IC50 was estimated directly. The method can be further decomposed into three steps. For a given compound, a random forest was first trained on all basal genes measured in untreated cells to select highly predictive genes. A second random forest was then trained using only these significant genes as features. Finally, a

third model was trained excluding the cell lines that were not associated with the giving compound determined from the second step. The authors evaluated their method on microarray data from 19 breast cancer cell lines and 7 glioma cell lines. For breast cancer, two drugs were tested, where the 19 cell lines were first divided into a sensitive group and a resistant group, and predicted IC50s were compared between the two groups with a t-test to see if the predictions are significantly different. For glioma, the proposed method was compared with a differentially expressed gene-based model by computing the coefficient of determination (R^2) between the predicted and observed IC50s. The highest average R^2 achieved across 37 drugs was 0.71.

The original CCLE project also preliminarily explored the power of machine learning techniques in revealing the correlation between genomic data and drug sensitivity (Barretina et al., 2012). In CCLE, the efficacy and potency of a drug (the drug response) were measured as the area over the dose-response curve, which is denoted as the activity area. Two types of approaches were tried for predicting drug response: a naïve Bayes classification for discrete sensitivity calls, and an Elastic net regression for continuous sensitivity measurements. Top predictive gene alterations for each drug were identified based on the coefficients assigned to the alterations by a trained model. Across their models, the origin of the cell line, referred to as the "lineage of the cell" line in the article, was recognized as a confounding factor. For certain cancer types, classifiers built using cell lines of the same cancer type usually outperformed classifiers built using all cell lines across all cancer types. Lineage also emerged as the predominant predictive feature for some compounds, where some lineages were more sensitive to certain compounds, which was also supported by clinical observations.

The larger size of cell line expression and drug response data also makes it possible to train deeper and integrative machine learning models for drug response prediction. For example,

Menden et al. implemented a single hidden layer perceptron model that combines both genomic features and chemical properties of drugs to predict the responses (logIC50) of cancer cell lines from the GDSC dataset (Menden et al., 2013b). The model achieved an average Pearson correlation (R_p), R^2, and root mean square error (RMSE) of 0.85, 0.72, and 0.83 across 111 drugs in the test dataset, respectively. Note that random forests could achieve comparable performances with R_p of 0.85, R^2 of 0.72, and RMSE of 0.84. As one of the earliest attempts in using neural network models for drug sensitivity prediction, this study set the baseline for future explorations of neural networks and deep learning techniques. Other works that also incorporated cheminformatics into network-based models to predict drug sensitivity include (Wei, Liu, Zheng, & Li, 2019; N. Zhang et al., 2015).

A systematic evaluation of 110,000 models on CCLE and GDSC data with different combinations of algorithms (seven algorithms tested), types of features, compounds being predicted, methods of summarizing compound sensitivity values, and types of target values (discretized or continuous response values) was carried out in 2014 and summarized in (Jang et al., 2014). In this evaluation, models predicting continuous drug responses were compared by R_p, and models predicting discrete response were compared by the area under the receiver operating characteristics curves (AUC). The contribution of each model factor (e.g., algorithm, types of input data) was measured by a multi-way ANOVA. For discrete models, AUC >70% was achieved for 22 of 24 compounds in CCLE, and 83 of 138 in GDSC. The evaluation results suggest that a model's performance is primarily explained by the type of input features being used and the choice of compounds being predicted, followed by the choice of algorithm. Gene expression data turned out to be the most informative data type, while a combination of multiple genomic data types usually only improved the model performance moderately. Predictions were more accurate for

pathway targeted compounds (e.g., MEK inhibitors). The results also suggest the use of Elastic net or ridge regression applied to continuous response can achieve a higher prediction accuracy while discretizing response measurements caused decreased model accuracies.

The NCI-DREAM drug sensitivity prediction challenge represents another community effort of systematic evaluation of different drug sensitivity prediction algorithms (Costello et al., 2014). The challenge provided DNA copy-number variation, transcript expression, mutations, DNA methylation, and protein abundance data for 35 breast cancer cell lines exposed to 28 therapeutic compounds. An independent test dataset was hidden from the participants, which was composed of the remaining 18 cell lines. 44 algorithms were submitted with their performances compared in the summarizing paper (Costello et al., 2014). The model performance was quantified as the weighted probabilistic concordance index (wpc-index). The algorithms that modeled nonlinear relationships between features and incorporated pathway information usually performed better. The best model was the Bayesian multitask multiple kernel learning (MKL) with a wpc-index of 0.583 and a balanced accuracy of 0.78. The second-best model leveraged the strength of random forests, with a wpc-index of 0.577. Consistent with the conclusions of (Jang et al., 2014) above, both the top 2 models are regression models rather than discrete response classification models. Gene expression data were also found to provide the best predictive power over other data types.

Efforts have also been put into developing more advanced machine learning tools for drug sensitivity prediction. For example, (Dong et al., 2015) built a predictor based on SVM and a recursive feature selection tool to predict drug sensitivity using data from CCLE. The drug sensitivity prediction task was treated as a classification problem, where the CCLE cell lines were divided into three groups, sensitive, resistant, and intermediate, according to their drug response

values (activity area). A SVM-REF (Support Vector Machine Recursive Feature Elimination) was trained for each drug for feature selection and drug response classification. The model achieved >= 80% accuracy for 10 drugs, and >=75% accuracy for 19 drugs. The highest accuracy was 91.75%, obtained for a target pathway compound, topoisomerase I inhibitor Irinotecan, followed by >85% for two MKE inhibitors, and 76%-87% for four EGFR inhibitors. The lowest accuracy was 69.35% for LBW242. The model was validated on an independent dataset from the CGP (Garnett et al., 2012), with a satisfactory performance achieved for three drugs, AZD6244, Erlotinib, and PD-0325901.

Instead of treating each gene as a feature, Ding et al (M. Q. Ding, Chen, Cooper, Young, & Lu, 2018) utilized deep AutoEncoder (AE) for learning latent representations from gene expression profiles of cell lines to predict drug sensitivity. The assumption is that the information extracted by deep learning models and preserved in the latent representations may reflect the activation state of cellular signaling pathways that are informative towards drug response outcomes. They trained a deep AE on gene expression data from the GDSC dataset and applied it on both the GDSC and CCLE datasets to get cell line latent representations; then they used the representations as input features to train Elastic net logistic regressions and SVMs for classifying drug response. For each drug, cell lines were divided into a sensitive group and a resistant group by applying the waterfall method. Their models achieved an average sensitivity of 0.82 and specificity of 0.82 per-drug basis, with exceptional performances obtained for 15 drugs with both sensitivity and specificity exceeding 0.98.

The use of deep neural network models for learning representations from high-dimensional data has become a popular trend for drug response prediction and drug repositioning in recent years. Chang et al. proposed a convolutional neural network (CNN) based model, called CDRscan,

for predicting cell line response to anticancer drugs (Chang et al., 2018). The CDRscan is an ensemble of five CNN-based models of slightly different architectures. Each model takes in the mutation profile of a cell line and the PaDEL fingerprint of a given drug, which is numeric representations of the drug's chemical structure (Yap, 2011), and predicts the IC50 of the drug. The mean IC50 across the five models is reported as the final prediction. All five models share a similar backbone architecture. Specifically, the mutation profile and the PaDEL fingerprint, are first processed independently by two CNNs for generating latent representations; the outputs of the two CNNs are then merged and fed into an additional CNN to yield an IC50 value. When training and testing CDRscan, each combination of a cell line and a drug was treated as an input instance. In total, 152,594 instances generated from mutation profiles from the CCLP database and drug response data from the GDSC database were used for training and validation, which contains 787 cell lines across 25 cancer types with IC50 measured for 244 drugs ($787 \times 244 > 152{,}594$ because not all drugs were tested for every cell line). For classification, a cell line was defined as sensitive against a drug if its $\ln(\mathrm{IC50}) < -2$. CDRscan achieved a R^2 of 0.843 and an AUC of 0.98 across all validation instances. Note that this overall AUC is affected by the composition of cell lines and drugs in the dataset, therefore it is not directly comparable with the drug level AUC reported in other studies. In addition, as also pointed out in the original article, the use of $\ln(\mathrm{IC50}) < -2$ as the sensitivity cutting threshold is more stringent than most other similar studies (Chang et al., 2018), which may also bias towards a higher instance AUC score for their model.

Instead of constructing a prediction model for every drug, Chiu et al. proposed DeepDR that learned latent representations from mutation and expression data with two neural network encoders and integrated them for simultaneously IC50 prediction for 256 drugs from GDSC (Chiu

et al., 2019). DeepDR achieved an overall prediction performance of Mean Squared Error (MSE) at 1.96 for log-scale IC50 and cell line centric R_p and Spearman ρ of 0.74-0.95 and 0.70-0.92.

Prior knowledge such as signaling pathway information has also been incorporated into models to boost model training and prediction. Wang et al. proposed a pathway-based prediction model that integrated gene expressions into pathway scores according to prior knowledge of signaling pathways and then used pathway scores to predict drug response (X. Wang, Sun, Zimmermann, Bugrim, & Kocher, 2019). Four scoring approaches for inferring pathway activity were tried, including two competitive scoring approaches, DiffRank (a new scoring approach proposed by Wang et al. in the same article), and GSVA, and two self-contained scoring approaches, PLAGE and Z-score. The pathway features were used to predict IC50 with an Elastic net regression. The model performance was measured as the MSE between the predicted IC50 and observed value. Overall, DiffRank produced more accurate predictions compared to other scoring approaches (best model for nine drugs and second-best for eight drugs).

Li et al. implemented a high-dimensional mixed linear regression model and applied to the CCLE dataset (Q. Li, Shi, & Liang, 2019). A mixture model was selected in order to address the population heterogeneity issue among the samples, which had been mostly overlooked by previous studies. Specifically, when building the model, samples were clustered into different groups, and different sets of drug sensitivity features were selected for each subpopulation. The model predicted the continuous activity area for each cell line given a drug, and the R_p and RMSE were computed for performance comparison. The highest correlation was 0.925, obtained for Nutlin-3, followed by 0.924 for L-685458.

Even though various machine learning techniques have been attempted for drug sensitivity prediction, most of the techniques are “shallow” in the sense that they only estimate the direct

correlations between genes and drug responses. The actual causal relationships, on the other hand, are often multi-hierarchical and buried in the intricate cellular signaling network. For previous works that utilized deep learning models like (M. Q. Ding et al., 2018), the focus was more on learning latent representations as a new set of features for improving drug sensitivity prediction, while the interpretability of the model architecture in respect of cellular signaling was less examined. The hierarchical structure of signaling network can be naturally mapped to the architecture of a deep learning model, which suggests deep learning models as a promising choice for revealing gene interactions. With the availability of large genomic and drug response datasets, to gain a deeper understanding of cell-drug interactions with deeper models is no longer impractical. Therefore, examining the utility of DGMs for learning latent representations and studying how they capture cellular signaling information is the focus of this project. We are expecting that the latent representations are more informative towards drug response compared to raw gene features, which can be used to significantly enhance the capability of drug sensitivity prediction and in turn, promote precision oncology.

2.2 Datasets for computational genomics and drug sensitivity prediction

In this section, we introduce the datasets we used in this thesis project and also briefly describe other datasets that are commonly used for computational genomics and drug sensitivity prediction studies. These include the L1000 dataset from the LINCS project that provides expression data for perturbed cell lines, the Cancer Genome Atlas (TCGA) dataset that provides omics data of real tumor samples, and the NCI-60, CCLE, and GDSC datasets that provide cell line drug response screen data.

2.2.1 The LINCS project and L1000 data

The LINCS project is an NIH Common Fund program that aims to create a network-based understanding of human biology by cataloging how human cells globally respond to chemical, genetic, and disease perturbations (Keenan et al., 2018). The goal is achieved by measuring cells' responses to various genetic and environmental stressors from different aspects of cellular phenotypes. The systematic experiments produced a dozen of types of data, including kinase-small molecule binding assay data, fluorescence images, cell growth assay data, protein secretion profiling data, mRNA profiling data, etc. Most of the data were made publicly available for research use.

The mRNA profiling data in LINCS were generated through a new gene-expression profiling method developed in the LINCS project, known as the L1000 assay (Subramanian et al., 2017). The L1000 assay, where the "L1000" indicates the 978 (~1000) landmark genes that are used to infer the entire transcriptome (>10,000 genes), was developed to obtain high-throughput expression profiles in a faster and less costly way. The landmark genes were selected following a data-driven procedure. First, a large, diverse dataset of 12,063 gene expression samples profiled on Affymetrix microarrays from the Gene Expression Omnibus (GEO) was assembled. Principal Component Analysis (PCA) was then applied to this dataset to reduce the dimension of samples and minimize batch effects that emerged through the mixture of samples from different tissue types with different physiologic conditions. In the resulted eigenspace of 386 components (90% variance was retained), consensus k-means clustering was performed in an iterative peel-off way to identify stable clusters of co-regulated genes. The centroid of the cluster obtained in each iteration was selected as a landmark gene. In total, 978 clusters were identified, and hence 978 landmark genes. These landmark genes were not found to be significantly enriched in any Gene Ontology (GO)

class except some generic categories like enzyme binding, protein kinase binding, catalytic activity, and ATP binding.

Meanwhile, 80 control genes, as empirically their expressions are invariant across samples, were also selected from this GEO dataset. These genes were used to normalized the expression level of the landmark genes, in order to reduce artifact produced during the assay. In concrete, the 80 genes were divided into 10 invariant levels, and the median expression of the 8 genes in each level form a calibration curve for each sample, which was then used as a reference to rescale the entire transcriptome using a power-law function.

The expressions of the 978 landmark genes were measured using a technique involving ligation-mediated amplification (LMA) followed by capturing the amplification products on fluorescently addressed microspheres. Each microsphere or bead was analyzed both for its color (denoting the landmark gene identity) and fluorescence intensity (denoting the gene expression abundance). Since only 500 fluorescent colors are commercially available, each color is used to measure two transcripts to avoid potential batch effects resulted from measuring all the 978 landmark genes in two runs. This was achieved by first coupling two genes to two beads of the same color, then mixing the two beads in a ratio of 2:1. This new bead was then hybridized with the sample templates and analyzed by the Luminex scanner, where two values were returned, one indicating the color and one indicating the intensity. The expression levels of the two genes were de-convolved from this intensity signal by plotting out the histogram of the intensity values where two peaks can be observed, one for each bead. The k-means clustering algorithm was used to identify the two clusters from the distribution, and the median expression value of each cluster is assigned as the expression level of the corresponding gene. Which pair of genes should be coupled

with the same color was optimized in advance to maximize the difference in their average expression level across the GEO dataset.

The expression levels of the 978 landmark genes were used to infer the expression levels of all the remaining, unmeasured genes. This was achieved by assuming that the expression level of an unmeasured gene *x* can be computed from the measured landmark genes *l* via a linear regression function:

$$x = w_0 + \sum_{i-1}^{978} w_i l_i \quad (2.1)$$

The weights w_s were estimated from the GEO dataset.

Except for the absolute expression level, a relative expression, in the form of a modified z-score, was also computed for each gene to reflect its differential expression level compared to unaffected genes under a specific stress condition (e.g. perturbagen). The L1000 assay was carried out on plates with 284 wells, where each well can incubate a sample with its perturbagen. Let *X* denote the vector of expression of gene x across all wells on the plate and MAD denote the median absolute deviations of *X*, then the differential expression of *x* in the *i*th sample can be computed as

$$z_i = \frac{x_i - median(X)}{1.4826 \times MAD(X)} \quad (2.2)$$

where 1.4826 is to make the denominator a consistent estimator of scale for normally distributed data.

In order to obtain robust expression data, each L1000 experiment (each combination of a cell line, perturbagen, dosage, exposure time, etc.) was typically repeated 3 times. A moderated z-score (MODZ) was computed for each experiment condition as a consensus signature. This was done by first computing a pairwise Spearman correlation matrix between the replicates in the space of landmark genes. The weights of each replicate were then computed as the sum of its correlations

to the other replicates. The weights were further normalized to have a sum of 1. The consensus signature is obtained as the weighted sum of replicate signatures.

The LINCS project currently provides five levels of L1000 gene expression data, inspired by the data format protocol of the TCGA database. These data levels reflect the different steps in the L1000 assay procedure (Figure 2.1). The five levels include:

1. Raw bead count and fluorescence intensity.
2. Deconvolved expression levels of 978 landmark genes.
3.
    a. Normalized expression levels of 978 landmark genes. Expression levels were normalized according to the 80 control genes.
    b. Expression levels of the entire transcriptome (12,328 genes) inferred from the landmark genes.
4. Differential expression levels of all genes.
5. Consensus signatures of the collapse of replicate experiments.

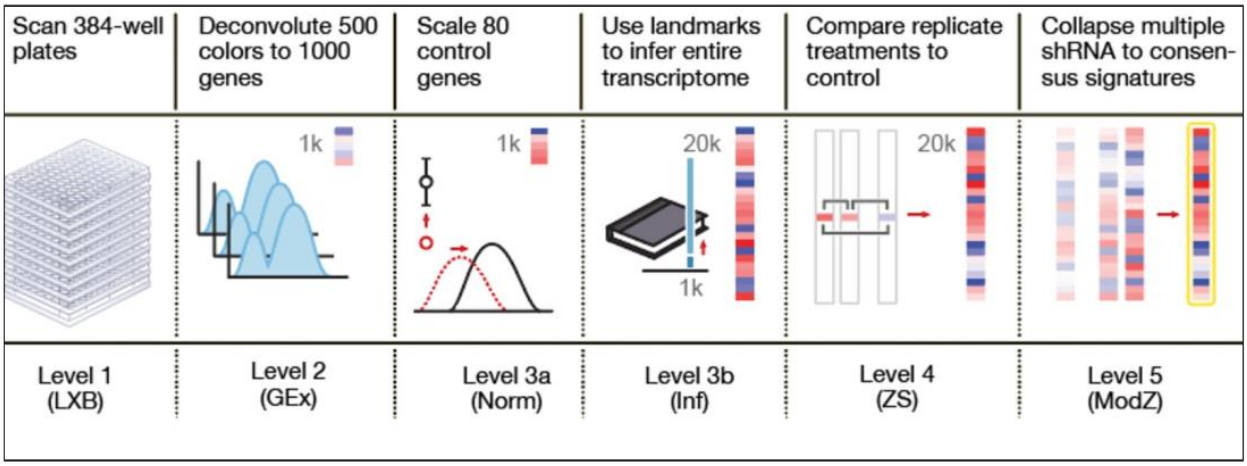


Figure 2.1. LINCS L1000 five data levels.

Figure 2c from (Subramanian et al., 2017).

The use of the L1000 assay dramatically increased the amount of gene expression data, which makes it possible to construct a 1000-fold larger connectivity map (CMap) than the previous one built with microarray data (Lamb et al., 2006; Subramanian et al., 2017). Such a CMap connects genes, drugs, and disease states through common gene-expression signatures, which can be used to discover mechanisms-of-action of drugs, reveal functional connections between diseases, genetic variants, and drugs, and propose new therapeutic targets. By 2017, phase II of LINCS CMap project has completed, and the L1000 datasets are publicly available from GEO. The datasets now contain over a million expression profiles with more than 10,000 different perturbagens.

Even though the datasets are still quite new, some attempts have been made to improve and extend the L1000 dataset and the CMap. For example, Duan et al. proposed using the characteristic direction method instead of the MODZ to compute L1000 signatures (Duan et al., 2016). The characteristic direction method was first proposed by Clark et al. as a geometric, multivariate approach to identify differentially expressed genes (DEGs) (Clark et al., 2014). The basic idea is that a multi-dimensional space is constructed for normal samples (or samples from the same class) with each dimension represent the expression of a gene. When a new sample comes, it is projected to the same gene space, and a hyperplane separating this sample and all the other samples is obtained, usually via Linear Discriminant Analysis (LDA). The direction of the normal vector of this hyperplane is just the characteristic direction of the new sample. The DEGs of this sample are determined by quantizing the contribution of the expression of each gene to the characteristic direction. This method was found to be more sensitive for identifying DEGs than univariate methods like fold change and Welch's t-test. Duan et al. showed that the characteristic

direction method significantly improved the signal-to-noise ratio compared with the MODS method. They developed a web-based search engine called L1000CDS^2, which can be used to predict drug targets by computing the cosine similarity between small-molecule signatures and single-gene perturbation signatures (Duan et al., 2016).

L1000 data and CMap have also been used in other applications, especially in guiding drug development. For example, Siavelis et al. used L1000 data and CMap as a drug repurposing tool to identify potential drugs for Alzheimer's disease (Siavelis, Bourdakou, Athanasiadis, Spyrou, & Nikita, 2015). Wang et al. proposed a machine learning classifier to predict adverse drug reactions (ADRs) by combining chemical structures and L1000 gene expression features (Z. Wang, Clark, & Ma'ayan, 2016). Iwata et al. used the L1000 data and CMap to elucidate the mode-of-action of bioactive compounds in a cell-specific way and predict therapeutic indications for 461 diseases (Iwata, Sawada, Iwata, Kotera, & Yamanishi, 2017). The approach they used consists of three steps: (1) identification of active pathways of a sample perturbed by a compound via enrichment analysis, (2) prediction of potential target proteins by comparing the expression signatures with those of other known perturbagens (disease or drug), and (3) prediction of new therapeutic indications by establishing connections between the compound in question and other diseases or drugs.

Even though some efforts have been made, the L1000 datasets are far from being fully explored. Particularly, less work has been done in examining the internal correlations among L1000 data or modeling the causal relationships between perturbagens and the resulted expression profiles. We aim to examine these aspects in this project.

2.2.2 TCGA

TCGA is a project supervised by the National Cancer Institute (NCI)'s Center for Cancer Genomics (CCG) and the National Human Genome Research Institute (NHGRI), which aims at molecularly characterizing primary tumors and matched normal samples to deepen our understanding of the genetic basis of cancer (program). The project began in 2006 with an initial focus on glioblastoma multiforme, lung cancer, and ovarian cancer. Phase II began in 2009 and expanded the genomic screen to 33 cancer types. By now, the TCGA project has generated genomic, epigenomic, transcriptomic, and proteomic data for over 20,000 samples (> 11,000 tumor samples), and has become one of the most popular data resources for the cancer research community.

The concluding project of the TCGA program, the Pan-Cancer Atlas (PANCAN) project, utilized the complete TCGA dataset to carry out comprehensive cross-cancer analyses. 27 papers have been published from this project, which further help gain insights into three themes of cancer, including the cell-of-origin patterns, oncogenic processes, and signaling pathways. In this thesis project, the PANCAN gene expression data were used to pre-train deep generative models to capture the general characteristics of real tumor samples.

2.2.3 NCI-60

The NCI-60 human tumor cell lines screen, starting from the late 1980s, represents the first effort in performing a systematic screen of drug response across cancer cell lines (Shoemaker, 2006). In NCI-60, 60 cancer cell lines, representing leukemia, melanoma, lung, colon, brain, ovary, breast prostate, and kidney cancers, were measured by NCI for their sensitivities towards

thousands of drugs. The similarities between response profiles of drugs across cell lines, quantified as the Pearson correlation coefficient, are often aligned with the similarities in the drug mechanism-of-actions (MOAs). The MOA of a new test compound can therefore be inferred by measuring its response profile and comparing it to known drugs. This is known as the COMPARE analysis.

2.2.4 GDSC

The GDSC project is an expansion of the Cancer Genome Project (CGP), sponsored by the Wellcome Trust Sanger Institute in The United Kingdom, which aims to improve cancer diagnosis, treatment, and prevention through molecularly characterizing cancer cell lines (Yang et al., 2012). The GDSC dataset provides comprehensive genomic profiling data for over 1,000 cell lines, with drug response data available for 453 compounds across most cell lines; these numbers are still fast-growing. A multivariate analysis of variance (MANOVA) was used in the original project to reveal correlations between drug sensitivity and genomic alterations in cancer. In our project, the GDSC data were used for training drug sensitivity prediction models.

2.2.5 CCLE

The CCLE dataset is another comprehensive open-access dataset of gene expression, genotype, and drug sensitivity data for human cancer cell lines. The data were generated through a collaboration between the Broad Institute, the Novartis Institutes for Biomedical Research, and the Genomics Institute of the Novartis Research Foundation. By 2019, the CCLE project has generated sequencing data for 1,457 cell lines with 24 anticancer drugs profiled across nearly 479

cell lines (Barretina et al., 2012). This dataset was primarily used in our project for testing drug sensitivity prediction models.

2.3 Deep generative models

Statistical models can be generally divided into two major types, generative models (GMs) and discriminative models (DMs). The two model types differ in the objective distributions they learn from the input data. Given an observable variable X and a target variable Y, a GM is to learn the joint distribution $P(X,Y)$ under some assumptions about the prior distribution $P(Y)$ and the generative relationships between X and Y, while a DM is to learn the conditional distribution $P(Y|X=x)$ or the posterior distribution. Analogously, a classifier based on a GM is referred to as a generative classifier, and a classifier based on a discriminative model is referred to as a discriminative classifier. Example generative classifiers include naïve Bayes classifier and linear discriminant analysis; example discriminative classifiers include logistic regression and SVM (SVM is sometimes considered as a classifier based on no specific model).

The GMs can be further divided into two groups based on the nature of the input data, the continuous GMs and the discrete GMs. In this project, we implemented five GMs, including AutoEncoder (AE), Variational AutoEncoder (VAE), and Vector-Quantized Variational AutoEncoder (VQ-VAE) for dealing with continuous input data, and Restricted Boltzmann Machine (RBM) and Deep Belief Network (DBN) for dealing with discrete input data. All these models are unsupervised GMs that learn the input data distribution $P(\vec{X})$ by reconstructing the data. The following sections give brief introductions of these models as well as other classic GMs.

2.3.1 AutoEncoder

AE is the foundation of most encoder-decoder-based GMs (Hinton & Salakhutdinov, 2006). An AE is a statistical model in the form of an artificial neural network (ANN), which is often used as a dimension reduction tool to learn reduced representations from the input data. In an ordinary ANN, the input layer corresponds to the observable variable X, and the output layer corresponds to the target variable Y. An ANN is typically trained using the gradient descent approach, where the derivatives of the classification or regression error between the output variable and the targets are backpropagated through the network to update the parameters iteratively. In order to learn a reduced representation of the input data X in an unsupervised manner, an AE uses X itself as the target and computes a reconstruction error for backpropagation. If the training converges, the resulted model learns to reconstruct the distribution of the input data accurately, and the hidden layer(s) are used to generate reduced representations of the input data.

An AE with a single hidden layer, as shown in Figure 2.2, can be further decomposed into two components: the input and hidden layers form the encoder of the AE, and the hidden and the output layer form the decoder of the AE. Traditionally, the weight matrix of the decoder is set as the transpose of the weight matrix of the encoder during training, known as the “tied-weights”. This constraint has been released in recent implementations, where the encoder and decoder have independent weights that are tuned separately.

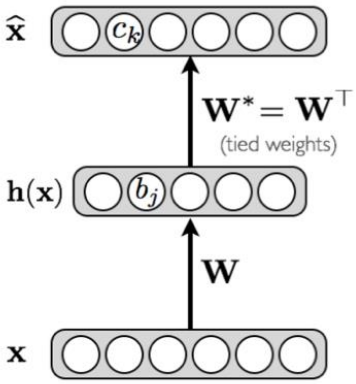


Figure 2.2. An auto-encoder with a single hidden layer.

An ANN with multiple hidden layers is called a deep neural network (DNN). Analogously, the deep version of an AE is known as the deep auto-encoder (DAE). Training a DNN is more challenging, as it is harder to initialize the parameters of a model in a way that will neither lead to a vanishing gradient nor converging to a poor local minimum. The first efficient algorithm for training a DAE was proposed by Hinton in 2006 (Hinton & Salakhutdinov, 2006). They solved the problem of training multiple layers by first modeling each pair of layers as an RBM and pre-training the model in a layer-wise manner; the RBMs are then unrolled to form the entire DAE (Figure 2.3) and a final fine-tune step is added to adjust all weights simultaneously.

A DAE can be interpreted in a hierarchical manner, where each hidden layer learns features of a different scale. For example, when using images as input data, the different layers are often found to represent basic shapes of different levels of complexity that compose an image. Typically, the hidden/higher the layer, the more elementary the shapes the layer represents (e.g., from parts of an object to straight lines of different orientations). Consequently, training a DAE will not only give us a reduced representation of the input data but also de-convolve the intractable data distribution into multiple sets of more interpretable hidden features.

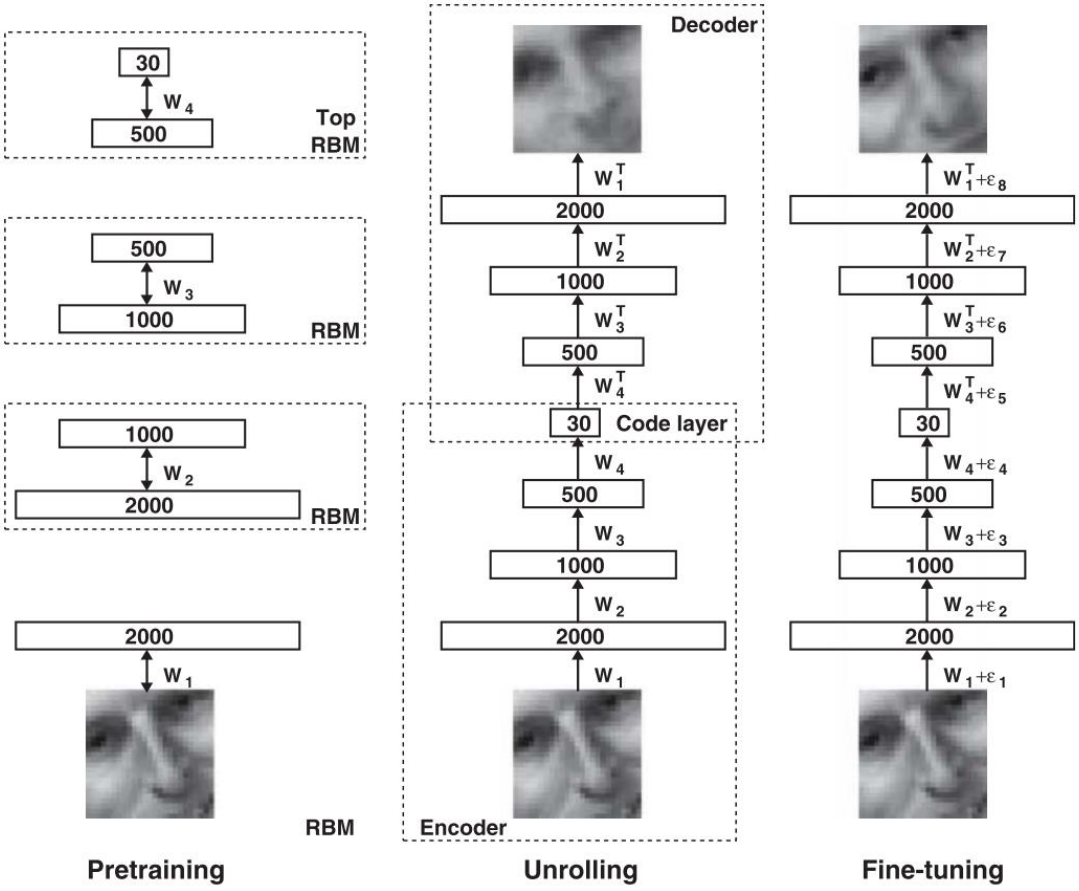


Figure 2.3. Three steps to train a DAE.

Figure 1 from (Hinton & Salakhutdinov, 2006).

The capability of AE in learning informative latent features can be further utilized by applying AE/DAE as a pre-training step for training supervised ANN/DNN models. Specifically, an AE/DAE is first trained with the input data X in an unsupervised manner for learning intrinsic characteristics from the data. Then a classification or regression layer, depending on the task, is added to the trained encoder of AE/DAE; this newly constructed model is fine-tuned using the labels of data Y. Including such a pre-training step often improves the performance of the model and makes the model more interpretable. This largely resolves the issue of traditional ANN/DNN

models behaving as black-boxes, which can achieve excellent task-oriented performance but with little interpretability.

2.3.2 VAE

The classic AE is very effective for dimension reduction and learning internal structures from the observed data. One shortcoming of AE is that, after training, the posterior distribution of the hidden variables is often intractable. This posterior distribution is to be used as the prior distribution to sample the hidden variables for generating new data, and its intractability makes AE less successful as a "generative" model. One way to resolve this problem is to estimate the posterior distribution of the latent variables with some empirical approaches like the kernel density estimation method. The problem remains, however, when we train two AEs independently with two subsets of data from the same distribution. In such cases, the resulted two models may have distinct distributions of hidden variables, even though they are expected to represent the same global distribution.

In 2014, Kingma and Welling exploited the idea of variational inference (also known as the variational Bayes approach) to approximate the posterior distribution of the latent variables and proposed VAE (Kingma & Welling, 2014) (a very similar idea was proposed in the same year by (Rezende, Mohamed, & Wierstra, 2014)). In VAE, the posterior distribution of the latent variable Z, $Q(Z|X)$, is approximated with a variational distribution $P(Z)$, which is typically restricted to a tractable family of distribution, like the Gaussian distribution. The Kullback-Leibler (KL) distance between $Q(Z|X)$ and $P(Z)$ is computed as a loss term to be minimized, which forces the posterior distribution to get close to the variational distribution through training. In the old expectation maximization (EM) training algorithm for variational inference, the KL distance is

reduced iteratively using the mean-field approach. The requirement of an analytical solution of expectation w.r.t the approximate posterior makes this approach intractable in general cases (Kingma & Welling, 2014). Stochastic gradient descent is another alternative, but it is not directly applicable to VAE as the sampling step of the latent variable is not derivable. This was resolved by Kingma and Welling by setting P as a normal distribution with diagonal covariance. This allows for the Gaussian re-parameterization trick to be used, where instead of sampling Z direction from P with $Z \sim N(\mu, \sigma^2)$ (μ and σ are estimated from the observed data), Z is sampled with $Z = \mu + \sigma\epsilon, \epsilon \sim N(0, 1)$. The re-parameterization trick gives a simple differentiable unbiased estimator of the latent variables, with which the stochastic gradient descent method can be applied. This whole training algorithm for VAE is called the stochastic gradient variational Bayesian approach (Kingma & Welling, 2014). Since the posterior distribution of the latent variables is pre-defined in VAE, generating new data becomes straightforward: sample a latent vector from $P(Z)$, and pass it through the decoder of VAE to obtain a new data instance.

In practice, the major difference between implementing an AE and a VAE is at the target function for training the model. In AE, the target function is just the reconstruction error between the output (the decoding of the encoded data $d\big(z_e(x)\big)$) and the input data (Equation 2.3). MSE is often used as the reconstruction error for continuous input data. In VAE, the KL distance between $Q(Z|X)$ and $P(Z)$ is added to the target function for updating the posterior distribution. (Equation 2.4).

$$L = l_r(x, d\big(z_e(x)\big)) \tag{2.3}$$

$$L = \ l_r(x, d(z_{e(x)})) + KL(q(z|x)||p(z)) \tag{2.4}$$

2.3.3 AAE

The adversarial autoencoder (AAE) is another AE extension that performs variational inference to approximate the posterior distribution of the latent variables (Makhzani, Shlens, Jaitly, Goodfellow, & Frey, 2015). The difference between VAE and AAE is that while VAE minimizes the KL distance between the posterior distribution of the latent variables $Q(Z|X)$ and the target variational distribution $P(Z)$ (Equation 2.4), AAE incorporates a discriminative network DN to compute an adversarial loss l_{ad} for distinguishing between $Q(Z|X)$ and $P(Z)$, inspired by the idea of the generative adversarial network (GAN) model (Goodfellow et al., 2014) (Equation 2.5, Figure 2.4 left).

$$L = l_r(x, d(z_e(x))) + l_{ad}(DN(q(z|x)), DN(p(z))) \quad (2.5)$$

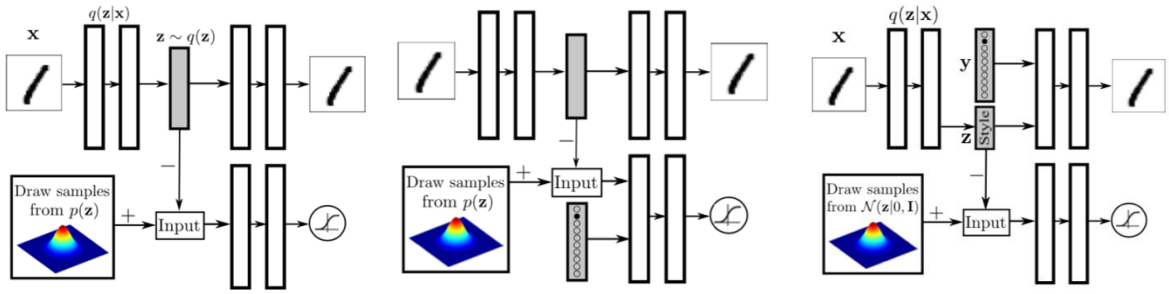


Figure 2.4. AAE architectures.

Figures 1, 3 and 6 from (Makhzani et al., 2015). The left panel shows the standard AAE. The middle panel shows how supervised information is incorporated into AAE by adding the one-hot representation of the data label to the input of the discriminative network. The right panel shows another way of incorporating the supervised information by adding the label to the decoder of the model.

AAE can be extended to semi-supervised and supervised learning by incorporating a one-hot representation of the data label to the input of the discriminative network to associate the label

of the data with a mode of the prior distribution (Figure 2.4 middle), or by concatenating the data label to the input of the decoder while using the discriminative network to learn the variance of data across all data classes. (Figure 2.4 right) (Makhzani et al., 2015).

2.3.4 VQ-VAE

In VAE and AAE, a continuous representation is learned for each input data instance. However, for many modalities in nature, for example, the words in a sentence, and the limited number of aberrant signaling pathways in a cancer sample, a discrete representation is usually more rational. To develop a model for learning discrete representations, one needs to address two problems: first, find a method to discretize the latent variable space; and second, find a method to sample a latent vector from the discretized latent variable space. The second problem was solved first, by representing the discrete latent variable distribution as an autoregressive distribution, and sampling the value of a latent variable depending on the other latent variables via ancestral sampling (Aäron Van Den Oord et al., 2016; Aaron Van Den Oord & Vinyals, 2017). For the first problem, Oord et al., inspired by the vector quantization (VQ) technique, proposed the VQ-VAE model in 2017, in which the latent variables are quantized by mapping to an embedding space composed of a limited number of codes (Aaron Van Den Oord & Vinyals, 2017). Specifically, during the forward computation, the input data is first passed through an ordinary encoder network to get its latent representation vector (or matrix if the latent representation has multiple channels). Each latent vector is then replaced by its nearest neighbor code in the embedding space and passed through the decoder to reconstruct the input data (Figure 2.5). In this way, each input instance is transformed into a (set of) discrete embedding code(s) rather than a continuous latent vector as in VAE. In summary, VQ-VAE differs from VAE in two ways: the encoder network outputs discrete,

rather than continuous representations; and the posterior distribution of the latent variables is learned rather than static.

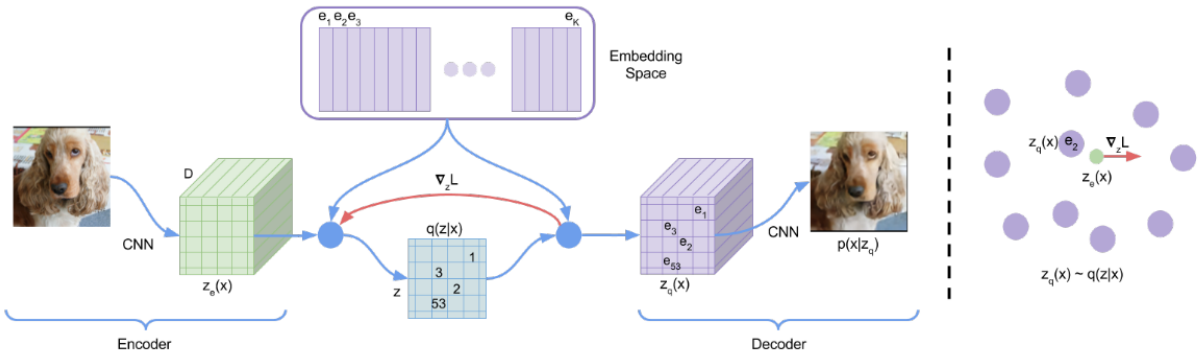


Figure 2.5. The VQ-VAE model.

Figure 1 from (Aaron Van Den Oord & Vinyals, 2017). The input image is first passed through the encoder $\boldsymbol{z_e(x)}$ to get its encoding matrix. The encoding matrix is then mapped to the embedding space by looking up the nearest neighbor code for each row vector. The nearest neighbor codes are used as a surrogate of the encoding matrix to be sent to the decoder $\boldsymbol{z_q(x)}$ to generate the reconstructed image. Both the encoder and decoder are ordinary CNNs.

To train a VQ-VAE, two additional terms are added to the objective function of AE to update the encoder, decoder, and the embedding space simultaneously. The objective function of VQ-VAE is

$$L = l_r(x, d(e_k)) + \|sg[z_e(x)] - e_k\|_2^2 + \beta\|z_e(x) - sg[e_k]\|_2^2 \tag{2.6}$$

$$k = argmin_j \left\|z_e(x) - e_j\right\| \tag{2.7}$$

Here sg is the stopgradient operator, which is defined as identity at the forward computation time and has zero partial derivatives when backpropagate. This is to constrain its operand to be a non-updated constant. The first term of the objective function is the reconstruction error as the models above, which helps the model learn the distribution of the input data. The

second term, called the VQ objective uses the L2 error to move the nearest code(s) e_k towards the latent representation $z_e(x)$. The third term, called the commitment loss, is to constrain the volume of the embedding space, making sure that the latent representation commits to a (set of) embedding code(s) rather than growing arbitrarily. The VQ objective and the commitment loss together update $z_e(x)$ and e_k in a bidirectional manner through training.

