Semi-Supervised Generative Adversarial Network for Gene Expression Inference

Kamran Ghasedi Dizaji*
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania
kamran.ghasedi@gmail.com

Xiaoqian Wang*
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania
xqwang1991@gmail.com

Heng Huang*
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania
heng.huang@pitt.edu

ABSTRACT

Gene expression profiling provides comprehensive characterization of cellular states under different experimental conditions, thus contributes to the prosperity of many fields of biomedical research. Although the rapid development of gene expression profiling has been observed, genome-wide profiling of large libraries is still expensive and difficult. Due to the fact that there are significant correlations between gene expression patterns, previous studies introduced regression models for predicting the target gene expressions from the landmark gene profiles. These models formulate the gene expression inference in a completely supervised manner, which require a large labeled dataset (*i.e.* paired landmark and target gene expressions). However, collecting the whole gene expressions is much more expensive than the landmark genes. In order to address this issue and take advantage of cheap unlabeled data (*i.e.* landmark genes), we propose a novel semi-supervised deep generative model for target gene expression inference. Our model is based on the generative adversarial network (GAN) to approximate the joint distribution of landmark and target genes, and an inference network to learn the conditional distribution of target genes given the landmark genes. We employ the reliable generated data by our GAN model as the extra training pairs to improve the training of our inference model, and utilize the trustworthy predictions of the inference network to enhance the adversarial training of our GAN network. We evaluate our model on the prediction of two types of gene expression data and identify obvious advantage over the counterparts.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Applied computing** → **Bioinformatics**;

KEYWORDS

Gene expression inference, Semi-supervised learning, Deep generative model.

ACM Reference Format:
Kamran Ghasedi Dizaji*, Xiaoqian Wang*, and Heng Huang. 2018. Semi-Supervised Generative Adversarial Network for Gene Expression Inference. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220114

1 INTRODUCTION

In the field of molecular biology, gene expression profiling is a powerful tool for measuring the expression pattern of tens of thousands of genes in a given cell or tissue. The rapid growth in high-throughput techniques has substantially reduced the cost of genome-wide profiling and enabled the profiling of gene expression in various biological states. Several data repositories have been constructed to store the gene expression profiles in versatile cellular conditions. For example, Gene Expression Omnibus (GEO) [10] is a functional genomics repository that collects curated gene expression profiles under different circumstances. ArrayExpress [4] archives well-annotated arrays and sequences based gene expression data from various species. The construction of these public gene expression databases provides a wealth of data resources that largely support global biomedical research.

Gene expression profiling provides a comprehensive view of cellular status and is therefore the basis for functional gene expression pattern characterization. The availability of rich gene expression data has contributed to the prosperity in many areas. In cancer classification, [40] analyzed the gene expression pattern among different breast cancer patients and detected important genes associated with clinical behavior. Recent landscape studies [12, 39] looked into gene expression data from different tumor types to reveal the cross-tissue cancer cluster structure, which enhanced the understanding of relations between and within different cancer types. Moreover, by looking into the gene expression levels in a post mortem brain tissue data, Richiardi *et al.* [31] analyzed the relations between genetic information and functional brain networks and identified genes linked with ion channels and synaptic function. Gene expression data has also been widely applied in drug-target network construction [43] and drug discovery [30], in which the characterization of different gene expression patterns in response to distinct perturbation of small molecules facilitates the analysis of drug mechanism and effect.

*To whom all correspondence should be addressed. This work was partially supported by U.S. NIH R01 AG049371, NSF IIS 1302675, IIS 1344152, DBI 1356628, IIS 1619308, IIS 1633753.

*K. Ghasedi and X. Wang made equal contributions to this paper.