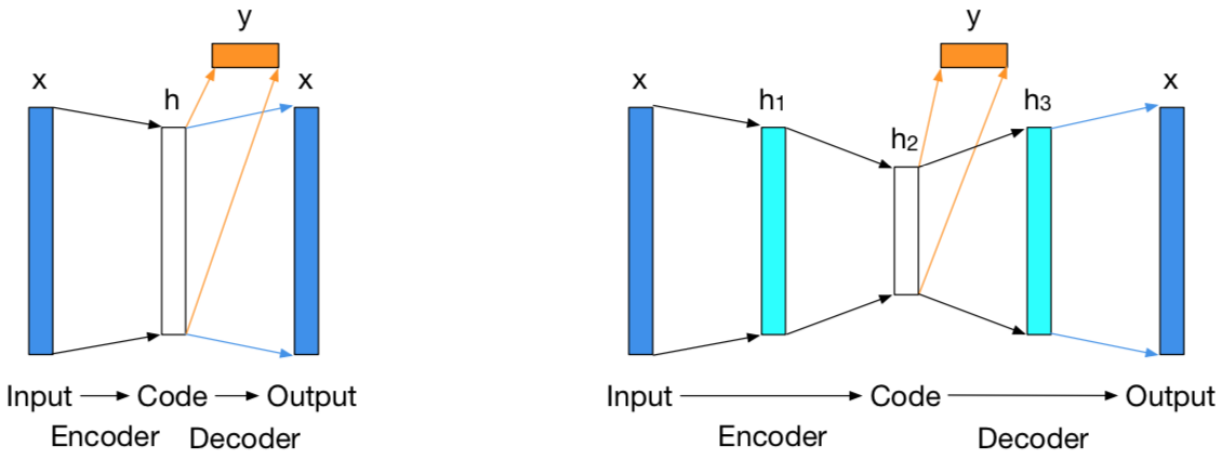
The mapping of a latent vector to the embedding space with a limited number of codes can be seen as a clustering step, where the codes are the centroid of each cluster, and latent vectors are assigned to clusters using the nearest neighbor strategy. As a result, the training of VQ-VAE can be boosted with exponential moving averages (EMA) of cluster assignment counts (Kaiser et al., 2018). Other improvements of the training algorithm (essentially an EM algorithm) and extension of the model have also been proposed. These include switching from a hard EM algorithm to a soft EM algorithm, which allows one encoding to be assigned to multiple clusters with different weights (Roy, Vaswani, Neelakantan, & Parmar, 2018), and solving the problem of index collapse, where only a small fraction of all the codes are updated and used, with the decomposed vector quantization technique (Kaiser et al., 2018). Currently, VQ-VAE is still under active research.

2.3.5 SAE

All the models above, including the standard AAE, are unsupervised autoencoder-based GMs. The supervised extensions of AAE that incorporate the data label information into the input of its discriminative network or decoder are still primarily used for data reconstruction and generation rather than used as multi-task models. One of the first models that jointly considers unsupervised data reconstruction and supervised regression/classification is the supervised autoencoder (SAE) (Le, Patterson, & White, 2018). The two learning tasks are solved at the same

time by adding a supervised loss l_s to the objective function of a standard autoencoder. The supervised loss is computed by predicting the target y of the input data x with an additional subnetwork using the latent representation generated by the encoder (Equation 2.8, Figure 2.6). The incorporation of two learning components in SAE improves the performance of each single learning task by exploiting the commonalities shared by related tasks. With this property, SAEs have been used to improve the supervised learning accuracy, learn meaningful representations for individual data, and improve the generalization performance of neural networks (Le et al., 2018; Ranzato & Szummer, 2008; Yuting Zhang, Lee, & Lee, 2016).

$$L = l_r\left(x, d(z_e(x))\right) + l_s(y, s(z_e(x))) \tag{2.8}$$



(a) (Linear) Supervised Autoencoder (b) Deep Supervised Autoencoder

Figure 2.6. The structure of SAE and deep SAE.

The Figure 1 from (Le et al., 2018).

2.3.6 RBM

The models have been introduced so far are all specific cases of directed acyclic graphs (DAGs). These models belong to a more general model class known as the probabilistic directed acyclic graphical model (this is sometimes considered as an equivalent term to Bayesian network, though the latter term emphasizes more on the conditional dependencies between variables). Another major category of graphical models is the undirected graphical model, also known as the Markov random field. One well-studied undirected graphical model is the Boltzmann machine (BM), which represents undirected dependencies between stochastic binary variables. In a BM we also have the observed or visible units, v, and hidden or latent units, h, just as in the directed graphical models (Figure 2.7). Connections are allowed between any units, including visible-to-visible connections, hidden-to-hidden connections, and visible-to-hidden connections. BM is one of the first GMs that are capable of learning internal representations, but its training is in general intractable.

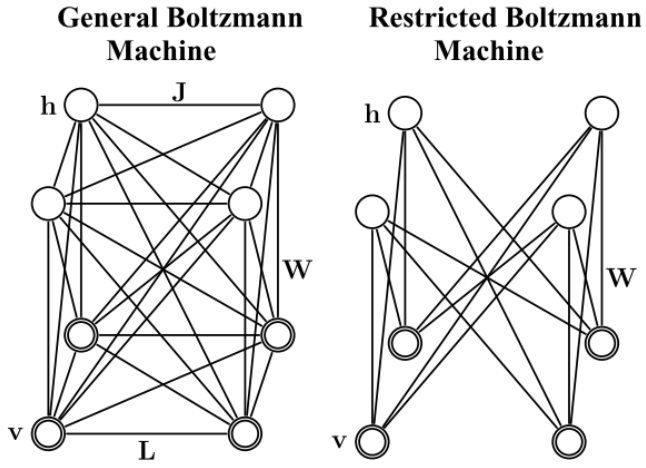


Figure 2.7. The structure of a general BM and a RBM.

Figure 1 of (Salakhutdinov & Hinton, 2009).

To reduce the complexity of training a general BM, restrictions are put on the structure of BM to only allow visible-to-hidden connections in the model. The resulting model is known as RBM (Smolensky, 1986) (Figure 2.7). Even though the idea of RBM can date back to 1986, an efficient training algorithm was not available until 2002, when Hinton proposed the contrastive divergence k algorithm (CD-k) (Hinton, 2002). The objective for training an RBM is to maximize the log-likelihood of the input data, or equivalently, minimize the average negative log-likelihood $-\log p(x)$. The partial derivative of the objective for the tth data to the parameter vector of the model θ is:

$$\frac{\partial -\log p(x^{(t)})}{\partial \theta} = E_h\left[\frac{\partial E(x^{(t)},h)}{\partial \theta}\middle| x^{(t)}\right] - E_{x,h}\left[\frac{\partial E(x,h)}{\partial \theta}\right] \tag{2.9}$$

The first term on the right-hand side, also called the positive phase, is data-dependent, where the expectation is taken w.r.t $P(h|x)$. The second term, the negative phase, is model-dependent, where the expectation is taken w.r.t $P(x,h)$. The second term is generally hard to compute, due to the exponential number of configurations of x and h. The key idea behind contrastive divergence is to replace the expectation in the second term with a point estimate at $\tilde{x}$, which is obtained by performing a Gibbs sampling starting with $x^{(t)}$. The k in CD-k is the number of iteration of the Gibbs sampling. Usually, $k = 1$ works well enough.

A single iteration in CD-1 takes the following steps:

1. Take an observed data v, compute the probabilities of the hidden variables and sample a hidden vector h from the Bernoulli distribution defined by the probabilities.
2. Take the outer product of v and h, and set it as the positive gradient.
3. With h, sample a reconstructed data v'.
4. With v', resample a hidden vector h' as in step 1.

5. Take the outer product of v' and h', and set it as the negative gradient.
6. Update weights, W, with the difference between the positive gradient and the negative gradient.
7. Update biases, c and b, with the difference between v and v', and difference between h and h', respectively.

The iterative update continues until the parameters converge or a stop criterion is met.

Though the ordinary RBM and its training algorithm are primarily for binary input data, RBM can be applied to real-valued data by using the Gaussian rectified transformation (Welling, Rosen-Zvi, & Hinton, 2005). In this case, the update rule for the hidden units remains the same as in the CD-k algorithm, while the update rule for a visible unit is sampling from a Gaussian distribution with a unit variance (identity covariance matrix) and a mean of $c + W^T h$. Learning an RBM with Gaussian units can be slow, and it is usually impractical to train a deep model with multiple real-valued RBMs. Instead, people train a Gaussian-to-binary RBM first to learn a binary representation from the continuous input data, then use the binary representation as input for training other deep models, like the DBM introduced below (Salakhutdinov & Hinton, 2009). Meanwhile, as mentioned in the AE section, RBM can be used to pre-train other deep models in a layer-wise manner, which facilitates learning features that preserve hierarchical information. Actually, a DAE can be seen as composed of two, symmetrical DBNs. This in turn makes the deep model more interpretable.

2.3.7 DBN

It may sound tempting to stack several RBMs together to obtain a deep model with stronger learning potential. The resulting model, however, is not a deep Boltzmann machine (DBM) as one

may have expected, but a DBN. Figure 2.8 shows the difference between DBM and DBN. In a DBN, all the edges between layers are directed except the ones between the two top layers. The top two layers form an ordinary RBM, while the lower layers form a sigmoid belief network. A DBN is often trained by first performing a greedy, layer-wise pre-training step with RBMs, using the hidden units of one RBM as the input data for training a higher-level RBM; then running a fine-tuning to optimize the parameters altogether. This final fine-tuning step is completed with a revised version of the CD-k algorithm, where the positive gradient and negative gradient are backpropagated for updating multiple hidden layers.

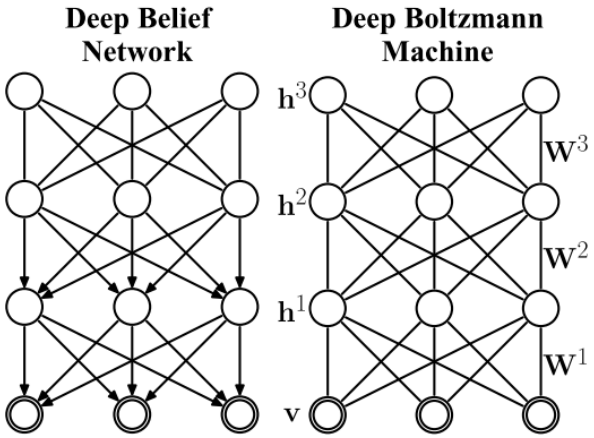


Figure 2.8. The structure of a general DBN and a DBM.

Figure 2 of (Salakhutdinov & Hinton, 2009).

On the other hand, a DBM, with its fewer constraints on the dependency direction between hidden units, is a more flexible and powerful model than a DBN. Its training procedure, however, is also more complex (Salakhutdinov & Hinton, 2009), thus is rarely used in general applications.

2.3.8 Adding regularizations

For all the models introduced in this section, regularization terms like sparsity or weights regularization can be added to the corresponding loss or objective function to create overfitting-resistant extensions. This results in the regularized AE (adding a L2 norm to the weights), the sparse AE (adding a KL distance between the Bernoulli distribution defined by the values of hidden nodes and the Bernoulli distribution defined by the desired hidden nodes activation proportion), and sparse RBM (Lee, Ekanadham, & Ng, 2008; Nair & Hinton, 2009). Some of these techniques were explored in this project.

2.4 Supervised learning algorithms for drug sensitivity prediction

Previous systematic evaluations of drug sensitivity prediction algorithms have shown that best-performed predictors were generally from three classes of models, that is, the Elastic net regression, random forest, and SVM. In this project, all three models were explored for predicting drug response or class based on the cell line representations learned by DGMs. The basic theories behind these three classes of models are introduced in this section.

2.4.1 Elastic net

An Elastic net is a regularized regression method, usually incorporated with linear regression or logistic regression models, which linearly combines the Lasso regularization (L1 penalty) and ridge regularization (L2 penalty). When using the Lasso regularization alone, the L1

term can be too strict for selecting variables, especially for high dimensional data and highly correlated feature variables. The inclusion of the L2 term helps overcome some of the limitations. The objective function for training an Elastic net is given in Equation 2.10, where βs are the coefficients/weights of variables to be estimated, and λs are the pre-defined coefficients for adjusting the strength of regularization. It has been further shown that the Elastic net can be reduced to the linear SVM, where the solution β defines the hyperplane obtained from a SVM for a binary classification problem transformed from the original regression task (Zhou et al., 2015).

$$\hat{\beta} = argmin_{\beta}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1) \quad (2.10)$$

In this project, we modeled the drug sensitivity prediction task as a binary classification problem and trained an Elastic net logistic regression model for each drug of interest to predict the probability of sensitivity of cell lines.

2.4.2 Random forest

The random forest is an ensemble learning method that fits a set of decision trees for both classification and regression tasks. The use of a set of trees helps reduce prediction variance and avoid the overfitting problem that is often observed when using a single decision tree (Ho, 1995). A random forest is typically trained using the bootstrap aggregation technique, where the trees are trained on different random samples drawn from the input data with replacement (also known as the bagging method). When doing classification, the mode of the classes of individual trees is returned; when doing regression, the mean prediction of the individual trees is returned. In this project, we used the random forest for predicting classes of drugs given drug perturbed cell line genomics data.

2.4.3 SVM

A SVM is a discriminative model that is primarily used for classification by constructing separation hyperplanes that maximize the margin between each pair of classes (Cortes & Vapnik, 1995). The data points that define the margins are called the "support vectors". A SVM is only dependent on the support vectors as the other data points would not affect the separation hyperplane when they are away from the margins. Suppose for a binary classification task with class labels 1 and -1, and the separation hyperplane is defined with $\omega * x - b = 0$, then the margin of class 1 is $\omega * x - b = 1$, and the margin of class -1 is $\omega * x - b = -1$. It can be derived that the distance between the margins is $\frac{2}{\|\omega\|}$, and the distance is to be maximized.

If the training data are linearly separable, solving a SVM can be written as an optimization problem as

$$\text{minimize } \|\vec{\omega}\|$$

$$\text{subject to } y_i(\vec{\omega}\vec{x_i} - b) \geq 1 \text{ for } i = 1, \dots n.$$

This is also known as the hard-margin SVM ("hard-margin" means no data point is allowed to lie on the wrong side of the margin). When the data are not linearly separable, the hinge loss function $max(0, 1 - y_i(\vec{\omega}\vec{x_i} - b))$ is introduced into the optimization problem and results in the soft-margin SVM as

$$\text{minimize } \left[\frac{1}{n}\sum_{i=1}^{n} max(0,1 - y_i(\vec{\omega}\vec{x_i} - b))\right] + \lambda\|\vec{\omega}\|^2.$$

In this case, misclassified data points can be, to some extent, tolerated. The above objective function can be rewritten as a constrained optimization problem with a differentiable objective function as

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n} \zeta_i + \lambda\|\omega\|^2$$

subject to $y_i(\omega \cdot x_i - b) \geq 1 - \zeta_i$ and $\zeta_i \geq 0$ for all i.

This is known as the primal problem. Its Lagrangian dual is

maximize $f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i \cdot x_j) y_j c_j$

subject to $\sum_{i=1}^{n} c_i y_i = 0$ and $0 \leq c_i \leq \frac{1}{2n\lambda}$.

The dot product between each pair of data in the dual problem allows the kernel trick to be played (replacing $(x_i \cdot x_j)$ in the above objective with $k(x_i \cdot x_j)$), with which non-linear hyperplanes can be learned for classifying the data.

The SVM can also be extended for regression tasks by solving the following optimization problem

minimize $\frac{1}{2}\|\omega\|^2$

subject to $y_i - wx_i - b \leq \epsilon$ and $wx_i + b - y_i \leq \epsilon$.

In this project, we also used SVM for predicting classes of drugs given drug perturbed cell line genomics data.

3.0 Chapter 1: learning cellular signaling component representations with network-based models

3.1 Introduction

Cancer is a complex genetic disease, mainly caused by somatic genome alterations (SGAs) that affect oncogenic processes (Croce, 2008). Such alterations include mutations, copy number alterations, DNA structure variants, epigenetic alterations, and other genomic variations (Vogelstein et al., 2013). Among the alterations, some are more causally related to the disease, known as the driver SGAs that activate the oncogenic process by perturbing genes in cellular signaling pathways that regulate homeostasis (Vogelstein et al., 2013).

Cancers are heterogeneous in that tumors originating from the same tissue often exhibit significantly different molecular and clinical phenotypes, leading to differences in responses to treatments and patient survival. This inter-tumor heterogeneity is due to distinct disease mechanisms that underlies the development of an individual tumor, which usually results from different compositions of pathway aberrations. Understanding disease mechanisms of an individual tumor and further identifying common patterns of disease mechanisms among a cohort will not only provide insights into cancer biology but can also promote personalized therapy.

So far, it remains a challenge to discover disease mechanisms of cancers solely based on SGA data for the following reasons: First, a tumor usually hosts from hundreds to over a thousand SGA events (Ciriello et al., 2013), among which the majority has a relatively low-occurrence frequency in a tumor cohort. As a result. it is difficult to find statistically significant patterns in SGA events. Second, among all the SGAs observed in a tumor, driver SGAs only take up a small

amount, whereas the majority of others is non-consequential regarding oncogenesis (passengers) (Carter et al., 2009; Ciriello et al., 2013; Dees et al., 2012; Mermel et al., 2011; Reva, Antipin, & Sander, 2011; Tamborero, Lopez-Bigas, & Gonzalez-Perez, 2013; Vogelstein et al., 2013; Weinstein et al., 2013; Zack et al., 2013). Identifying driver SGAs of an individual tumor is at best challenging, which in turn makes it even more difficult to find co-occurrence patterns among driver SGAs. Third, an oncogenic pathway can be perturbed by different SGAs affecting distinct members of the pathway (Weinstein et al., 2013). For example, the phosphoinositide 3-kinase (PI3K) pathway can be aberrantly activated by mutation/amplification of *PIK3CA*, mutation/deletion of *PTEN*, or mutation of *AKT1* (Lin et al., 2014; Vivanco & Sawyers, 2002), and so on. There is no simple way to determine whether two distinct SGA events observed in two different tumors are affecting a common pathway. Consequently, it is hard to use SGA data to determine which pathways are aberrant in a tumor and to further identify combination patterns of pathway aberrations.

On the other hand, gene expression profiles have been widely applied to identify molecular subtypes of cancers originating from a common tissue through clustering analysis. In many cases, such transcriptome-based subtypes present different outcomes and different responses to therapies (C. G. A. Network, 2012; C. G. A. R. Network, 2011; Verhaak et al., 2010). While it is relatively straightforward to identify differentially expressed genes in these cancer subtypes, it is unclear which pathways drive their differential expression. Furthermore, the subtyping results can be heavily influenced by cell-type-specific expressions, leading to subtypes that are divided based on the origins of cells rather than disease mechanisms. For example, some breast cancer subtypes are based on the cell of origin, such as basal vs. luminal cells. In general, it would be ideal to identify modules of genes whose expressions are regulated by a common oncogenic pathway and use the

modules to discover combination patterns of pathway aberrations and classify tumors according to disease mechanisms rather than the tissue of origin.

In this chapter, we preliminarily explore the use of a graph-based machine learning framework for learning disease-mechanism informative representations, in the form of DEG modules, from genomics data. The framework we propose here transfers the information from genetic alterations to clinical outcomes via examining the expression modules that reflect the status of transcriptomic program perturbations.

The framework takes as input the causal inferences produced by a Bayesian causal learning algorithm we developed and referred to as the Tumor-specific Causal Inference algorithm (TCI) (Cai et al., 2018). TCI estimates the causal relationships between SGAs and somatic genome alterations (DEGs) within an individual tumor (Figure 3.1), which enables us identify DEG modules that are each regulated by a common SGA and use it as a signature of the pathways perturbed by the SGA. Specifically, we developed a network-based framework to construct a DEG network in which DEGs are connected by weighted edges if they are co-regulated by a common set of SGAs. We then applied spectral clustering on the network to identify modules of DEGs where the members share driver SGAs. We used the expression status of a DEG module as a surrogate measure of the aberration status of the corresponding regulatory pathways and represented a tumor as a vector in pathway space that reflects the combination of pathway aberrations in the tumor. With these pathway representative feature vectors, we identified subgroups of tumors sharing similar aberration patterns that exhibit different survival outcomes.

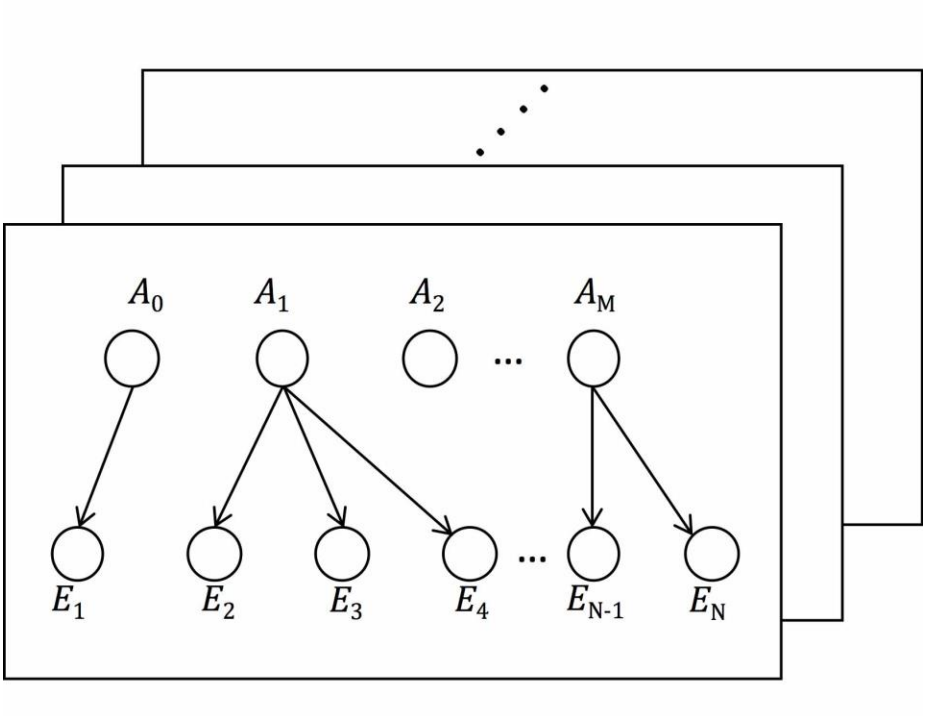


Figure 3.1. The diagram of the TCI algorithm.

Each plate represents a tumor sample. Based on a causal Bayesian network model, TCI infers causal relationships between genes that carry somatic alterations (A) and genes that are differentially expressed (E). A_0 designates all the factors other than gene alterations (e.g., the cellular environment). Each E receives one, and only one, A as its cause, and each A can be the parent of multiple Es.

We evaluated this computational framework on breast cancer (BRCA) and glioblastoma multiformes (GBM) data, and we report the results here. The same approach can be applied to other cancer types, with minor modifications. This pilot project represents the first step towards the goal of better cellular state representation learning with machine learning techniques.

3.2 Methods

3.2.1 Significant TCI causal inference generation

The machine learning framework we developed takes the tumor-specific inferences of TCI as input to construct DEG networks. The TCI algorithm is a Bayesian Causal Network model, which models the SGAs and the DEGs as a bipartite graph and adds edges between the two gene sets that represent causal relationships (Cai et al., 2018). In particular, for each tumor sample, the algorithm assigns each DEG one, and only one, SGA as its cause by ranking all candidate SGAs based on the BDeu score (Heckerman, Geiger, & Chickering, 1995); each SGA can be assigned to multiple DEGs (Figure 3.1). The biological intuition behind this setup is that the differential expression of a gene is mainly due to the direct interaction between this gene and a single SGA; all indirect interactions between the gene and other SGAs are relatively minor if the direct interaction is recognized. On the other hand, one SGA can affect the expression status of multiple genes at the same time.

We collected omics data of 5,097 tumors across 16 cancer types (includes 891 BRCA tumors and 144 GBM tumors) from The Cancer Genome Atlas (TCGA) dataset (program). A gene is considered a somatic alteration carrier if one or more somatic mutations (SM), or somatic DNA copy number alterations (SCNA), were observed on it; a gene is recognized as a DEG if its expression level significantly deviates from the mean of its expression distribution in healthy tissue. We applied TCI to each of these tumors and identified tumor-specific causal relationships between SGAs and DEGs (Cai et al., 2018). The TCI causal inferences were further filtered through a series of empirical standards to obtain robust and significant results to be used as the input data for our framework. The filtering standards we used are:

- A SGA-DEG causal relationship is considered valid if its posterior probability is larger than the posterior probability estimated in a random permutated experiment.
- A SGA is called a driver in a tumor if TCI assigns it to be a cause of 5 or more DEGs in the tumor.
- A SGA is called a significant driver if it is called a driver in 30 or more tumors AND it is called driver in at least 25% of tumors where it is observed as a SGA.
- A SGA-DEG is called a significant causal relationship if the SGA is a significant driver AND the DEG is caused by this SGA in at least 50 tumors OR 20% of the tumors where the SGA is called a driver.

Some tumor samples had no inference left after the filtering step and were excluded from the following experiments. Consequently, the significant TCI inferences we used for BRCA and GBM analyses were from 874 BRCA tumor samples and 143 GBM tumor samples, respectively. For a more detailed overview of the data generation and processing procedure, please refer to the original TCI paper (Cai et al., 2018).

3.2.2 DEG module identification

3.2.2.1 DEG network construction

The TCI significant inferences were used to construct DEG networks in the form of a weighted, undirected graph. When constructing the graph for a single cancer type, the corresponding subsets of significant inferences were extracted. Each node in this graph represents a DEG that was identified in more than 10% of the tumors. Edges were added between DEG pairs where the two DEGs were co-regulated in the same tumor by the same SGA. The edge weight is defined as the frequency of the co-regulation, which equals the number of tumors in which the co-

regulation took place. The weighted, undirected graph was represented in the form of a symmetric affinity matrix, where the affinity in row i column j is the edge weight between DEG_i and DEG_j.

3.2.2.2 Spectral clustering

In our framework, we use spectral clustering to identify modules from the DEG networks. Specifically, we implemented a new extension of spectral clustering, which was derived from the algorithm described by Ng, 2002 (Ng, Jordan, & Weiss, 2002). In our implementation, a DEG network affinity matrix is first converted to a pseudo-distance matrix by taking the inverse of each affinity value. The distance matrix is then transformed into an optimized affinity matrix with the Gaussian kernel, as shown in Equation 3.1

$$A_{ij} = \exp\left(D_{ij}^2/2\sigma^2\right) \tag{3.1}$$

Here D_{ij} and A_{ij} are the pseudo-distance and optimized affinity between DEG_i and DEG_j. The standard deviation σ of the Gaussian kernel is selected based on the distribution of pseudo-distances to convert short distances to high affinities and suppress long distances (0.05 for BRCA and 0.1 for GBM). The remaining steps are identical to steps 2-6 in the standard spectral clustering algorithm (Ng et al., 2002). In brief, a Laplacian matrix is computed from the optimized affinity matrix, from which the k largest eigenvectors are extracted to project the data into a k dimensional feature space. The data points are then clustered via the *k*-means algorithm.

With the use of *k*-means, the spectral clustering result partially depends on the choice of k and the random initialization of the *k*-centres of clusters. To determine the value of k (i.e., the number of DEG modules), we first tried consensus spectral clustering with $k = 5, 10, 15, 20$. We then narrowed down to the range between the two adjacent ks that gave the most stable consensus matrices, and tried each k from the range. For generating the consensus matrix of each k, a spectral

clustering was repeated independently for 100 times with different random initializations. The value of k was selected such that further increasing k would result in modules that were unstable, with significant overlaps across modules on the consensus matrix. Such overlaps indicated that the data points that were assigned to two different modules were often clustered into the same group across independent runs. This suggests that the two modules should be merged and the k being used was too large. The final module assignments we used were generated by running the clustering algorithm one more time with the selected k.

3.2.3 Survival analysis

3.2.3.1 Survival feature dataset construction

When constructing the dataset for survival analyses, each DEG module identified by spectral clustering was treated as a single feature and was represented with the mean of the expression levels of all DEGs in the module. This representation can be seen as a surrogate measure for the aberration status of the signaling pathway that regulates genes in each module. Other clinical features of interest (e.g. age at diagnosis, etc.) were also added. For BRCA, the gene expression, clinical, and survival data used were from the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) project (Curtis et al., 2012), accessed through the Synapse repository (synapse.sagebase.org, ID syn1688369). For GBM, the microarray gene expression data and clinical data were downloaded from TCGA through the Firehose browser of the Broad Institute. The DEG modules were obtained with TCI inferences that were produced using RNA-seq data from the TCGA database, and some DEGs were not available from the METABRIC expression data or the TCGA GBM microarray data. As a result, the number of DEGs used to compute each DEG module feature was smaller than the original number of DEGs in each module.

We refer to these DEGs as effective DEGs (Table 3.1 and Table 3.2). From the METABRIC clinical data, we extracted eight features and added these to our BRCA dataset -- the age at diagnosis, size of tumor, grade of disease, lymph node assessment, tumor histology type, ER status, PR status, and Her2 status. For tumor histology type we only considered three factor levels, including IDC-TUB, IDC-MUC, and IDC-MED. For GBM, patient age at diagnosis was extracted and added to our dataset as the only clinical feature. Since clinical features typically took different scales of values, all features (including DEG module features) were normalized across patients by subtracting the mean of values and dividing by the standard deviation. The final survival dataset took the form of a table in which each patient was represented with a feature vector, a survival time, and a binary value indicating the death status (0 for alive and 1 for dead). The BRCA survival feature dataset contained 1,981 patients and the GBM dataset contained 524 patients.

3.2.3.2 Patient groups identification

Patient groups were identified using consensus PAM clustering. We used the consensus clustering function from the R package *ConsensusClusterPlus* (Wilkerson & Hayes, 2010), version 1.38.0. The number of patient groups was determined using the consensus matrix and the area under the consensus cumulative distribution function curve (AUCDFC). This was done by clustering with the number of groups that varies from 2 to 15 (200 resamplings for GBM, 100 resamplings for BRCA) and selecting the point at which there was no significant overlap between the resulting groups on the consensus matrix and at which further increases the number of groups would not lead to a significant increase in the AUCDFC.

3.2.3.3 Patient groups survival models

We used the R package *survival* (Therneau) version 2.41.3 to generate the Kaplan-Meier plots of patient groups, run log-rank tests, and do Cox regressions. The prediction performances of Cox regression models were compared by computing the C-index between the model outputs and the true survival times (Harrell Jr, Lee, & Mark, 1996; Pencina & D'Agostino, 2004).

3.2.4 Code availability

The source code for spectral clustering has been deposited to GitHub and available at https://github.com/evasnow1992/SpectralClustering.

3.3 Results

3.3.1 DEG modules

We first applied the TCI on the target datasets (BRCA or GBM, see Methods) to obtain tumor-specific causal inferences between SGAs and DEGs. From these inferences, we identified a set of driver SGAs and their signature DEGs (Cai et al., 2018). We then constructed a network of the signature DEGs to represent the co-regulation relationships among the DEGs. Specifically, each node in the network represents a DEG, and an edge was added between two DEGs if they were co-assigned to the same SGA by TCI in at least one-tenth of the tumors. The edge weight is proportional to the number of tumors in which the pair were co-regulated by a common driver SGA (note that the regulator SGA for a pair of DEGs can be different in different tumors; see

Methods). Our assumption is that the higher frequency that two DEGs are co-regulated by a common set of SGAs, the higher probability that these two DEGs are regulated by the same upstream signaling pathway perturbed by these SGAs. The DEG networks of BRCA and GBM were constructed using TCI results from 874 BRCA and 143 GBM tumors, respectively. The resulting networks contained 1,747 DEG nodes for BRCA and 3,576 DEG nodes for GBM.

We next identified modules of DEGs using the spectral clustering algorithm (Ng et al., 2002), where each module consists of a set of DEGs that are likely co-regulated by a common pathway. Specifically, we repeatedly performed spectral clustering with different random initializations of cluster centers (see Methods) and then conducted a consensus clustering analysis by pooling the results and identifying DEGs that were consistently assigned to a common module. Using this approach, we identified 7 stable DEG modules for BRCA and 15 for GBM. The module size varies from a few DEGs to over hundreds of DEGs (Figure 3.2, Supplementary Table S3.1, and Table S3.2). In comparison, when other traditional clustering methods such as hierarchical clustering were used, the resulting DEG modules were inconsistent across independent runs with different random initializations (Figure 3.3). This supports the advantage of using spectral clustering for identifying stable DEG modules.

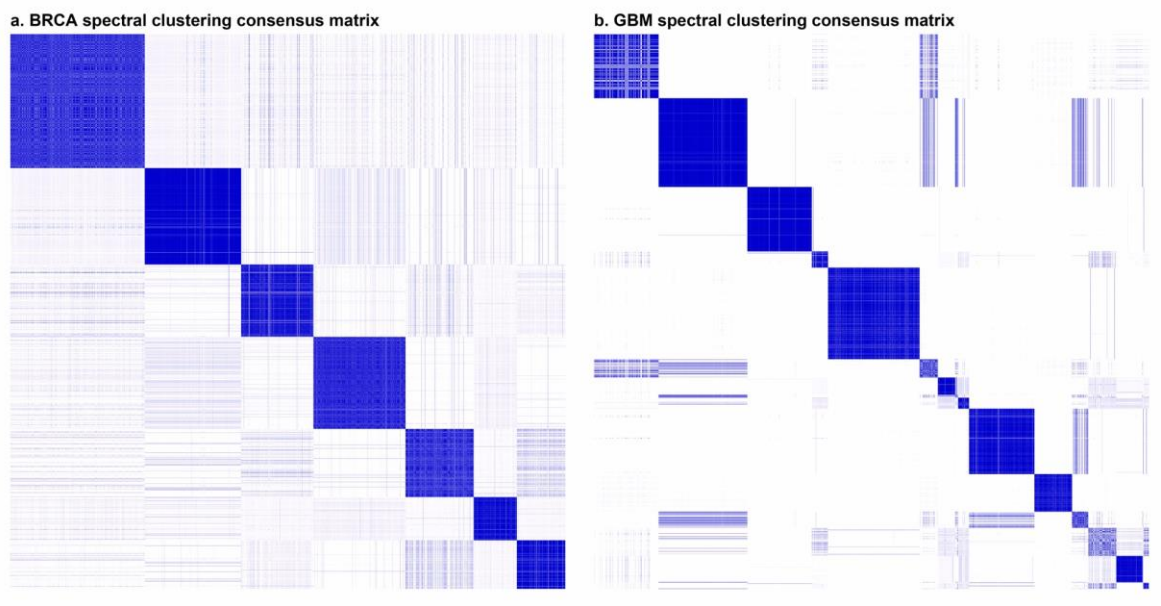


Figure 3.2. The consensus matrices of spectral clustering for identifying DEG modules.

Spectral clustering was generated with 100 independent repeats of runs. The higher the frequency two DEGs were clustered into the same module, the darker blue the corresponding spot on the matrix. Each block sitting on the diagonal corresponds to a DEG module. The low overlapping across blocks indicates that spectral clustering was able to identify robust modules.

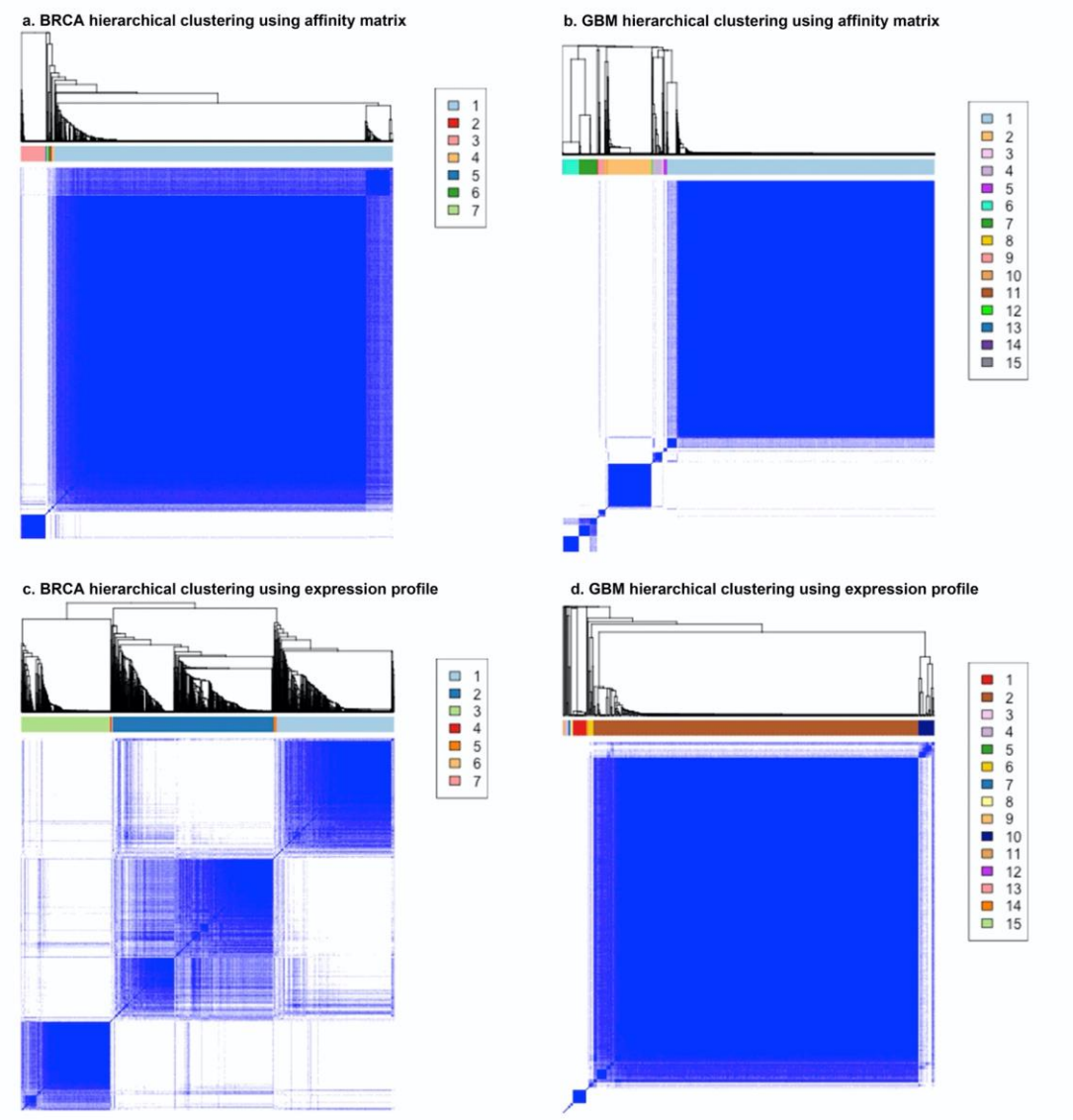


Figure 3.3. The consensus matrices of hierarchical clustering for identifying DEG modules.

(a) and (b) hierarchical clustering using affinity matrix. (c) and (d) hierarchical clustering using expression profile with 1 - Pearson correlation as distance. The higher the frequency two DEGs are clustered into the same module, the darker blue the corresponding spot on the matrix. Each block sitting on the diagonal corresponds to a DEG module. The single dominant module composed of the majority of genes in (a), (b) and (d), and the overlapping across modules in (c) suggest that hierarchical clustering was unable to untangle the correlations between DEGs to identify robust modules, no matter whether the correlations were measured as the co-regulation frequency (affinity) or expression profile distance.

To understand what function module each DEG module may represent, we ran a gene set overlap analysis on each DEG module against all gene sets in the Molecular Signature Database (MSigDB) (Liberzon et al., 2011). The top 10 overlapping gene sets, ranking by the hypergeometric distribution p-value, are listed in Supplementary Table S3.3 and Table S3.4. All BRCA DEG modules are correlated with some cancer-related gene sets, and most of them (modules 1, 3, 4, 5, 6, and 7) significantly overlap with breast cancer subtype-specific gene sets. For example, module 1 contains genes down-regulated in the luminal B subtype and genes up-regulated in the basal-like subtype.

For GBM, half of DEG modules overlap with tissue-specific gene sets, including those of neuron, synapse, and brain. Among the other modules, module 3 stands out with its enrichment of genes in MODULE_84, GO_IMMUNE_SYSTEM_PROCESS, and GO_IMMUNE_RESPONSE that represent immune and inflammatory responses. We hypothesize that module 3 represents a functional module that interacts with the immune system, which when it becomes defective helps a tumor escape immune surveillance. This conjecture, however, requires additional experimental confirmation studies.

3.3.2 Candidate pathways underlying DEG modules

Each DEG module contains a group of genes that are frequently co-regulated by the same set of SGAs. Accordingly, we extracted the SGAs that underlie the co-regulation for each DEG module. We call an SGA a dominant SGA of a DEG module if it is responsible for over 10% of the co-regulations between DEG pairs in the module. Although for each module there can be hundreds of SGAs that contribute to the DEG co-regulations, usually no more than three SGAs turn out to be dominant. The dominant SGAs together take up about 90% or more of all the co-

regulation instances. Furthermore, different DEG modules present distinct dominant SGAs, except a few members overlapped (Table 3.1 and Table 3.2). This indicates that each DEG module likely results from a different upstream signaling pathway that is perturbed by a few major drivers.

Table 3.1. The composition of DEG modules of BRCA.

Module Index	# of DEG	# of Effective DEGs	Dominant SGAs (Prop. of Co-regulation)
Module 1	288	259	*CDH1* (60.2%), *GATA3* (20.8%), *PIK3CA* (12.4%)
Module 2	225	202	*PTEN* (66.1%), *PIK3CA* (19.6%)
Module 3	155	138	*ZFHX4* (44.7%), *RYR2* (22.9%)
Module 4	302	281	*GATA3* (92.7%)
Module 5	214	184	*ERBB2* (58.0%), *PIK3CA* (17.4%)
Module 6	135	124	*TP53* (96.4%)
Module 7	428	387	*PIK3CA* (90.5%)

Table 3.2. The composition of DEG modules of GBM.

Module Index	# of DEG	# of Effective DEGs	Dominant SGAs (Prop. of Co-regulation)
Module 1	413	255	*TP53* (99.69%)
Module 2	128	72	*PTEN* (50.0%), *SEC61G* (47.6%)
Module 3	529	347	*CDKN2A* (98.5%)
Module 4	170	81	*MARCH9* (97.9%)
Module 5	599	347	*PTEN* (98.3%)
Module 6	425	255	*SEC61G* (98.8%)
Module 7	11	7	*EGFR* (68.9%), *TP53* (31.0%)
Module 8	242	150	*CDKN2B-AS1* (94.8%)
Module 9	165	88	*AGAP2-AS1* (58.1%), *CHIC2* (41.4%)
Module 10	71	42	*CDKN2B* (94.2%)
Module 11	428	260	*EGFR* (97.9%)
Module 12	85	62	*CDKN2A* (69.2%), *PTEN* (30.1%)
Module 13	142	88	*GSX2* (75.0%), *RYR2* (11.4%)
Module 14	123	74	*MTAP* (94.5%)
Module 15	45	26	*TTN* (91.9%)

For BRCA, all dominant SGAs, except *ZFHX4* and *RYR2*, are well-known drivers of BRCA (Ciriello et al., 2015; Curtis et al., 2012; C. G. A. Network, 2012; Stephens et al., 2012). *ZFHX4* was found to play a role in maintaining tumor cell state in GBM (Chudnovsky et al., 2014); our previous experimental study also indicates that *ZFHX4* regulates the expression of certain target genes as predicted by the TCI algorithm (Cai et al., 2018). On the other hand, while some studies suggest that alterations on *RYR2* are likely passenger events, TCI consistently discovered

that SGAs in *RYR2* have impacts on certain DEGs. Therefore, we propose *ZFHX2* and *RYR2* to be novel drivers for BRCA.

For GBM, most dominant SGAs are known drivers of this cancer type (Brennan et al., 2013; Hou & Ma, 2014; Verhaak et al., 2010) except *MARCH9*, *AGAP2-AS1* (*AGAP2* antisense RNA 1), *CHIC2*, *GSX2*, *RYR2*, *MTAP*, and *TTN*. For these exceptions, excluding *MARCH9* and *TTN*, previous works support that they are potential novel drivers of GBM. Specifically, *AGAP2-AS1* and *GSX2* are known to be associated with neuron system development (Waclaw, Wang, Pei, Ehrman, & Campbell, 2009; Xia et al., 2003) and, therefore, alterations on these genes could be exclusive drivers of GBM. *CHIC2* has been found to be associated with myeloid leukemia (Pardanani et al., 2003), and *MTAP* has been proposed as a tumor suppressor for BRCA (Christopher, Diegelman, Porter, & Kruger, 2002). For *MARCH9*, on the other hand, we consider it to be a passenger because it is on the same chromosome location 12q14.1 as *AGAP2-AS1*; they are frequently co-affected by genomic alteration events. *TTN* was found to be associated with BRCA and other cancer types (Greenman et al., 2007; Toss & Cristofanilli, 2015), but it is generally considered as a passenger as its long polypeptide structure may bias its mutation frequency (C. G. A. R. Network, 2011).

Based on the dominant SGAs, we can infer what signaling pathway or function module each DEG module represents. *CDH1* and *GATA3* are the first two dominant SGAs of BRCA's DEG module 1, and they are also two well-known drivers of BRCA (Ciriello et al., 2015; C. G. A. Network, 2012). 50.1% of TCGA BRCA samples (891 samples from the input data of TCI) have mutations in *CDH1*, *GATA3*, or *PIK3CA*, which suggests module 1 as the most associated function module with the disease mechanism of BRCA. With dominant SGAs *PTEN* and *PIK3CA*, DEG modules 2 and 7 represent the PI3K/Akt signaling pathway, which is known as one of the most

commonly activated pathways in cancer (Liu, Cheng, Roberts, & Zhao, 2009). The sharing of the dominant SGA *PIK3CA* across modules 1, 2, 5, and 7 suggests that although each module is considered to perform a relatively independent function, they are communicating with each other through interactions within a common signaling pathway. Module 3 contains two novel drivers, *ZFHX4* and *RYR2*, which cover 44.7% and 22.9% edges (pairs of DEGs) respectively. This may represent a novel functional module that would support the development of BRCA for some subgroups of patients (dominant SGAs found in 18.2% samples). Module 4 has only one dominant SGA, *GATA3*, which represents the module resulting from a single driver rather than from the interactions between multiple drivers like module 1. Module 5, with its most dominant SGA being *ERBB2*, represents another important signaling pathway in BRCA, the ErbB/HER signaling pathway (Stern, 2000). Module 6, on the other hand, represents the most commonly inactivated pathway in cancer, the p53 pathway (Joerger & Fersht, 2016). Therefore, some of the BRCA DEG modules are more representative of general cancer signaling pathways, whereas the others are more specific to this cancer type.

Similarly, in GBM, module 1 represents the p53 pathway. Modules 2, 5, and 12, sharing the dominant SGA *PTEN*, communicate with each other through the PI3K/Akt signaling pathway. Modules 3, 8, 10, and 12, with the most dominant SGA being *CDKN2s* (commonly deleted in GBM) (Brennan et al., 2013), represent function modules controlled by the cell cycle process. Modules 6, 7, and 11, with dominant SGAs being *SEC61G* and *EGFR* that were found specifically amplified in GBM (Hou & Ma, 2014; Kleihues & Ohgaki, 1999), represent the EGF/EGFR pathway. Modules 4, 9, 13, and 14, which have the most novel drivers, are potentially newly discovered functional modules that guide tumor development for some subgroups of GBM patients (dominant SGAs found in 19.7%, 28.9%, 24.6%, and 39.4% samples, respectively).

3.3.3 Identification of patient subgroups based on DEG module status

Based on the hypothesis that the expression status of a DEG module reflects the state of the pathway that regulates this module, we partitioned the BRCA and GBM patients into subgroups, using the expression status of the DEG modules as features. To this end, we used the BRCA data from the METABRIC project (Curtis et al., 2012), which has relatively complete gene expression and survival data of close to 2,000 breast cancer patients. The BRCA feature dataset used for clustering patients consists of the constructed DEG module features and 8 clinical features that are correlated with survival outcomes (see Methods). For GBM, we used the gene expression and clinical data provided by the TCGA. The GBM feature dataset consists of the constructed DEG module features and age at diagnosis (the only clinical feature we considered, see Methods). Patient subgroups were identified using Partitioning Around Medoids (PAM, also known as k-medoids) consensus clustering, as consensus clustering generally produces more robust and consistent clusters (Swift et al., 2004). PAM was selected, for it provides a center of each resulting group with which new data can be classified, an advantage compared to the hierarchical clustering and it is generally more robust to noise and outliers than k-means (RDUSSEEUN, 1987).

When all clinical features and DEG modules were used, 5 and 6 patient groups were identified for BRCA and GBM, respectively (Figure 3.4a and Figure 3.5a. Supplementary Table S3.5 and Table S3.6). The Kaplan-Meier curves of patient groups (Figure 3.4b and Figure 3.5b) show that different patient groups have different survival patterns. On average, BRCA patients have higher survival rates than GBM patients. This is consistent with the longer mean survival time of BRCA (2,951 days for our dataset) than GBM (510 days for our dataset). The p-value of the log-rank test for survival difference is $< 2x10^{-16}$ for BRCA and $8.96x10^{-6}$ for GBM, which suggests a significant difference between the survival distributions of the patient groups. For

BRCA, group 1 has the best survival outcome, and group 5 has the worst survival outcome (Figure 3.4b). For GBM, groups 4 and 5 have nearly twice the survival chance at the beginning compared to the other four groups (Figure 3.5b).

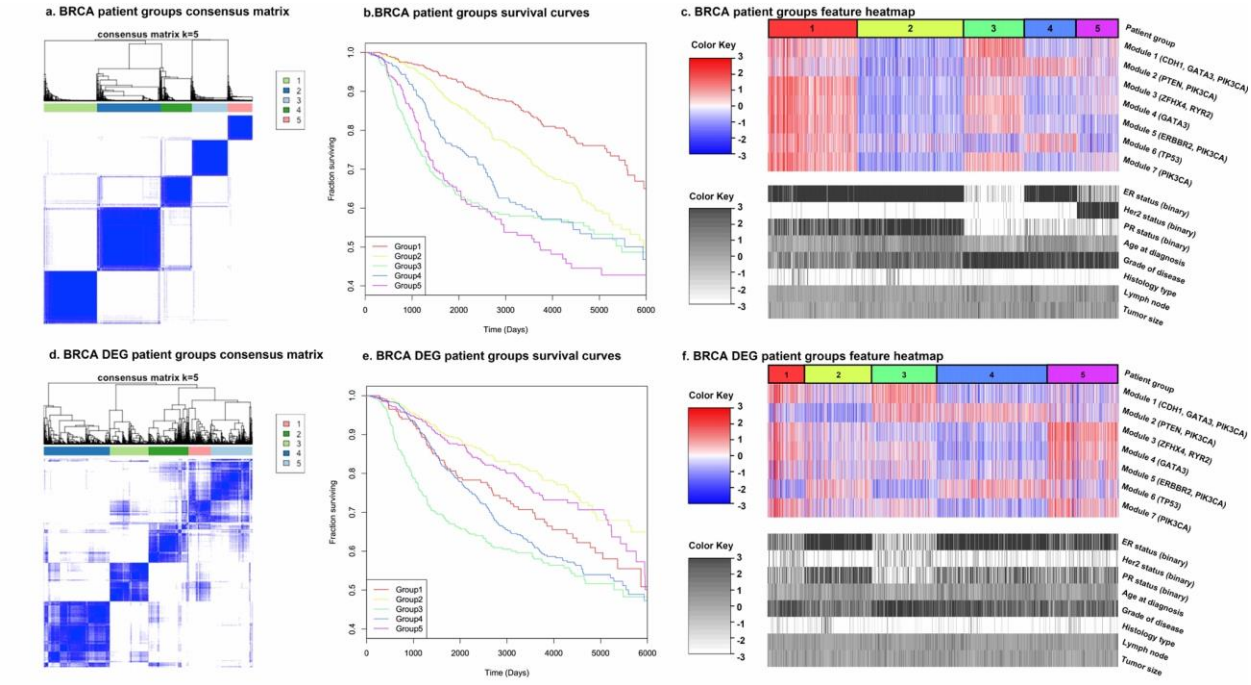


Figure 3.4. The consensus matrices of PAM consensus clustering for identifying patient groups for BRCA, the survival curves of the resulting patient groups, and the feature heatmaps.

Patient groups were identified using all DEG modules and clinical features. DEG patient groups were identified using only DEG modules. For the heatmaps, the features were normalized across all patients. Values above 3 and below -3 are compressed into 3 and -3, respectively. The dominant SGAs of each DEG module are listed by the module index. The values of the clinical features of each DEG patient group are also given as a reference.

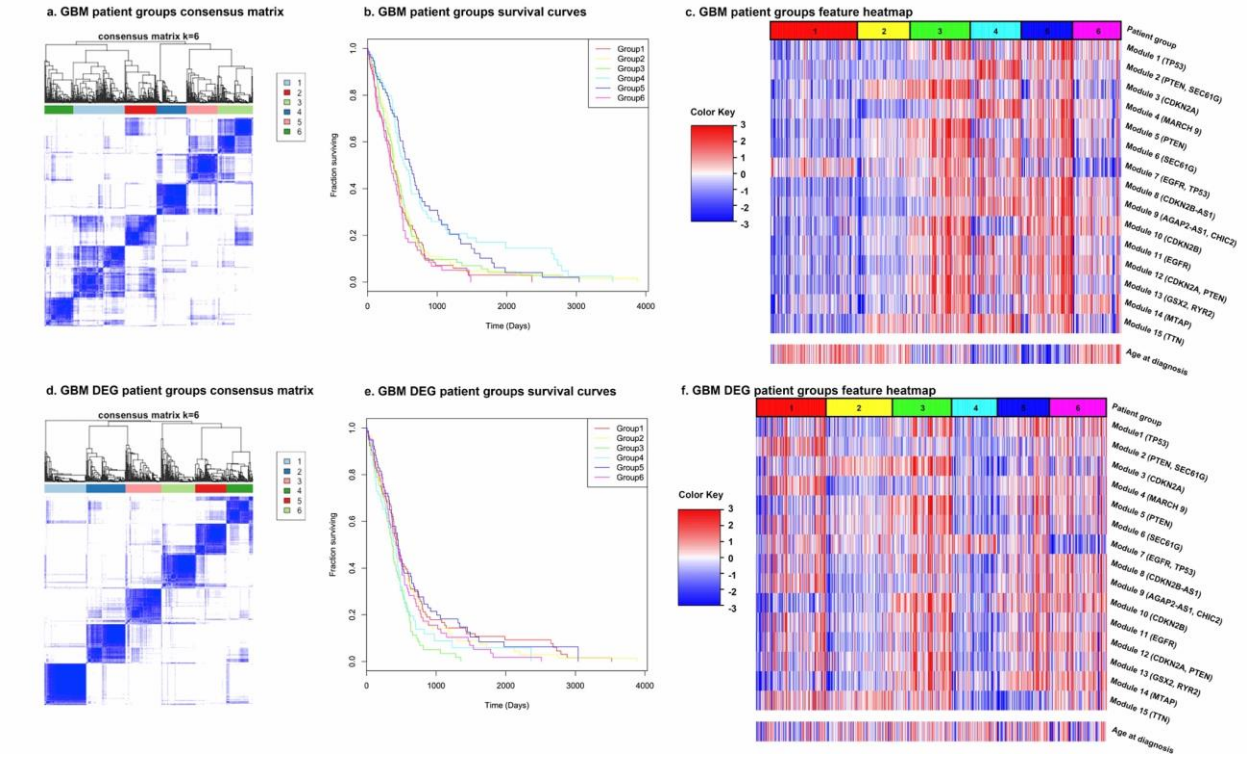


Figure 3.5. The consensus matrices of PAM consensus clustering for identifying patient groups for GBM, the survival curves of the resulting patient groups, and the feature heatmaps.

Patient groups were identified using all DEG modules and clinical features. DEG patient groups were identified using only DEG modules. For the heatmaps, the features were normalized across all patients. Values above 3 and below -3 are compressed into 3 and -3, respectively. The dominant SGAs of each DEG module are listed by the module index. The values of the clinical features of each DEG patient group are also given as a reference.

Figure 3.4 and Figure 3.5 also display the correlation between the features used in the PAM consensus clustering and the resulting patient groups as heatmaps. For BRCA (Figure 3.4c), groups 1 and 2 have all clinical features alike and benign, which resulted in their significantly better survival outcomes compared to the other groups. The difference between their survival curves (Figure 3.4b) is explained by their distinct patterns in DEG modules, with group 1 having significantly higher values than group 2. Group 3, the patient group with the second-worst survival outcome (Figure 3.4b), is a typical triple-negative group, with all three gene markers, estrogen

receptors (ER), progesterone receptors (PR), and human epidermal growth factor receptor-2 (Her2) as negative. Group 4, with similarly lower DEG module values as group 2, distinguishes itself from group 2 with mainly PR negative patients and its high values in DEG module 2 (dominant SGAs *PTEN* and *PIK3CA*); its grade of disease is also higher, which resulted in its relatively lower survival chance. Group 5, having the worst survival outcome, contains most patients as Her2+. In summary, the survival of BRCA subgroups is strongly related to their clinical features such as age and protein-based biomarkers (ER, PR, and Her2). Given the similar clinical features, the pattern in DEG modules determines the survival difference.

For GBM (Figure 3.5c), groups 1 and 2 both contain older patients, which is associated with poor survival outcomes. Except that group 1 has a specifically high value in module 7 (dominant SGAs *EGFR*, *TP53*) compared to group 2. Groups 3 and 4 distinguish themselves with their different distributions of DEG module values, especially in their reversed pattern in DEG modules 1-5. Although they both contain younger patients, their different values in DEG modules suggests that they have different combinations of signaling pathways being defective, which resulted in a much higher survival fraction of group 4 than group 3 (Figure 3.5b). Group 5 contains most of the youngest patients, giving it the second-best survival outcome. Group 6, having the lowest average value in module 7, contains mostly older patients, making it indistinguishable from groups 1, 2, and 3 from a survival aspect. It can be seen that the age at diagnosis is the strongest indicator of GBM prognostic, which agrees with previous studies (Lamborn, Chang, & Prados, 2004; Le Mercier et al., 2012; Piccolo & Frey, 2013; Roldan-Valadez et al., 2016; Wangaryattawanich et al., 2015; Ya Zhang, Li, Peng, & Wang, 2016). Given the similar patient ages, the pattern of DEG modules explains the differences between survival outcomes.

We next compared BRCA patient groups discovered by our approach with the PAM50 subtypes (C. G. A. Network, 2012) to see if these two patient classification standards align with each other (Figure 3.6 and Table 3.3). Each one of the five patient groups has a single dominant PAM50 subtype (overlapping proportion > 50%). Groups 1 and 2 are mainly composed of luminal A patients (Figure 3.6a). Specifically, luminal A and luminal B together make up over 90% of group 2. Group 4 is enriched in luminal B patients, followed by luminal A (Figure 3.6a). Thus, groups 1, 2, and 4 together re-divide the PAM50 luminal A and luminal B subtypes into three groups (Figure 3.6b). The discovery of multiple subtypes in luminal/ER+ groups has been reported in previous studies (Curtis et al., 2012; C. G. A. Network, 2012), which supports that a refinement of the luminal subtypes is necessary. In addition, we also found that most ILC (invasive lobular carcinoma) patients and IDC (invasive ductal carcinoma) + ILC patients were clustered in patient groups 1, 2 and 4 (55.8%, 17.0% and 16.3%, respectively for ILC; 42.2%, 27.8%, and 17.8% for IDC+ILC). This agrees with previous studies that ILC patients are mostly ER+ tumors classified as luminal A subtype (Ciriello et al., 2015). Group 3, the triple-negative group, is dominated by basal-like patients (Figure 3.6a), as basal-like tumors typically have negative ER, PR, and Her2 (C. G. A. Network, 2012). Group 5, the Her2+ group, is enriched in Her2 patients as expected (Figure 3.6a). It is known that BRCA survival differs by subtype, and shortest survival is generally observed among Her2+ and basal-like subtypes (Carey et al., 2006); this agrees with our observations of patient groups 3 and 5 on the Kaplan-Meier plot (Figure 3.4b). No patient group is mainly composed of normal-like patients. The p-value of survival difference between the PAM50 subtypes is $< 2x10^{-16}$. Therefore, both the PAM50 subtypes and our BRCA patient groups can efficiently divide BRCA patients into significantly different survival groups.

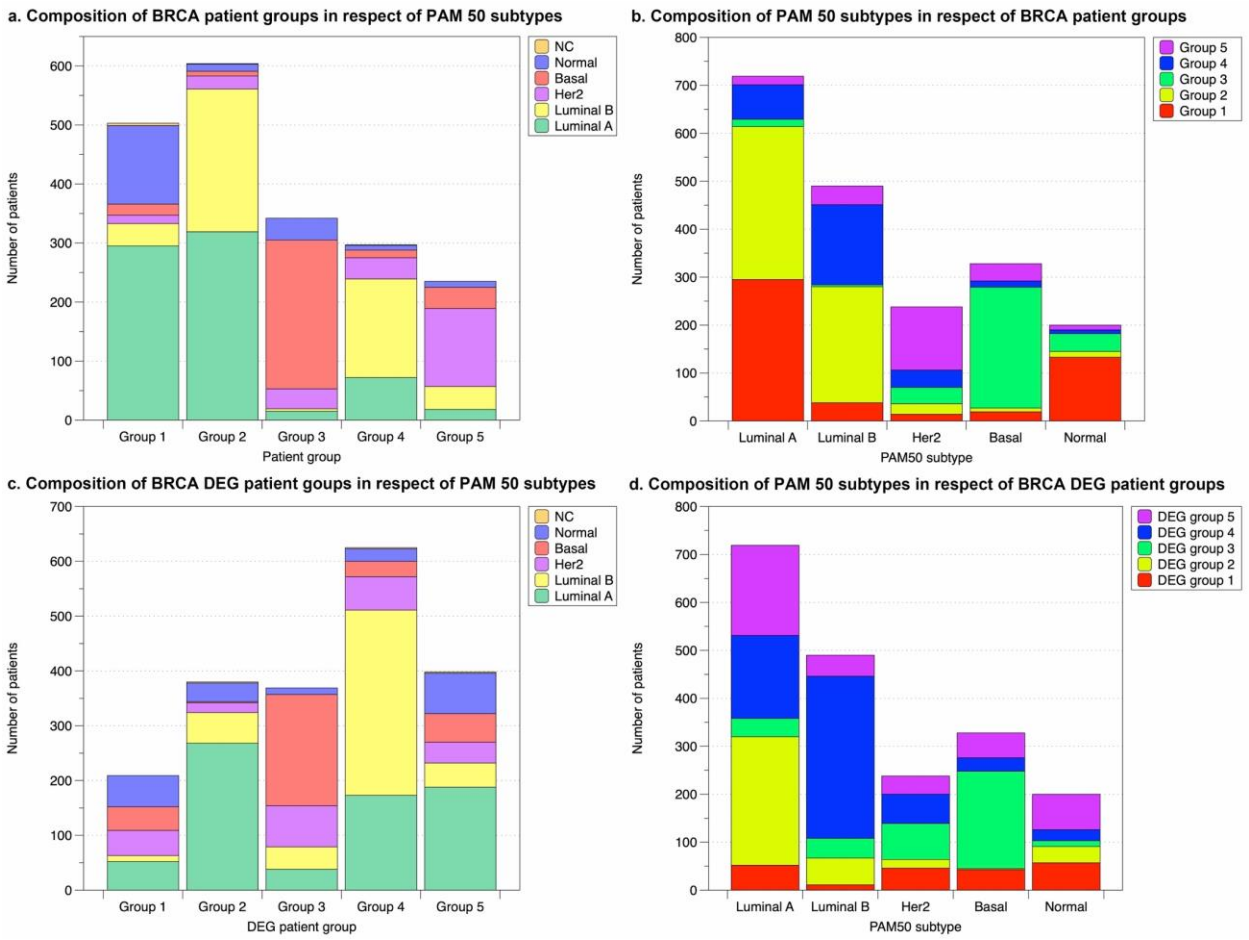


Figure 3.6. The comparison between BRCA patient groups and DEG patient groups with the PAM50 subtypes.

(a) and (c) show the composition of patient groups/DEG patient groups in respect of PAM50 subtypes. (b) and (d) show the composition of PAM50 subtypes in respect of patient/DEG patient groups.

Table 3.3. The overlap between BRCA patient groups and PAM50 classification.

Group index	# of Patients	Luminal A	Luminal B	Her2	Basal	Normal	NC
Patient group 1	503	**58.60%**	7.60%	2.80%	3.80%	26.40%	0.80%
Patient group 2	604	**52.80%**	40.10%	3.60%	1.30%	2.00%	0.20%
Patient group 3	342	4.40%	1.20%	9.90%	**73.70%**	10.80%	0.00%
Patient group 4	297	24.20%	**56.20%**	12.10%	4.40%	2.70%	0.30%
Patient group 5	235	7.70%	16.60%	**56.20%**	15.30%	4.30%	0.00%
DEG patient group 1	209	24.9%	5.3%	22.0%	20.6%	27.3%	0.0%
DEG patient group 2	380	**70.5%**	14.7%	4.7%	0.5%	8.9%	0.5%
DEG patient group 3	369	10.3%	11.1%	20.3%	**55.0%**	3.3%	0.0%
DEG patient group 4	625	27.7%	**54.1%**	9.8%	4.5%	3.7%	0.3%
DEG patient group 5	398	47.2%	11.1%	9.5%	13.1%	18.6%	0.5%

We compared our GBM patient groups with the four GBM subtypes established by TCGA, 2010 (Verhaak et al., 2010) (Figure 3.7and Table 3.4). Group 1 is mainly composed of Classical patients (Figure 3.7a). Recall that group 1 has positive values in DEG module 7 (Figure 3.5c), where the most dominant SGA is *EGFR*. *EGFR* was found to be highly amplified in the classical subtype, which supports the correlation between this subtype and patient group 1 (Verhaak et al., 2010). Groups 2 and 3 are both enriched in mesenchymal patients (Figure 3.7a). These two groups consist of patients with different age ranges and DEG module distributions (Figure 3.5c), which suggests intrinsic subgroups exist in mesenchymal patients. Group 5 is mainly composed of proneural patients, and nearly half of the patients in group 6 are also proneural (Figure 3.7a). No patient group we identified is strongly enriched in neural patients. The neural subtype has been considered as normal tissue contamination, thus it is not an intrinsic subtype of GBM (Q. Wang et al., 2017). The p-value of the log-rank test of GBM TCGA subtypes is 0.06, significantly higher than that achieved by our GBM patient groups (8.96×10^{-6}), which indicates that our GBM patient groups are more survival indicative compared to the TCGA subtypes.

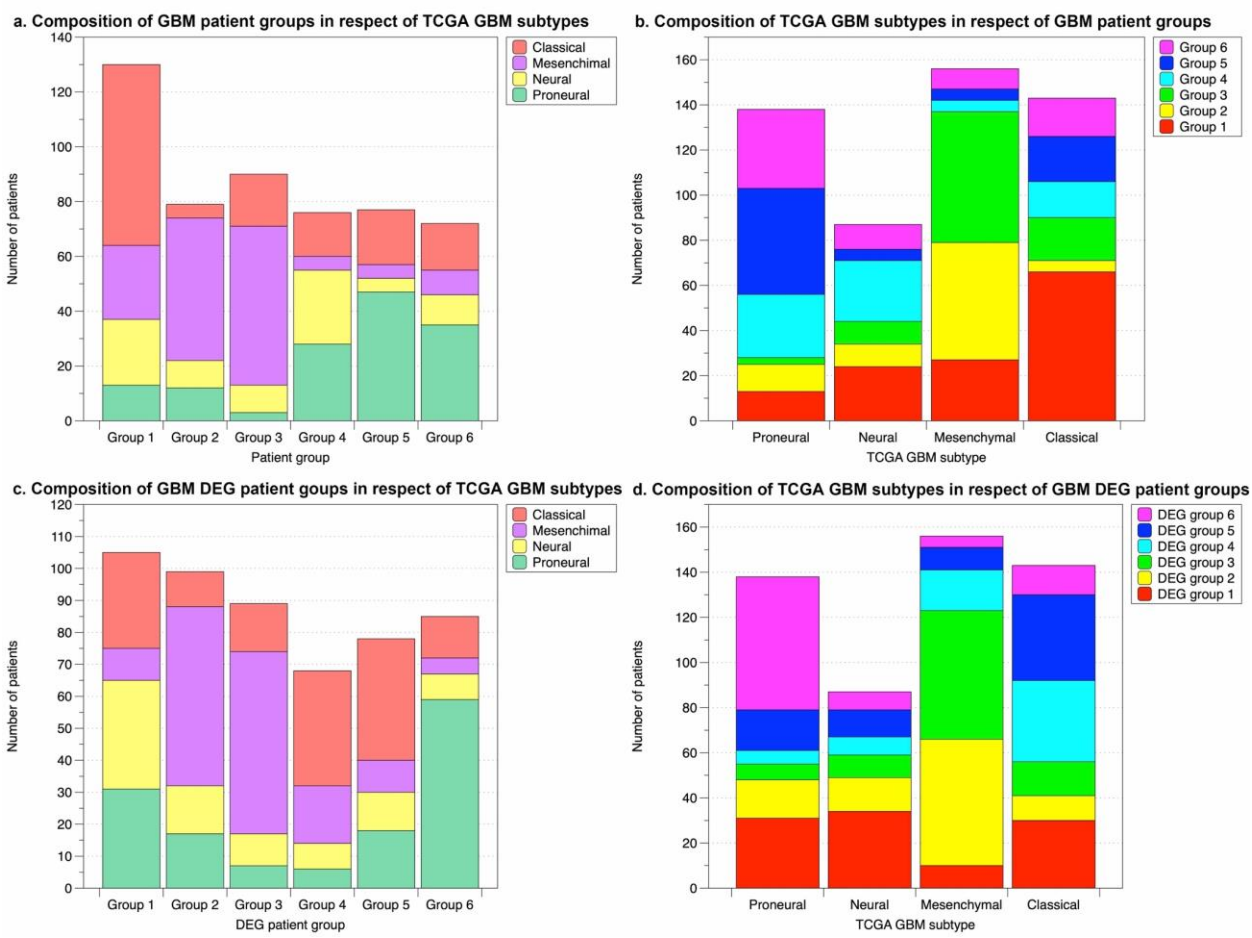


Figure 3.7. The comparison between GBM patient groups and DEG patient groups with the TCGA GBM subtypes.

(a) and (c) show the composition of patient groups/DEG patient groups in respect of TCGA GBM subtypes. (b) and (d) show the composition of TCGA GBM subtypes in respect of patient/DEG patient groups.

Table 3.4. The overlap between GBM patient groups and GBM TCGA subtypes.

Group index	# of Patients	Proneural	Neural	Mesenchymal	Classical
Patient group 1	130	10.00%	18.50%	20.80%	**50.80%**
Patient group 2	79	15.20%	12.70%	**65.80%**	6.30%
Patient group 3	90	3.30%	11.10%	**64.40%**	21.10%
Patient group 4	76	36.80%	35.50%	6.60%	21.10%
Patient group 5	77	**61.00%**	6.50%	6.50%	26.00%
Patient group 6	72	48.60%	15.30%	12.50%	23.60%
DEG patient group 1	105	29.5%	32.4%	9.5%	28.6%
DEG patient group 2	99	17.2%	15.2%	**56.6%**	11.0%
DEG patient group 3	89	7.9%	11.2%	**64.0%**	16.9%
DEG patient group 4	68	8.8%	11.8%	26.5%	**52.9%**
DEG patient group 5	78	12.8%	15.4%	23.1%	48.7%
DEG patient group 6	85	**69.4%**	9.4%	5.9%	15.3%

To examine the power of genetic features alone in predicting patient survival outcome, a second PAM consensus clustering of patients was completed using only the DEG modules as features. This also gave rise to a division of BRCA data into 5 patient groups, and a division of GBM data into 6 patient groups (Figure 3.4d and Figure 3.5d. Supplementary Table S3.7 and Table S3.8). For simplicity, from now on we will refer to these patient groups as the DEG patient groups. Although the survival curves of these DEG patient groups are relatively similar to each other and regress to the average survival, they are still significantly different (log-rank test p-value 8.60e-12 and 9.75e-03 for BRCA and GBM, respectively, Figure 3.4e and Figure 3.5e). The correlations between all features and DEG patient groups are less obvious (Figure 3.4f and Figure 3.5f), but two BRCA groups (1 and 3) preserve the patterns as having most patients as ER- and PR-, even though ER and PR status were excluded from DEG patient group identification. DEG patient group 3, the most comparable group to the original triple-negative group (patient group 3), is also the group that has the worst survival curve (Figure 3.4e). For GBM, DEG patient group 1, having a similar distribution in DEG modules, especially in DEG modules 1-5, as the original patient group

4, is also the one that has the best overall survival time (Figure 3.5f). Comparisons of the DEG patient groups with known subtypes (PAM50 for BRCA and TCGA subtypes for GBM) were also carried out and shown in Figure 3.6, Figure 3.7. Table 3.3, and Table 3.4.

Even though the DEG patient groups were obtained without including any clinical feature that was involved in defining the subtypes, the correlation between DEG patient groups and subtypes still exists. For example, BRCA DEG patient groups 2, 3, and 4 have a single dominant PAM50 subtype, where group 3 is enriched in Basal subtype patients as expected (Figure 3.6 c and d). GBM DEG patient groups 2, 3, 4, and 6 have a single dominant TCGA subtype, where the mesenchymal subtype is again divided into two subgroups (Figure 3.7 c and d). All these suggest that DEG modules alone can identify patient subgroups of distinct genetic aberration patterns with significantly different survival outcomes.

3.3.4 Cox Regression models

In order to evaluate the contribution of each feature towards survival estimation, we trained a Cox regression model using all features as covariates for all patients as a whole and for each patient group separately (Table 3.5). To compare clinical features and DEG modules, we also trained a Cox regression model using only clinical features and only DEG modules for all patients and for each DEG patient group (Table 3.6).

For BRCA, the all-patients model that received the highest concordance index (C-index) is the model trained using all covariates. Its C-index, 0.724, is higher than previously reported Cox regression models trained using only clinical and subtype information (0.67) (Parker et al., 2009). For the patient-group-specific models, each patient group has a different combination of clinical features as significant (Wald-test p-value <0.05). The DEG modules that are generally significant

across all-patient and DEG patient groups are modules 1, 2 and 5. Modules 1 and 2 are positively correlated with the hazard rate, and module 5 is negatively correlated with the hazard rate. These partially explain the survival curves we observed above. With high value in module 2, patient group 4 has a much lower survival fraction compared to patient group 2, even though their other DEG modules are comparable. The lower average value in module 2 also resulted in a better survival outcome of DEG patient groups 2, 5 and 1. Note that the dominant SGAs of module 2 are *PTEN* and *PIK3CA*; a high value in this module represents activation of the PI3K/Akt signalling pathway that is known to be related to ILC (Ciriello et al., 2015).

Table 3.5. The Cox regression models trained for BRCA and GBM for all patients and for each specific patient group, with different combinations of covariates.

Cox regression model	BRCA Significant covariates (coefficient)	C-index	GBM Significant covariates (coefficient)	C-index
All patients-all covariates	ER status (-0.118) Her2 status (0.122) age at diagnosis (0.196) tumour histology type (-0.226) lymph node assessment (0.264) size of tumour (0.169) module 2 (0.201) module 5 (-0.205)	0.724	age at diagnosis (0.486) module 1 (-0.496) module 4 (0.374) module 11 (0.738)	0.665
Patient group 1-all covariates	tumour histology type (-0.504) lymph node assessment (0.687) module 2 (0.450)	0.665	age at diagnosis (0.475) module 7 (-0.479) module 11 (1.816)	0.684
Patient group 2-all covariates	Her2 status (0.335) age at diagnosis (0.473) lymph node assessment (0.179) size of tumour (0.455) module 1 (0.408) module 2 (0.270) module 5 (-0.497)	0.701	age at diagnosis (0.496) module 2 (0.689) module 12 (1.400)	0.688
Patient group 3-all covariates	tumour histology type (-0.605) lymph node assessment (0.354)	0.680	age at diagnosis (0.620)	0.624
Patient group 4-all covariates	ER status (-0.508) PR status (-0.354) age at diagnosis (0.311) lymph node assessment (0.368) size of tumour (0.118) module 2 (0.378)	0.717	module 4 (1.079) module 8 (1.796) module 9 (-1.066)	0.707
Patient group 5-all covariates	lymph node assessment (0.226) size of tumour (0.248)	0.680	age at diagnosis (0.384) module 1 (-1.840) module 2 (-1.144) module 3 (1.862) module 5 (-1.609) module 6 (-1.711) module 11 (2.743) module 12 (1.343)	0.759
Patient group 6-all covariates	NA		age at diagnosis (0.789) module 1 (-1.196) module 5 (1.355) module 11 (1.666)	0.720

Table 3.6. The Cox regression models trained for BRCA and GBM with different combinations of covariates.

	BRCA		GBM	
Cox regression model	Significant covariates (coefficient)	C-index	Significant covariates (coefficient)	C-index
All patients-clinical features	ER status (-0.141) Her2 status (0.124) PR status (-0.110) age at diagnosis (0.193) grade of disease (0.122) tumor histology type (-0.230) lymph node assessment (0.261) size of tumor (0.185)	0.711	age at diagnosis (0.465)	0.646
All patients-DEG modules	module 1 (0.276) module 2 (0.316) module 5 (-0.421)	0.635	module 1 (-0.381) module 4 (0.339) module 9 (-0.401) module 11 (0.566) module 13 (-0.302)	0.593
DEG patient group 1-all covariates	lymph node assessment (0.420) size of tumor (0.244)	0.788	age at diagnosis (0.460) module 8 (1.440) module 9 (-0.755)	0.706
DEG patient group 2-all covariates	PR status (-0.413) age at diagnosis (0.626) lymph node assessment (0.465) size of tumor (0.170)	0.765	age at diagnosis (0.722) module 11 (1.767) module 12 (0.197)	0.705
DEG patient group 3-all covariates	tumor histology type (-0.654) lymph node assessment (0.319) module 1 (0.409)	0.683	module 15 (-0.608)	0.608
DEG patient group 4-all covariates	PR status (-0.148) age at diagnosis (0.206) lymph node assessment (0.181) size of tumor (0.133)	0.663	age at diagnosis (0.885) module 4 (-2.084) module 8 (2.983) module 11 (2.336)	0.736

Table 3.6 continued

Cox regression model	BRCA Significant covariates (coefficient)	C-index	GBM Significant covariates (coefficient)	C-index
DEG patient group 5-all covariates	ER status (-0.494) age at diagnosis (0.281) lymph node assessment (0.634) size of tumor (0.204) module 2 (0.392)	0.766	age at diagnosis (0.851) module 2 (-1.127) module 4 (1.429) module 5 (1.164)	0.712
DEG patient group 6-all covariates	NA		age at diagnosis (0.453) module 1 (-0.972) module 11 (2.038)	0.715
DEG patient group 1-clinical features	Her2 status (0.245) lymph node assessment (0.414) size of tumor (0.252)	0.762	None	0.652
DEG patient group 2-clinical features	PR status (-0.356) age at diagnosis (0.693) lymph node assessment (0.430)	0.756	age at diagnosis (0.583)	0.655
DEG patient group 3-clinical features	tumor histology type (-0.662) lymph node assessment (0.310) size of tumor (0.103)	0.688	None	0.574
DEG patient group 4-clinical features	Her2 status (0.134) PR status (-0.148) age at diagnosis (0.186) lymph node assessment (0.166) size of tumor (0.140)	0.649	age at diagnosis (0.405)	0.635
DEG patient group 5-clinical features	ER status (-0.418) age at diagnosis (0.259) lymph node assessment (0.567) size of tumor (0.271)	0.749	age at diagnosis (0.559)	0.676
DEG patient group 6-clinical features	NA		age at diagnosis (0.410)	0.648

Table 3.6 continued

	BRCA		GBM	
Cox regression model	Significant covariates (coefficient)	C-index	Significant covariates (coefficient)	C-index
DEG patient group 1-DEG modules	module 4 (-0.989) module 5 (-0.890)	0.660	module 8 (1.276) module 9 (-0.804)	0.652
DEG patient group 2-DEG modules	None	0.617	module 11 (1.390)	0.651
DEG patient group 3-DEG modules	module 1 (0.481)	0.579	module 15 (-0.575)	0.601
DEG patient group 4-DEG modules	module 2 (0.206) module 5(-0.336)	0.596	module 9 (-1.454) module 11 (1.954)	0.693
DEG patient group 5-DEG modules	module 2 (0.588)	0.640	module 5 (1.001)	0.626
DEG patient group 6-DEG modules	NA		module 11 (2.150)	0.700

Unlike BRCA, where clinical features dominate survival estimation, most GBM Cox regression models contain several DEG modules as significant covariates. The most common significant DEG module across patient groups and DEG patient groups is module 11, with its dominant SGA *EGFR*. *EGFR* has been used as the primary marker in distinguishing between GBM patients and it was found to interact with multiple signaling pathways in GBM (Mischel et al., 2003). In addition to module 11, the set of significant DEG modules are mostly mutually exclusive across patient groups. In other words, even though GBM patients generally share similarly undesirable survival outcomes, their survival rates can be explained by different combinations of genetic features. This suggests that each of them took a different disease mechanism in their tumor developments. For example, module 7, the smallest DEG module with dominant SGA *EGFR* and *TP53*, has a high diversity across patients. This module represents the result of the communications between the Glioma pathways (KEGG map05214), which are known to explain the disease mechanism for both primary and secondary GBM (Mao, LeBrun, Yang, Zhu, & Li, 2012). In addition, the C-index of GBM Cox regression models is higher in the patient-group-specific model

than in the overall model, which also supports the idea that different patient groups underwent different disease development procedures that should not be mixed. Three patient groups, 4, 5 and 6 (together containing 225 patients), have a C-index over 0.7, which is higher than a previously reported Cox regression model trained on a subset of TCGA GBM patients using clinical and imaging features (0.69) (Mazurowski, Desjardins, & Malof, 2013).

3.4 Discussion

In this preliminary work, we designed and evaluated a graph-based computational framework, which utilizes the causal inferences between SGAs and DEGs for constructing expression and signaling state representations, in the form of DEG modules. The DEG modules reflect the major transcriptomic programs that are perturbed in a cancer type and are informative towards clinical outcome predictions. Indeed, we have shown that different combinations of DEG modules divided BRCA and GBM patients into subgroups that exhibit significantly different survival patterns. Since the identification of DEG modules was driven by estimates of causal relationships between SGA and DEG events, our approach provides underlying mechanistic information for each cancer subgroup, and such information can potentially be used to guide future targeted therapy in a pathway-oriented fashion. This differentiates our method from previous approaches of using gene expression data to discover cancer subtypes.

We used the spectral clustering algorithm to identify DEG modules from the DEG networks. The major advantage of using spectral clustering algorithm compared to other clustering methods is that spectral clustering identifies modules with high data connectivity but not necessarily with high data compactness (Braun, Leibon, Pauls, & Rockmore, 2011; Von Luxburg,

2007). In our case, since the DEG networks were constructed based on co-regulations between DEGs, we emphasized more on identifying modules that connect sequences of DEGs (high connectivity) rather than modules with a high direct correlation between any pair of DEGs (high compactness). Sequences of DEGs may each represent a cascade of aberrant signaling resulting from upstream perturbed genes. Two DEGs that are indirectly connected through a subsequence of other genes are very likely controlled by the same regulatory network, thus are functional related. In addition, our DEG networks are relatively dense (766,444 edges for BRCA, 1,567,144 edges for GBM), where classical hierarchical clustering or k-means would fail to identify robust modules (Figure 3.3 visualizes the consensus matrices of hierarchical clustering). Spectral clustering, on the other hand, can still find stable and consistent modules across different independent random initializations as we have shown above.

For general clustering or communication detection algorithms, features with the highest diversity across data are often given a higher priority to be used to cut between observations, which maximizes both the distance between observations of different resulting clusters and similarity between observations in the same cluster. For gene expression data, tissue-specific genes are often more diverse across samples than other globally expressed genes. Consequently, using solely gene expression data or genetic signatures like PAM50 for discovering cancer subtypes often leads to a division of subtypes based on cell-of-origin. The approach we adopted to identify patient groups with a combination of clinical features and DEG modules, however, does not suffer from this problem. For example, none of the BRCA patient groups or DEG patient groups is overwhelmingly dominated by a single PAM50 subtype that is related to a cell type. The division of ILC and IDC+ILC in patient groups 1, 2, and 4 also supports that our patient groups are not simply tissue-specific divisions. Nonetheless, our patient grouping results are not simply random. Each patient

group presents a distinct pattern of DEG modules, where the modules reflect the status of the signaling pathway perturbations that drives tumorigenesis. The patient groups also present distinct survival outcomes. For GBM, in particular, our patient groups are even more survival correlated compared to the TCGA subtypes. All these suggest that our approach is robust to tissue-specific-expressions and can identify subtypes that are disease mechanism and prognostic indicative.