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220114

Despite the fast development and widespread application of gene expression analysis, profiling of large libraries in different chemical conditions is still difficult and expensive [26]. How to effectively measure the expression level of more than 20,000 genes in the human genome for large-scale profiles still remains a key issue. Based on previous findings [16, 27, 35], there is a common observation that similarity patterns exist in the expression profile of different genes, such that genes with similar functions indicate similar mechanisms in response to various experimental conditions. As it is pointed in the clustering analysis on single cell RNA-seq in [27, 35], genes in the same clusters exhibit similar expression pattern across different cellular states. Given such correlation structure among gene expression profiles, it is reasonable to assume that even a small number of genes can be informative to approximate the message in the entire genome. To identify such subset of informative genes, researchers from the Library of Integrated Network-Based Cellular Signatures (LINCS) Program (http://www.lincsproject.org/) picked ~1000 genes from the entire genome, which contain ~80% of message delivered in the whole transcriptome. These set of genes with correlation information and predictive power are referred to as landmark genes.

Based on the above findings, one feasible and cost-effective strategy for large-scale gene expression profiling is to measure the expression profile of only landmark genes and then estimate the remaining target gene expression through an appropriate predictive model. Therefore, it is essential to construct effective computational methods to infer the target gene expression profiles from the landmark genes. One most straightforward model is multi-task linear regression, where the estimation of one target gene from the landmark gene expression is formulated as one task. The linear regression model has been applied in the LINCS program. The LINCS program generated the landmark gene expression of ~1.3 million profiles using L1000 technology, and adopt the linear regression model to infer the expression of the remaining target genes.

However, the regulatory network among genes is complicated, linear models do not have enough capacity to capture the non-linear relationship of the gene expression profiles [15]. Kernel models provide a way to introduce flexibility in representing the non-linear relations among gene expression, but they suffer from high computational burden thus are not applicable to large-scale problems. In contrast, deep learning models are scalable and highly flexible, and have been widely applied to different biological problems, such as prediction of specificities of RNA-binding proteins in alternative slicing [1, 23], regulatory mechanism of histone modifications in gene expression [36], protein structure prediction [38], predicting the function of non-coding DNA [44] and population stratification detection [32]. The remarkable predictive power and flexibility of the deep learning model makes it a powerful alternative for effective prediction large-scale gene expression profiles.

Recently, [5], Chen *et al.* applied deep learning models to the gene expression inference problem. They used a fully connected multilayer perceptron to study the non-linear association among genes and achieved better results than linear methods, which validates the effectiveness of deep learning models in the gene expression inference problem. However, previous methods still suffers from several problems: 1) traditionally, gene expression inference is formulated as a regression problem, where the computational models attempt to approximate the conditional probability distribution of target genes given landmark genes, but do not consider their joint distribution, thus have limited predictive power; 2) previous methods formulate the gene expression inference in a totally supervised manner, where only profiles with both landmark and target gene expression measurements (which we call as "labeled" data according to the notations in previous paper [41]) are involved in the training process. However, since the measurement of only landmark genes are much cheaper, there are a lot more profiles with the measurement of only landmark genes (which we call as "unlabeled" data according to the notations in [41]) are not used in the training process.

In order to solve these problems, we propose a novel semi-supervised generative adversarial network (abbreviated as Semi-GAN) for gene expression inference. Our model is inspired by the inpainting problem in computer vision applications, where the goal is to fill in the missing part in a corrupted image based on the known image context and the learned distribution over the entire image. Here we regard the target gene expression as the missing part in a profile and the goal is to fill in the missing given the landmark gene information (*i.e.,* context). We propose to construct a deep generative model that approximate the joint distribution of landmark and target genes. By doing this, we analyze the overall distribution and correlation among genes which improves the inference. Moreover, we formulate our model in a semi-supervised manner that incorporates the profiles with only landmark genes into the training process. The use of the unlabeled section of data can improve the learning of landmark gene distribution and also strengthens the inference of target gene expression.

- Proposing a novel semi-supervised framework for gene expression inference;
- Introducing the collaborative training of our GAN and inference network;
- Outperforming alternative supervised models with significant margins on two datasets according to different evaluation metrics.

We organize the remaining part of paper in the following order: Firstly we review the recent progress in gene expression inference and deep learning. Then in Section 3 we propose the motivation of our model and also introduce the architecture details. Next in Section 4, we present the experiments where we validate our mode on the inference of two gene expression datasets. In the end we conclude our paper in Section 5.