In BRCA, clinical features seem to be more informative towards survival than DEG modules when doing patient grouping and Cox regression. One of the reasons is that certain clinical features are in fact molecular features, including the ER, Her2, and PR status, which are not independent of the DEG modules. For example, the Her2 expression status is correlated with the expression status of the DEG module driven by dominant SGAs *ERBB2* and *TP53*. As a result, the corresponding DEG modules became less significant in the Cox regression due to the redundant information they provide. The decrease in C-index when DEG modules were excluded (Table 3.6), and the irreplaceable role that DEG modules play in GBM survival estimation, however, support that these DEG module features preserved independent pathway-oriented information that clinical features did not capture.

DEG modules were discovered by constructing a DEG graph from the SGA-DEG causal inferences and simply represented as the average expression value of all genes in the module. In order to learn more comprehensive and interpretable cellular state representations from genomic data, in the next chapter, we switch to more advanced models for representation learning, the deep generative models.

4.0 Chapter 2: learning to encode cellular responses to systematic perturbations with deep generative models

4.1 Introduction

A cellular signaling system is a network-based signal processing machine that detects changes in the internal or external environment, encodes these changes as cellular signals, and eventually transmits these signals to effectors to adjusts cellular responses. Such cellular responses often lead to changes in transcriptomic programs (Azeloglu & Iyengar, 2015; Radhakrishnan, Halász, Vlachos, & Edwards, 2010; Weng, Bhalla, & Iyengar, 1999). The transcriptomic changes can be utilized in turn to investigate the cellular signaling system, which is an important task in system biology. A common approach is to systematically perturb a cellular system with genetic or pharmacological perturbagens and monitor transcriptomic changes in order to reverse engineer the system and gain insights into how cellular signals are encoded and transmitted. This approach has been employed in many large-scale systems biology studies, e.g., the yeast deletion library (Giaever & Nislow, 2014), the Connectivity Map project (Lamb, 2007; Lamb et al., 2006), and most recently, the Library of Integrated Network-based Cellular Signatures (LINCS) (Keenan et al., 2018; Subramanian et al., 2017).

Among these system studies, the LINCS project is arguably the most comprehensive systematic perturbation dataset currently available, in which multiple cell lines were treated with over tens of thousands of perturbagens (e.g., small molecules or single gene knockdowns), followed by monitoring gene expression profiles using a new technology known as the L1000 assay (Subramanian et al., 2017). Previous studies involving LINCS mainly used the L1000 data

to investigate the mechanism-of-action (MOA) of drugs and to promote clinical translation of MOA information (Donner, Kazmierczak, & Fortney, 2018; Iwata et al., 2017; Lamb et al., 2006; Pabon et al., 2018; Siavelis et al., 2015; Subramanian et al., 2017; Z. Wang et al., 2016; Woo et al., 2019). Few studies, however, utilized the data to model the cellular signaling system as a network-based information processor.

Modeling the signaling system as an information processing network would enable examining in detail the paths that different perturbagens take to regulate or disturb cellular activities. To see this, when signaling components at different levels of a signaling cascade are perturbed, the resulting expression data would present compositional statistical structures that can be hard to reverse-engineer. For instance, perturbing an upstream signaling molecule will likely subsume the effect of perturbing its downstream molecules. Capturing such a compositional statistical structure requires advanced graphical models that are capable of representing hierarchical relationships among signaling components (Figure 4.1), which were rarely explored in previous studies.

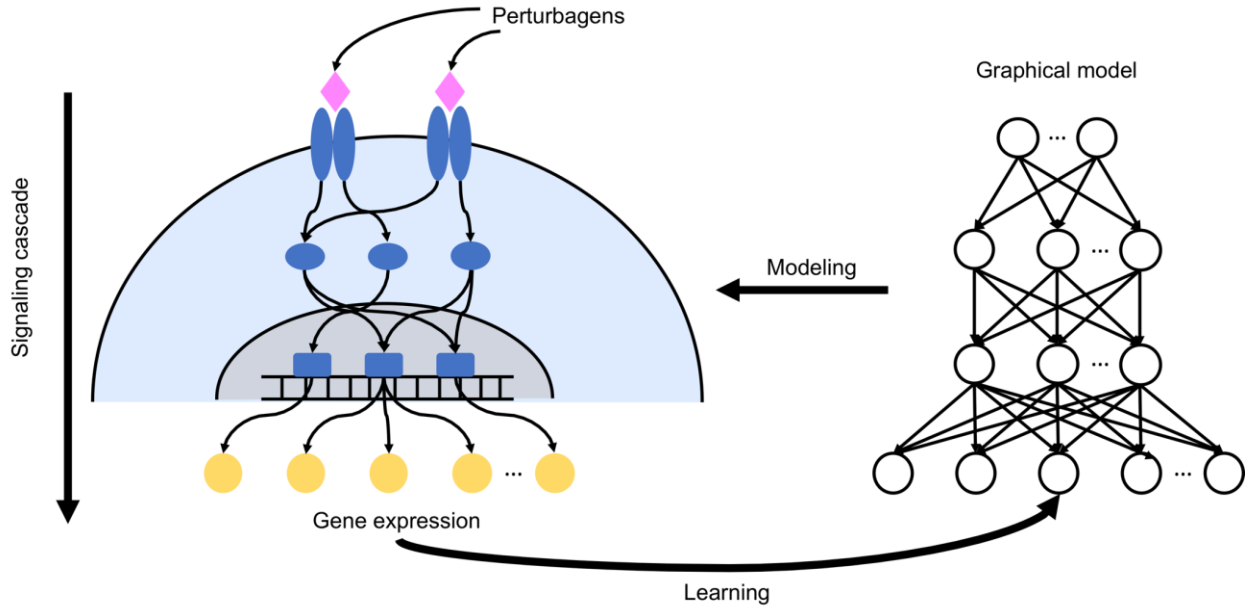


Figure 4.1. Modeling cellular signaling system with graphical model.

In this chapter, we present the deep generative models (DGMs) we developed to understand how perturbagens affect the cellular signaling system and lead to changes in the gene expression profile.

DGMs are a family of deep learning models that employ a set of hierarchically organized latent variables to learn the joint distribution of a set of observed variables. After training, DGMs are capable of generating simulated data that preserve the same compositional statistical structure as the training data. The hierarchical organization of latent variables in DGMs is naturally suitable for representing cellular signaling cascades and detecting compositional statistical patterns derived from perturbing different components of the cellular system. The capability to "generate" samples similar to the training data is also of particular interest in that if a model can accurately regenerate transcriptomic data produced under different perturbations, the model should have learned a comprehensive representation of the cellular signaling system. Such representations would shed light on the MOAs through which perturbagens impact different cellular processes.

In this chapter, we investigate the utility of two DGMs, the variational autoencoder (VAE) (Figure 4.2a) (Hinton & Salakhutdinov, 2006; Kingma & Welling, 2014; Rezende et al., 2014) and a new model designed by us, the supervised vector-quantized variational autoencoder (S-VQ-VAE) (Figure 4.2b), in modeling the cellular signaling system using the L1000 data. We show that the VAEs can reconstruct the distribution of the L1000 data accurately and generate new data that are indistinguishable from the real data. We demonstrate that by adding a supervised learning component to vector-quantized VAE (VQ-VAE) (Aaron Van Den Oord & Vinyals, 2017), we can summarize the common features of a family of drugs into a single embedding vector and use these vectors to reveal relationships between different families of drugs. Latent representations learned by VAEs for samples perturbed by different types of perturbagen can also enhance the drug-gene target prediction compared to using conventional supervised models. To our knowledge, this is the first study that systematically investigates the power of DGMs for learning how cellular signals are processed in response to perturbations, and our findings support the use of deep generative models as a powerful tool in modeling cell signaling systems.

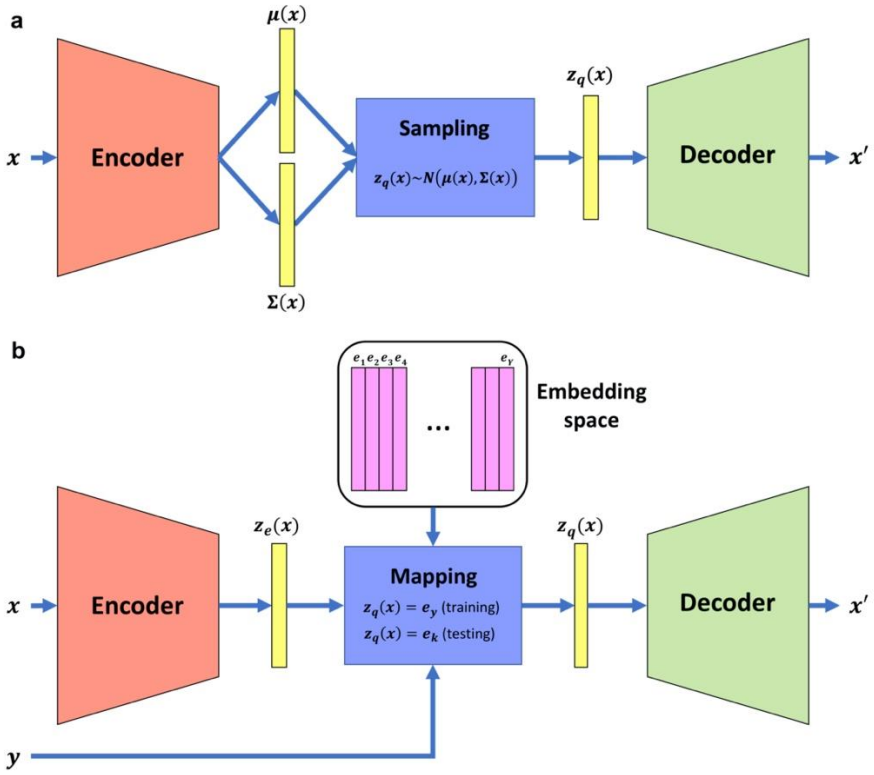


Figure 4.2. The VAE model and S-VQ-VAE model.

(a) The architecture of VAE. The encoder and decoder are two sub-neural networks. An input case is transformed into a mean vector $\boldsymbol{\mu(x)}$ and a covariance vector $\boldsymbol{\Sigma(x)}$ by the encoder, from which the encoding vector $\boldsymbol{z_q(x)}$ is sampled and fed to the decoder to reconstruct the input case. The distribution of the encoding vector is trained to follow a prior standard normal distribution. (b) The architecture of S-VA-VAE. S-VQ-VAE is an extension of VQ-VAE where the training of the embedding space is guided by the label of the input data. Similar to VAE, an input case is first transformed into an encoding vector $\boldsymbol{z_e(x)}$ by the encoder. During training, the encoding vector is replaced by the embedding vector $\boldsymbol{e_y}$ designated to represent the label *y* of data to reconstruct the input case. The embedding vector is updated according to the reconstruction error. During testing, the encoding vector is replaced by the nearest neighbor embedding vector $\boldsymbol{e_k}$.

4.2 Methods

4.2.1 Data

We constructed several datasets for training our DGMs, consisting of different combinations of samples treated with two major types of perturbagens, the small molecule perturbagen (SMP) and the genetic perturbagen (GP). The SMP dataset was extracted from the Gene Expression Omnibus (GEO) dataset GSE70138 (Subramanian, 2015), which contains the level 5 L1000 expression data (moderate z-scores) of the 978 landmark genes of 85,183 samples from seven major cell lines treated with small molecules (Table 4.1). The GP dataset was from the GEO dataset GSE106127 (Subramanian, 2017), which contains the level 5 data of 116,782 samples from nine major cell lines with gene knockdowns (Table 4.1). A cell line is considered as a major cell line if the cell line had over 10,000 samples.

We performed principal component analysis (PCA) on the two datasets, and the distributions of samples in the first two principal components are shown in Figure 4.3. By comparing the scatter plot of the SMP dataset with its density contour (Figure 4.3), we can see that the group of samples on the right of the scatter plot is an outlier group with a high variance but low density. This group contains 4,649 samples treated with two proteasome inhibitors, bortezomib, and MG-132. Therefore, in the third dataset, the SMGP dataset that merges the SMP dataset with the GP dataset, these outlier samples are excluded. The removal of outliers results in comparable distributions between SMP samples and GP samples (Figure 4.3), which enables the use of the SMGP dataset for training a VAE model to reveal connections between small molecules and knocked down genes.

Table 4.1. LINCS datasets and major cell lines.

GEO ID	Dataset content	# of sample	# of perturbagen	# of drug/gene name	# of perturbagen class (PCL)	# of cell line	Major cell line name	Major cell line type	# of sample in cell line
GSE70138	LINCS phase II L1000 dataset, mainly small molecular perturbation	118050	2170	1826	991 perturbagens in 171 classes	41	MCF7	breast adenocarcinoma	13476
							A375	malignant melanoma	12740
							PC3	prostate adenocarcinoma	12719
							HT29	colorectal carcinoma	12529
							HA1E	normal kidney	12481
							YAPC	pancreatic carcinoma	10621
							HELA	large intestine adenocarcinoma	10617
GSE106127	LINCS L1000 RNAi and CRISPR dataset. Corresponds to genetic perturbational signatures of shRNAs and CRISPR reagents, that exist in GEO series GSE70138 and GSE92742.	119013	18413	4320	NA	15	VCAP	prostate carcinoma	17098
							A375	malignant melanoma	13121
							PC3	prostate adenocarcinoma	13061
							HA1E	normal kidney	12957
							A549	non-small cell lung cancer	12691
							HT29	colorectal carcinoma	12305
							HCC515	lung cancer	11985
							MCF7	breast adenocarcinoma	11869
							HEPG2	hepatocellular carcinoma	11695

*A major cell line is a cell line that has over 10,000 samples.

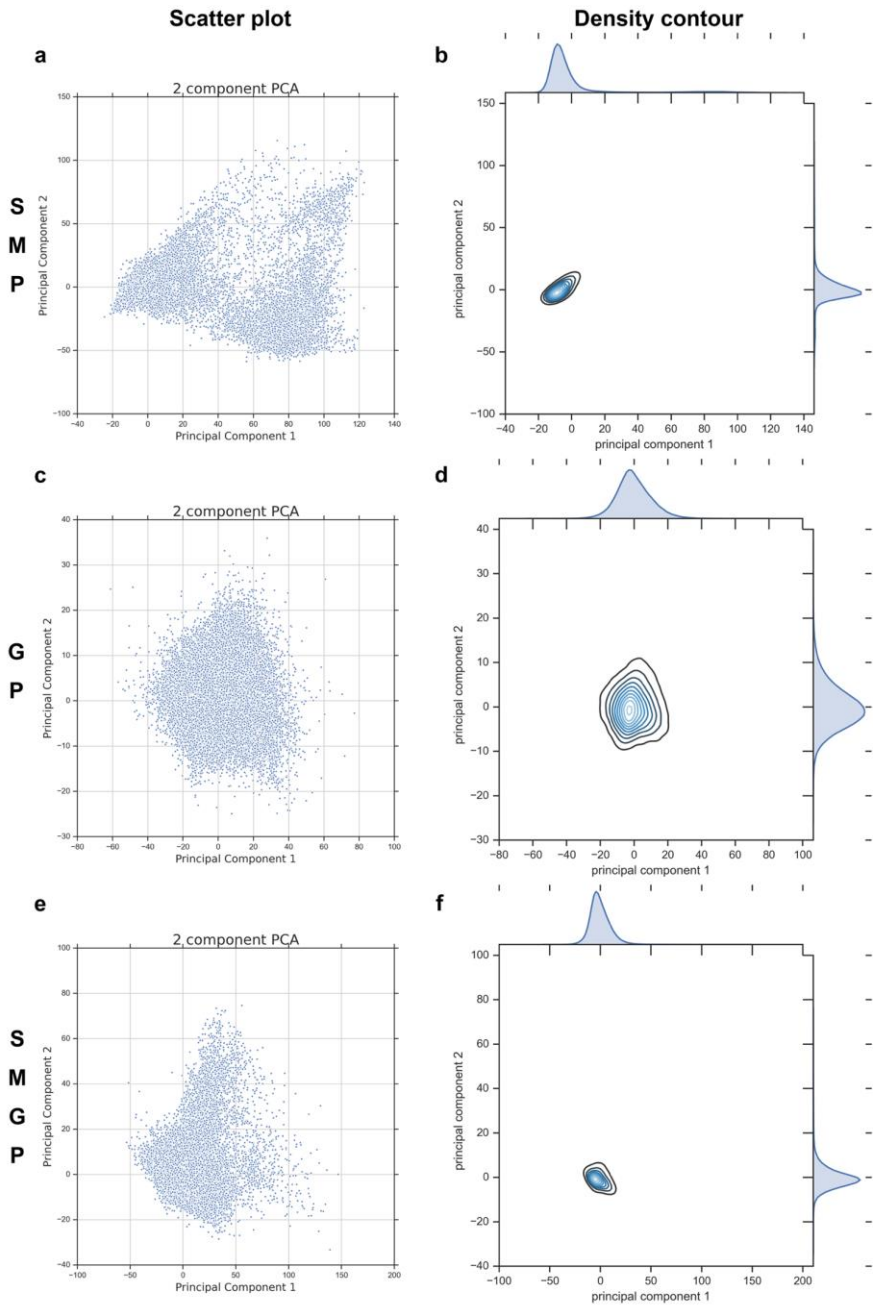


Figure 4.3. PCA two components scatter plots and density contour plots of three input datasets.

(a and b) Scatter plot and density contour of the SMP dataset. The outlier group on the right of the scatter plot is composed of 4,649 samples treated with bortezomib and MG-132. Both these SMPs are proteasome inhibitors. (c and d) The scatter plot and density contour of the GP dataset. (e and f) The scatter plot and density contour of a combination of the SMP and GP datasets (the SMGP dataset), excluding the outlier group of proteasome inhibitors.

We extracted a subset from the SMP dataset, which contains 12,079 samples treated with 204 small molecules from 75 perturbagen classes (PCLs) defined by the LINCS project (Subramanian et al., 2017). We call this subset the SMP with Class information (SMC) dataset. The PCL information was extracted from the Supplementary Table S7 of the L1000 paper (Subramanian et al., 2017). The SMC dataset was used to train Logistic Regression (LR) and Support Vector Machine (SVM) models for predicting PCLs of samples based on cellular representations learned from VAEs.

We also extracted a subset from the SMC dataset to learn PCL representations with S-VQ-VAE. We call this dataset as SMCNP dataset, where we excluded the samples treated with the proteasome inhibitor MG-132 (bortezomib was not given a PCL label, and thus had been excluded from the SMC dataset). This subset contained 9,769 samples treated with small molecules from 75 PCLs.

4.2.2 S-VQ-VAE Model

S-VQ-VAE is a new DGM designed in this study for learning a vector representation (embedding) for each PCL. The model was extended from the standard VQ-VAE (Aaron Van Den Oord & Vinyals, 2017) by adding a supervised mapping step to guide the training of the embedding space. Like VQ-VAE, a S-VQ-VAE is composed of three parts, an encoder neural network to generate the encoding vector $z_e(x)$ given an input vector x, an embedding space to look up the discrete representation $z_q(x)$ based on $z_e(x)$, and a decoder neural network to reconstruct the input data from $z_q(x)$ (Figure 4.2b). Suppose that the encoder encodes the input data to a vector of length D, the embedding space E is then defined as $E \in R^{Y \times D}$, where Y is the number of different classes of the input data. In our case, Y corresponds to the number of PCLs. Each of the

Y embedding vectors of dimension D is designated to learn a global representation of one of the classes. In forward computation, an input x is first converted to its encoding vector $z_e(x)$, which will be used to update the embedding space. In the training phase, $z_e(x)$ is replaced with $z_q(x) = e_y$ to pass to the decoder, where e_y is the embedding vectors of the class y of x. In the testing phase, $z_e(x)$ is replaced by its nearest code $z_q(x) = e_k$ with

$$k = argmin_j \|z_e(x) - e_j\| \tag{4.1}$$

Note that we are not assuming a uniform distribution of the embedding vectors as in the ordinary VQ-VAE (Aaron Van Den Oord & Vinyals, 2017). Instead, the distribution of codes is determined by the input data with its discrete class labeling governing by a multinomial distribution.

In order to design a model that can learn individual representations through data reconstruction as well as learn a global representation for each class in a supervised manner, the objective function of S-VQ-VAE contains a reconstruction loss to optimize the encoder and decoder (first term in equation (4.2)), and a dictionary learning loss to update the embedding space (second term in equation (4.2)). The form of reconstruction loss can be selected based on the data type, and here we used the mean square error (MSE). Following the training protocol of standard VQ-VAE (Aaron Van Den Oord & Vinyals, 2017), we chose VQ as the dictionary learning algorithm, which computes the l_2 error between $z_e(x)$ and e_y thus updating the embedding vector towards the encoding vector of an input case of class y. To control the volume of the embedding space, we also added a commitment loss between $z_e(x)$ and e_y to force the individual encoding vector towards the corresponding global embedding vector (third term in equation (4.2)).

$$L = l_r\left(x, d(e_y)\right) + \|sg[z_e(x)] - e_y\|_2^2 + \beta\|z_e(x) - sg[e_y]\|_2^2 - I(k \neq y)(\|sg[z_e(x)] - e_k\|_2^2 + \gamma\|z_e(x) - sg[e_k]\|_2^2) \tag{4.2}$$

In addition to making the encoding vectors and the embedding vectors converge, we added two additional terms to force the encoding vector of an input data to deviate from the nearest embedding vector e_k if $k \neq y$ (i.e., to minimize misclassification with nearest neighbor). As given in equation (2), the fourth term is another VQ objective which updates the embedding vector of the mis-class. The final term, called the divergence loss, expands the volume of the embedding space to allow different classes to diverge from each other.

Coefficients are applied to the commitment loss (β) and divergence loss (γ) to control the strength of regularization over the embedding space volume. According to preliminary experiments using coefficients from [0, 1], the performance of the model is quite robust to these coefficients. For generating the results presented in this chapter, we used $\beta = 0.25$, and $\gamma = 0.1$. Note that the mapping step with either the class label or nearest neighbor has no gradient defined for it. As in training VQ-VAE, we approximate the gradient in a manner similar to the straight-through estimator (Bengio, Léonard, & Courville, 2013), by passing the gradient from the reconstruction loss from $z_q(x)$ directly to $z_e(x)$.

As a generative model, S-VQ-VAE can also be used to generate new data from the distribution of the training data. The data generation process is composed of two steps, similar to the ancestral sampling method. First, sample a target class y from the distribution of classes of the input data. Second, sample an encoding vector $z \sim N(e_y, \sigma^2)$, where σ^2 is the covariance matrix of hidden variables estimated from the training data of class y. A new sample of class y can then be generated by passing z to the decoder of S-VQ-VAE. The generation process reflects another advantage of S-VQ-VAE compared to unsupervised GMs: we can determine what type of content the new data should present rather than interpret it afterward.

In this chapter, we only utilized the global representation learning function of S-VQ-VAE. The test phase and the new data generation function of S-VQ-VAE were not examined here. To see how S-VQ-VAE can be used as a general generative model, please refer to our tutorial of S-VQ-VAE at https://github.com/evasnow1992/S-VQ-VAE, where we provide an example applying S-VQ-VAE on a benchmark machine learning dataset, the MNIST handwritten digits data (LeCun, Cortes, & Burges, 1998).

4.2.3 Model Architecture and Training Setting

The VAE model we implemented has three hidden layers in its encoder and three hidden layers in its decoder; the third hidden layer of the encoder is shared by both the encoder and decoder parts via a sampling step (Figure 4.4) and is also called the top hidden layer. The structure of the encoder is 978-1000-1000-100, where it has 978 nodes in the input layer, each corresponding to a landmark gene in the LINCS data, 1000 nodes in the first and second hidden layers, and 100 nodes in the third hidden layer (Figure 4.4). The structure of the decoder is just the reverse of the encoder. We only included the 978 landmark genes as input data to avoid redundant information from the inferred expression levels of other genes. The number of hidden layers and the number of nodes on each layer were determined based on preliminary experiments with a wide range of model architectures. Specifically, we tried architectures from 978-500-15 to 978-2000-1000-200 to select a model with as a simple structure as possible and with a low training error. Based on our previous experience, a three hidden layer model with 1000-1500 nodes on the first hidden layer, ~1000 nodes on the second hidden layer, and small bottleneck on the third hidden layer usually performs the best (Chen, Cai, Chen, & Lu, 2016; M. Q. Ding et al., 2018). The model achieved the best overall performance in this study has a structure of 978-1000-1000-100.

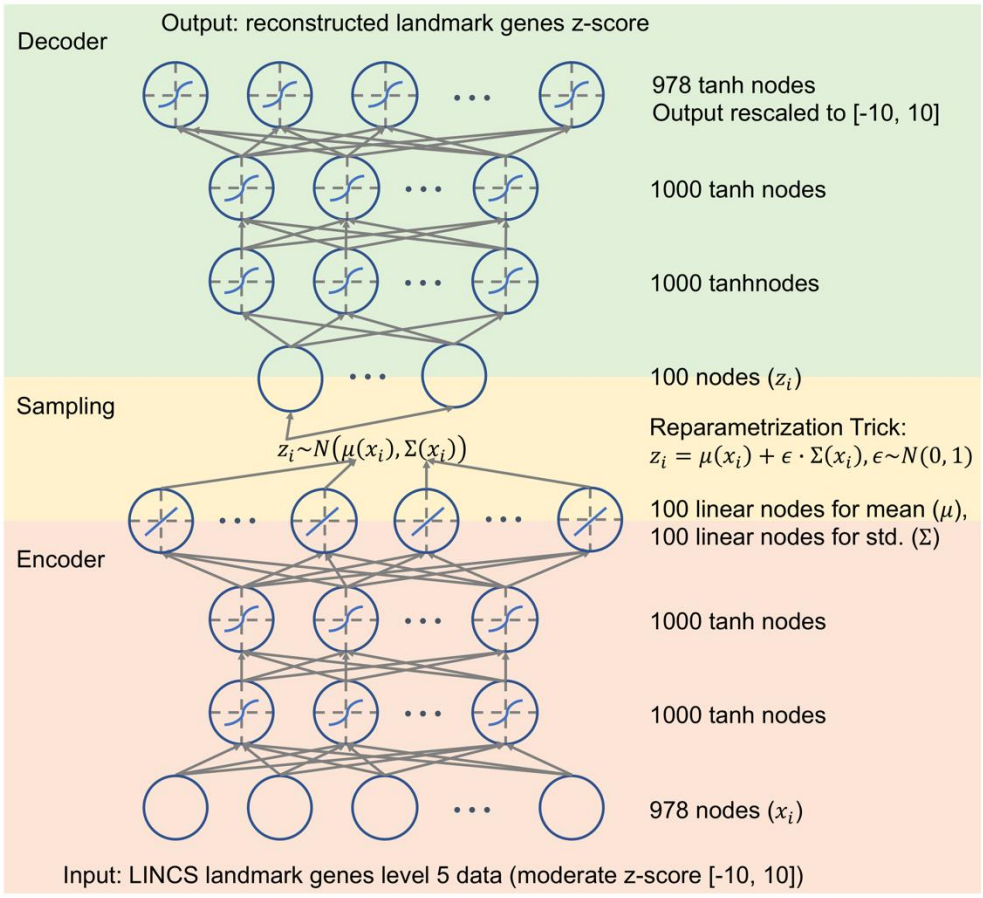


Figure 4.4. The architecture of VAE.

The encoder is composed of three hidden layers, where the third hidden layer defines the distribution from which the encoding vector z_i is sampled. The reparameterization trick is applied to generate a differentiable estimate of z_i, which allows gradient descent to be used for training the model. The decoder has a reversed architecture as the encoder.

We use a standard normal distribution, $N(0, 1)$, as the prior distribution of the top hidden layer variables $p(z)$ of VAE. The input data of our models are the L1000 level 5 gene expression data of range [-10, +10]. In order to preserve the sign information of the input data, where a positive value indicates high-expression of a gene and a negative value indicates low-expression of a gene,

we use the tangent function as the activation function for all hidden layers. Note that the tangent function will map a real number to [-1, +1], while our input data are of range [-10, +10]. To reconstruct the input data, the outputs of the last layer of the decoder are rescaled to [-10, +10] before computing the reconstruction loss (Figure 4.4).

The loss/target function for training a general VAE is

$$L = l_r(x, d(z_{e(x)})) + KL(q(z|x)||p(z)) \tag{4.3}$$

where the first term is the reconstruction loss and the second term is the KL-distance between the posterior distribution of the top hidden variables $q(z|x)$ given the input data and the prior variational distribution $p(z)$. We use the MSE as the reconstruction loss.

We trained three VAE models using the SMP, GP, and SMGP datasets independently. Each model was trained on 9/10 (random split) of the data and validated on the other 1/10 data. All models were trained for 300 epochs, with batch size 512 and learning rate 1e-3 (Table 4.2). To generate a new sample, we first sampled from the multi-variate $N(0, 1)$ distribution to get an encoding vector, then passed the vector through the decoder of the VAE to get a new data instance.

Table 4.2. The data reconstruction performance of VAE models and S-VQ-VAE model on training data and validation data.

Model type	Train data	Training loss	Validation loss
VAE	SMP	1.081	1.113
VAE	GP	0.861	0.864
VAE	SMGP	0.995	1.002
S-VQ-VAE	SMCNP	1.725	1.831

*The models were trained on 9/10 of the data and validated on the other 1/10 of the data. The reported losses are MSE between the reconstructed data and the input data.

The S-VQ-VAE model we implemented has a single hidden layer of 1,000 nodes in its encoder. The decoder has the reversed architecture of the encoder. As in VAE, we also use the

tangent activation function for S-VQ-VAE and rescale the data from [-1, 1] to [-10, 10] before computing the reconstruction loss. The number of hidden layers and hidden nodes were selected based on preliminary experiments with architectures from one to two hidden layers and 200 to 1500 hidden nodes in each layer. The embedding space contains 75 codes, one for each PCL. The model was trained on 9/10 (random split) of the SMCNP dataset for 900 epochs, with batch size 256, and learning rate 1e-4. The model was validated on the other 1/10 data (Table 4.2).

4.2.4 Mixing Score of Binary-categorical Data

To quantize the mixing level of the two types of data (real expression profiles vs. generated expression profiles in our case), we defined a mixing score for a *k*-clustering result of binary-categorical data as follows. Suppose the total number of data to be clustered is N. For a cluster i, the number of data in this cluster of one category is denoted as p_i, and the number of data of the other category is denoted as q_i. Then the mixing score for a *k*-clustering result is defined as

$$MS_k = \frac{\sum_{i=1}^{k} \max(p_i, q_i)}{N} \tag{4.4}$$

This score equals the average proportions of data from the category that dominates each cluster. The mixing score is of range [0.5, 1], where 0.5 indicates the two categories on average mixing evenly in the k clusters, and 1 indicates the two categories are cleanly separated among the k clusters. The mixing score, by definition, tends to increase with the number of clusters k used to stratify the data.

4.2.5 PCL Prediction

Seven different types of sample representations were evaluated as predictors for predicting the PCL label of the small molecule that treated each SMC sample via LR, random forest, naïve Bayes classifier, and SVM. We only report the results of LR and SVM below as these two models consistently outperformed the others, where LR achieved the best validation performance while SVM was significantly more tolerant of overfitting. The seven representations included the raw expression profile, the latent representations from three encoder layers of the SMGP-trained VAE, the 12 signature nodes values, and the latent representations from two decoder layers of the SMGP-trained VAE (the top hidden layer of the encoder is shared with the decoder, thus there are only two independent decoder layers). The latent representation of a layer of a sample was obtained by feeding the expression profile of the sample to a trained VAE and extracting the values of hidden nodes on the desired layer.

The prediction accuracy reported in this study was obtained by doing 10-fold cross-validation across SMC data. Specifically, the SMC data were randomly split into 10 subsets. In each iteration, an independent model was trained on 9 of the subsets and validated on the 10^{th} subset. The reported accuracy is the average validation accuracies over the 10 models.

4.2.6 Drug-Target Identification

We extracted drug-target relationships from the ChEMBL database (Gaulton et al., 2011) referring to Table 1 of Pabon et al. (Pabon et al., 2018), which included 16 drugs tested in all the seven major cell lines in the SMP dataset. Here we considered different LINCS drug IDs with the same drug name as the same perturbagen.

For predicting the gene targets for each drug, we first extracted samples treated with the drug from the SMP dataset. Then for each sample, we computed the Pearson correlations between the representation of the sample and the corresponding representations of all 116,782 samples from the GP dataset. The genes knocked down in the GP samples were ranked according to the Pearson correlations, and the rank of the top known target gene was recorded. Finally, the best (lowest) top rank and mean top rank across all samples treated with the same drug were computed and used to compare different types of representations. Similar to the PCL classification task, seven types of sample representations were compared based on the top rank and mean rank.

4.2.7 Program Language, Packages, and Softwares

VAE and S-VQ-VAE models were implemented in Python2.7 using the library *PyTorch* 0.4.1 (Paszke et al., 2017). Adam optimizer was used for updating the models. PCA analysis functions, LR models, and SVM models were from the Python library *Scikit-learn* 0.21.3 (Pedregosa et al., 2011). For LR, we used a random seed 0 for shuffling data and solver "lbfgs" (Limited-memory BFGS) for multi-classification. For SVM we used a random seed 0 and default settings for the other hyper-parameters. Distance computation functions, including Euclidean distance and Pearson correlation, were from the Python library *SciPy* 1.3.1 (Virtanen et al., 2019). We used the Euclidean distance for revealing general associations between expression profile representations and we used the Pearson correlation in drug-target prediction for emphasizing the orientation consistency between representations. Hierarchical clustering and heatmap visualization were carried out with the Python library *Seaborn* 0.9.0 (Waskom, 2018). The code for preprocessing LINCS data, training VAE and S-VQ-VAE models, and carrying out model analyses are available at https://github.com/evasnow1992/DeepGenerativeModelLINCS. S-VQ-

VAE PCL representation graph visualization and community detection were accomplished with software *Gephi* 0.9.2 (Bastian, Heymann, & Jacomy, 2009). The community detection algorithm being used was the Louvain algorithm developed by Blondel et al. (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) and was run with randomization (for better decomposition), using edge weights, and resolution 1 (for detecting smaller communities).

4.3 Results

4.3.1 Modeling Cellular Transcriptomic Processes with VAE

We carried out a series of model comparison experiments and selected the architecture based on model complexity, reconstruction error, and other aspects of performance (see Methods). In the architecture we selected to generate the results shown here, the input and output layers contained 978 nodes, each corresponding to one of the 978 landmark genes in the L1000 expression profile. The internal architecture was composed of three hidden layers in its encoder and three hidden layers in its decoder. The encoder and decoder share the top hidden layer, thus in total of five hidden layers. The encoder part contained 1000, 1000, and 100 hidden nodes, respectively (Figure 4.4); the decoder had a reverse architecture as the encoder.

We trained three VAE models on the SMP, GP, and SMGP datasets, respectively (Table 4.1, Table 4.2). We first examined whether the models captured the distribution of the input data by generating new data and comparing their distribution with that of the original input data. For each of the three models, we randomly generated 10,000 samples and projected them with 10,000 randomly selected original training samples into the first two components PCA space (Figure 4.5).

From the scatter plots in Figure 4.5 (a, d and g), we can see that the VAE-generated data points take up a similar space in the PCA plot as the input data for all three datasets. The consistencies in the centroid, shape, and range of the density contour indicate that the VAE models are able to reconstruct the distribution of the input data (Figure 4.5).

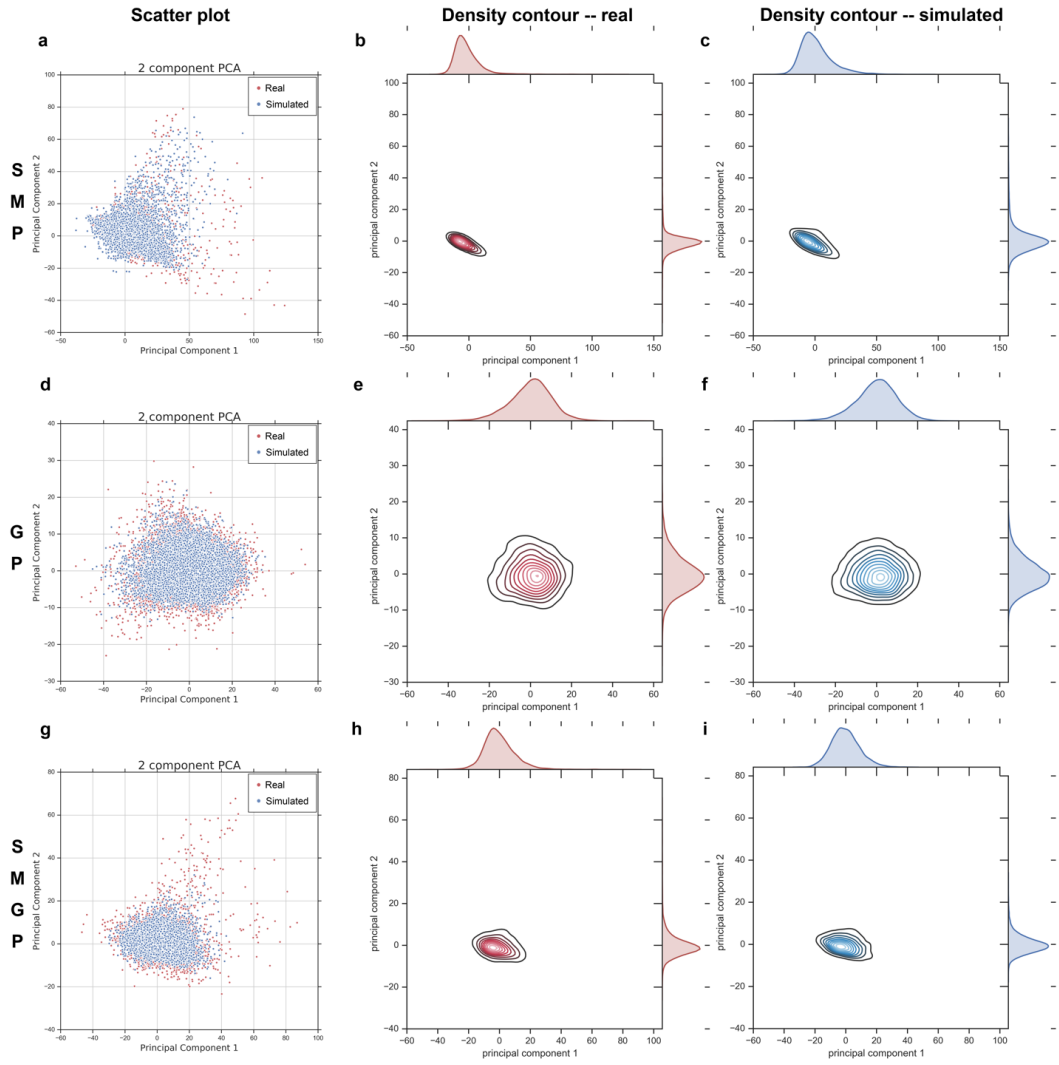


Figure 4.5. Simulated data of VAE vs. original input data.

(a) Scatter plot of simulated data (blue points) generated by SMP-trained VAE and the original SMP data (red points) in the space of the first two PCA components. (b) The density contour of the real data in (a). (c) The density contour of the simulated data in (a). (d) Scatter plot of simulated data (blue points) generated by GP-trained VAE and the original GP data (red points) in the space of the first two PCA components. (e) The density contour of the real data in (d). (f) The density contour of the simulated data in (d). (g) Scatter plot of simulated data (blue points) generated by SMGP-trained VAE and the original SMGP data (red points) in the space of the first two PCA components. (h) The density contour of the real data in (g). (i) The density contour of the simulated data in (g).

We then performed hierarchical clustering analyses to examine whether the newly generated data are indistinguishable from real data. Using 2,000 randomly generated samples and 2,000 randomly selected original samples, we conducted hierarchical clustering with *1-Pearson correlation* as the distance metric (Figure 4.6). We cut the dendrogram at 10 clusters and computed a mixing score (see Methods) to examine whether the generated data and original data were similarly distributed across clusters. For binary-categorical data, a mixing score is of range [0.5, 1], which gives the average proportion of data from the dominant category in each cluster. A mixing score of 0.5 indicates an even mixture of the two categories of data within all clusters, and a score of 1 indicates a clear separation between the two categories across clusters. For each of the three VAE models, this process of sample generation, selection, and mixing score computation was repeated 50 times. The mean mixing score was 0.594 for SMP-trained VAE with a 95% confidence interval (CI) of [0.589, 0.598], 0.586 for GP-trained VAE with a 95% CI of [0.581, 0.591], and 0.603 for SMGP-trained VAE with a 95% CI of [0.598, 0.608]. These mixing scores indicate that neither real data nor simulated data exhibit dominance in individual hierarchical clusters. Therefore, the generated data cannot be distinguished from the real data via hierarchical clustering.

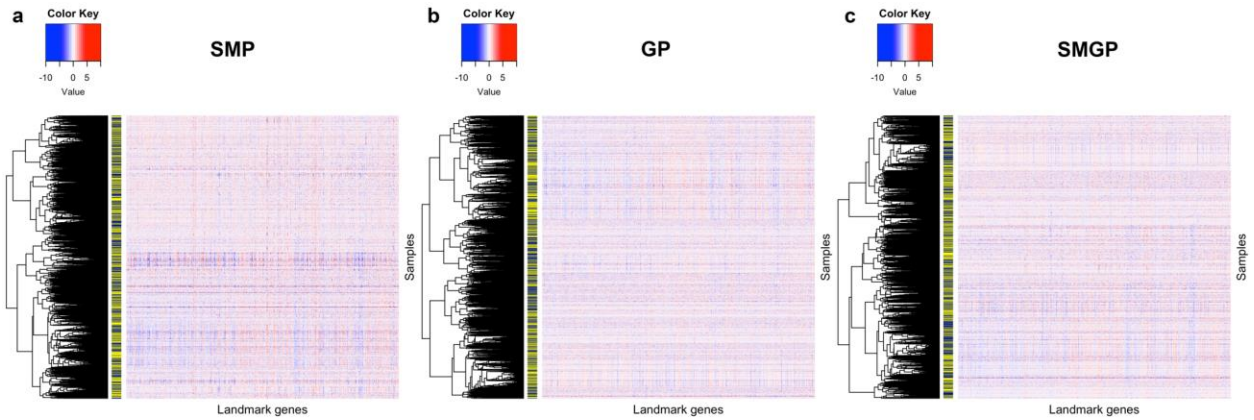


Figure 4.6. Hierarchical clustering of simulated data generated by trained VAEs vs. real data from the corresponding training datasets.

(a-c) The plot of 2,000 data generated with SMP-trained VAE, GP-trained VAE, and SMGP-trained VAE, respectively, and 2,000 real data from the corresponding training datasets. In the row color bar of the heatmap, blue indicates the original data and yellow indicates the simulated data. The evenly mixing of the original data and simulated data suggests that the simulated data are indistinguishable from the original data by doing a hierarchical clustering using *1-Pearson Correlation* as distance.

4.3.2 A Few Signature Nodes Encodes the Primary Characteristics of an Expression Profile

To gain a better understanding of how VAEs encode the distribution of diverse input data, we next examined the activation patterns of hidden nodes on different layers of the SMGP-trained VAE model. We paid special attention to the top hidden layer of 100 nodes that serves as an "information bottleneck" for compressing the original data. This layer is of particular interest as it is also used as the starting point for the generation of new samples.

For each sample from the SMC dataset, we computed an encoding vector by feeding the expression profile through the encoder of the SMGP-trained VAE to the top hidden layer ($z_q(x)$ in Figure 4.2a). We found that 12 out of 100 nodes in the encoding vector had a high variance in

activation values across samples (Figure 4.7a). The average values of these nodes show a clear bimodal distribution with one mode formed by the 12 nodes and one mode formed by the others. For the SMC dataset, the average absolute value of these 12 nodes is 0.885 vs. 0.048 of the other hidden nodes, (two-sided t-test p-value < 1e-10). Across the SMGP training dataset as a whole, the average absolute value of the 12 nodes is 0.503 and vs. 0.044 of the other nodes (p < 1e-10).

For an ordinary VAE model, the prior distribution of the encoding vector is a standard normal distribution with a mean vector $\mu(x) = 0$ and a diagonal covariance matrix $\Sigma(x) = diag(1)$ (Figure 4.2a). An element of the encoding vector should shrink towards 0 during training unless it is driven by data to deviate from 0. As a result, the significantly high absolute values taken by these 12 hidden nodes suggest that they encode major signals of input data. From now we denote these 12 hidden nodes as the signature nodes.

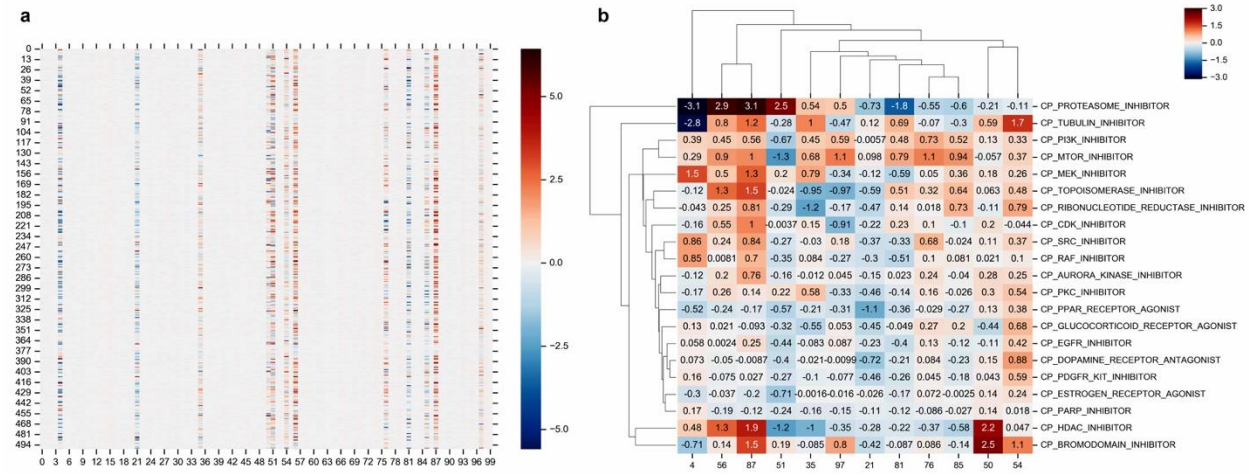


Figure 4.7. Signature nodes on the top hidden layer of SMGP-trained VAE.

(a) The heatmap of the 100 hidden nodes of the top hidden layer for 500 random selected SMGP samples. The pseudo-colors represent the values of elements in the encoding vectors. (b) The average of signature nodes for samples treated with major PCLs.

We investigated whether the patterns of these 12 signature nodes reflect the MOA of drugs by examining their association with PCLs. A PCL was considered a major perturbagen class if at least 150 samples were treated with perturbagens of that class. Using this definition, there are 21 major PCLs. For each major PCL, we fed the samples treated with perturbagens of the class through the trained VAE encoder and took the average signature node values across samples as a vector representation of the PCL. As shown in Figure 4.7b, different PCLs presented different patterns in the signature nodes.

We further examined whether the representations of each PCL revealed similarities between PCLs via hierarchical clustering analysis (Figure 4.7b). PCLs that were closely clustered tend to share similar MOAs (Figure 4.7b). For example, the mTOR inhibitor and PI3K inhibitor were grouped together according to their consistent activation directions (positive vs. negative) for most signature nodes, and they are both known to impact the PI3K/AKT signaling pathway (O'Reilly et al., 2006), where the mTOR is a downstream effector of PI3K. Other examples include the grouping of Src inhibitor and Raf inhibitor, where Src is known to activate Ras-c, which in turn activates Raf in the Raf-MEK-ERK pathway (Moon et al., 2002); the grouping of topoisomerase inhibitor and ribonucleotide reductase inhibitor that both impact the DNA replication process; and the grouping of Aurora kinase inhibitor and PKC inhibitor, where Aurora kinases are essential to mediate PKC-MAPK signal to NF-κB/AP-1 pathway (Noh et al., 2015). These observations support the idea that the 12 signature nodes preserve crucial information of the expression profile resulting from a small molecule perturbation of the cellular signaling system.

To further demonstrate that the primary characteristics of an expression profile are encoded in the 12 signature nodes, we generated new expression profiles to simulate samples treated with a target PCL by manipulating values of the signature nodes according to the PCL patterns found

in the previous experiment. Specifically, we preset the signature nodes to values similar to the average values of training samples treated with the target PCL as shown in Figure 4.7b and randomly initialized the other hidden nodes from a standard normal distribution. In this manner, we randomly generated 500 new samples for eight major PCLs. We then compared the randomly generated samples against real samples to see whether their nearest neighbors in real samples were from the target PCL (Figure 4.8a). The signature node patterns used to generate samples and the similarities of these samples to real samples and associated PCLs are shown in Figure 4.8b-i.

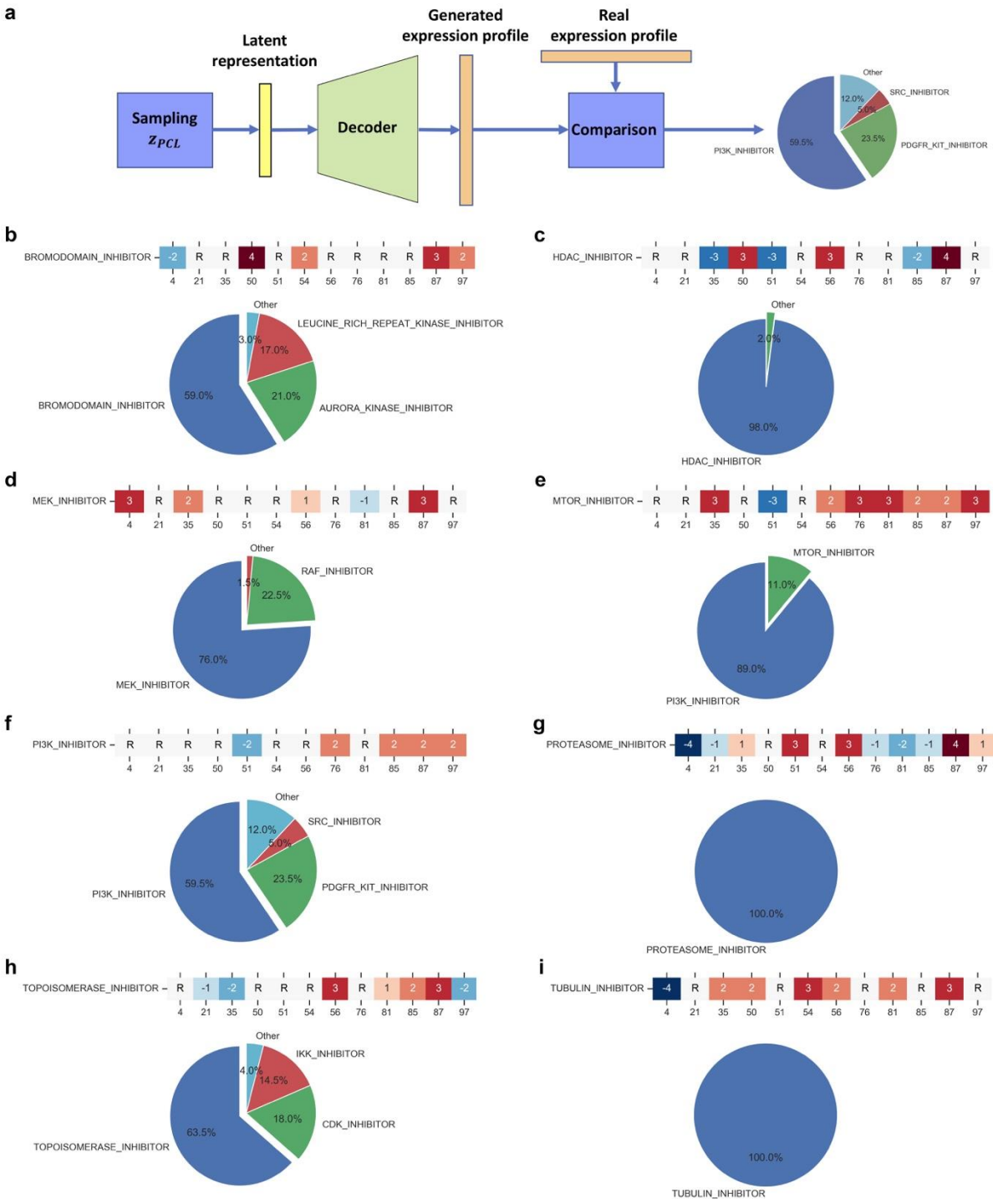


Figure 4.8. Comparison of data generated based on the signature pattern of PCLs with real data.

(a) Diagram illustrating the procedure for generating new data from the signature pattern of a PCL. First, an encoding vector is initialized where the signature nodes are set according to the signature pattern of real samples from the given PCL; the non-signature nodes are randomly initialized by sampling from a standard normal

distribution. The vector is then fed through the decoder of the SMGP-trained VAE, and a new expression profile is generated. The new data are compared to real data by computing the nearest neighbor based on Euclidean distance, to see if the new data are closely related to real data of the given PCL. (b-i) The composition of real data nearest neighbors of new data generated from latent representations simulating different PCLs. "R" indicates the value of the signature node is not specified but randomly initialized as non-signature nodes.

In most cases, more than half of the generated data had their nearest neighbors from the target PCLs. For proteasome inhibitor and tubulin inhibitor specifically (Figure 4.8g and i), 100% of the generated data were nearest neighbors of real samples from the same PCL, which was repeatedly observed across independent runs. This agrees with the PCL clustering outcome in Figure 4.7b, where proteasome inhibitor and tubulin inhibitor were found as outliers from the other PCLs with their distinct signature node patterns.

We also noted that the specific value of each signature node did not matter as long as the value correctly reflects the direction, i.e., positive or negative, of the node for a given PCL. This suggests that the major characteristics of a PCL can potentially be encoded into only 12 bits of information. The only pattern that did not have over half of the generated samples mapped to the target PCL was the mTOR inhibitor (Figure 4.8e). Most of the samples generated using mTOR signature nodes were nearest neighbors to PI3K inhibitor-treated samples. This is reasonable, as mTOR inhibitors act downstream on the same pathway of PI3K inhibitors. For this reason, the former's effects can be in many cases replicated by the latter. This observation also supports the conclusion that each signature node pattern reflects a specific cellular signaling process, which, after decoding, generates an expression profile that reflects how the signaling is perturbed.

4.3.3 Learning Global Representations of PCLs with S-VQ-VAE

The signature node representations of PCLs discussed above were obtained by averaging over samples treated with small-molecule perturbagens from a PCL. In order to learn a unique, stable global representation for each PCL, we designed another DGM, the S-VQ-VAE, which utilizes the PCL label of small molecules that treated the cells to partially supervise the training process. S-VQ-VAE was extended from VQ-VAE by utilizing the vector-quantized (VQ) technique to discretize the encoding vector space into multiple mutually exclusive subspaces represented by a limited number of embedding vectors and projecting data from each class into its pre-assigned subspace (Figure 4.2b, see Methods). After training, each embedding vector learns to summarize the global characteristics of a class of data. Here, we used S-VQ-VAE to learn an embedding vector with a dimension of 1000 for representing each of the 75 PCLs in the SMC dataset (Table 4.2).

We utilized the embedding vectors to reveal similarity and potential functional relationships between PCLs by comparing each PCL to all the others to identify its nearest neighbor based on the Pearson correlation. The nearest neighbor relationships between PCLs are visualized as a directed graph (Figure 4.9), in which a directed edge indicates that the source node is the nearest partner to the target node. We also applied the Louvain algorithm (Blondel et al., 2008) to detect the community (aka, clusters) among PCLs, and members of different communities are indicated as pseudo-colors (Figure 4.9). The modularity score of the communities is 0.875, which indicates significantly denser connections existing between members within communities compared to a randomly assigned network of the same set of PCLs.

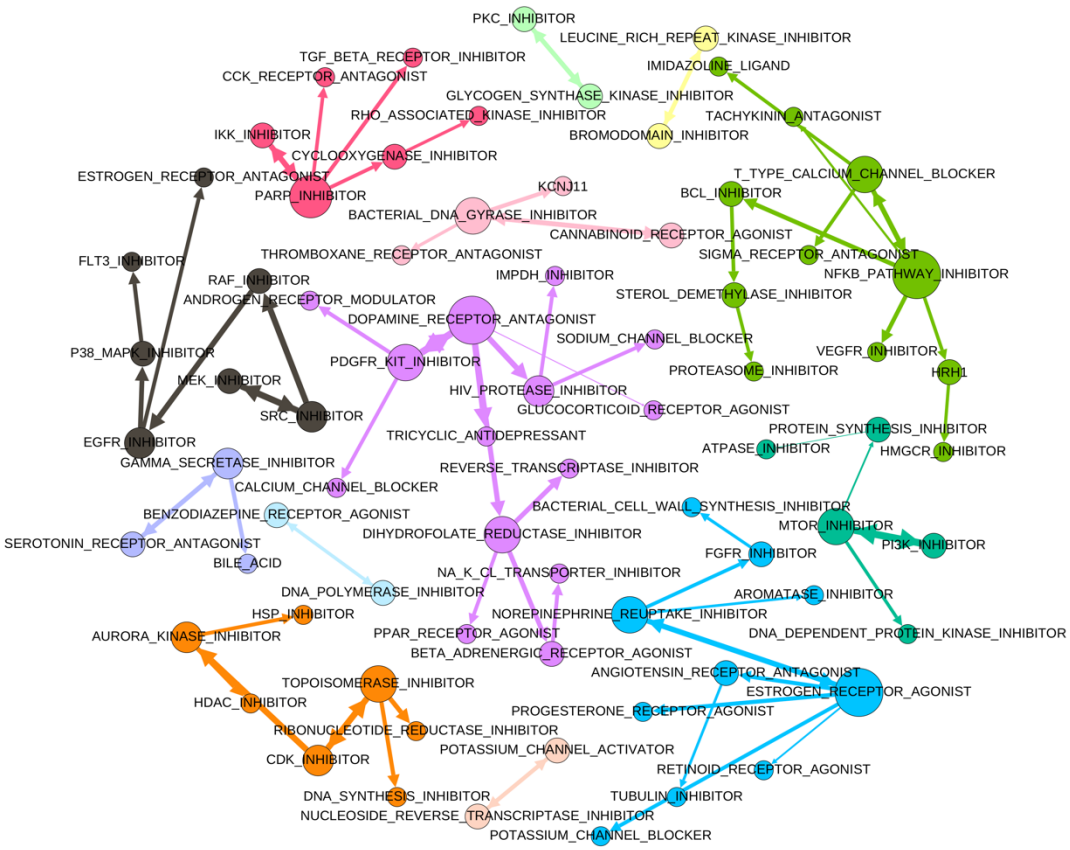


Figure 4.9. Similarities between PCLs revealed with global PCL representations learned by S-VQ-VAE. A directed edge in the graph indicates that the source node is the nearest node to the target code based on the Pearson correlation between the corresponding representations. The node size is proportional to the out-degree. The edge width is proportional to the correlation. The color of a node indicates the community the node belongs to.

Some strong relationships, like bi-directional connections and thick edges, are observed in Figure 4.9. Many such relationships correspond to well-documented shared MOAs between the drugs in the connected PCLs. These include the relationships that have also been revealed with the signature node representations above, e.g., the functional similarity between mTOR inhibitors and PI3K inhibitors (O'Reilly et al., 2006), and the relationship between MEK inhibitors, Src inhibitors, and Raf inhibitors (Moon et al., 2002). Other strong connections were observed between CDK inhibitors and topoisomerase inhibitors, which may reflect coordinated response to mitosis

inhibition and DNA damage induction (Peyressatre, Prével, Pellerano, & Morris, 2015; Weinberg, 2013), between Aurora kinase inhibitors and HDAC inhibitors which both impact the histone deacetylase pathway (Y. Li et al., 2006), and between gamma-secretase inhibitors, serotonin receptor antagonists, and bile acid that affect amyloid precursor protein processing and lipid metabolism (Pimenova, Thathiah, De Strooper, & Tesseur, 2014; Watanabe et al., 2010).

The members of a PCL community also shed light on the high-level functional theme of the community. For example, the black community on the left of Figure 4.9 with Raf, Src, MEK, and EGFR related PCLs may represent the drug effects transmitted through the EGFR-RAS-RAF-MEK signaling cascade. The orange community (bottom left of Figure 4.9), consisting of inhibitors of Aurora kinase, HDAC, CDK, topoisomerase, ribonucleotide reductase, and DNA synthesis, may represent the signaling transduction for regulating DNA duplication and mitosis. The blue community (bottom right of Figure 4.9), with estrogen, progesterone, norepinephrine, and angiotensin may represent the comprehensive effects of perturbing hormones. These findings indicate that the global representations learned with S-VQ-VAE preserve crucial information that reveals the functional impact of different PCLs.

4.3.4 The VAE Latent Representations Preserve PCL-Related Information

The latent variables at different levels of the hierarchy of a DGM may encode cellular signals with different degrees of complexity and abstraction (Chen et al., 2016). Therefore, we next investigated the information preserved in the latent variables of different hidden layers of the SMGP-trained VAE. To do this, we first represented the SMC samples with seven types of representations, including the raw expression profiles, the latent representations obtained from the five hidden layers of the VAE, and the 12 signature node values (see Methods). We then used these

representations to predict the PCL label of the small molecule used to treat each sample by training two multi-classification models, LR and SVM. As shown in Table 4.3, the highest test prediction accuracy was achieved by using the raw expression profiles as input data for both LR and SVM (0.5922 and 0.5273 respectively). This was followed by the latent representations of samples extracted from the first hidden layer of the VAE encoder (0.5096 for LR and 0.4528 for SVM). The lowest accuracy was obtained using the 12 signature node values as input data (0.3814 for LR and 0.3615 for SVM). Nonetheless, the highest test accuracy achieved with latent representation, 0.5096, was nearly 10 times higher than guessing at random from the 75 unevenly distributed PCLs, 0.0543. These results indicate that although there was information loss with respect to the classification task as the representations become more abstract with deeper hidden layers, the latent representations preserved significant information from the original input data.

Table 4.3. Performance of PCL classification with different sample representations as input data.

Representation type	LR test accuracy	SVM test accuracy
raw	0.5922	0.5273
encoder layer 1	0.5096	0.4528
encoder layer 2	0.4461	0.3881
encoder layer 3	0.4098	0.4232
signature nodes	0.3814	0.3615
decoder layer 1	0.4002	0.4082
decoder layer 2	0.3994	0.4085

*The accuracies were obtained with 10-fold cross validation. LR: logistic regression; SVM: support vector machine.

4.3.5 The VAE Latent Representations Enhance Drug-Target Identification

Combining SMP and GP data can help establish connections between the MOAs of small molecules and genetic perturbations, which further help reveal the targets of small molecules (Lamb, 2007; Pabon et al., 2018). A simple approach is to examine whether a pair of perturbagens

(a small molecule and a genetic perturbation) leads to similar transcriptomic profiles, or more intriguingly, similar latent representations that reflect the state of the cellular system. Given a known pair of a drug and its target protein, we assumed that treatment with the drug and knockdown of the gene of the protein would result in a similar transcriptomic response reflected as raw data or VAE-derived latent representations.

To test the assumption, we extracted 16 FDA-approved drugs and their gene targets from the ChEMBL database that are also available in LINCS data (Gaulton et al., 2011; Pabon et al., 2018) (Table 4.4). We computed the Pearson correlations between the representations (either the raw expression profile or a latent representation) of each SMP sample treated by one of the drugs and the corresponding representations of all GP samples. The GP samples and their knockdown genes were then ranked according to the correlations to obtain the ranks of known target genes, in a manner similar to an information retrieval task (Figure 4.10a).

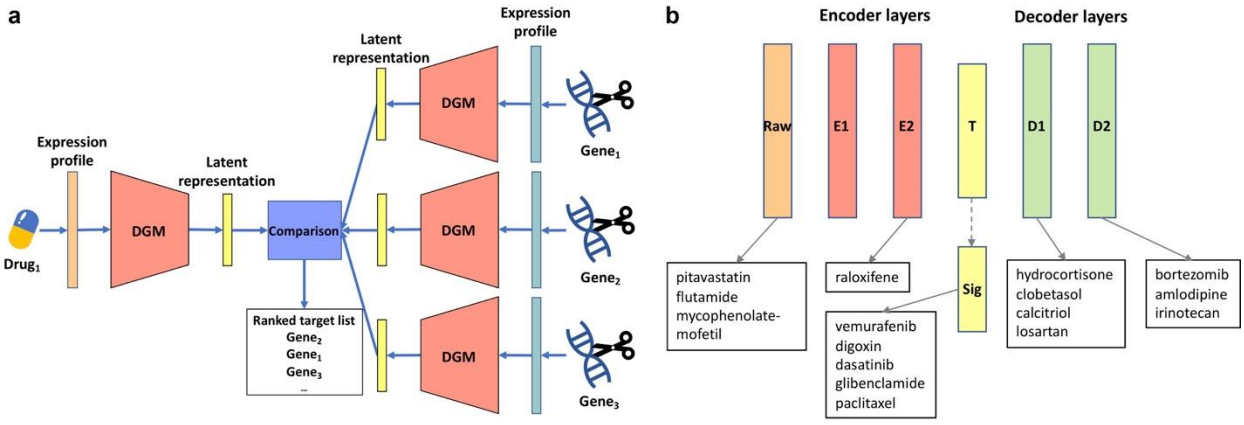


Figure 4.10. Drug-target prediction with different sample representations for 16 FDA-approved drugs.

(a) Diagram illustrating the approach for drug-target prediction. For a given drug, samples treated with the drug are fed to the SMGP-trained VAE to obtain latent representations from different layers of the encoder and decoder. All GP samples are also fed to the VAE to obtain corresponding latent representations and compared with the SMP samples by computing the Pearson correlation. For a given type of representation, genes are ranked according to the correlations to the representation of the given drug, and the ranks of the top known target of the drug are recorded. (b) The representation type that achieved the best matching (lowest mean rank) of the top known target gene for each drug.

Table 4.4. The mean of rank of the top known target for 16 FDA-approved drugs from drug-target prediction with different types of representation.

Drug	Target	# of SMP	# of GP	Raw	E1	E2	T	Sig.	D1	D2
pitavastatin	HMGCR	42	45	**376.40**	962.69	1930.62	806.81	706.74	827.60	688.00
bortezomib	PSMB10, PSMA3, PSMA1, PSMA5, PSMB7, PSMB5, PSMA8, PSMB1	2339	196	20.32	48.12	45.94	65.69	96.78	49.09	**19.13**
hydrocortisone	NR3C1	42	36	4605.67	3759.6	2221.74	3019.07	3339.52	**2122.33**	2392.12
vemurafenib	BRAF	81	108	1030.01	981.65	1215.56	779.80	**664.57**	970.10	1060.69
flutamide	AR	42	29	**5111.36**	7178.43	8167.55	9172.00	11013.07	13934.31	13047.52
clobetasol	NR3C1	42	36	5221.50	5025.12	3236.19	5060.95	5628.48	**3136.02**	4185.90
digoxin	ATP1A3, FXYD2, ATP1B1	42	83	595.62	431.10	492.21	497.86	**336.48**	590.71	405.69
mycophenolate-mofetil	IMPDH2	42	27	**4594.12**	6514.81	5928.67	6630.00	5091.55	8631.02	4683.26
dasatinib	LCK, YES1	204	175	942.70	743.19	694.73	681.38	**641.82**	649.23	658.83
amlodipine	CACNA1D	42	27	7798.79	4797.38	5290.60	5252.71	5159.55	5872.90	**4295.33**
calcitriol	VDR	42	27	5293.00	5859.21	6924.50	5877.14	6128.60	**5263.52**	5856.98
glibenclamide	KCNJ11	42	27	6074.93	9643.98	8653.05	7038.45	**3969.43**	7179.50	7104.81
paclitaxel	TUBB6, TUBA1A, TUBB2A, TUBB2C	42	105	464.17	548.00	781.81	640.90	**412.62**	421.00	831.40
losartan	AGTR1	42	24	4841.31	4767.88	4162.24	3865.98	3871.10	**3656.90**	4469.76
irinotecan	TOP1	42	27	5079.93	5317.57	4332.83	4459.55	4248.19	3999.52	**3915.17**
raloxifene	ESR2	42	36	4686.60	3236.07	**1262.83**	2078.57	2383.14	1595.02	1757.98

*Data related to Figure 4.10. The value for a given drug and a type of representation is the mean rank of the drug target gene(s) among all genes when retrieved and ranked according to the similarity between the representation of drug and representations of samples with gene knockdowns. The lowest mean rank is bolded for each drug. # of SMP: the number of SMP samples treated with the drug; # of GP: the number of GP samples with target gene knocked down; E: encoder layer; T: top hidden layer; Sig.: signature nodes; D: decoder layer.

We compared different types of representations to identify which were more effective in assigning a higher rank to target genes. As shown in Table 4.4 and Table 4.5, for different drugs, different representations achieved the best target-retrieval performance as reflected by top rank and mean rank. This suggests that VAEs can encode the impact of different drugs within different layers in the hierarchy that potentially reflect the relative level of drug-target interactions in the cellular signaling network. Figure 4.10b summarizes the mean rank results where each drug is assigned to the representation layer that produced the best mean rank of its top known target. Most drugs have their best performance achieved with VAE-learned latent representations rather than the raw expression profiles, and for five drugs, the best performance was achieved with the 12-signature-node-representation. Table 4.5 gives the best rank of the top known target for each drug, which is comparable to the Table 1 from Pabon et al. (Pabon et al., 2018). Even though our approach is essentially an unsupervised learning method based purely on expression data, 13 out of 16 drugs received an equal or better rank than from the previous state of the art random forest model trained with a combination of expression and protein-protein interaction features (Pabon et al., 2018) (bolded in Table 4.5).

Table 4.5. The rank of the top known target for 16 FDA-approved drugs from drug-target prediction with different types of representations.

Drug	Target	Raw	E1	E2	T	Sig.	D1	D2
pitavastatin	HMGCR	2	1	1	1	1	2	1
bortezomib	PSMB10, PSMA3, PSMA1, PSMA5, PSMB7, PSMB5, PSMA8, PSMB1	1	1	1	1	1	1	1
hydrocortisone	NR3C1	72	35	37	12	11	3	1
vemurafenib	BRAF	1	1	1	1	1	1	1
flutamide	AR	36	14	39	78	160	29	163
clobetasol	NR3C1	415	10	2	7	16	13	4
digoxin	ATP1A3, FXYD2, ATP1B1	71	2	9	30	15	23	11
mycophenolate-mofetil	IMPDH2	12	342	31	28	21	18	44
dasatinib	LCK, YES1	25	2	5	5	7	3	2
amlodipine	CACNA1D	62	243	134	111	116	97	58
calcitriol	VDR	917	164	341	661	211	1018	335
glibenclamide	KCNJ11	275	497	336	579	307	482	536
paclitaxel	TUBB6, TUBA1A, TUBB2A, TUBB2C	14	71	49	20	18	92	98
losartan	AGTR1	238	190	370	53	115	253	57
irinotecan	TOP1	308	653	331	18	13	326	504
raloxifene	ESR2	114	115	53	139	72	105	65

*The value for a given drug and a type of representation is the top rank of the drug target gene(s) among all genes when retrieved and ranked according to the similarity between the representation of drug and representations of samples with gene knockdowns. A lower rank is better. Drugs with the lowest rank equal to or lower than the rank reported by Pabon et al. (Pabon et al., 2018) are bolded. E: encoder layer; T: top hidden layer; Sig: signature nodes; D: decoder layer.

4.4 Discussion

In this chapter, we examined the utility of DGMs, specifically VAE and S-VQ-VAE, for learning representations of the states of cells treated with different perturbagens in the LINCS project. We showed that the trained VAE and S-VQ-VAE models were able to accurately regenerate transcriptomic profiles almost indistinguishable from the input data. These results are

intriguing because they suggest that the DGMs have captured signals of cellular processes underlying the statistical structures of the data. Such capability is highly desirable as it provides a means to investigate how responses to diverse environmental changes are processed by the cellular system as signals.

Cellular signaling systems are essentially signal coding machines, and training a model capable of mimicking the behaviors of cellular signaling systems is a critical step of using contemporary artificial intelligence technologies to advance systems biology. A more intriguing future direction is to investigate whether the latent variables of DGMs can be mapped to the signals encoded by real biological entities like proteins or pathways as suggested by previous research in simpler organism systems (Chen et al., 2016). This may require designing more interpretable deep learning models that integrate information from multiple platforms. Particularly, additional information such as genetic perturbations can be utilized to facilitate establishing a mapping between biological entities and latent variables.

During our development of the models, we compared VAE with other DGMs, including restricted Boltzmann machines (Salakhutdinov & Hinton, 2009), deep belief networks (Salakhutdinov & Hinton, 2009), deep autoencoders (Hinton & Salakhutdinov, 2006), and VQ-VAEs (Aaron Van Den Oord & Vinyals, 2017). VAE outperformed all of these DGMs in capturing the expression data distribution. However, in its original form, VAE cannot utilize additional information aside from data passed via the input layer. The S-VQ-VAE model is an early attempt toward the goal of combining different information sources. It utilizes additional label information to facilitate the learning of global representations, but essentially it does not directly combine multiple types of data nor realize a fully interpretable multi-task learning. More directions of model design remain to be explored.

As cells are the basic unit of life, a complete model for understanding cellular signaling systems would represent a major breakthrough in both machine learning and systems biology, with profound implications for cell biology, pharmacology, drug development, and precision medicine. In the next chapter, we further examine the utility of DGMs for learning representations from genomics data for predicting drug sensitivities for both cell lines and real patients.

5.0 Chapter 3: drug sensitivity prediction with deep generative models and transfer learning

5.1 Introduction

Cancers are heterogeneous in that tumors originating from the same tissue are often driven by different disease mechanisms. This inherent heterogeneity results in differentiated drug responses across patients of the same cancer type. For many widely used non-specific chemotherapy drugs, only a small fraction of patients respond to the drugs while for the others the drugs do not bring any obvious benefits. As a result, increasing demand has emerged for computational tools that can predict patient drug response in advance to help select drugs based on patient-specific information. Such tools promise to significantly improve survival outcomes and reduce unnecessary side-effects resulted from the indiscriminate adoption of standard therapy protocols for all patients.

Most traditional drug response prediction models were developed in vitro based on cell line data (Barretina et al., 2012; Costello et al., 2014; Dong et al., 2015; Encyclopedia, 2015; Jang et al., 2014; Q. Li et al., 2019; Menden et al., 2013a; Riddick et al., 2010). These models were typically supervised regression or classification models that predict drug sensitivity by learning direct correlations between genomic data and drug response outcomes. The use of only cell line data and the shallow architecture of these models all limited their potential to be deployed in the clinical environment and benefit patients.

With genomic data growing at an unprecedented speed, deep neural network models (DNNs) have attracted more attention these days for solving medical problems. Among DNNs, a

branch known as the deep generative model (DGM) has proved very successful in solving many of the hardest machine learning problems. Instead of modeling the direct correlation between input features and outcomes, DGMs are typically used to learn a latent representation of the input features in an unsupervised learning manner, where the latent representations are then used to train a second supervised model to predict the targets. The latent representations condense the information from the raw features, and thus are often more informative of the outcome than are the raw features.

In this chapter, we compare six types of DGMs for learning latent representations from expression and mutation data and predicting drug sensitivity of cell lines and real patients. The DGMs we examined are AutoEncoder (AE) (Hinton & Salakhutdinov, 2006), Variational AutoEncoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014), and four new DGMs we designed in this dissertation project, including Res-AutoEncoder (ResAE), Res-Variational AutoEncoder (ResVAE), Redundant Input AutoEncoder (RIAE), and Redundant Input Variational AutoEncoder (RIVAE) (Figure 5.1). AE and VAE learn latent representations by directly reconstructing expression profiles. ResAE, RIAE, ResVAE, and RIVAE are extensions of AE and VAE, respectively, which combine mutation data for learning representations from expression data. Concretely, ResAE and ResVAE incorporate mutation data into the model by parallelly adding transformed mutation data as residues to different hidden layers of AE and VAE, as inspired by the Residual neural Network (ResNet) (He, Zhang, Ren, & Sun, 2016). RIAE and RIVAE incorporate mutation into the model by redundantly concatenating the mutation data to different hidden layers of AE and VAE, as inspired by the Redundant Input Neural Network (RINN) (Young, 2020).

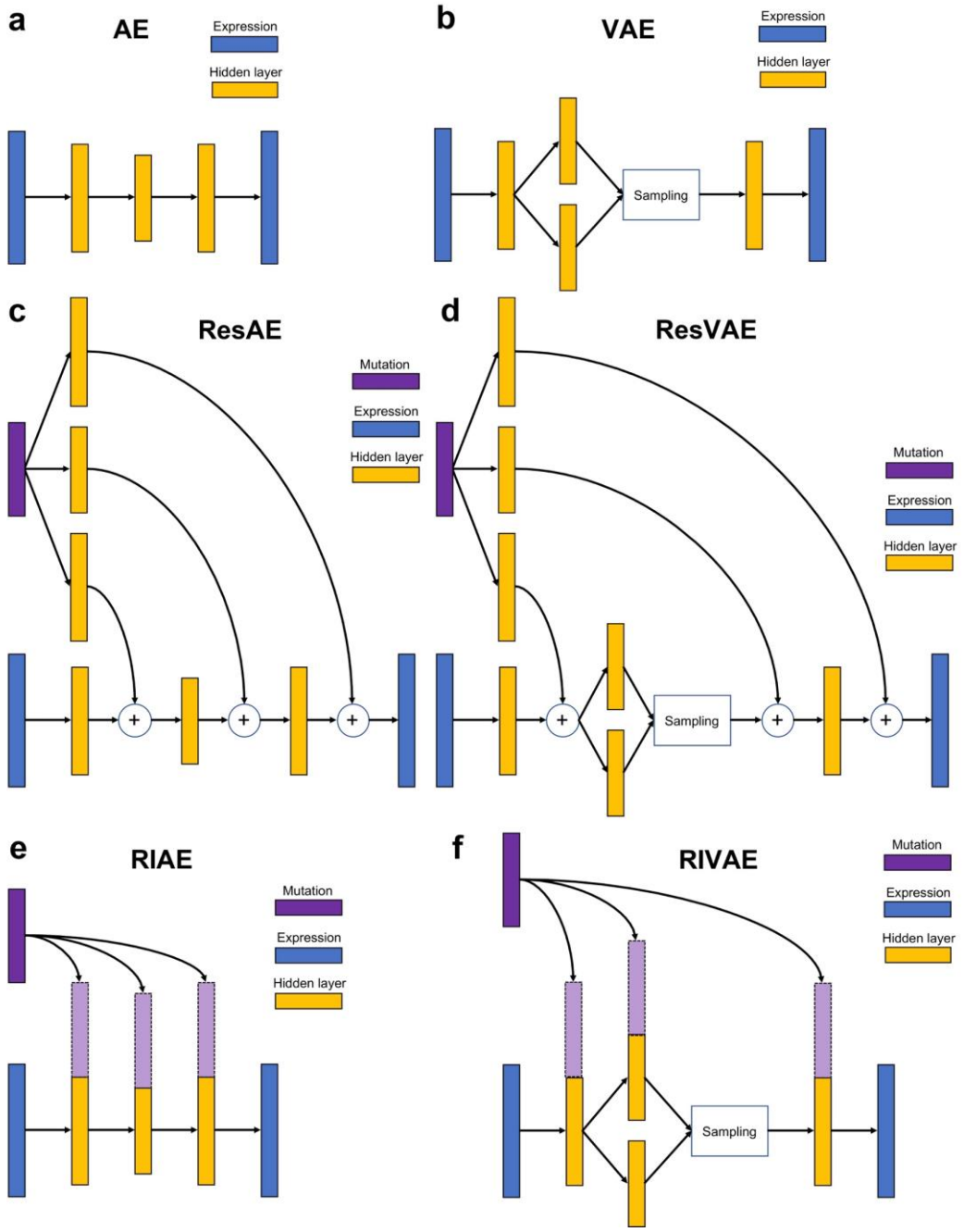


Figure 5.1. Generative models used for learning latent representations from genomic data.

The learned representations from different hidden layers of DGMs were used to train Elastic Net Logistic Regression models (ENLRs) for predicting GDSC and CCLE cell lines and TCGA patients as sensitive vs. resistant to various drugs. We show that for most drugs from GDSC

and CCLE, latent representations achieved better prediction performance, from the aspect of Area Under the ROC Curve (AUC) score, than using the raw expression profiles. In addition, VAE-based models generally worked better than AE-based models in learning drug sensitivity informative latent representations, among which ResVAE performed the best from many aspects of model evaluation. The representations of mutated genes learned with ResVAE can also be used to reveal functional associations between genes. The ENLR selected significant hidden nodes are highly correlated with drug response outcomes, where some nodes can be sued as strong indicators for drug sensitivity. The ENLRs trained with cell line data can be transferred to make predictions for real patients and divide patients into different survival groups.

Our results support DGMs as a powerful tool for modeling correlations among expression and mutation data and learning latent representations for solving drug response prediction problem. The models we developed can be deployed as clinical decision support tools for promoting precision oncology.

5.2 Methods

5.2.1 Data

We used the GDSC and CCLE datasets for learning representations of cell lines and training drug sensitivity prediction models. The GDSC data were downloaded from the GDSC data portal of the Sanger Institute (https://www.cancerrxgene.org/). The dataset contains expression data and mutation data of 1018 cell lines, and drug response data of 320 drugs and 175 drugs from GDSC1 and GDSC2, respectively. The expression data are Robust Multichip Average

(RMA) normalized microarray data. The CCLE data were downloaded from the CCLE data portal of the Broad Institute (https://portals.broadinstitute.org/ccle). The dataset contains expression data, mutation data, and drug response data for 1019 cell lines and 24 drugs. The expression data are RNAseq normalized count (transcript per million (TPM)) data. We applied a log-transformation on the expression data before further data preprocessing. In order to leverage the power of transferring learning, we also tried pretraining DGMs with a large number of expression profiles from the TCGA database. This pretraining PANCAN dataset was composed of RNAseq expression data for 10,332 tumor samples across 33 cancer types downloaded from the UCSC Xena TCGA data portal (https://xenabrowser.net/datapages/).

When training ResAE, ResVAE, RIAE, and RIVAE, we incorporated the mutation data into the model. The intuition is to enhance the ability of DGMs in learning latent representations from expression data by utilizing information from the causal relationships between somatic mutations and gene expression outcomes. The binary mutation states of 143 genes were used as input data for ResAE, ResVAE, RIAE, and RIVAE, where the genes were selected by overlapping the mutation data across TCGA, GDSC, and CCLE. For cell lines or TCGA samples with missing mutation information, we filled 0 (assuming no mutation) for all genes.

In order to examine whether the cell-line trained drug sensitivity prediction models can be transferred to real patients, we first tested our models on lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC) patients from TCGA. We download the drug usage data of 179 LUAD and 144 LUSC patients from the Genomic Data Commons Data Portal (GDC, https://portal.gdc.cancer.gov/). The corresponding survival data were downloaded from the UCSC Xena data portal. To rule out other clinical confounders except for chemotherapy drugs that may

affect survival outcome, we only included patients that were given drugs for adjuvant therapy. This resulted in a lung cancer test dataset with drug usage and survival information for 182 patients.

We further tested our models on a larger TCGA patient set with drug usage and overall response outcomes. This dataset was provided by (Z. Ding, Zu, & Gu, 2016) as supplementary information, which contains drug usage and response information for 1,197 TCGA patients across 28 cancer types using 152 drugs. Among the 152 drugs, 32 were available in GDSC1 for which we have trained ENLRs as sensitivity prediction models. We filtered out patients that have no drug with an ENLR or have no expression data available. This left us with 880 patients across 15 cancer types using 31 drugs.

5.2.2 Drug response data binarization

We treated drug sensitivity prediction as a binary classification task where all cell lines were assigned into sensitive or resistant classes based on the area under the curve (AUC) score (for GDSC data) or activity area (for CCLE data). The AUC score in the GDSC drug response data is the normalized area under the relative availability to the drug concentration curve (dose-response curve). The lower the AUC, the more sensitive the cell line to the drug. The activity area in the CCLE drug response data is the area over the drug-response curve, and the higher the activity area, the more sensitive the cell line to the drug. We used the waterfall approach described in the CCLE study (Barretina et al., 2012) to find the threshold to discretize the drug response measures (AUC or activity area). The approach we used is as follows. First, sort the measurements to generate the rank-ordered plot. If the curve appears linear (Pearson correlation > 0.95), then the median value is used as the threshold to binarize the response. If the curve is unilinear, estimate the major inflection point of the curve as the point on the curve with the maximal distance to the straight line

dawn between the start and end points of the curve. The value of the inflection point is used as the threshold. Unlike the original waterfall approach, we did not define an intermediate class of cell lines with moderate response values. Instead, all cell lines are assigned as either sensitive or resistant to a drug based on the threshold.