2 RELATED WORK

2.1 Gene Expression Inference

Although rapid progress has been observed in high-throughput sequencing and analysis techniques, genome-wide expression profiling for large-scale libraries under different disturbance remains expensive and difficult [26]. Therefore, how to keep a low budget while the informative measurement in gene expression profiling remains a key issue. Previous studies have detected a high degree of correlation among gene expression such that genes with similar function preserved similar expression patterns under different experimental circumstances. Due to the correlation structure existing in gene expression patterns, even a small number of genes

can provide a wealth of information. Shah *et al.* [35] found that a random collection of 20 genes captured ~50% of the relevant information throughout the genome. Recent advances in RNA-seq [16, 27] also support the notion that a small number of genes are abundant enough to approximately depict the overall information throughout the transcriptome.

Researchers from the LINCS program assembled GEO data on the basis of Affymetrix HGU133A microarray to analyze the gene correlation structure and identify the subset of informative genes to approximate the overall information in genome. They collected the expression profiles from a total of 12,063 genes and determined the maximum percentage of correlation information can be recovered given a specific number of genes. The calculation of recovery percentage is based on the comparable rank from the Kolmogorov-Smirnov statistic. According to the LINCS analysis, researchers found that only 978 genes were capable of restoring 82% of the observed connections across the entire transcriptome [20]. The set of 978 genes have been characterized as landmark genes and can be used to deduce the expression of other target genes in different cell types under various chemical, genetic and disease conditions.

2.2 Deep Neural Networks

In recent years, deep learning has shown remarkable results in wide range of applications, such as computer vision [22], natural language processing [7], speech recognition [17], and even biological science[9]. The impressive capability of deep models is due to efficient and scalable leaning of discriminative features from raw data via multi-layer networks. Among different models, Goodfellow *et. al.* proposed a powerful generative model, called generative adversarial networks (GAN) [14], especially in computer vision tasks. In particular, GAN consists of two sub-networks, a generator and a discriminator, and aims to play minimax game between these networks. While the generator's goal is to fool the discriminator by synthesizing realistic images from arbitrary distribution (i.e. random noise), the discriminator tries to distinguish between the real and synthesized (i.e. fake) images. GAN model is applied to different tasks, including image generation [8, 19], image translation [45], semi-supervised image classification [33], image inpainting [29, 42], also speech enhancement [28] and drug discovery [3].

We also adopt GAN architecture in our model in order to learn the joint distribution of landmark and target genes. In one view, our model on inferring the target genes from landmark genes is similar to image inpainting methods [29, 42], in which the goal is to deduce the missing part in a corrupted image. Pathak *et al.* [29] employed the autoencoder architecture, where the encoder maps the corrupted image to a latent variable, and the decoder recovers the original image without damage. The framework attempted to reduce the reconstruction loss as well as the adversarial loss such that the recovered images followed similar distribution as real images. In another view, our work is similar to the semi-supervised image classification methods [6, 33], in which the task is to predict categorical labels of input image data. For instance in [6], GAN is utilized to learn the joint distribution of image and categorical labels in order to improve the classification task by the synthesized image-label pairs. However, our proposed model has major differences compared to the previous works. First, our task is semi-supervised regression on non-structured gene data, which is different from supervised inpainting and structured image data. Moreover, our generative model is unique in comparison with other models, since we train it using adversarial, reconstruction and translation loss functions.

3 GENERATIVE NETWORK FOR SEMI-SUPERVISED LEARNING

3.1 Problem Definition

In the gene expression inference problem, we use vector $\mathbf{x}$ to denote the landmark gene expression profile and vector $\mathbf{y}$ for the target gene expression. $\Omega_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{n_l}$ collects the labeled profiles where the measurement for both landmark and target genes are available, while $\Omega_u = \{\mathbf{x}_j^u\}_{j=1}^{n_u}$ corresponds to the unlabeled profiles with the expression of only landmark genes measured. Usually we have $n_u \gg n_l$, since the measurement of only landmark genes is much cheaper than all the genes in the entire transcriptome. Our goal is to construct a model, which appropriately predicts the target gene expression using a small set of labeled genes (*i.e.* paired landmark and target genes) and a large set of unlabeled genes (*i.e.* landmark genes).

3.2 Motivation

In previous works, the inference of target gene expression is formulated as a multi-task regression, where predicting the expression of each target gene in $\mathbf{y}$ via landmark genes $\mathbf{x}$ is one regression task. The regression framework is usually formulated in a fully supervised manner, such that a large set of labeled data is required to efficiently train the regression model. However in our problem, collecting the whole gene expression profiles (*i.e.* paired landmark and target genes $(\mathbf{x}, \mathbf{y})$) is much more expensive than the the landmark genes $\mathbf{x}$ alone. In order to address this issue and benefit from the plentiful unlabeled profiles, we propose a semi-supervised learning framework to take advantage of both labeled and unlabeled profiles and use the unlabeled data to strengthen the learning. Our proposed model consists of an inference network and a GAN sub-model. Generally, we consider the GAN sub-model to learn the joint distribution $p(\mathbf{x}, \mathbf{y})$, and the inference network to learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$. We provide a collaboration framework between the GAN and inference networks, such that the GAN generates the approximated paired samples $(\hat{\mathbf{x}}^z, \hat{\mathbf{y}}^z)$ as reliable extra labeled data for training the inference network, and the approximated pairs $(\mathbf{x}^u, \hat{\mathbf{y}}^u)$ by the inference network improves the adversarial training of the GAN network.

In particular, our GAN network includes two generators G_x and G_y to synthesize both landmark genes $\hat{\mathbf{x}}^z$ and target genes $\hat{\mathbf{y}}^z$ from a shared random input $\mathbf{z}$ respectively, and three discriminators D_x, D_y, D_{xy} to distinguish between the real and fake data $\mathbf{x}^u$ vs. $\hat{\mathbf{x}}^z$, $\mathbf{y}^l$ vs. $\hat{\mathbf{y}}^z$, and $(\mathbf{x}^l, \mathbf{y}^l)$ vs. $(\hat{\mathbf{x}}^z, \hat{\mathbf{y}}^z)$ respectively. In addition to adversarial loss, we use a reconstruction and a translation loss functions to help training of our generators. To do so, we consider a network to learn the inverse mapping of G_x, where the input is the landmark genes and the output has the same dimension of $\mathbf{z}$. Using this inverse network I_x, we define a reconstruction loss function for unlabeled data $\mathbf{x}^u$ through $I_x \rightarrow G_x$ pathway,