5.2.3 ResAE and ResVAE

We implemented six major different types of DGMs for learning latent representations from expression data. Among the models, AE and VAE are classic DGMs that have been widely adopted in deep learning application areas (Hinton & Salakhutdinov, 2006; Kingma & Welling, 2014; Rezende et al., 2014). Here, we used AE and VAE to learning representations by reconstructing the expression profiles of samples.

To enhance the representation learning using causal information between mutation genes and gene expressions, we developed four hybrid models based on AE and VAE. Among them, ResAE and ResVAE were developed inspired by ResNet (He et al., 2016). Unlike ordinary deep neural network models (DNN) that only establish connections between adjacent layers, ResNet utilizes skip connections to jump over layers, thus to shorten the path for backpropagation and prevent vanishing gradients. For example, suppose we have a single skip from layer $l-2$ to l, the activations a^l of the layer l would be

$$a^l = g(Z^l + a^{l-2}) \tag{5.1}$$

Here g is the activation function for layer l, and $Z^l = W^{l-1,l} \cdot a^{l-1} + b^l$ where $W^{l-1,l}$ is the weight matrix between layer $l-1$ and l. With this formula, the skipped layer $l-1$ can be seen as learning the residual Z^l between the a^{l-2} and the pre-activation of layer l, thus comes the name ResNet. An additional weight matrix can be used to learn the weights for the skipped connection,

which allows a flexible dimension matching between the start and destination layers. This leads to the HighwayNet (Srivastava, Greff, & Schmidhuber, 2015) with

$$a^l = g(Z^l + W^{l-2,l} \cdot a^{l-2}) \tag{5.2}$$

Parallel skips originating from the same layer and successively connecting to later layers can also be added with

$$a^l = g(Z^l + \sum_{k=2}^{K} W^{l-k,l} \cdot a^{l-k}) \tag{5.3}$$

which gives rise to another extension known as the DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2017).

In our case, to incorporate the mutation data into the expression-data-learning AE and VAE models, we followed a similar structure of DenseNet where we parallelly transform the input mutation data vector (*M*) with weight matrices and add them to different hidden layers of AE and VAE (Figure 5.1 c and d). The activations a^l of a hidden layer l is

$$a^l = g(Z^l + W^{m,l} \cdot M) \tag{5.4}$$

This can be seen as using the transformed mutation data as residuals to supplement Z^l to be used as the pre-activation of layer *l*.

We tried different constraints on the weight matrices for transforming the mutation data, including assigning an independent weight matrix for each hidden layer, assigning and tying the weight matrix for all corresponding hidden layers in encoder and decoder (tied-weights), assigning and tying weight matrices to a subset of corresponding hidden layers in encoder and decoder, and only assigning independent weight matrices to encoder or decoder hidden layers. The assumption behind tying the weights of corresponding encoder and decoder layers is that the symmetric architectures of encoder and decoder may suggest they model the expression data generation

process in a reversed manner. Therefore, the contribution of the mutation data on the lth encoder layer should be the same as on the lth to last decoder layer. The tied-weights helps reduce the number of parameters for tuning and accelerate training convergence.

The difference between ResAE and ResVAE is that ResVAE uses a VAE architecture as its backbone. As in an ordinary VAE, the variational inference over the posterior distribution of the latent variables is performed on the top hidden layer of the encoder in ResVAE.

In general, we are expecting that the incorporation of mutation data in ResAE and ResVAE will help the model learn more robust and generalizable expression-based latent representations. The weight matrices that map between the mutation vector to each hidden layer may also help reveal the contribution of mutated genes to each hidden node, thus improve the interpretability of the model.

5.2.4 RIAE and RIVAE

Another two hybrid models we designed are RIAE and RIVAE, where instead of adding the transformed mutation data to different hidden layers, the mutation data are concatenated to hidden layers, as inspired by the RINN (Young, 2020).

RINN is a DNN specifically designed to model the causal relationships between genomic alterations and gene expressions (Young, 2020). Similar to our motivation of using DGM to model the signaling network, RINN learns the hierarchical causal relationships between signaling network components through using the mutation profiles to predict the gene expression outcomes. Unlike the traditional DNNs where the input data are only fed in through the input layer, RINN allows the input mutation profiles to directly interact with all hidden layers through concatenations, thus comes the name of "Redundant Input". This design is based on the assumption that different

alterations affect functional components on different levels of a cellular signaling cascade. Allowing direct connections between the "causes", mutations, to the "outcomes", latent variables make it possible for RINN to model any direct or indirect causal relationships between signaling network components.

In our case, since we also want to utilize the causal relationships between mutations and gene expression to enhance signaling network representation learning, we followed the same idea of RINN and developed RIAE and RIVAE. In RIAE and RIVAE, the activations a^l of a hidden layer l is

$$a^l = g(W^{l-1+m,l} \cdot cat(a^{l-1}, M) + b^l) \tag{5.5}$$

where $W^{l-1+m,l}$ is of dimension $D(l) \times (D(l-1) + D(M))$. For RIVAE, the variational inference is performed on the top hidden layer as in an ordinary VAE.

5.2.5 Model architectures and training settings

We used the same expression gene set as the input features for all our DGMs. This gene set contained 3,024 genes, extracted from the GDSC expression data based on the median absolute deviation (MAD) of expression level, the outlier sum statistics for differential gene expression, and the Hartigan dip test for expression bimodality. Specifically, a gene is included if its expression data cross all cell lines have MAD > 1, outlier sum statistics > 1100, and Hartigan dip test p-value < 0.2. After intersecting with the TCGA, GDSC, and CCLE datasets to include genes that had expression data in all three datasets, 2,758 genes remained in the final input gene set.

The DGMs were implemented in Python3 with the *Pytorch* library 1.3.1 (Paszke et al., 2017). We used the tangent function as the activation function for all hidden layers. All DGMs were trained on 9/10 of the data and validated on the rest, with a learning rate of 1e-4 and batch

size 64. We tried over 200 different combinations of training settings, including different model architectures, training datasets, epoch numbers, and L1-norm regularizations. We compared the models based on reconstruction loss (Supplementary Table S5.1).

In order to estimate the usability of the DGMs in the following drug sensitivity prediction task, we also trained ordinary Logistic Regression models (LRs) with latent representations from each DGM and compared DGMs based on validation AUC average over GDSC2 drugs (Figure 5.2a). Concretely, for each drug from the GDSC2 dataset, we trained a set of LRs, each using a type of representation (raw expression profile or latent representations) from the given DGM. The LR was trained in a 10-fold cross-validation manner, where in each iteration, an independent LR is trained on 9/10 of the cell lines and made predictions for the other 1/10 cell lines. After the 10 iterations, the validation predictions were collected and combined for all cell lines, and an AUC score was computed. Different types of representations were compared based on the AUC score to determine the best representation type for predicting the given drug. The best AUC score achieved for each drug was collected and average over all drugs as a metric for comparison between DGMs (Supplementary Table S5.1). For each DGM, we also computed the proportion of drugs that achieved their best prediction AUCs using the raw input features rather than a latent representation. The lower the raw proportion, the relatively more informative the learned latent representations for drug sensitivity prediction.

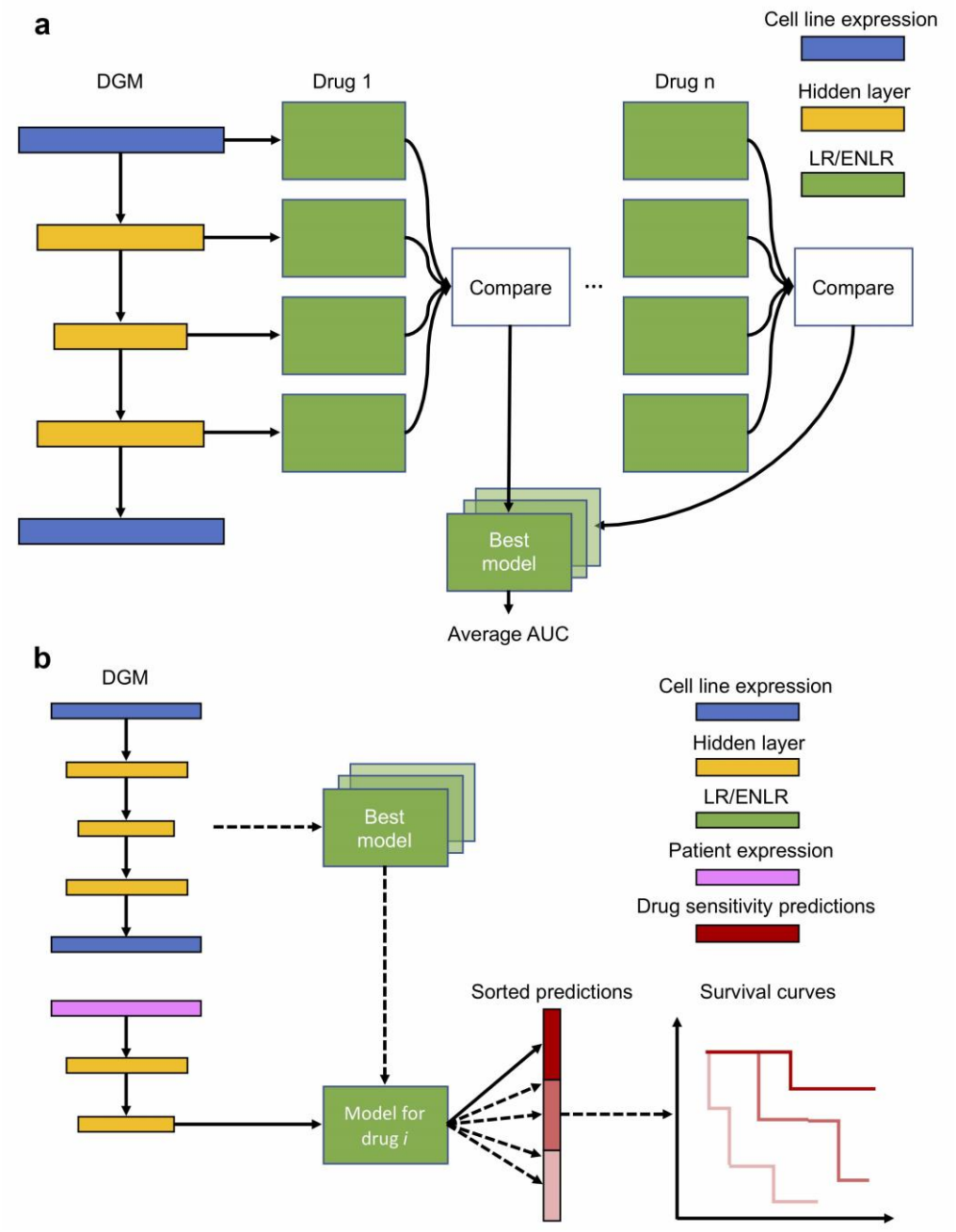


Figure 5.2. Diagrams of drug sensitivity prediction model construction and application.

(a) For each drug, use the raw expression profiles of cell lines and latent representations from different hidden layers of a given DGM to train a set of prediction models. Select the best model for each drug with the highest AUC score. (b) For a drug of interest, extract the best prediction model for it, and input the corresponding latent representations for real patients to generate drug sensitivity predictions. Divide the patients into sensitive, intermediate, and resistant groups based on the predictions and compare their survival outcomes.

5.2.6 Elastic net logistic regression for drug sensitivity prediction

The learned latent representations from DGMs as well as the raw expression profile were used as input features to train ENLRs to predict drug sensitivity (Figure 5.2a). The models were implemented in R 3.6.2 with *glmnet* library 3.0.2 (Friedman, Hastie, & Tibshirani, 2010). The objective function of ENLR in *glmnet* is the negative binomial log-likelihood function as

$$\min_{(\beta_0,\beta)} -\left[\frac{1}{N}\sum_{i=1}^{N} y_i \cdot (\beta_0 + x_i^T\beta) - \log\left(1 + e^{\beta_0 + x_i^T\beta}\right)\right] + \lambda[\frac{(1-\alpha)||\beta||_2^2}{2} + \alpha\left|\left|\beta\right|\right|_1] \quad (5.6)$$

When $\alpha = 1$ the model reduces to the Lasso regularization and when $\alpha = 0$ the model reduces to the Ridge regularization.

For each drug in GDSC1 and GDSC2, we trained multiple models independently on different types of representations. The models were trained with 25-cross validation using $alpha = 0.5$ and $\lambda \in [e^{-8}, e^{-1}]$. Lambda was selected by *glmnet* to maximize the validation AUC score of prediction. We used a random seed of 42 to allow repeat experiments. For each drug, the representation (either a latent representation or the raw expression profile) gives the best validation AUC was selected, and then validation AUCs and F1 scores were averaged overall all drugs to compare between DGMs. Since GDSC1 and GDSC2 tested different sets of drugs on a different number of cell lines using different techniques, models trained on GDSC1 and GDSC2 were compared separately.

The ENLRs were tested on an independent subset of CCLE drugs that overlaps with GDSC1 or GDSC2. 16 out of 24 CCLE drugs were available in GDSC1 and used to test GDSC1 trained DGMs, and 12 out of 24 CCLE drugs were available in the GDSC2 and used to test GDSC2 trained DGMs.

5.2.7 Survival analysis of lung cancer patients with drug response predicted by cell line-trained ENLRs

We applied the GDSC-trained ENLRs on TCGA lung cancer patients to see if the predicted drug responses are correlated with the patients' survival outcomes (Figure 5.2b). Specifically, for each set of chemotherapy drugs of interest, we extract the subset of lung cancer patients that were given at least one of the drugs for adjuvant therapy. For each patient that took a drug, we used the corresponding ENLR to predict the drug sensitivity probability. We then sorted the patients according to their predictions and marked the top 1/3 patients as sensitivity to the drug, middle 1/3 as intermediate, and tail 1/3 as resistant. Finally, we divided patients into three mutual-exclusive groups, sensitive, intermediate, and resistant with the following rules.

1. A patient is classified into the sensitive group if he is predicted as sensitive to at least one of the drugs he was given.

2. A patient is classified into the resistant group if he is NOT predicted as sensitive to any drug AND is predicted as resistant to at least one of the drugs he was given.

3. All other patients are classified as the intermediate group.

We used the Python library *lifelines* 0.23.9 (DOI: 10.5281/zenodo.3727281) to generate the Kaplan-Meier curves of the three response groups and compare them using the log-rank test to see if the curves are significantly different.

5.2.8 Prediction analyses of PANCAN patients with drug responses predicted by cell line-trained ENLRs

We also tested our ENLRs on 880 TCGA patients of 15 cancer types extracted from (Z. Ding et al., 2016) (see Data). The patients were labeled with four different types of response: “Complete Response”, “Partial Response”, “Stable Disease”, and “Clinical Progressive Disease”. We considered the first two as indicators of a responder and the last two as indicators of a non-responder, and we binarized the ground-truth response label accordingly. For each patient, we made predictions for all the drugs he took that we have corresponding GDSC1-trained ENLR models. The predictions were in the form of probability of sensitivity. The patient is represented by the highest probability among all drugs he took (highest prob.) as well as the probability calculated through a noisy-or mechanism across the drugs (noisy-or prob.). We then computed AUC and F1 scores from the ground-truth label and the predicted probability as a measure to evaluate our ENLRs. A threshold of 0.5 was used to binarized the predictions when computing F1 scores and conducting independent tests.

5.2.9 Code availability

The Python code for preprocessing raw data, training DGMs, generating latent representations, and doing model and survival analyses, R code for training ENLRs and prediction drug responses, and trained DGMs are available upon request.

5.3 Results

5.3.1 Non-paranormal transformation for merging expression data from different resources

The GDSC expression data are microarray data, while the expression data from the CCLE and TCGA are RNA-seq data. The different measuring platforms and preprocessing procedures resulted in incompatible distributions across the three expression datasets (Figure 5.3a). In order to allow a DGM trained on one dataset to be transferrable to the others, we applied a nonparanormal (NPN) transformation on each dataset for distribution normalization. As shown in Figure 5.3 a and b, after the nonparanormal transformation, the distribution of TCGA, GDSC, and CCLE expression data became compatible. In Figure 5.3b, there is an outlier group in the TCGA data with higher first component values. This group consists of 879 samples from three neuron related cancer types, the Lower Grade Glioma (LGG), Glioblastoma (GBM), and Pheochromocytoma and Paraganglioma (PCPG) (Figure 5.3c). These cancer types have distinct expression profiles compared to other cancer types due to their special origin of cells. Therefore,

we excluded this outlier group from the training dataset, which resulted in a consistent distribution across all three data resources as shown in Figure 5.3d.

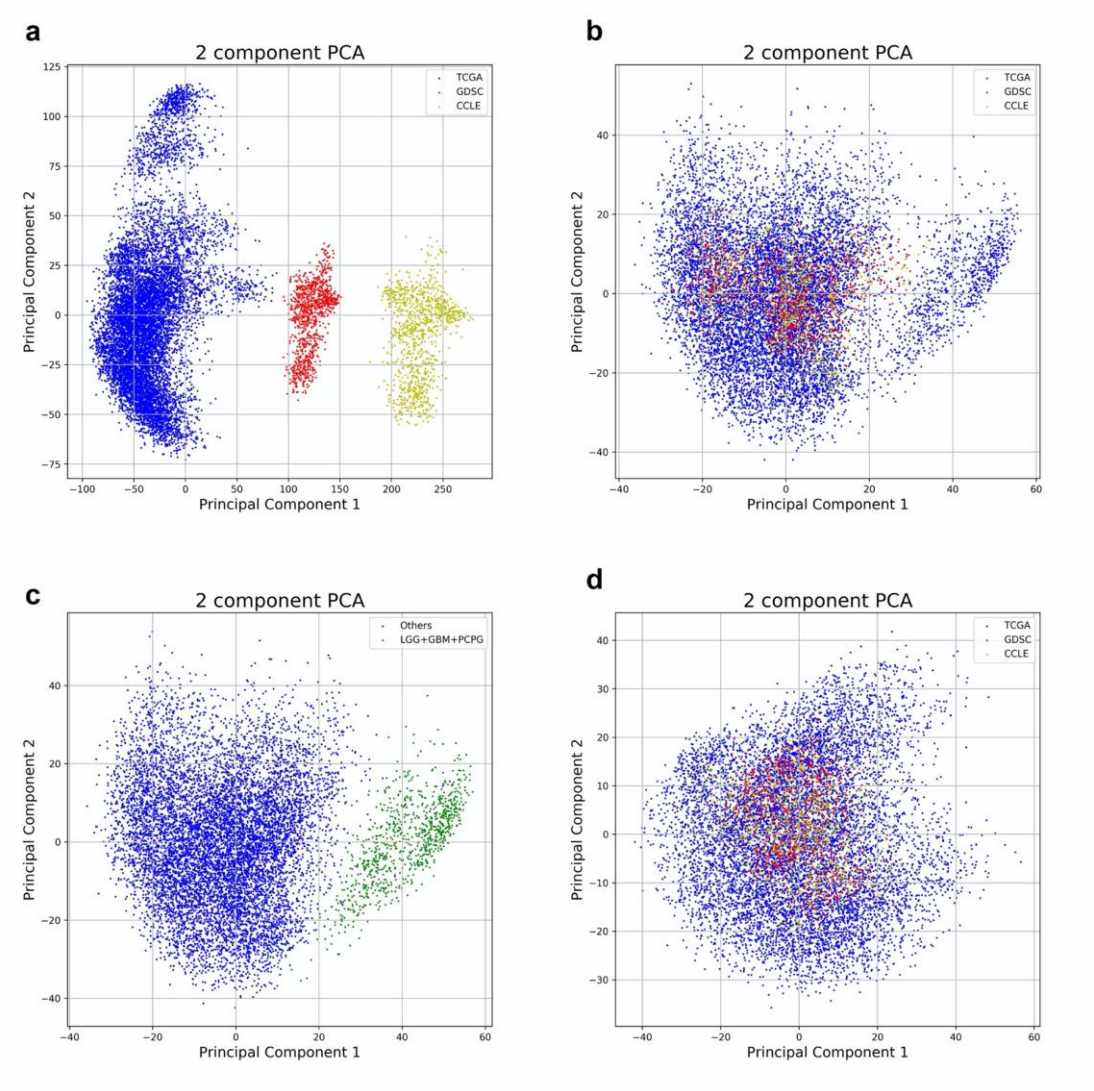


Figure 5.3. NPN transformation of TCGA, GDSC, and CCLE expression data.

(a) the two-component PCA plot of raw TCGA, GDSC, and CCLE expression data. (b) the PCA plot of TCGA, GDSC, and CCLE expression data after NPN transformation. (c) the PCA plot of TCGA expression data with the outlier group of neuron-related cancer types highlighted. (d) the PCA plot of TCGA, GDSC, and CCLE expression data after NPN transformation and excluding the outlier group.

In addition to the overall distribution compatibility, we next examined whether NPN transformation preserves the tumor-type-specific characteristics of samples. We compared the cell lines of GDSC with samples in TCGA and cell lines in CCLE by computing the cosine distance (1 - *cosine similarity*) between their expression profiles, and then examined the nearest neighbor relationships before and after NPN transformation. We are expecting that NPN transformation should not reduce the proportion of GDSC cell lines where their nearest neighbors in TCGA or CCLE are of the same tumor type. To determine whether two samples are of the same tumor type, we utilized the TCGA cancer type labels of GDSC cell lines to match between GDSC and TCGA samples and the tumor type description keywords to match between GDSC and CCLE cell lines. The keyword pairs we used are summarized in Supplementary Table S5.2.

Before NPN transformation, 247 out of 1018 (24.26%) GDSC cell lines had their nearest neighbors in TCGA of the same cancer type label. This number increased to 303 (29.76%) after the transformation. The GDSC and CCLE cell lines are already aligned well in the original expression data space, even though they were measured on different platforms. Specifically, 794 (78.00%) GDSC cell lines had a CCLE nearest neighbor of the same tumor type, and 789 out of 1019 (77.43%) CCLE cell lines had a GDSC nearest neighbor of the same tumor type. The similar cell line composition between GDSC and CCLE is also supported by the similar contour of the distributions of the raw expression profiles as shown in Figure 5.3a. After the transformation, the numbers were further increased, with 832 (81.73%) GDSC cell lines had a matched CCLE nearest neighbor and 791 (77.63%) CCLE cell lines had a matched GDSC nearest neighbor. Therefore, NPN transformation not only preserves the tumor-type-specific information of expression profiles but also improves the expression profile matching across datasets by normalizing the data distribution.

5.3.2 Model selection for learning latent representations

We trained over 200 DGMs with different training settings (see Methods). We also preliminarily tested the potential of the latent representations learned by these models for drug sensitivity prediction by training LR models using GDSC2 drug response data (see Methods). The performances of the majority of the models are summarized in Supplementary Table S5.1. We compared the models based on their validation reconstruction losses and LR AUC scores. From these experiments, we obtained the following observations.

1. VAE-based models had higher reconstruction loss than AE models, and adding L1 regularization of weight matrices would decrease reconstruction loss for AE-based models but increase loss for VAE-based models (Figure 5.4a).

2. The architecture that generally achieved a lower loss across all model types contained three hidden layers with 1000, 500, 1000 hidden nodes respectively (Figure 5.4b).

3. Pre-training the models with TCGA data then fine-tuning with GDSC data generally reduces the reconstruction loss compared to using the GDSC data alone (Figure 5.4c).

4. Incorporating mutation data did not significantly impact the reconstruction loss compared to using expression data alone (Figure 5.4d).

5. VAE and its extension models generally achieved higher AUC scores compared to AE models, when using their latent representations to train LRs for drug sensitivity predictions (Figure 5.4e, see Methods).

6. L1 regularization helped improve LR prediction performance (Figure 5.4f).

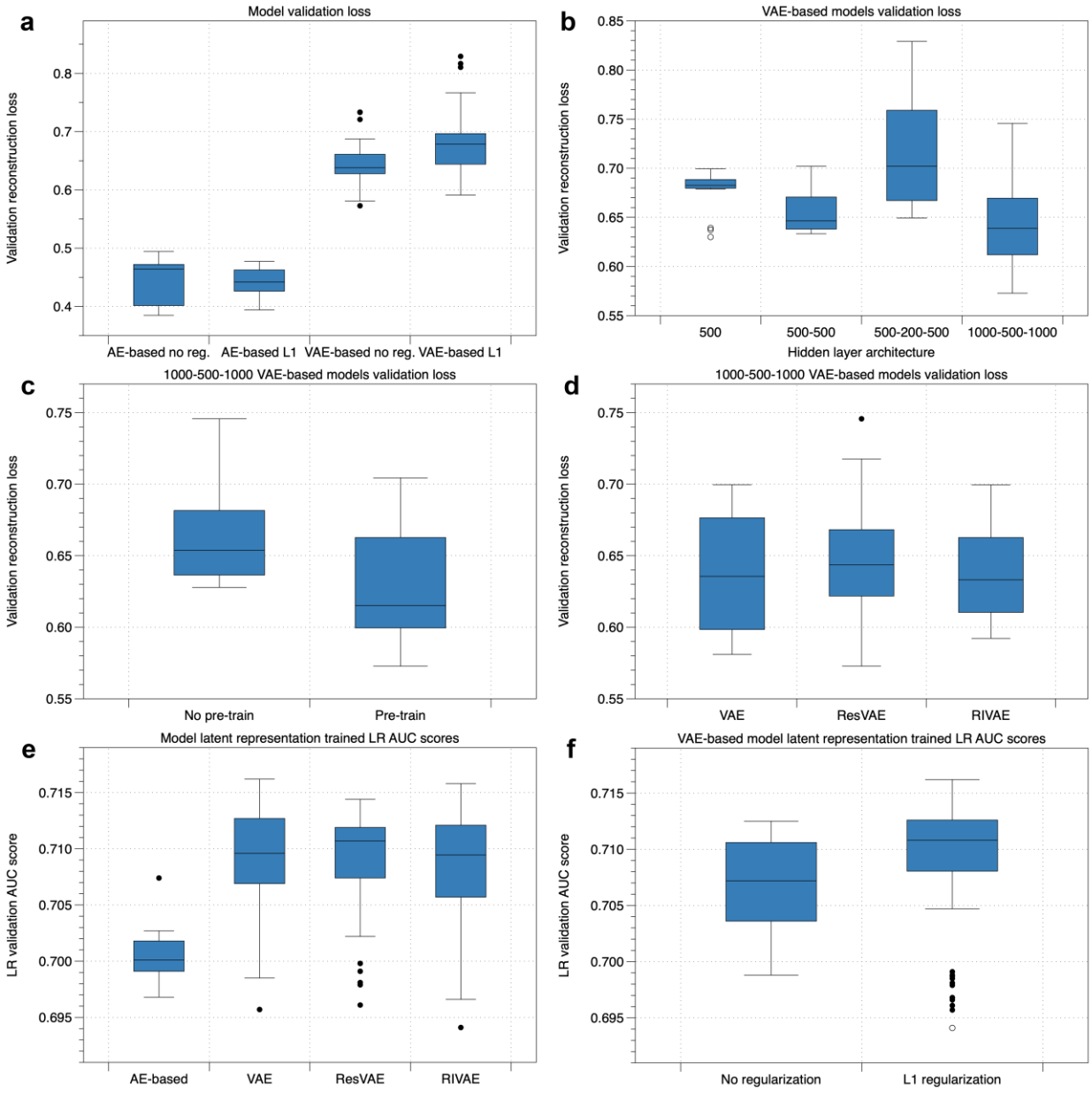


Figure 5.4. Performance comparison for all trained DGMs from the aspects of GDSC expression profile reconstruction loss and ordinary logistic regression AUC score.

The reason VAE-based models had higher reconstruction loss compared to AE-based models is due to the variational inference step performed on the top hidden layer of VAE encoder. This variational inference can be seen as a strong regularization operation. According to our previous experience with the LINCS data (introduced in Chapter 4), the variational inference helps

VAEs learn more concise and robust latent representations compared to AEs, which are often more informative and noisy tolerant for a secondary prediction task. This is supported by the higher LR AUC scores for VAE-based models compared to AE-based models (Figure 5.4e).

Among the VAE-based models, ResVAE consistently outperformed the ordinary VAE and RIVAE from the aspect of LR AUC scores when the other aspects of model setting up (e.g, architecture, training data, and regularization) are similar (Figure 5.4e and Supplementary Table S5.1). Based on these observations, in the following analyses, we mainly focused on VAE-based models, especially ResVAE, with a backbone architecture of 1000-500-1000.

5.3.3 Weight matrices for transforming mutation data revealing pathway information

Adding linearly transformed mutation data in ResAE and ResVAE not only helps reduce the validation reconstruction loss for the input expression data but also improve the interpretability of the model. The weight matrices ($W^{h\times M}$) learned to transform the mutation data provide a numeric vector for representing each mutated gene. This vector, the same dimension as the target hidden layer the transformed mutation data is added to, quantifies how much the mutated gene contributes to hidden nodes. Comparing the vectors between the mutated genes will help reveal the functional correlations between the genes. Our assumption is that the more similar the vectors of two genes, the more similar the pattern in which they affect the hidden nodes and the expression reconstruction process, and the more functional correlated the two genes are.

To test this assumption, we computed the cosine similarity between the vectors of each pair of the 143 input mutation genes and retrieved the top 3 nearest neighbors for each gene. Cosine similarity was chosen over other distance measurements to emphasize the direction consistency between the numeric vectors. We represented each gene by the corresponding vector from the

mutation transformation weight matrix for the first hidden layer. Then we counted the number of genes that had their top 1 nearest neighbor gene or at least one of the top 3 nearest neighbor genes from the same oncogenic signaling pathway. The ground truth of oncogenic signaling pathway assignments was extracted from Figure 2 of (Sanchez-Vega et al., 2018). Among the 143 mutation genes, 57 had their signaling pathway information available (in total 10 pathways), so we focused on these genes in particular.

In Table 5.1 we show the counting results of ResAEs and ResVAEs trained with tied-weight (see Methods). We can see that ResVAE representations are generally much better than ResAEs for revealing functional correlations between genes, with significantly more genes having their nearest neighbors of the same pathways. In addition, models trained or pre-trained with TCGA are better than models trained with GDSC alone. This again supports the usability of larger, real tumor data for revealing oncogenic pathway information with deep graphical models.

Table 5.1. Mutation genes with nearest neighbors from the same oncogenic signaling pathway.

DGM[a]	# of genes with top 1 nearest neighbor of the same pathway	# of genes with at least one of top 3 nearest neighbor of the same pathway
ResAE_1000_500_1000_0_TCGA_EN	4	14
ResAE_1000_500_1000_0_TCGA_DE	5	16
ResAE_1000_500_1000_0_TCGA_ALL	7	12
ResAE_1000_500_1000_0_GDSC_FT_EN	4	14
ResAE_1000_500_1000_0_GDSC_FT_DE	5	13
ResAE_1000_500_1000_0_GDSC_FT_ALL	6	15
ResAE_1000_500_1000_0_GDSC_ALL	1	8
ResVAE_1000_500_1000_0_TCGA_EN	19	30
ResVAE_1000_500_1000_0_TCGA_DE	10	24
ResVAE_1000_500_1000_0_TCGA_ALL	16	31
ResVAE_1000_500_1000_0_GDSC_FT_EN	20	30
ResVAE_1000_500_1000_0_GDSC_FT_DE	11	23
ResVAE_1000_500_1000_0_GDSC_FT_ALL	17	27
ResVAE_1000_500_1000_0_GDSC_ALL	5	14
ResVAE_1000_500_1000_0_TCGA_FI	**21**	**31**
ResVAE_1000_500_1000_0_GDSC_FT_FI	18	26
ResVAE_1000_500_1000_0_GDSC_FI	7	13

[a]Model nomenclature: ModelType + [ModelArchitecture] + L1Coef. + TrainData + MutationDataLayer. GDSC: model trained using GDSC data alone. GDSC_FT: model pre-trained with TCGA data then fine-tuned with GDSC data. TCGA: model trained using TCGA data alone. EN: mutation data added to all encoder hidden layers. DE: mutation data added to all decoder hidden layers. ALL: mutation data added to all hidden layers. FI: mutation data added to the first encoder hidden layer and last decoder hidden layer.

The model with the highest number of genes in Table 5.1 (21 for top 1 and 31 for top 3 nearest neighbors) is ResVAE_1000_500_1000_0_TCGA_FI, which is a ResVAE model trained using the TCGA data and have the transformed mutation data added to the first hidden layer in the encoder and the last hidden layer in the decoder with tied-weights. With representations learned with this model, many mutated genes had all their top 3 nearest neighbors from the same pathway (Figure 5.5). For example, the top 3 nearest neighbors of *BRAF* were *NRAS*, *NF1*, and *KRAS*, and the top 3 nearest neighbors of *MAP2K1* were *BRAF*, *NRAS*, and *HRAS*. All these genes are from the RTK/RAS pathway (light blue in Figure 5.5), where the products of *NRAS* and *KRAS* directly

interact with *BRAF*, and *BRAF* directly interacts with *MAP2K1*. Another closely bounded group of genes included *AKT1*, *PIK3CA*, *PTEN*, and *PIK3R1*. They were mutually the top 3 nearest neighbors to each other, except for *PIK3CA* for which the top 1 nearest neighbor was *TP53*. These four genes are from the PI3K/Akt pathway (yellow in Figure 5.5) and are directly interacting with each other. Therefore, even though the weight matrix of a ResVAE was randomly initialized and the mutated genes were fully connected to all hidden nodes in a layer, without incorporating any prior knowledge of gene products interaction topology, the weight matrix was able to learn the inner correlations among the mutations, which in turn reveal the functional associations between the mutation genes.

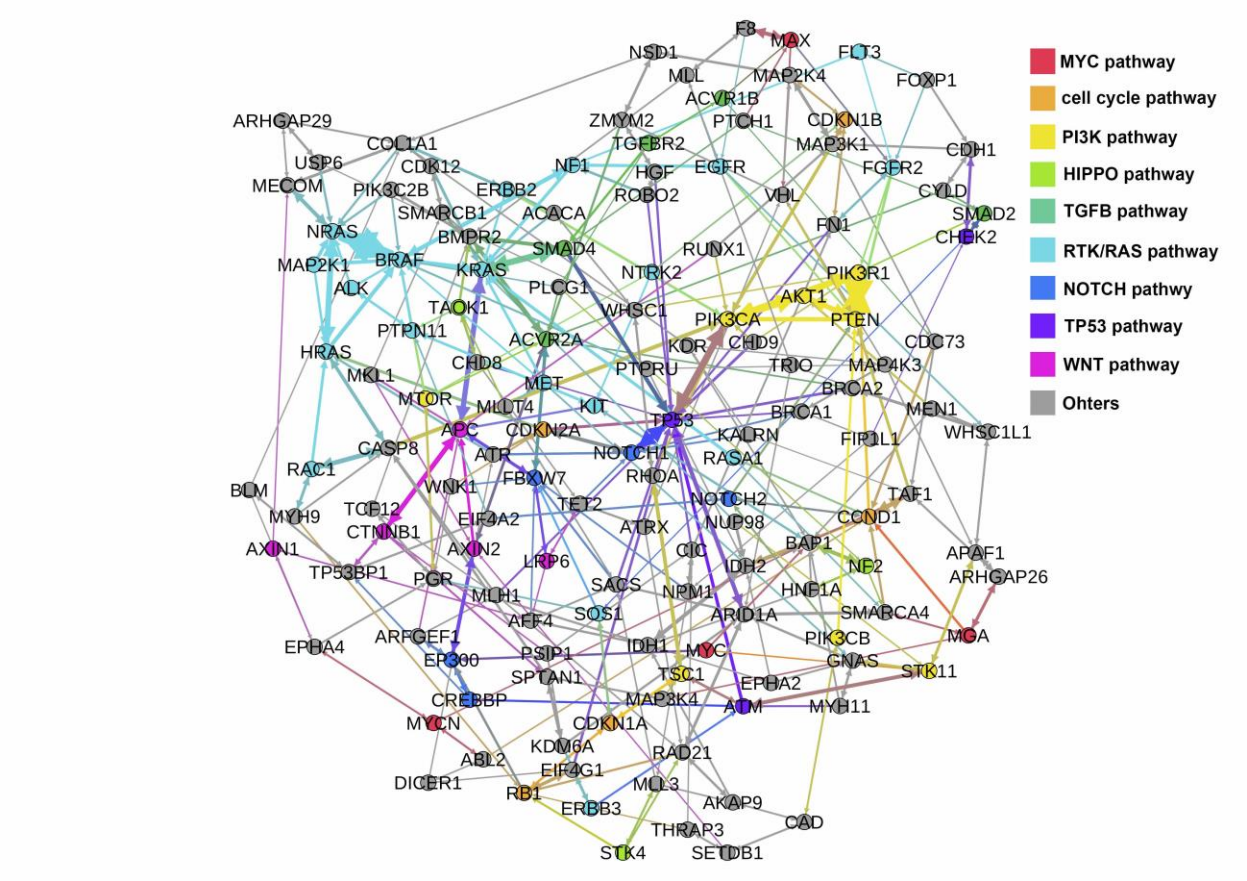


Figure 5.5. Mutation genes and their top 3 nearest neighbors identified based on DGM-learned representations.

We also tested other models without tied-weights, as well as using vectors from weight matrices for other hidden layers. In general, the number of mutated genes with pathway-related top nearest neighbors were lower in these cases. Introducing tied-weights reduces the number of parameters of a model and helps the model converge faster. These tied-weight models were later found to be more robust for generating latent representations for drug sensitivity prediction, which suggests the tied-weight as a more efficient architecture constraint for combining mutation data with expression data.

The effects of mutation data also differ across hidden layers, among which the effect to the first encoder hidden layer seems to the strongest. It was previously found that the first hidden layer of a deep learning model for learning expression data may correspond to transcription factors that directly control the expression of genes (Chen et al., 2016). Therefore, incorporating the mutation data to the first hidden layer could be more efficient to transmit the information than the other layers. This was supported by the model with the highest counts of genes mentioned above, ResVAE_1000_500_1000_0_TCGA_FI, which only added the transformed mutation data to the first hidden layer of encoder and symmetrically the last hidden layer of decoder.

5.3.4 Drug sensitivity prediction for cell lines

Based on the validation reconstruction loss and the AUC score of classic LR models, we selected 59 DGMs to generate latent representations for training ENLRs for drug sensitivity prediction and compare their performances. These included models representing different model types (AE-based and VAE-based), different architectures, and different training settings (Supplementary Table S5.3). For each DGM, we trained and selected an ENLR for each drug in the GDSC2 response data (Figure 5.2a), and computed the average AUC and F1 score across drugs to compare between DGMs (see Methods). We then tested the ENLRs on 12 CCLE drugs available in GDSC2 (Table 5.2). The validation and test performances of the 59 DGMs are shown in Supplementary Table S5.3. Figure 5.6 compares the performances of ENLRs trained based on different DGMs. Similar to what we have observed with LRs, ENLRs trained with VAE-based representations generally achieved higher AUC and F1 scores than AE-based representations (Figure 5.6 a and b). The TCGA pre-training step, on the other hand, even though it helped reduce the reconstruction loss for GDSC expression data, it did not help predict GDSC drug sensitivity

compared to DGMs trained using GDSC data alone (Figure 5.6 c and d). The outcomes for CCLE test drugs were more controversial, where DGMs trained or pre-trained with TCGA achieved higher AUC and F1 scores, which suggests these models are more generalizable to new data (Figure 5.6 c and d). Similarly, adding regularization did not help ENLRs in GDSC validation performance but improve CCLE test performance (Figure 5.6 e and f). This inconsistency between GDSC validation and CCLE test outcomes may partially due to the limited number of CCLE test drugs and may also suggest overfitting among GDSC-trained DGMs.

Table 5.2. Drug name correspondence between GDSC and CCLE.

GDSC name	Available GDSC dataset	CCLE name
Erlotinib	GDSC1/GDSC2	Erlotinib
Irinotecan	GDSC2	Irinotecan
Lapatinib	GDSC1/GDSC2	Lapatinib
Nilotinib	GDSC1/GDSC2	Nilotinib
Nutlin-3a (-)	GDSC1/GDSC2	Nutlin-3
PD0325901	GDSC1/GDSC2	PD-0325901
Palbociclib	GDSC1/GDSC2	PD-0332991
Crizotinib	GDSC1/GDSC2	PF2341066
PLX-4720	GDSC1/GDSC2	PLX4720
Paclitaxel	GDSC1/GDSC2	Paclitaxel
Sorafenib	GDSC1/GDSC2	Sorafenib
Topotecan	GDSC2	Topotecan
Tanespimycin	GDSC1	17-AAG
Saracatinib	GDSC1	AZD0530
Selumetinib	GDSC1	AZD6244
PHA-665752	GDSC1	PHA-665752
Panobinostat	GDSC1	Panobinostat
NVP-TAE684	GDSC1	TAE684

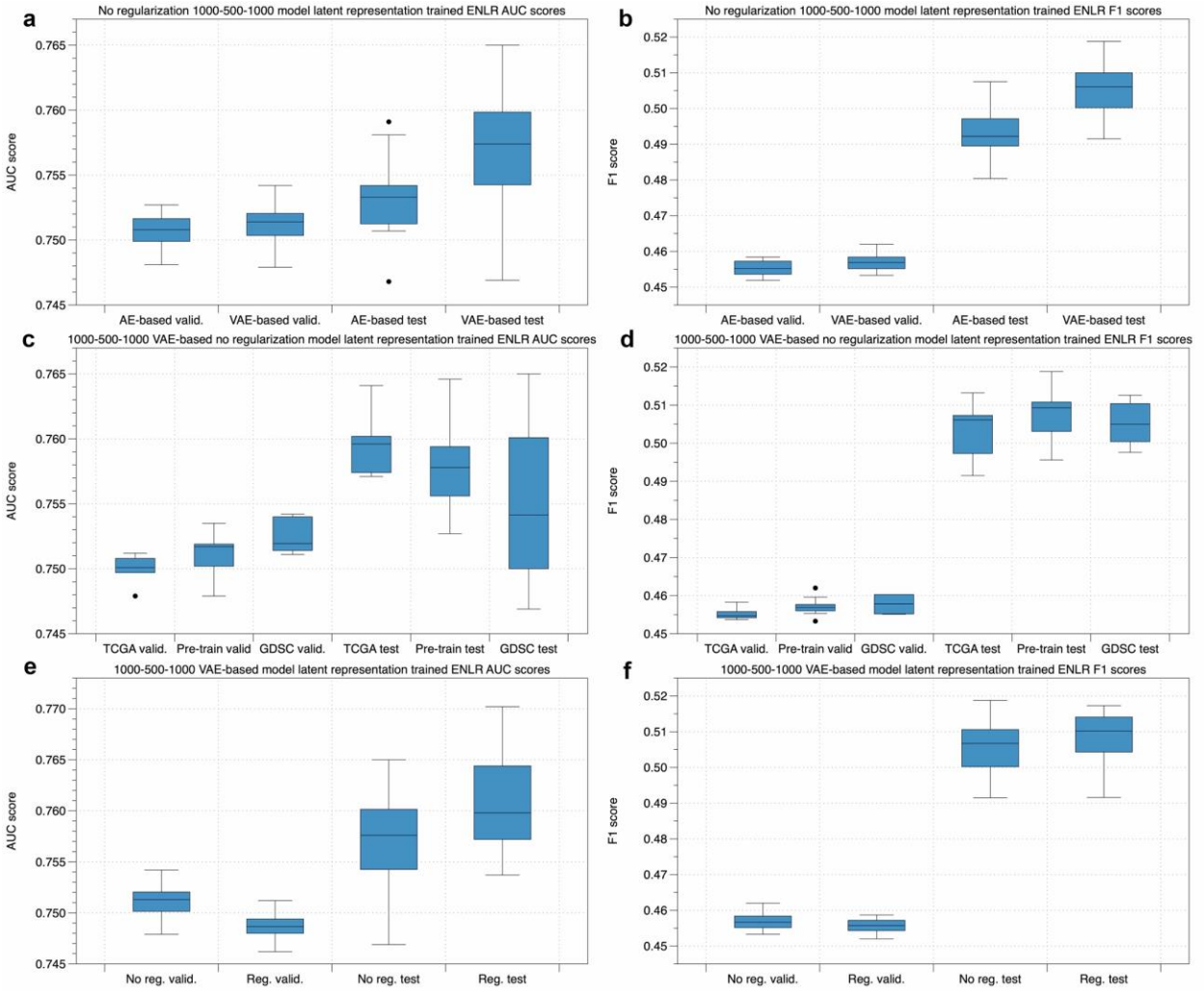


Figure 5.6. Performance comparison of ENLRs trained with latent representations from DGMs.