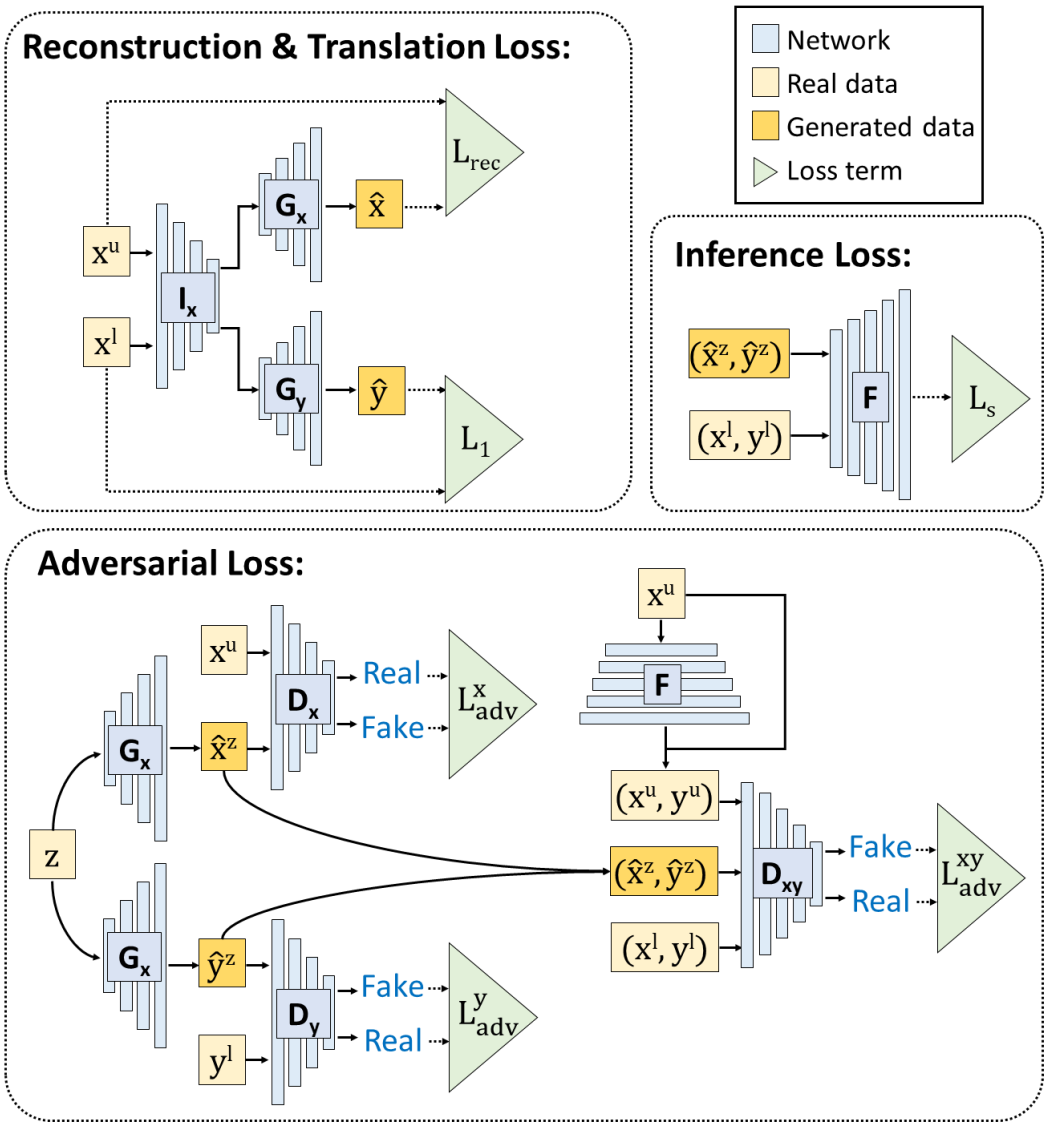


Figure 1: Illustration of the SemiGAN architecture for gene expression inference. SemiGAN consists of two sub-models, a GAN network and an inference network. x and y denote the profile for landmark and target gene expression. z is a random variable drawn from a prior distribution $p(\mathbf{z})$, working as the input for generators. The labeled profiles $(\mathbf{x}^l, \mathbf{y}^l)$ are drawn from the joint distribution $p(\mathbf{x}, \mathbf{y})$ while the unlabeled profiles $\mathbf{x}^u$ come from the distribution $p(\mathbf{x})$. Our GAN network includes two generators, G_x and G_y, generating $\hat{\mathbf{x}}^z$ and $\hat{\mathbf{y}}^z$ to fool three discriminators, D_x, D_y and D_{xy}. These networks are mainly trained by adversarial loss functions (bottom). We also construct an inverse network I_x to encode x in the generators latent space, and use the reconstruction and translation losses to help the training of generators and the inverse network (top left). Finally, we build an inference network F to estimate the the output y given x. We use $\mathcal{L}_s$ to measure the loss between the ground truth and the predictions (top right). The inference and GAN networks have collaborative relation, such that predictions of F enhances the training of generators, and the generated data is utilized to improve the learning of the inference network.

and a translation loss function for labeled data $(\mathbf{x}^l, \mathbf{y}^l)$ through $I_x \rightarrow G_y$ pathway. Note that theses two loss functions are helpful in adversarial training of our generator networks, and aid generating large-dimension and unstructured gene data by avoiding mode collapse issue and using side information. Furthermore, we employ the inference network F to map the landmark gene expressions to the target gene expressions. For clarification purpose, we plot the

architecture of our model, called SemiGAN, along with the applied loss functions in Fig. 1.

3.3 Semi-Supervised GAN Model

As mentioned, SemiGAN has two generators and three discriminator networks. Following we show the adversarial loss functions corresponding to the pairs of generator and discriminator networks. The min-max adversarial loss for training the generator network G_x and discriminator network D_x is formulated as:

$$\min_{G_x}\max_{D_x}\ \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[\log(D_x(\mathbf{x}))\big]+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1-D_x(G_x(\mathbf{z}))\big] \quad (1)$$

where the goal is to learn the distribution of $p(\mathbf{x})$ via G_x, and generate realistic fake landmark gene samples.