The above experiments were carried out using the GDSC2 drug response data as GDSC2 adopted a new type of assay to measure cell viability that is more robust than GDSC1. The number of drugs and cell lines tested in GDSC2, however, was much smaller compared to GDSC1. Therefore, we also tried training ENLRs on GDSC1 response data with latent representations from eight DGMs selected based on their validation performances on GDSC2. We tested the ENLRs for 16 CCLE drugs available in GDSC1 (Supplementary Table S5.4, Table 5.2). Compared to

GDSC2, both the validation and test scores were lower for GDSC1 ENLRs (Supplementary Table S5.4).

Table 5.3 summarizes the drug sensitivity prediction performance for GDSC2 and GDSC1-trained ENLRs. The number of drugs with validation/test AUC above a threshold for the best DGM or across all DGMs is shown in Table 5.3. We selected the best single DGM for GDSC validation and CCLE test according to the average AUC score across all drugs. The best DGM for both GDSC2 and GDSC1 validation is ResVAE_1000-500-1000_0_GDSC_ALI (average validation AUC 0.7542 for GDSC2 and 0.7348 for GDSC1), which is a ResVAE with mutation data independently transformed and added to all hidden layers. The best DGMs for CCLE test are ResVAE_1000-500-1000_50_GDSC_FT_ALL when training with GDSC2 (average test AUC 0.7702) and ResVAE_1000_500_1000_0_GDSC_FT_ALL when training with GDSC1 (average test AUC 0.6997). Both are ResVAEs with mutation data added to all hidden layers with tied-weight and pre-trained with TCGA data, except that ResVAE_1000-500-1000_50_GDSC_FT_ALL uses L1 regularization with a coefficient of 50 for all hidden layers. The involvement of tied-weight, pretraining and L1 regularization again supports that these measures helped prevent overfitting and enhanced the generalizability of GDSC-trained ResVAE on new datasets.

For the 10 CCLE drugs that are both available in GDSC2 and GDSC1 (Table 5.2), the average AUC is 0.7649 for GDSC2-trained models and 0.6546 for GDSC1-trained models. This suggests that GDSC-trained models can be transferred to CCLE effectively, where GDSC2 response data are better aligned with CCLE data.

Table 5.3. Drug sensitivity prediction performance summary.

Training dataset	Validation/test dataset (# of drugs)	Model for generating latent representations	# of drugs with latent representations achieving the best prediction performance	Average AUC	# of drugs with AUC > 0.7	# of drugs with AUC > 0.75	# of drugs with AUC > 0.8	# of drugs with AUC > 0.85	# of drugs with AUC > 0.9
GDSC2[a]	GDSC2 (158)[b]	ResVAE_1000-500-1000_0_GDSC_ALI	110	0.7542	117	82	45	17	3
		Across all models	139	0.7813	137	106	61	31	4
GDSC2	CCLE (12)[c]	ResVAE_1000-500-1000_50_GDSC_FT_ALL	7	0.7702	10	9	5	0	0
		Across all models	10	0.7869	11	9	5	1	0
GDSC1	GDSC1 (320)	ResVAE_1000-500-1000_0_GDSC_ALI	241	0.7348	212	129	68	15	2
		Across all models	273	0.7516	248	152	85	23	3
GDSC1	CCLE (16)[c]	ResVAE_1000_500_1000_0_GDSC_FT_ALL	12	0.6997	9	5	1	0	0
		Across all models	16	0.7164	9	6	2	0	0

[a]For GDSC2 response data, we only included drugs that had been tested over 100 cell lines for training robustness.

[b]For validation, the reported performance is for 25-fold cross validation.

[c]When testing on the CCLE data set we only made predictions for drugs that were available in the training dataset.

We also combined all available DGMs for GDSC2 and GDSC1 respectively to retrieve the overall best ENLR model for each drug (Table 5.3). The highest average AUC we achieved was 0.7813 for GDSC2 (158 drugs), 0.7516 for GDSC1 (320 drugs), and 0.7869 for CCLE (12 drugs). For GDSC2, when all available models were taken into consideration, 137 out of 158 drugs got an AUC above 0.7 (86.71%); this proportion is much higher than a previously best classification model (83 out of 138 or 60.14%) evaluated with a similar metric (Jang et al., 2014). We also counted the number of drugs for which the best performance was achieved with latent representations rather than the raw expression profiles. For the majority of drugs (139 out of 158 for GDSC2 and 273 out of 320 for GDSC1), the best ENLRs were trained with latent representations, which proves that DGMs learned to condense information from the raw input expression data.

We also compared our best GDSC1 drug sensitivity prediction model, ResVAE_1000-500-1000_0_GDSC_ALI + ENLR, with four other DNN based approaches. The first one is the previously state-of-art model, DeepDR, for drug sensitivity prediction (Zeng et al., 2019). The second one is directly training a DNN in a supervised manner for sensitivity prediction. The DNN is composed of 2-4 hidden layers with 20-2500 hidden nodes per layer, and the architecture is optimized for each drug. The third approach is replacing ResVAE with an ordinary AutoEncoder (AE), following the same setups as in (M. Q. Ding et al., 2018), and the last one is replacing ResVAE with and an ordinary VAE of the same backbone architecture. The average 25-fold cross-validation AUC for the 320 GDSC1 drugs are 0.6695 for DeepDR, 0.7056 for DNN, 0.7250 for AE-ENLR, and 0.7319 for VAE-ENLR, all lower than the best ResVAE-ENLR model (Figure 5.7).

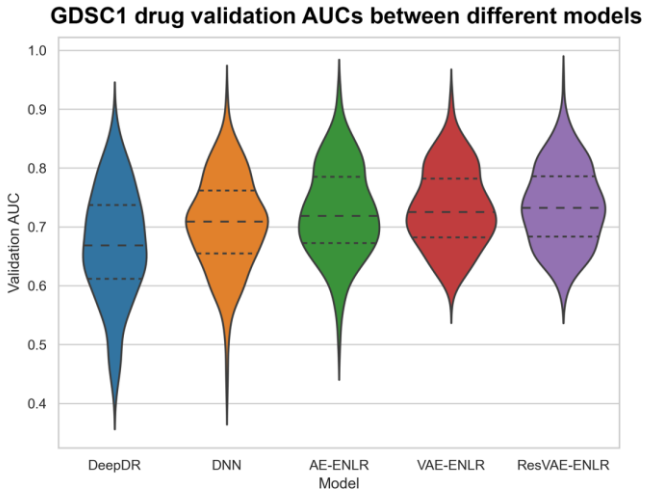


Figure 5.7. Violin plots of GDSC1 drug 25-fold cross-validation AUCs of different drug sensitivity prediction models.

Since for all validation and test datasets, the single models that achieved the highest average AUC scores are ResVAE models (Table 5.3), in the following model analyses, we only focus on these ResVAEs for latent representation generations.

5.3.5 Drug specific cell line latent representations cluster cell lines into groups of distinct response rates

ENLRs use the elastic net regularization to tease out the features that are most correlated to the targets being predicted. Informative features will be assigned with a non-zero coefficient after training, while others are simply ignored in the final predictive model. In this way, even though we used the same type of cell line representations to train ENLRs for all drugs, different drugs may turn out to use different sets of genes or hidden nodes for sensitivity predictions. These

drug-specific feature sets in-turn generate drug-specific cell line representations that can be used to reveal how the genes or hidden nodes are correlated with the drug response.

With this assumption, we extracted GDSC cell line representations, both raw input expression profiles and latent representations from three hidden layers of ResVAE_1000-500-1000_0_GDSC_ALI, to visualize the correlations between the ENLR selected covariates and drug response outcomes. ResVAE_1000-500-1000_0_GDSC_ALI was used as this is the best single DGM for drug sensitivity predictions for both GDSC1 and GDSC2 (Table 5.3). For each drug, we selected the ENLR trained with the representation type that achieved the highest cross-validation AUC and then extracted the significant covariates for this representation type. We constructed the drug-specific representations of cell lines by only including the significant covariates, and clustered cell lines into groups accordingly to see if they are aligned with the responder group.

Figure 5.8 and Figure 5.9 show the top 4 drugs, from the aspect of cross-validation AUC, of GDSC1 and GDSC2 as examples. Different drugs achieved the best AUC with different representation types (raw expression or different hidden layers). Yet, in most cases, cell lines are clustered such that responders tend to get close to each other. For example, Lapatinib from GDSC 1 has the responding cell lines gathering at the top cluster (Figure 5.8a). Venotoclax from GDSC1 also has most responders in the same cluster at the top, with particularly high and low values of hidden nodes in the middle columns that present a reversed pattern compared to non-responding cell lines (Figure 5.8b). A similar trend can also be observed in VNLG-124 in GDSC1 (Figure 5.8c) and Venotoclax in GDSC2 (Figure 5.9c). In these cases, most responders are in the same cluster that split from the majority of the other non-responding cell lines at a high level in the dendrogram, which presents distinct representation patterns, and are not specifically correlated with tissue type (Figure 5.8 and Figure 5.9).

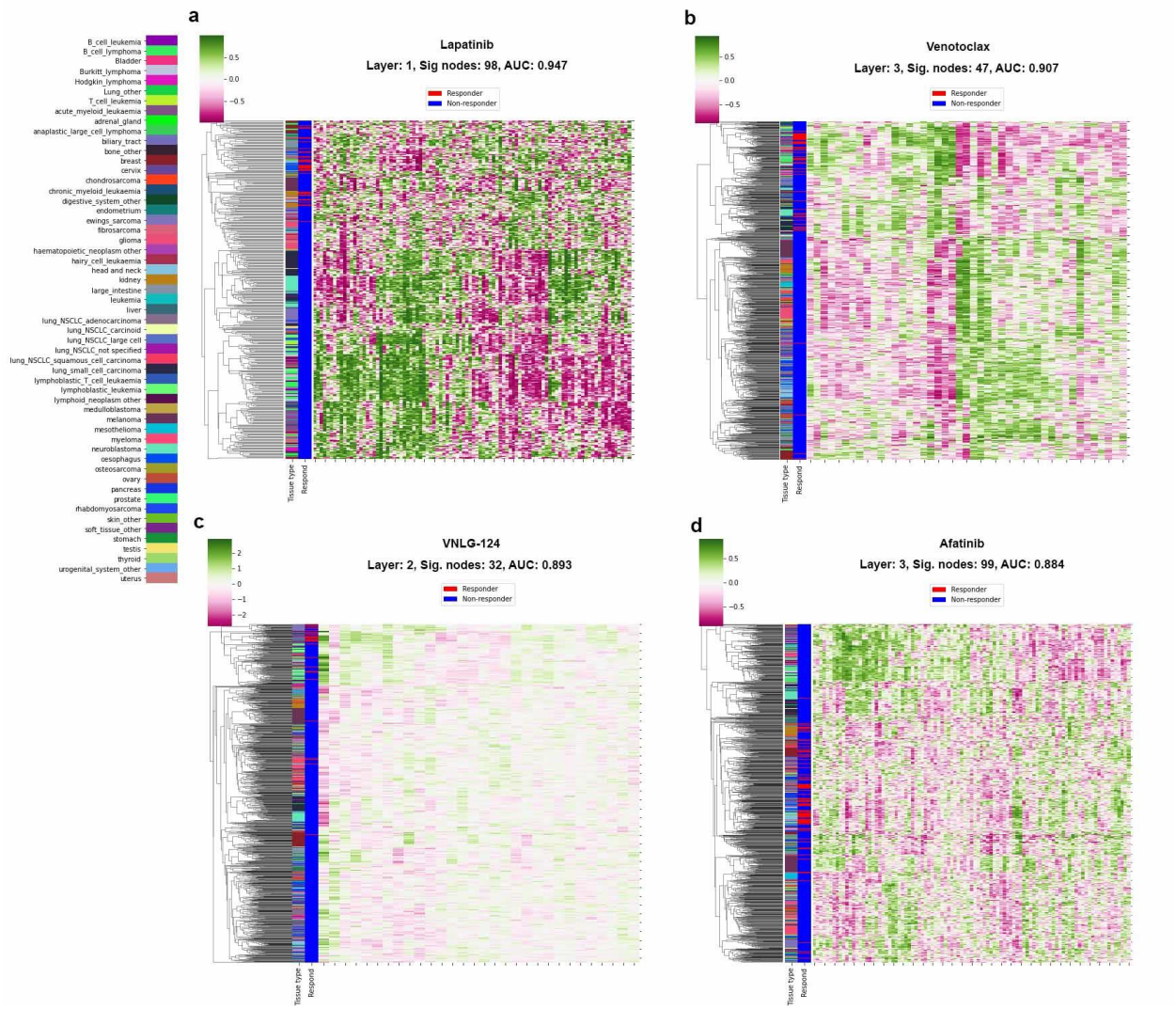


Figure 5.8. Drug-specific cell line representations and cell line clustering for top 4 drugs from GDSC1-trained ENLRs.

The top 4 drugs are the drugs with the highest AUC from GDSC1-trained ENLRs. Each cell line is represented by the subset of genes or hidden nodes that received a non-zero coefficient from the ENLR trained with the representation type that gave the best AUC across all representation types (row expression and latent representations)

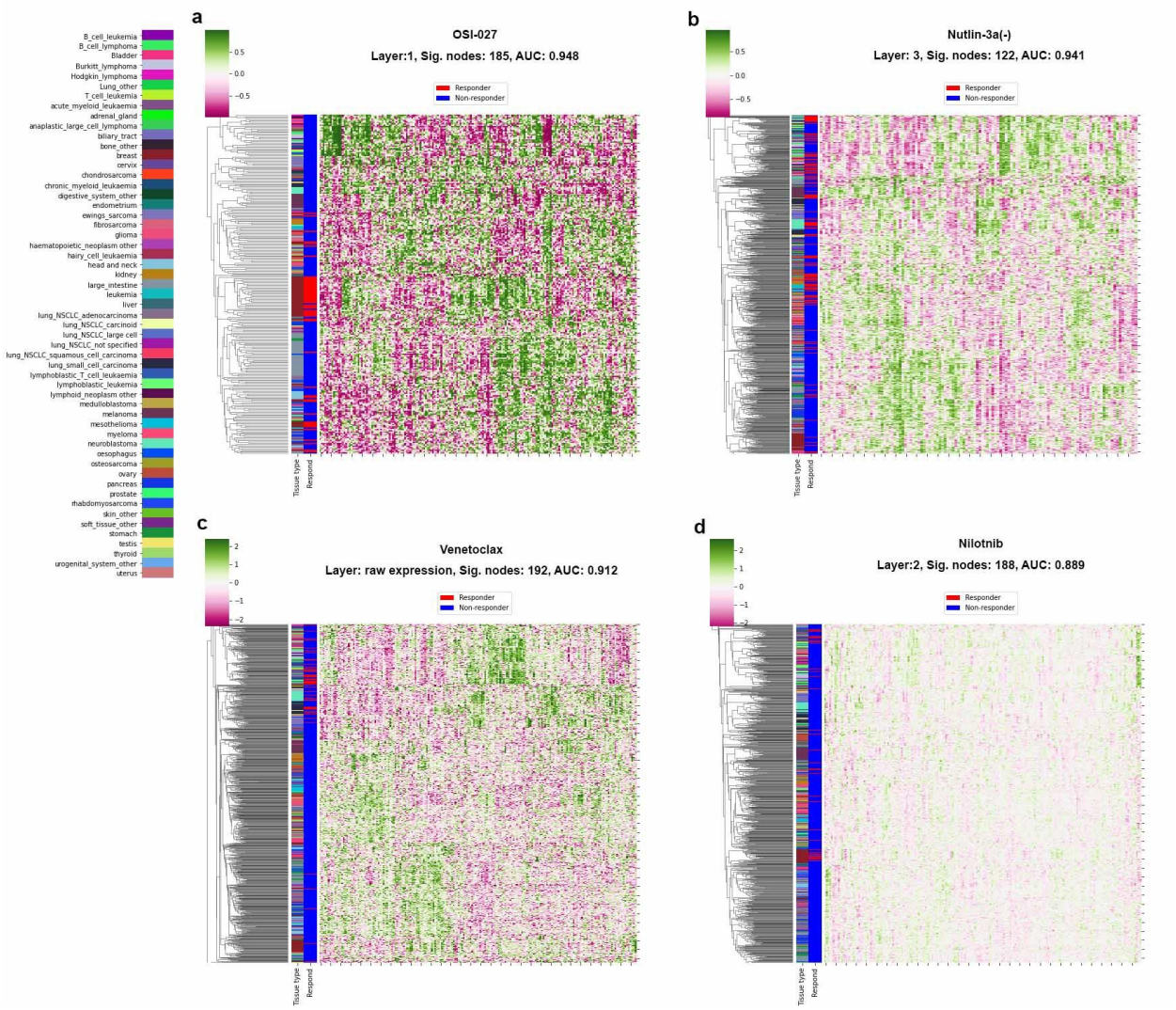


Figure 5.9. Drug-specific cell line representations and cell line clustering for top 4 drugs from GDSC2-trained ENLRs.

The top 4 drugs are the drugs with the highest AUC from GDSC2-trained ENLRs. Each cell line is represented by the subset of genes or hidden nodes that received a non-zero coefficient from the ENLR trained with the representation type that gave the best AUC across all representation types (row expression and latent representations)

In other cases, Afatinib in GDSC1 (Figure 5.8d) and Nutlin-3a(-) in GDSC2 (Figure 5.9b) specifically, subgroups exist in non-responding cell lines or responding cell lines. For Afatinib,

there are two major subgroups of non-responders, one on the top with particularly high values in hidden nodes clustered on the left and low values in hidden nodes clustered on the right; one on the bottom with low values in the hidden nodes on the left (Figure 5.8d). Despite that they are not responding to Afatinib, the distinct hidden node patterns of the two groups suggest that molecular subtypes exist in these non-responders, and they may potentially present different sensitivities to other drugs. For Nutlin-3a(-), an outlier group was clustered on the top. Even though this group has a similar responder proportion as the other few clusters below, it has a distinct latent representation pattern with significantly low values of hidden nodes clustered on the left and high values of hidden nodes in the middle. As a result, this responder-enriched group can be more easily distinguished from non-responders. On the other hand, the other responding cell lines are less discriminative from the aspect of latent representations compared to non-responding cell lines, thus it becomes harder to attribute their sensitivities to the hidden nodes.

In the above cases, the responding cell lines are not specifically correlated with tissue type. This is not the case for OSI-027 in GDSC2 (Figure 5.9a), where most responders are breast cancer cell lines (deep red). OSI-027 is a small molecule that targets mTORC1 and mTORC2 (Mateo et al., 2016) and has been used in clinical trials for the treatment of advanced solid tumors or lymphoma. It is not specifically known to work on breast cancer, but the heatmap we show here indicates that breast cancer cell lines present a unique latent representation pattern and are more likely to respond to this drug compared to other tissue types. Similarly, Niltonib was mainly used to treat chronic myelogenous leukemia, but it also presents a higher responding rate for the breast cell lines (Figure 5.9d); these cell lines also present a distinct latent representation pattern for them to be clustered together. These tissue-specific observations, though not directly resulted from the use of ResVAE, are still valuable for guiding drug usage repurposing, thus we point them out here.

5.3.6 Pathway specific cell line latent representations correlate with pathway targeting drug sensitivity

Instead of examining each drug independently, we moved on to the groups of drugs that target the same signaling pathways to see if the responding cell lines to these drugs share a similar latent representation profile. This can, in turn, help identify hidden nodes that are specific indicators of the state of signaling pathways.

In this case, we examined drugs in GDSC1 as GDSC1 contains more drugs than GDSC2 as well as tested them on more cell lines. From the top 50 drugs in GDSC1 we predicted using ResVAE_1000-500-1000_0_GDSC_ALI and ENLR, we identified drugs that targeting four cancer-driving signaling pathways, including EGFR pathway, ERK/MAPK pathway, PI3K/mTOR pathway, and RTK pathway. For drugs that target the same pathway, we combined the significant covariates from the ENLRs trained on the first hidden layer representations and generated pathway-specific representations of cell lines. The first hidden layer was chosen as this is often the most informative layer for drug sensitivity prediction judging from both validation AUCs and representation heatmaps. Figure 5.10 visualizes the representations and the clustering of cell lines. As we have expected, cell lines that are sensitive to a drug are more likely to respond to other drugs with the same target. For EGFR, PI3K/mTOR, and RTK pathways, responding cell lines tend to be clustered together according to the latent representations. For PI3K/mTOR and RTK pathways in particular, whether a cell line is responding to the drugs is strongly correlated with a few hidden nodes (Figure 5.10c and d). Specifically, having high values in hidden nodes 730, 614, 947, and/or 728 and low values in 76, 787, 596, and/or 304 would confidently indicate a cell line as sensitive to drugs targeting the PI3K/mTOR pathway (Figure 5.10c). Having high values in nodes 944, 444, 425, 251, 558, and/or 639 and low values in nodes 371, 274, 787, 247, 508, and/or

686 would likely make a cell line responding to drugs targeting the RTK pathway (Figure 5.10d). These hidden nodes capture the pathway status information from the expression profile and can serve as sensitivity indicators for judging whether a new sample may respond to the pathway targeting drugs in the future.

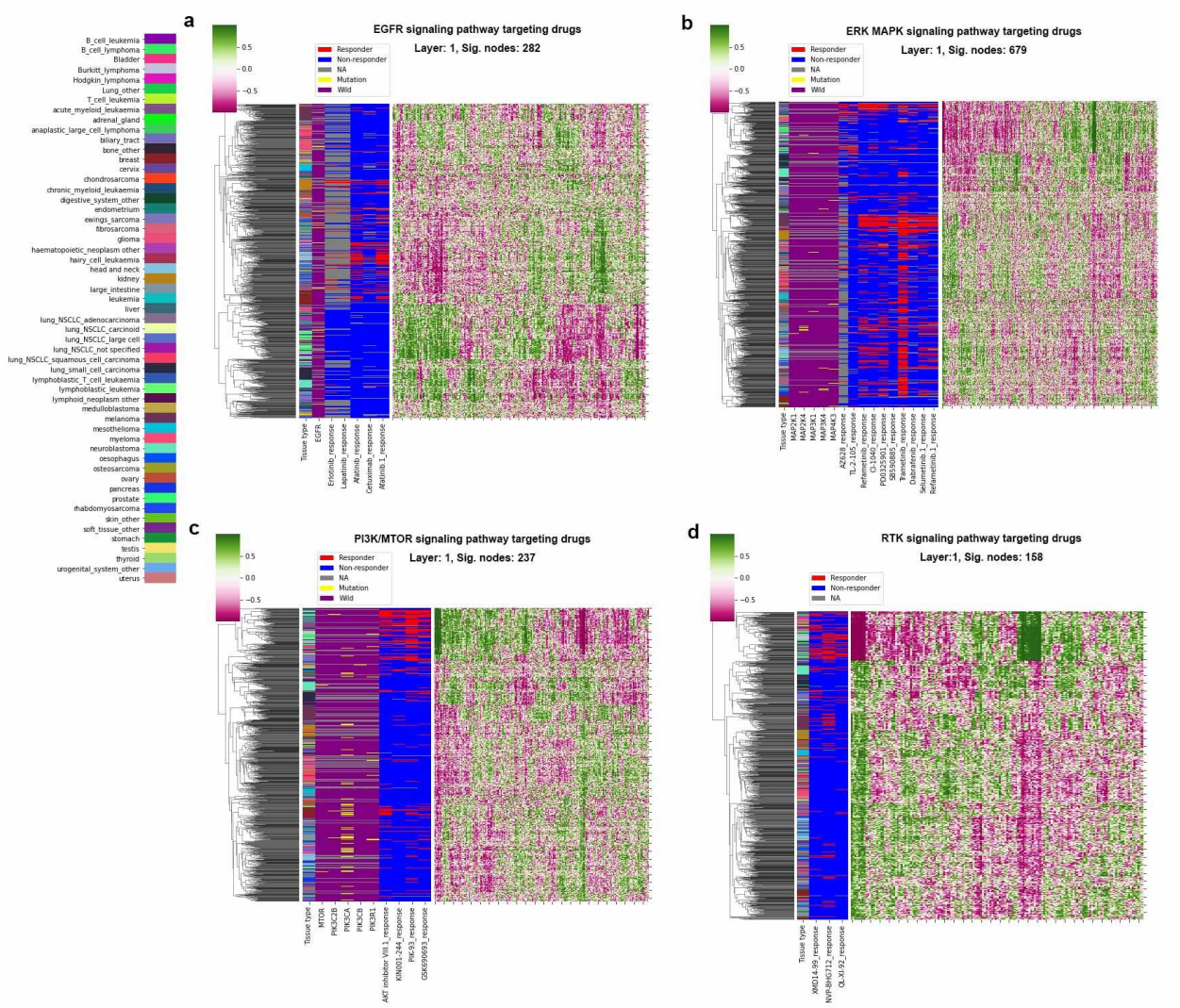


Figure 5.10. Signaling pathway targeting drugs latent representations and response clustering for GDSC1.

Such dense clustering of responders and strong indicators of sensitivity, however, does not exist with the ERK MAPK targeting drug representations (Figure 5.10b), where responders

distribute across all cell lines. The only subgroup that is more enriched in responders is aligned well with the tissue type melanoma (brown, middle rows in Figure 5.10b). The ERK/MAPK signaling pathway is known to be essential in the progress of melanoma skin cancer (Savoia, Fava, Casoni, & Cremona, 2019), therefore, melanoma cell lines are naturally more sensitive to the pathway targeting drugs. Despite the less clear separation between responding groups, distinct inherent representation patterns still exist in different clusters of cell lines that mix various tissue types. Therefore, tissue type or cell of origin is not the only factor that affects the expression profile and determines a tumor's drug sensitivity. More fundamental disease mechanism effects, as revealed by the latent representation, provide crucial information towards whether a tumor is responding or not to a drug.

Together with the responding outcomes and tissue types, we also show the mutation state of genes in the corresponding signaling pathway in Figure 5.10. These include *EGFR* for EGFR signaling pathway, *MAP2K1*, *MAP2K4*, *MAP3K1*, *MAP3K4*, and *MAP4K3* for ERK/MAPK signaling pathway, and *MTOR*, *PIK3C2B*, *PIK3CA*, *PIK3CB*, *PIK3R1* for PI3K/mTOR. Despite the direct participation of these genes in the signaling pathways, the gene mutation states are not strongly correlated with the sensitivities of cell lines to the pathway targeting drugs. For the PI3K/mTOR pathway in particular (Figure 5.10c), the PIK3CA mutation is even more frequent among non-responding cell lines than responding cell lines. When using the mutation states as markers and predicting a cell line as sensitive to the drugs if it carries at least one of the related mutations, the average positive prediction value (PPV) is 0.2062 for the five EGFR pathway targeting drugs, 0.2125 for the ten ERK/MAPK pathway targeting drugs, and 0.1744 for the four PI3K/mTOR pathway targeting drugs. The average false omission rate (FOR) is 0.1037 for EGFR pathway, 0.2095 for ERK/MAPK pathway, and 0.1508 for PI3K/mTOR pathway. The low PPVs

and relatively high FOR suggests that mutations may not be the best markers for guiding drug assignment for cancer patients, despite its being overwhelmingly adopted in clinic. On the other hand, the drug-specific latent representations learned by our models are more informative towards drug responses, which can serve as better indicators for treatment selection.

Observing the descent alignment of response profiles of drugs targeting the same pathway, as shown in the sidebars of Figure 5.10 a, c, and d, it may be tempting to train a single ENLR for each signaling pathway by combining data of drugs that target the pathway. The increment in the training data should improve the generalizability of the resulting ENLR. The important assumption underlying is that drugs that share a similar targeted process should lead to a similar sensitivity outcome to the same set of cell lines. This assumption, however, does not always hold. Figure 5.11 visualizes the pairwise Pearson correlations between 320 drugs in GDSC1. A correlation was computed using the response profiles of two drugs across all cell lines that the two drugs were both tested on. The drugs were then clustered based on the pairwise correlations (1- *Pearson correlation* as distance) and the targeted pathways of interest were highlighted. Except for the EGFR pathway targeting drugs that are highly pairwise correlated and hence clustered together (yellow in Figure 5.11), the other drugs generally do not share a similar response profile for them to be grouped. This suggests different mechanisms of action (or off-target effects) exist among drugs that are supposed to target the same cellular process.

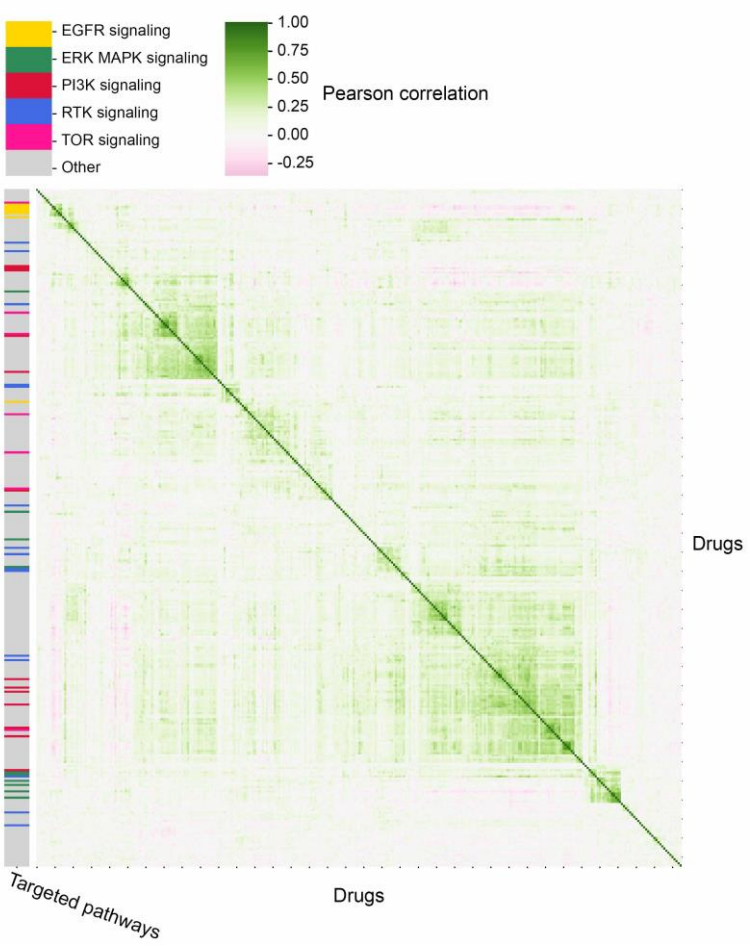


Figure 5.11. Pairwise Pearson correlation between GDSC1 drugs.

Correlations were computed using the response profiles of each pair of drugs over cell lines that both drugs were tested on. Drugs are clustered based on the correlations.

Nonetheless, we gave it a try to train an ENLR for EGFR pathway, ERK/MAPK pathway, PI3K/mTOR pathway, and RTK pathway separately by combining cell line representations and labels of drugs that target the pathway. Since a cell line could be tested on multiple drugs, when combining drugs, we randomly jittered the representations a little to avoid exact repeats in the training data. We then computed the 25-fold validation AUC for each drug by training and applying the pathway ENLR iteratively to see if the training data augmentation improves the performance for member drugs. The outcomes, however, were generally undesirable, thus are not

shown here. Though a few drugs received a slightly higher AUC and hence benefited from the increment of training data, the others experienced a significant drop in AUC compared to using the drug-wise ENLRs.

The weak correlations between drugs targeting the same process, as shown in Figure 5.11, suggest that the inherent diversity in the drug effects cannot be simply ignored, and combining their data to train a single model is statistically inappropriate. One explanation is that drugs target the same pathway by targeting different members on the pathway. A cell line that responds to a drug due to a misfunction of a member protein may not respond to another drug that targets the same pathway but through a different member upstream or downstream that cannot eliminate the impact of the problem protein. Drug off-target effects and other side-effects may also contribute to the inconsistent sensitivity outcomes for the same cell line. Consequently, we did not move forward training pathway-level predictive models and stick with drug-wise models for our following analyses.

5.3.7 Transfer cell line sensitivity prediction models to cancer patients

So far we have shown that DGM latent representations together with ENLR are very effective in predicting drug sensitivities for cell lines. We next examined whether these in vitro models can be applied to real patients.

We first selected lung cancer patients (LUAD + LUSC) from TCGA with drug usage information as a test patient group. Since drug response data are not available for many of these patients, in order to determine whether the predictions correctly reflect the drug response status of patients, we compared the survival outcomes of patients that were predicted as sensitive vs. resistant by our models. Our assumption is that a patient that is predicted as sensitive to a drug

prescribed to the patient is more likely to survive longer compared to a patient that also took the drug but is predicted as resistant. We only included patients that took chemotherapy drugs as adjuvant therapy to further rule out other confounders that may impact the survival outcome other than drug usage.

For LUAD and LUSC specifically, we focused on four drugs, Cisplatin, Pemetrexed, Paclitaxel, and Vinorelbine, that are commonly used in the treatment protocol for lung cancer and have been tested in GDSC1. This resulted in a dataset with 64 adjuvant LUAD and LUSC patients, where 42 were given Cisplatin, 19 were given Pemetrexed, 18 were given Paclitaxel, and 10 were given Vinorelbine (some patients were prescribed multiple drugs). We used our GDSC1-trained ENLRs to generate the predictions and classify patients into three groups, sensitive, intermediate, and resistant (see Methods and Figure 5.2b). We compared the ENLRs trained with latent representations from different DGMs to see if any DGMs would result in a division of patients with significantly different survival curves. It turned out that for the eight DGMs that we trained GDSC1-based ENLRs (Supplementary Table S5.4), the predicted sensitivity probabilities for these four drugs were always negatively correlated with the hazard rate of survival if running a Cox Regression. A negative correlation with the hazard rate means the higher the probability a patient is predicted as sensitive to the drug, the higher the probability the patient survives longer. Thus, our drug sensitivity predictions correctly reflect the effectiveness of taking drugs to improve survival outcomes.

Among the models, predictions generated with ENLRs trained with latent representations from ResVAE_1000_500_1000_0_GDSC_FI resulted in the most significant survival division of the patients, with a multi-variate log-rank test p-value of 0.0298. As shown in Figure 5.12a, the sensitive group survived significantly longer than the intermediate group, followed by the resistant

group. If only the two most significant drugs, Cisplatin and Pemetrexed were included, the patients were divided into groups as shown in Figure 5.12b, with a multi-variate log-rank test p-value of 0.0464. We repeated the whole experiments multiple times by independently training ResVAE of the same architecture with different random initializations and training downstream ENLRs to regenerate predictions. The patient division outcome may be slightly different across runs, but the trend of significantly different survival curves for the three patient groups remained the same. Therefore, our ResVAE latent representation trained ENLRs can successfully discriminate lung cancer patients into different drug response groups that are correlated with survival outcomes.

ResVAE_1000_500_1000_0_GDSC_FI is a ResVAE model trained using only GDSC data with mutation data adding to the first and last hidden layers. Its outstanding performance in this transferring learning supports that the information of the mutation data is more efficiently incorporated into the model through the hidden layer that is the closest to the expression data; the same conclusion as we drew in analyzing mutation representations above.

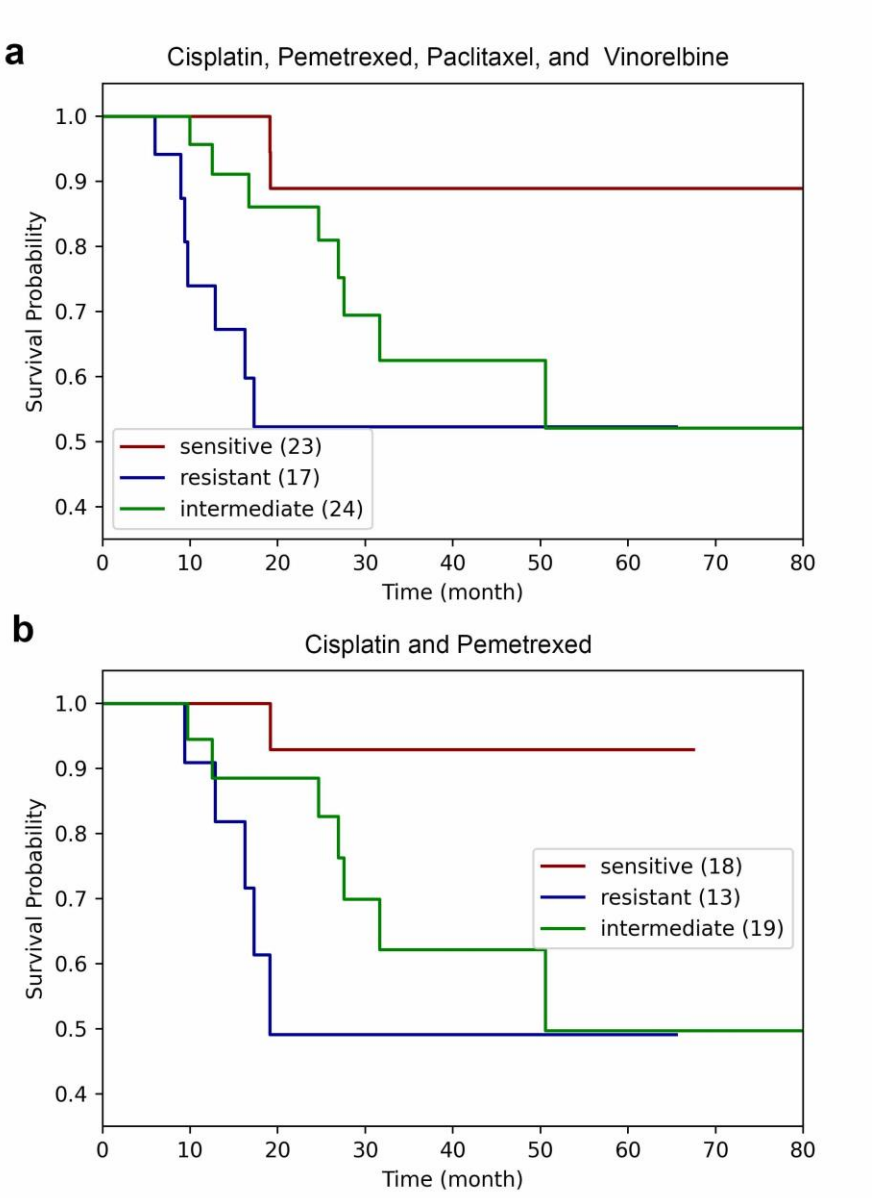


Figure 5.12. Kaplan-Meier curves of three drug response groups of lung cancer patients that took chemotherapy drugs for adjuvant therapy.

(a). Patient groups divided by predictions of Cisplatin, Pemetrexed, Paclitaxel, and Vinorelbine. (b). Patient groups divided by predictions of Cisplatin and Pemetrexed.

We moved on to apply our ResVAE_1000_500_1000_0_GDSC_FI and ENLRs on a PANCAN TCGA dataset that contains drug usage and overall treatment response information of

880 patients from 15 cancer types (see Methods). In total, 31 drugs were prescribed for these patients that were also tested in GDSC1. For each patient, we make sensitivity predictions for all the drugs he took that we also had a prediction model. A patient was then represented by two measures, one is the highest sensitivity probability (max prob.) among his drug predictions, which represents the probability that the most probable drug had worked; one is the noisy-or sensitivity probability (noisy-or prob.) calculated from his predictions, which represents the probability that at least one drug had worked if all drugs worked independently. A patient was then marked as sensitive if the measure, max prob. or noisy-or prob, is above 0.5, and resistant if equal or below 0.5. We compared the survival curves between the sensitive and resistant patient groups as shown in Figure 5.13. Patients that were predicted as sensitive to their treatments survived significantly longer than patients predicted as resistant, with a log-rank test p-value 0.0020 for max prob. and 0.0207 for noisy-or porb., respectively. This again demonstrates the utility of our models on real patients.

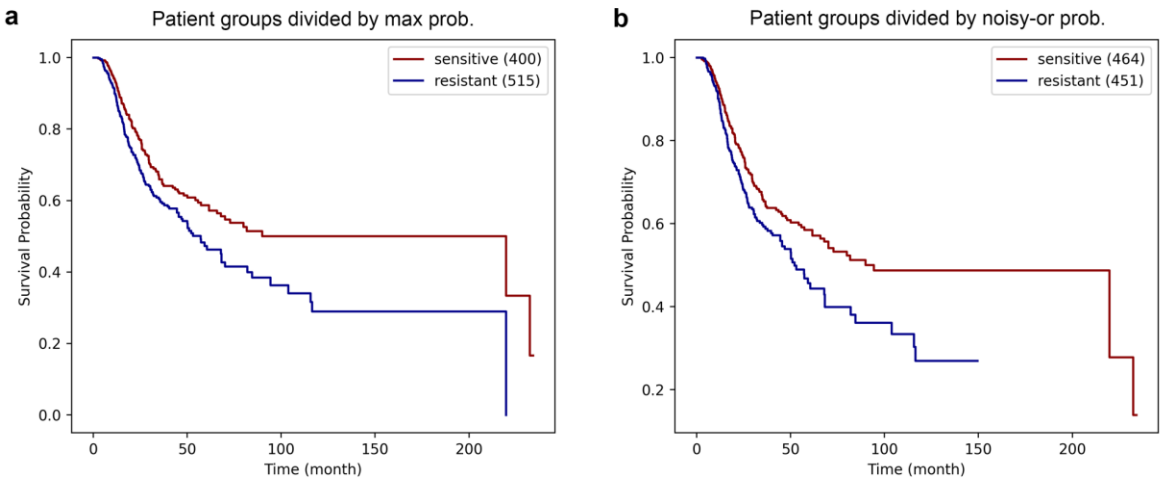


Figure 5.13. Kaplan-Meier curves of TCGA patient groups divided by max prob. vs. noisy-or prob.

We also computed the AUC score between the prediction probabilities and the clinical response outcomes provided by the dataset. The AUC scores are 0.5478 for the highest prob. and 0.5451 for the noisy-or prob. The AUCs are not themselves very impressive for three reasons. First, among the 152 drugs taken by the patients, only 31 were tested in GDSC for which we can make predictions. Patients that have a positive disease outcome ('Complete Response' or 'Partial Response', see Methods) may respond to other drugs they took that we have no corresponding models. As a result, our models are incapable of predicting these patients as responders. Second, the overall responding rate of these TCGA patients, 66.45%, is much higher than the average GDSC1 drug sensitivity rate, 23.78%. In other words, the label distribution of these patients is not statistically compatible with the GDSC1 data we used to train the ENLRs, which underlies the hardness for transferring the models for this specific dataset. Indeed, a higher proportion of patients were predicted as non-responders (resistant) than should be (Table 5.4). Nonetheless, the true positive rate (69.83%) is significantly higher than the false-negative rate (62.97%) and the contingency table passes the Chi-square independency test with a p-value of 0.0335. In other words, a patient predicted as sensitive to his treatment is more likely to be a real responder than a patient predicted as resistant.

Table 5.4. Contingency table of TCGA patients true response state and predicted response state from noisy-or probabilities.

Number of patients	Predicted responder	Predicted non-responder	Sum
True responder	324	284	608
True non-responder	140	167	307
Sum	464	451	915[a]

[a]The number is larger than the number of patients 880 as some patients took multiple therapeutic regimens through the treatment and each resulted in a different outcome. We made predictions for each of such cases.

The third reason that the overall performance over TCGA patients is less impressive is due to the mixing of cancer types. Patients of different cancer types have very different survival expectations due to the nature of the tumors. They also underwent different treatment protocols and were given different drugs, for some of which our models generate more accurate predictions. Indeed, after we examined each major cancer types independently, we noticed that our models work better for certain cancer types, particularly for LUSC. Our predictions divided LUSC patients into survival-related groups as shown in Figure 5.14, which agrees with our analyses of the lung cancer patient set above. In fact, the patient groups we identified are more correlated with the survival outcomes compared with the clinical response label, where the log-rank test p-value is 0.1307 for clinical labeled groups and 0.0024 for our prediction-divided groups.

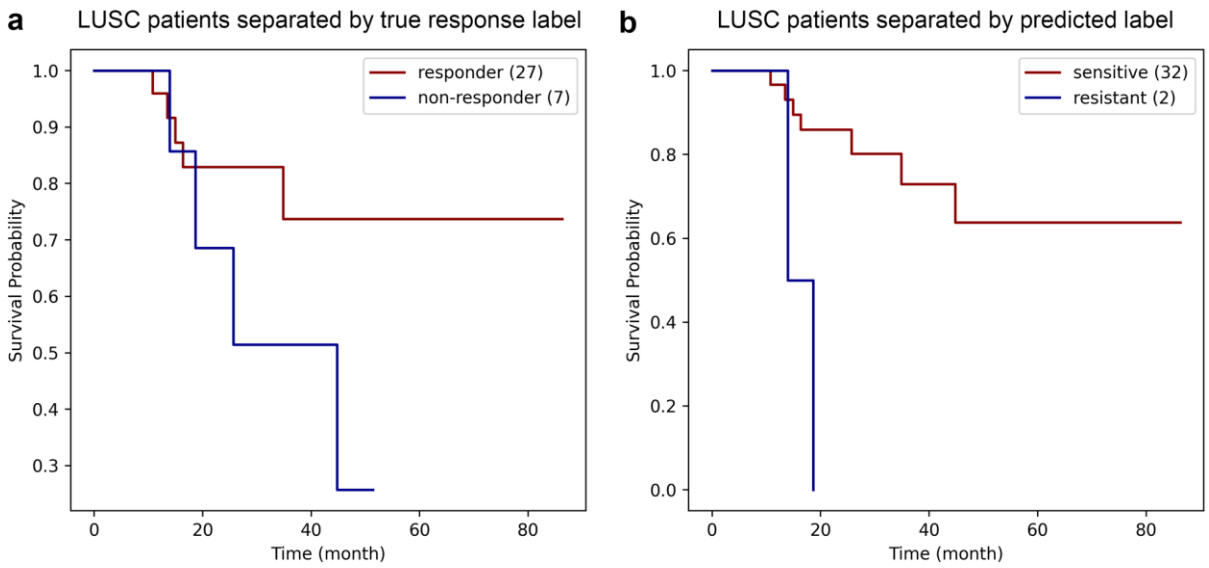


Figure 5.14. Kaplan-Meier curves of LUSC patient groups divided by true response outcome labels vs. noisy- or predicted response labels.

5.4 Discussion

In this chapter, we systematically examined the potential of DGMs in learning latent representations from expression data for drug sensitivity prediction. We compared six different types of DGMs, including the ordinary AE and VAE, and four extension models, ResAE, ResVAE, RIAE, and RIVAE. We incorporated mutation data into the extension models to utilize the causal relationships between gene mutations and expressions to enhance feature learning from expression data. We showed that adding the mutation information significantly improved both the representation learning and drug sensitivity prediction performances, where ResVAEs outperformed the other DGMs on all tasks we measured.

The most efficient way of incorporating mutation data into a ResVAE for reconstructing expression data is by adding the mutation information into the first and last hidden layers of the model. This is supported by the pathway-informative mutation gene representations learned from ResVAE_1000_500_1000_0_TCGA_FI (the suffix "_FI" indicates the mutation data is added to the first and last hidden layers, same below), the drug-response-related cell line representations constructed from the first hidden layer of ResVAE_1000-500-1000_0_GDSC_ALI (Figure 5.8, Figure 5.9, and Figure 5.10), and the best model, ResVAE_1000_500_1000_0_GDSC_FI, for making predictions for real patients. The first and last hidden layers are directly connected with the input and output expression data layers, which may represent the effects of transcript factors or signaling pathways in real cells (Chen et al., 2016). As a result, the gene expression regulatory information from mutation data is more efficiently transmitted to the DGM by interacting with these layers.

The pathway-informative mutation gene representations and the capability of latent representations for clustering cell lines into groups of distinct patterns also suggest that ResVAE, though as a deep neural network model, is not simply a black-box. Although no pathway or disease mechanism information was directly incorporated into the ResVAE, the mutation gene representations learned from the model are more similar if the genes are members of the same signaling pathway; the change of pathway status in an input sample is also strongly correlated to the value of a few hidden nodes on the first hidden layer (Figure 5.10). Even though a clear mapping between biological functional modules and a ResVAE model architecture is not yet established, these observations provide ResVAE with a moderate level of interpretability, which is not available with a classic DGM like AutoEncoder.

In the meantime, the latent representations learned from ResVAE turned out to be better aligned with drug sensitivities than the mutation state of drug-targeting-pathway genes, despite that mutations have been commonly used as markers for guiding drug assignment. Both the model interpretability and the stronger drug response indicative information equip ResVAE with a high potential for clinic use.

We used the latent representations of cell line expression data generated by DGMs to train ENLRs to predict binary drug response for GDSC and CCLE cell lines. We showed that for most drugs, the models trained on GDSC cell lines can be transferred to CCLE cell lines. Our models achieved an average AUC > 0.78 for both GDSC2 drugs and CCLE test drugs. For the GDSC drugs that were consistently receiving low AUC scores, most of them were either highly biased labeled, with a small fraction of sensitive training cell lines (e.g. < 5%), or only tested on a small number of cell lines, both of which prevent obtaining a robust ENLR. We also noticed that for certain drugs, like Nutlin-3, the GDSC validation AUC was always significantly higher than the CCLE test AUC (>0.9 for Nutlin-3a in GDSC2 and < 0.7 in CCLE), which suggest a failed transfer. Nutlin-3a, specifically, was contradictorily reported among the hardest or easiest predicted drugs when only considering CCLE data (Q. Li et al., 2019; X. Wang et al., 2019; N. Zhang et al., 2015). The relatively high AUC for Nutlin-3 in GDSC and the failure of applying the GDSC trained Nutlin-3 model to CCLE may suggest an inconsistency between GDSC and CCLE measurements, which should be cautiously treated in further studies.

The ultimate goal of developing drug response prediction models is to apply them on real patients to guide drug selection and improve treatment protocol. In this study, we briefly examined the potential of transferring the models to real patients by classifying TCGA patients into different drug response groups and compared their survival outcomes. As we have expected, the patients

that were classified as sensitive to chemotherapy drugs had significantly better survival outcomes than patients that took the same drugs but were predicted as resistant. This supports that our drug sensitivity prediction models can be transferred to make predictions for real patients and improve personalized cancer therapy.

Most DGMs we used to generate the final results shown here were trained on GDSC cell line data. We also tried pre-training DGMs with TCGA data to take advantage of transferring learning with a large dataset. This pre-training step reduced the loss for reconstructing the GDSC expression profile and resulted in more robust latent representations for predict drug response on the test CCLE dataset. The GDSC-alone-trained DGMs, on the other hand, are better at predicting GDSC cell line drug responses. A similar inconsistency between the validation GDSC data and test CCLE data was observed for regularization, where adding regularization to DGMs helped improve test performance but not validation. These observations suggest that the DGMs may be slightly overfitted on GDSC data. Like other deep learning models, the deep hierarchy and flexible architecture guarantee DGMs with great power to learn the distribution of the input data, which can easily overfit on small datasets compared to traditional shallow models. This, however, also indicates that the learning potential of DGMs has yet been fully exploited given the limited amount of data. We are expecting that with the fast-growing of genomic and drug response data, we will be able to learn more informative representations for drug response prediction in the short future.

6.0 Discussion, conclusions and future work

In this dissertation project, we demonstrated the utility of machine learning techniques, especially deep generative models (DGMs), for learning cellular signaling state representations and predicting drug responses.

We first designed a graph-based machine learning framework, which uses tumor-specific causal relationships between somatic genetic alterations (SGAs) and differentially expressed genes (DEGs) to identify DEG modules that each represent the differential expression outcomes resulted from an aberrant signaling pathway. We applied the model on TCGA and METABRIC data and showed that the identified DEG modules are indicative of the disease mechanism of each tumor, which can be utilized to divide cancer patients into subtypes of different survival outcomes.

The DEG modules were simply represented as the average expression level of all genes in a module. To learn more interpretable cellular state representations that capture comprehensive information from expression data, we switched to DGMs for representation learning. We implemented the variational autoencoder (VAE) and designed a new model, the supervised vector-quantized variational autoencoder (S-VQ-VAE) for learning individual and global representations from the LINCS L1000 expression data. We found that the trained VAEs can accurately reconstruct the distribution of input expression profiles and generate sample-specific latent-representations that enhance drug-gene target predictions. The drug class representations learned by S-VQ-VAE can reveal mechanism-of-action correlations between different drugs, which can be further utilized for guiding drug development and drug re-purposing.

After observing the utility of DGMs for refining information from expression data, we moved on to apply DGMs for drug sensitivity prediction, which will directly promote precision

oncology. In this case, we designed four new DGMs, ResAE, ResVAE, RIAE, and RIVAE, that incorporate the mutation data to further enhance representation learning and improve model interpretability. We compared them with ordinary autoencoder (AE) and VAE for learning representations from GDSC and CCLE expression data. We showed that for most drugs of interest, DGM latent representations produced higher AUC compared to raw expression data when using elastic net logistic regressions (ENLRs) for drug sensitivity prediction. The mutation gene representations learned from new models also revealed functional correlations, from the aspects of oncogenic signaling pathways, between the mutated genes. We applied the cell line-trained ENLRs to real patients and showed that patients predicted as resistant to the drugs they were given had significantly worse survival outcomes compared to patients predicted as sensitive. This represents a successful transfer learning from in vitro data to in vivo applications. Our drug sensitivity prediction models, therefore, have the potential to be deployed as a decision-supporting tool to facilitate personalized drug selection.

From the graph-based module detection framework to more advanced DGMs, step by step, our results support the value of machine learning techniques for representation learning from genomic data and their potential to promote precision oncology. Thus, the methods introduced here represent a step towards more domain-specific, comprehensive, and interpretable deep learning models for use in biomedical research and clinical applications.

One major shortcoming of general deep learning models is their interpretability. In many cases, an uncontroversial interpretation of the architecture of a deep generative model or a clear one-to-one mapping between model structural components and cellular function components is not available. We partially resolved this problem by designing new DGMs including S-VQ-VAE, ResVAE, etc, for learning interpretable representations from expression and mutation data. We

understand, however, the connections between our models and a real cellular signaling system are often indirect, and developing more interpretable DGMs is the main focus in our future work. This may be realized by designing hybrid model architectures that incorporate new deep learning techniques like multi-attention and/or graph neural networks and integrate the signaling network topology information into the model as prior knowledge.

In this dissertation project, we preliminarily tried applying drug sensitivity prediction models trained on cell line data to real patients, which produced promising results. To obtain a more robust computational tool for serving patients, another major direction for future work is to train DGMs and drug sensitivity prediction models directly on data collected from real tumors. We are currently working on this direction to prepare a large dataset for new model development.

Appendix A Supplementary tables

The supplementary tables cited in chapter 3 and chapter 5 have been deposited in D-Scholarship@pitt and are available at http://d-scholarship.pitt.edu/cgi/export/40249/HTML/d-scholarship-40249.html

Bibliography

Azeloglu, E. U., & Iyengar, R. (2015). Signaling networks: Information flow, computation, and decision making. *Cold Spring Harbor perspectives in biology, 7*(4), a005934.

Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., . . . Sonkin, D. (2012). The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature, 483*(7391), 603.

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. *Third international AAAI conference on weblogs and social media*.

Bellamy, W. T. (1992). Prediction of response to drug therapy of cancer. *Drugs, 44*(5), 690-708.

Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *https://arxiv.org/abs/1308.3432*.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment, 2008*(10), P10008.

Braun, R., Leibon, G., Pauls, S., & Rockmore, D. (2011). Partition decoupling for multi-gene analysis of gene expression profiling data. *BMC bioinformatics, 12*(1), 497.

Brennan, C. W., Verhaak, R. G., McKenna, A., Campos, B., Noushmehr, H., Salama, S. R., . . . Berman, S. H. (2013). The somatic genomic landscape of glioblastoma. *Cell, 155*(2), 462-477.

Cai, C., Cooper, G., Lu, K., Ma, X., Xu, S., Zhao, Z., . . . Clark, N. (2018). Systematic discovery of the functional impact of somatic genome alterations in individual tumors through tumor-specific causal inference. *bioRxiv*, 329375.

Carey, L. A., Perou, C. M., Livasy, C. A., Dressler, L. G., Cowan, D., Conway, K., . . . Edmiston, S. (2006). Race, breast cancer subtypes, and survival in the Carolina Breast Cancer Study. *Jama, 295*(21), 2492-2502.

Carter, H., Chen, S., Isik, L., Tyekucheva, S., Velculescu, V. E., Kinzler, K. W., . . . Karchin, R. (2009). Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations. *Cancer research, 69*(16), 6660-6667.

Chang, Y., Park, H., Yang, H.-J., Lee, S., Lee, K.-Y., Kim, T. S., . . . Shin, J.-M. (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.

Chen, L., Cai, C., Chen, V., & Lu, X. (2016). Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC bioinformatics, 17*(1), S9.

Chiu, Y.-C., Chen, H.-I. H., Zhang, T., Zhang, S., Gorthi, A., Wang, L.-J., . . . Chen, Y. (2019). Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC medical genomics, 12*(1), 18.

Christopher, S. A., Diegelman, P., Porter, C. W., & Kruger, W. D. (2002). Methylthioadenosine phosphorylase, a gene frequently codeleted with p16cdkN2a/ARF, acts as a tumor suppressor in a breast cancer cell line. *Cancer research, 62*(22), 6639-6644.

Chudnovsky, Y., Kim, D., Zheng, S., Whyte, W. A., Bansal, M., Bray, M.-A., . . . Thiru, P. (2014). ZFHX4 interacts with the NuRD core member CHD4 and regulates the glioblastoma tumor-initiating cell state. *Cell reports, 6*(2), 313-324.

Ciriello, G., Gatza, M. L., Beck, A. H., Wilkerson, M. D., Rhie, S. K., Pastore, A., . . . Kandoth, C. (2015). Comprehensive molecular portraits of invasive lobular breast cancer. *Cell, 163*(2), 506-519.

Ciriello, G., Miller, M. L., Aksoy, B. A., Senbabaoglu, Y., Schultz, N., & Sander, C. (2013). Emerging landscape of oncogenic signatures across human cancers. *Nature genetics, 45*(10), 1127-1133.

Clark, N. R., Hu, K. S., Feldmann, A. S., Kou, Y., Chen, E. Y., Duan, Q., & Ma'ayan, A. (2014). The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC bioinformatics, 15*(1), 79.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Cortés-Ciriano, I., H Mervin, L., & Bender, A. (2016). Current trends in drug sensitivity prediction. *Current pharmaceutical design, 22*(46), 6918-6927.

Costello, J. C., Heiser, L. M., Georgii, E., Gönen, M., Menden, M. P., Wang, N. J., . . . Mpindi, J.-P. (2014). A community effort to assess and improve drug sensitivity prediction algorithms. *Nature biotechnology, 32*(12), 1202.

Croce, C. M. (2008). Oncogenes and cancer. *New England journal of medicine, 358*(5), 502-511.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., . . . Yuan, Y. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature, 486*(7403), 346-352.

Dees, N. D., Zhang, Q., Kandoth, C., Wendl, M. C., Schierding, W., Koboldt, D. C., . . . Mardis, E. R. (2012). MuSiC: identifying mutational significance in cancer genomes. *Genome research, 22*(8), 1589-1598.

Ding, M. Q., Chen, L., Cooper, G. F., Young, J. D., & Lu, X. (2018). Precision oncology beyond targeted therapy: Combining omics data with machine learning matches the majority of cancer cells to effective therapeutics. *Molecular Cancer Research, 16*(2), 269-278.

Ding, Z., Zu, S., & Gu, J. (2016). Evaluating the molecule-based prediction of clinical drug responses in cancer. *Bioinformatics, 32*(19), 2891-2895.

Dong, Z., Zhang, N., Li, C., Wang, H., Fang, Y., Wang, J., & Zheng, X. (2015). Anticancer drug sensitivity prediction in cell lines from baseline gene expression through recursive feature selection. *BMC cancer, 15*(1), 489.

Donner, Y., Kazmierczak, S. p., & Fortney, K. (2018). Drug repurposing using deep embeddings of gene expression profiles. *Molecular pharmaceutics, 15*(10), 4314-4325.

Dry, J. R., Pavey, S., Pratilas, C. A., Harbron, C., Runswick, S., Hodgson, D., . . . Cockerill, M. (2010). Transcriptional pathway signatures predict MEK addiction and response to selumetinib (AZD6244). *Cancer research, 70*(6), 2264-2273.

Duan, Q., Reid, S. P., Clark, N. R., Wang, Z., Fernandez, N. F., Rouillard, A. D., . . . Hafner, M. (2016). L1000CDS 2: LINCS L1000 characteristic direction signatures search engine. *NPJ systems biology and applications, 2*, 16015.

Encyclopedia, T. C. C. L. (2015). Consistency of drug profiles and predictors in large-scale cancer cell line data. *Nature, 528*(7580), 84.

Fojo, T. (2016). *Precision oncology: a strategy we were not ready to deploy.* Paper presented at the Seminars in oncology.

Frieboes, H. B., Edgerton, M. E., Fruehauf, J. P., Rose, F. R., Worrall, L. K., Gatenby, R. A., . . . Cristini, V. (2009). Prediction of drug response in breast cancer using integrative experimental/computational modeling. *Cancer research, 69*(10), 4484-4492.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software, 33*(1), 1.

Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., . . . Soares, J. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature, 483*(7391), 570.

Garraway, L. A., Verweij, J., & Ballman, K. V. (2013). Precision oncology: an overview. *J Clin Oncol, 31*(15), 1803-1805.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., . . . Al-Lazikani, B. (2011). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research, 40*(D1), D1100-D1107.

Giaever, G., & Nislow, C. (2014). The yeast deletion collection: a decade of functional genomics. *Genetics, 197*(2), 451-465.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). *Generative adversarial nets.* Paper presented at the Advances in neural information processing systems.

Greenman, C., Stephens, P., Smith, R., Dalgliesh, G. L., Hunter, C., Bignell, G., . . . Stevens, C. (2007). Patterns of somatic mutation in human cancer genomes. *Nature, 446*(7132), 153-158.

Harrell Jr, F. E., Lee, K. L., & Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine, 15*(4), 361-387.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning, 20*(3), 197-243.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation, 14*(8), 1771-1800.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504-507.

Ho, T. K. (1995). *Random decision forests.* Paper presented at the Proceedings of 3rd international conference on document analysis and recognition.

Hou, J. P., & Ma, J. (2014). DawnRank: discovering personalized driver genes in cancer. *Genome medicine, 6*(7), 56.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). *Densely connected convolutional networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Iwata, M., Sawada, R., Iwata, H., Kotera, M., & Yamanishi, Y. (2017). Elucidating the modes of action for bioactive compounds in a cell-specific manner by large-scale chemically-induced transcriptomics. *Scientific reports, 7*, 40164.

Jang, I. S., Neto, E. C., Guinney, J., Friend, S. H., & Margolin, A. A. (2014). Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data. In *Biocomputing 2014* (pp. 63-74): World Scientific.

Joerger, A. C., & Fersht, A. R. (2016). The p53 pathway: origins, inactivation in cancer, and emerging therapeutic approaches. *Annual review of biochemistry, 85*, 375-404.

Kaiser, Ł., Roy, A., Vaswani, A., Pamar, N., Bengio, S., Uszkoreit, J., & Shazeer, N. (2018). Fast Decoding in Sequence Models Using Discrete Latent Variables. *arXiv preprint arXiv:1803.03382*.

Keenan, A. B., Jenkins, S. L., Jagodnik, K. M., Koplev, S., He, E., Torre, D., . . . Lachmann, A. (2018). The library of integrated network-based cellular signatures NIH program: system-level cataloging of human cells response to perturbations. *Cell systems, 6*(1), 13-24.

Kingma, D. P., & Welling, M. (2014). Stochastic gradient VB and the variational auto-encoder. *Second International Conference on Learning Representations, ICLR*.

Kleihues, P., & Ohgaki, H. (1999). Primary and secondary glioblastomas: from concept to clinical diagnosis. *Neuro-oncology, 1*(1), 44-51.

Lamb, J. (2007). The Connectivity Map: a new tool for biomedical research. *Nature Reviews Cancer, 7*(1), 54.

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., . . . Ross, K. N. (2006). The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, *313*(5795), 1929-1935.

Lamborn, K. R., Chang, S. M., & Prados, M. D. (2004). Prognostic factors for survival of patients with glioblastoma: recursive partitioning analysis. *Neuro-oncology*, *6*(3), 227-235.

Le, L., Patterson, A., & White, M. (2018). *Supervised autoencoders: Improving generalization performance with unsupervised regularizers.* Paper presented at the Advances in Neural Information Processing Systems.

Le Mercier, M., Hastir, D., Lopez, X. M., De Neve, N., Maris, C., Trepant, A.-L., . . . Salmon, I. (2012). A simplified approach for the molecular classification of glioblastomas. *PLoS one*, *7*(9).

LeCun, Y., Cortes, C., & Burges, C. J. (1998). The MNIST database of handwritten digits, 1998. *URL https://deepai.org/dataset/mnist*, *10*, 34.

Lee, H., Ekanadham, C., & Ng, A. Y. (2008). *Sparse deep belief net model for visual area V2.* Paper presented at the Advances in neural information processing systems.

Li, Q., Shi, R., & Liang, F. (2019). Drug sensitivity prediction with high-dimensional mixture regression. *PLoS one*, *14*(2), e0212108.

Li, Y., Kao, G. D., Garcia, B. A., Shabanowitz, J., Hunt, D. F., Qin, J., . . . Lazar, M. A. (2006). A novel histone deacetylase pathway regulates mitosis by modulating Aurora B kinase activity. *Genes & development*, *20*(18), 2566-2579.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., & Mesirov, J. P. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, *27*(12), 1739-1740.

Lin, G.-S., Chen, Y.-P., Lin, Z.-X., Wang, X.-F., Zheng, Z.-Q., & Chen, L. (2014). STAT3 serine 727 phosphorylation influences clinical outcome in glioblastoma. *International journal of clinical and experimental pathology*, *7*(6), 3141.

Liu, P., Cheng, H., Roberts, T. M., & Zhao, J. J. (2009). Targeting the phosphoinositide 3-kinase pathway in cancer. *Nature reviews Drug discovery*, *8*(8), 627-644.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial Autoencoders. *arXiv preprint arXiv:1511.05644*.

Mao, H., LeBrun, D. G., Yang, J., Zhu, V. F., & Li, M. (2012). Deregulated signaling pathways in glioblastoma multiforme: molecular mechanisms and therapeutic targets. *Cancer investigation, 30*(1), 48-56.

Mateo, J., Olmos, D., Dumez, H., Poondru, S., Samberg, N. L., Barr, S., . . . Tan, D. S. (2016). A first in man, dose-finding study of the mTORC1/mTORC2 inhibitor OSI-027 in patients with advanced solid malignancies. *British journal of cancer, 114*(8), 889-896.

Mazurowski, M. A., Desjardins, A., & Malof, J. M. (2013). Imaging descriptors improve the predictive power of survival models for glioblastoma patients. *Neuro-oncology, 15*(10), 1389-1394.

Menden, M. P., Iorio, F., Garnett, M., McDermott, U., Benes, C. H., Ballester, P. J., & Saez-Rodriguez, J. (2013a). Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLoS one, 8*(4).

Menden, M. P., Iorio, F., Garnett, M., McDermott, U., Benes, C. H., Ballester, P. J., & Saez-Rodriguez, J. (2013b). Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLoS one, 8*(4), e61318.

Mermel, C. H., Schumacher, S. E., Hill, B., Meyerson, M. L., Beroukhim, R., & Getz, G. (2011). GISTIC2. 0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome biology, 12*(4), R41.

Mischel, P. S., Shai, R., Shi, T., Horvath, S., Lu, K. V., Choe, G., . . . Liau, L. M. (2003). Identification of molecular subtypes of glioblastoma by gene expression profiling. *Oncogene, 22*(15), 2361-2373.

Moon, S.-K., Lee, H.-Y., Li, J.-D., Nagura, M., Kang, S.-H., Chun, Y.-M., . . . Lim, D. J. (2002). Activation of a Src-dependent Raf–MEK1/2–ERK signaling pathway is required for IL-1α-induced upregulation of β-defensin 2 in human middle ear epithelial cells. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research, 1590*(1-3), 41-51.

Nair, V., & Hinton, G. E. (2009). *3D object recognition with deep belief nets.* Paper presented at the Advances in neural information processing systems.

Network, C. G. A. (2012). Comprehensive molecular portraits of human breast tumours. *Nature, 490*(7418), 61.

Network, C. G. A. R. (2011). Integrated genomic analyses of ovarian carcinoma. *Nature, 474*(7353), 609.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). *On spectral clustering: Analysis and an algorithm.* Paper presented at the Advances in neural information processing systems.

Noh, E.-M., Lee, Y.-R., Hong, O.-Y., Jung, S. H., Youn, H. J., & Kim, J.-S. (2015). Aurora kinases are essential for PKC-induced invasion and matrix metalloproteinase-9 expression in MCF-7 breast cancer cells. *Oncology reports, 34*(2), 803-810.

O'Reilly, K. E., Rojo, F., She, Q.-B., Solit, D., Mills, G. B., Smith, D., . . . Ludwig, D. L. (2006). mTOR inhibition induces upstream receptor tyrosine kinase signaling and activates Akt. *Cancer research, 66*(3), 1500-1508.

Pabon, N. A., Xia, Y., Estabrooks, S. K., Ye, Z. F., Herbrand, A. K., Suss, E., . . . Bar-Joseph, Z. (2018). Predicting protein targets for drug-like compounds using transcriptomics. *PLoS computational biology, 14*(12), e1006651. Retrieved from <Go to ISI>://WOS:000454835100043

Pardanani, A., Ketterling, R. P., Brockman, S. R., Flynn, H. C., Paternoster, S. F., Shearer, B. M., . . . Cools, J. (2003). CHIC2 deletion, a surrogate for FIP1L1-PDGFRA fusion, occurs in systemic mastocytosis associated with eosinophilia and predicts response to imatinib mesylate therapy. *Blood, 102*(9), 3093-3096.

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., . . . Hu, Z. (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology, 27*(8), 1160.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., . . . Lerer, A. (2017). Automatic differentiation in pytorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.

Pencina, M. J., & D'Agostino, R. B. (2004). Overall C as a measure of discrimination in survival analysis: model specific population value and confidence interval estimation. *Statistics in medicine, 23*(13), 2109-2123.

Peyressatre, M., Prével, C., Pellerano, M., & Morris, M. (2015). Targeting cyclin-dependent kinases in human cancers: from small molecules to peptide inhibitors. *Cancers, 7*(1), 179-237.

Piccolo, S. R., & Frey, L. J. (2013). Clinical and molecular models of glioblastoma multiforme survival. *International journal of data mining and bioinformatics, 7*(3), 245.

Pimenova, A. A., Thathiah, A., De Strooper, B., & Tesseur, I. (2014). Regulation of amyloid precursor protein processing by serotonin signaling. *PLoS one, 9*(1), e87014.

Prasad, V. (2016). Perspective: the precision-oncology illusion. *Nature, 537*(7619), S63.

Prasad, V., Fojo, T., & Brada, M. (2016). Precision oncology: origins, optimism, and potential. *The Lancet Oncology, 17*(2), e81-e86.

program, T. c. g. a. The cancer genome atlas research network. Retrieved from https://www.cancer.gov/tcga

Radhakrishnan, K., Halász, Á., Vlachos, D., & Edwards, J. S. (2010). Quantitative understanding of cell signaling: the importance of membrane organization. *Current opinion in biotechnology, 21*(5), 677-682.

Ranzato, M. A., & Szummer, M. (2008). *Semi-supervised learning of compact document representations with deep networks.* Paper presented at the Proceedings of the 25th international conference on Machine learning.

RDUSSEEUN, L. K. P. J. (1987). Clustering by means of medoids.

Reva, B., Antipin, Y., & Sander, C. (2011). Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic acids research, 39*(17), e118-e118.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *https://arxiv.org/abs/1401.4082*. Retrieved from https://arxiv.org/abs/1401.4082

Riddick, G., Song, H., Ahn, S., Walling, J., Borges-Rivera, D., Zhang, W., & Fine, H. A. (2010). Predicting in vitro drug sensitivity using Random Forests. *Bioinformatics, 27*(2), 220-224.

Roldan-Valadez, E., Rios, C., Motola-Kuba, D., Matus-Santos, J., Villa, A. R., & Moreno-Jimenez, S. (2016). Choline-to-N-acetyl aspartate and lipids-lactate-to-creatine ratios together with age assemble a significant Cox's proportional-hazards regression model for prediction of survival in high-grade gliomas. *The British journal of radiology, 89*(1067), 20150502.

Roy, A., Vaswani, A., Neelakantan, A., & Parmar, N. (2018). Theory and Experiments on Vector Quantized Autoencoders. *arXiv preprint arXiv:1805.11063*.

Rubio-Perez, C., Tamborero, D., Schroeder, M. P., Antolín, A. A., Deu-Pons, J., Perez-Llamas, C., . . . Lopez-Bigas, N. (2015). In silico prescription of anticancer drugs to cohorts of 28 tumor types reveals targeting opportunities. *Cancer cell, 27*(3), 382-396.

Salakhutdinov, R. R. J. s., & Hinton, G. E. (2009). Deep Boltzmann Machines. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 448-455.

Sanchez-Vega, F., Mina, M., Armenia, J., Chatila, W. K., Luna, A., La, K. C., . . . Saghafinia, S. (2018). Oncogenic signaling pathways in the cancer genome atlas. *Cell, 173*(2), 321-337. e310.

Savoia, P., Fava, P., Casoni, F., & Cremona, O. (2019). Targeting the ERK signaling pathway in melanoma. *International journal of molecular sciences, 20*(6), 1483.

Shen, L., Kondo, Y., Ahmed, S., Boumber, Y., Konishi, K., Guo, Y., . . . Issa, J.-P. J. (2007). Drug sensitivity prediction by CpG island methylation profile in the NCI-60 cancer cell line panel. *Cancer research, 67*(23), 11335-11343.

Shoemaker, R. H. (2006). The NCI60 human tumour cell line anticancer drug screen. *Nature Reviews Cancer, 6*(10), 813.

Siavelis, J. C., Bourdakou, M. M., Athanasiadis, E. I., Spyrou, G. M., & Nikita, K. S. (2015). Bioinformatics methods in drug repurposing for Alzheimer's disease. *Briefings in bioinformatics, 17*(2), 322-335.

Smolensky, P. (1986). *Information processing in dynamical systems: Foundations of harmony theory*. Retrieved from

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Highway networks. *arXiv preprint arXiv:1505.00387*.

Staunton, J. E., Slonim, D. K., Coller, H. A., Tamayo, P., Angelo, M. J., Park, J., . . . Weinstein, J. N. (2001). Chemosensitivity prediction by transcriptional profiling. *Proceedings of the National Academy of Sciences, 98*(19), 10787-10792.

Stephens, P. J., Tarpey, P. S., Davies, H., Van Loo, P., Greenman, C., Wedge, D. C., . . . Bignell, G. R. (2012). The landscape of cancer genes and mutational processes in breast cancer. *Nature, 486*(7403), 400-404.

Stern, D. F. (2000). Tyrosine kinase signalling in breast cancer: ErbB family receptor tyrosine kinases. *Breast cancer research, 2*(3), 176.

Subramanian, A. (2015). L1000 Connectivity Map perturbational profiles from Broad Institute LINCS Center for Transcriptomics LINCS PHASE II. *Gene Expression Omnibus GSE70138* *https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138*. Retrieved from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138

Subramanian, A. (2017). Datasets Used in Evaluation Of RNAi And CRISPR Technologies By Large Scale Gene Expression Profiling In The Connectivity Map. *Gene Expression Omnibus GSE106127* *https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE106127*. Retrieved from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE106127

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., . . . Asiedu, J. K. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell, 171*(6), 1437-1452. e1417.

Swift, S., Tucker, A., Vinciotti, V., Martin, N., Orengo, C., Liu, X., & Kellam, P. (2004). Consensus clustering and functional interpretation of gene-expression data. *Genome biology, 5*(11), R94.

Szakács, G., Annereau, J.-P., Lababidi, S., Shankavaram, U., Arciello, A., Bussey, K. J., . . . Reimers, M. (2004). Predicting drug sensitivity and resistance: profiling ABC transporter genes in cancer cells. *Cancer cell, 6*(2), 129-137.

Tamborero, D., Lopez-Bigas, N., & Gonzalez-Perez, A. (2013). Oncodrive-CIS: a method to reveal likely driver genes based on the impact of their copy number changes on expression. *PLoS one, 8*(2).

Tannock, I. F., & Hickman, J. A. (2016). Limits to personalized cancer medicine. *N Engl J Med, 375*(13), 1289-1294.

Therneau, T. A package for survival analysis in S R-package version 2. 2013. In.

Toss, A., & Cristofanilli, M. (2015). Molecular characterization and targeted therapeutic approaches in breast cancer. *Breast cancer research, 17*(1), 60.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., . . . Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 6306-6315.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). *Attention is all you need.* Paper presented at the Advances in Neural Information Processing Systems.

Verhaak, R. G., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., . . . Mesirov, J. P. (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell, 17*(1), 98-110.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . Bright, J. (2019). SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python. *https://arxiv.org/abs/1907.10121*.

Vivanco, I., & Sawyers, C. L. (2002). The phosphatidylinositol 3-kinase–AKT pathway in human cancer. *Nature Reviews Cancer, 2*(7), 489-501.

Vogelstein, B., Papadopoulos, N., Velculescu, V. E., Zhou, S., Diaz, L. A., & Kinzler, K. W. (2013). Cancer genome landscapes. *Science, 339*(6127), 1546-1558.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing, 17*(4), 395-416.

Waclaw, R. R., Wang, B., Pei, Z., Ehrman, L. A., & Campbell, K. (2009). Distinct temporal requirements for the homeobox gene Gsx2 in specifying striatal and olfactory bulb neuronal fates. *Neuron, 63*(4), 451-465.

Wang, Q., Hu, B., Hu, X., Kim, H., Squatrito, M., Scarpace, L., . . . Li, Y. (2017). Tumor evolution of glioma-intrinsic gene expression subtypes associates with immunological changes in the microenvironment. *Cancer cell, 32*(1), 42-56. e46.

Wang, X., Sun, Z., Zimmermann, M. T., Bugrim, A., & Kocher, J.-P. (2019). Predict drug sensitivity of cancer cells with pathway activity inference. *BMC medical genomics, 12*(1), 15.

Wang, Z., Clark, N. R., & Ma'ayan, A. (2016). Drug-induced adverse events prediction with the LINCS L1000 data. *Bioinformatics, 32*(15), 2338-2345.

Wangaryattawanich, P., Hatami, M., Wang, J., Thomas, G., Flanders, A., Kirby, J., . . . Luedi, M. M. (2015). Multicenter imaging outcomes study of The Cancer Genome Atlas glioblastoma patient cohort: imaging predictors of overall and progression-free survival. *Neuro-oncology, 17*(11), 1525-1537.

Waskom, M. (2018). Seaborn. *Zenodo https://doi.org/10.5281/zenodo.883859*. Retrieved from https://github.com/mwaskom/seaborn/tree/v0.9.0

Watanabe, H., Akasaka, D., Ogasawara, H., Sato, K., Miyake, M., Saito, K., . . . Hondo, T. (2010). Peripheral serotonin enhances lipid metabolism by accelerating bile acid turnover. *Endocrinology, 151*(10), 4776-4786.

Wei, D., Liu, C., Zheng, X., & Li, Y. (2019). Comprehensive anticancer drug response prediction based on a simple cell line-drug complex network model. *BMC bioinformatics, 20*(1), 44.

Weinberg, R. (2013). *The biology of cancer*: Garland science.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., . . . Network, C. G. A. R. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics, 45*(10), 1113.

Weller, M. (1998). Predicting response to cancer chemotherapy: the role of p53. *Cell and tissue research, 292*(3), 435-445.

Welling, M., Rosen-Zvi, M., & Hinton, G. (2005). Advances in Neural Information Processing Systems 17. In: MIT Press, Cambridge, MA.

Weng, G., Bhalla, U. S., & Iyengar, R. (1999). Complexity in biological signaling systems. *Science, 284*(5411), 92-96.

Wilkerson, M. D., & Hayes, D. N. (2010). ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. *Bioinformatics, 26*(12), 1572-1573.

Woo, G., Fernandez, M., Hsing, M., Lack, N. A., Cavga, A. D., & Cherkasov, A. (2019). DeepCOP: deep learning-based approach to predict gene regulating effects of small molecules. *Bioinformatics*.

Xia, C., Ma, W., Stafford, L. J., Liu, C., Gong, L., Martin, J. F., & Liu, M. (2003). GGAPs, a new family of bifunctional GTP-binding and GTPase-activating proteins. *Molecular and cellular biology, 23*(7), 2476-2488.

Yang, W., Soares, J., Greninger, P., Edelman, E. J., Lightfoot, H., Forbes, S., . . . Thompson, I. R. (2012). Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research, 41*(D1), D955-D961.

Yap, C. W. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry, 32*(7), 1466-1474.

Young, J. (2020). DEEP LEARNING FOR CAUSAL STRUCTURE LEARNING APPLIED TO CANCER PATHWAY DISCOVERY. In.

Zack, T. I., Schumacher, S. E., Carter, S. L., Cherniack, A. D., Saksena, G., Tabak, B., . . . Mermel, C. H. (2013). Pan-cancer patterns of somatic copy number alteration. *Nature genetics*, *45*(10), 1134-1140.

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., & Cheng, F. (2019). deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics*, *35*(24), 5191-5198.

Zhang, N., Wang, H., Fang, Y., Wang, J., Zheng, X., & Liu, X. S. (2015). Predicting anticancer drug responses using a dual-layer integrated cell line-drug network model. *PLoS computational biology*, *11*(9).

Zhang, Y., Lee, K., & Lee, H. (2016). *Augmenting supervised neural networks with unsupervised objectives for large-scale image classification.* Paper presented at the International Conference on Machine Learning.

Zhang, Y., Li, A., Peng, C., & Wang, M. (2016). Improve glioblastoma multiforme prognosis prediction by using feature selection and multiple kernel learning. *IEEE/ACM transactions on computational biology and bioinformatics*, *13*(5), 825-835.

Zhou, Q., Chen, W., Song, S., Gardner, J. R., Weinberger, K. Q., & Chen, Y. (2015). *A reduction of the elastic net to support vector machines with an application to GPU computing.* Paper presented at the Twenty-Ninth AAAI Conference on Artificial Intelligence.