The adversarial loss for training the generator network G_y and discriminator network D_y is formulated as:

$$\min_{G_y}\max_{D_y}\ \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\big[\log(D_y(\mathbf{y}))\big]+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1-D_y(G_y(\mathbf{z}))\big] \quad (2)$$

where the goal is to learn the distribution of $p(\mathbf{y})$ using G_y, and generate realistic fake target gene samples.

The min-max adversarial loss for training the networks D_{xy}, G_x, G_y is formulated as:

$$\begin{aligned}\min_{G_x,G_y}\max_{D_{xy}}\ &\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\log(D_{xy}(\mathbf{x},\mathbf{y}))\big]\\&+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1-D_{xy}(G_x(\mathbf{z}),G_y(\mathbf{z}))\big]\end{aligned} \quad (3)$$

where the goal is to learn the corresponding relationship between the paired landmark and target gene expressions. Note that we consider the shared random input $\mathbf{z}$ for both generators to learn the joint distribution of landmark and target genes as $p(\mathbf{x},\mathbf{y}|\mathbf{z})$. In addition to the labeled data $(\mathbf{x}^l,\mathbf{y}^l)$, we suppose $(\mathbf{x}^u,F(\mathbf{x}^u))$ as the real paired data in the above loss function, when the predictions of inference network are good enough after a few training epochs.

The auxiliary reconstruction loss function for training the inverse network I_x and the generator network G_x is:

$$\min_{I_x,G_x}\ \mathbb{E}_{(\mathbf{x})\sim p(\mathbf{x})}\big[\|\mathbf{x}-G_x(I_x(\mathbf{x}))\|_1\big] \quad (4)$$

The auxiliary translation loss function for training I_x and G_y is:

$$\min_{I_x,G_y}\ \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\|\mathbf{y}-G_y(I_x(\mathbf{x}))\|_1\big] \quad (5)$$

We also help training of the inverse network with the following loss:

$$\min_{I_x}\ \mathbb{E}_{(\mathbf{z})\sim p(\mathbf{z})}\big[\|I_x(G_x(\mathbf{z}))-\mathbf{z})\|_1\big] \quad (6)$$

The loss function for training the inference network F is:

$$\begin{aligned}\min_{F}\ &\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\|\mathbf{y}-F(\mathbf{x})\|_1\big]+\mathbb{E}_{(\mathbf{z})\sim p(\mathbf{z})}\big[\|G_y(\mathbf{z})-F(G_x(\mathbf{z}))\|_1\big]\\&+\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[\|F(\mathbf{x}\oplus\mathbf{e})-F(\mathbf{x}\oplus\mathbf{e}')\|^2\big]\end{aligned} \quad (7)$$

where the first term is the ℓ_1-norm loss using the original labeled data $(\mathbf{x}^l,\mathbf{y}^l)$, the second term is the ℓ_1-norm loss using the pseudo-labeled data $(\hat{\mathbf{x}}^z,\hat{\mathbf{y}}^z)$ synthesized by the generators, and the last term is the consistency loss that requires similar outputs for an input with different added noises $\mathbf{e}$ and $\mathbf{e}'$. It is worth mentioning that we empirically get better results using the ℓ_1-norm loss for the first two terms compared to the ℓ_2-norm loss, which shows the advantages of robust ℓ_1-norm loss in the gene expression problem.

In our gene expression completion problem, we adopt a variant of mini-batch SGD methods to update the parameters in the networks for an efficient and stable update. We summarize the optimization steps of SemiGAN in Algorithm 1 by considering the empirical approximation of the expectations in the aforementioned loss functions.

Algorithm 1 Optimization of SemiGAN via mini-batch SGD method.

Input: Labeled gene expression dataset $\Omega_l=\{(\mathbf{x}_i^l,\mathbf{y}_i^l)\}_{i=1}^{n_l}$ which corresponds to the profiles with the measurement of both landmark and target genes; and $\Omega_u=\{\mathbf{x}_j^u\}_{j=1}^{n_u}$ representing the profiles with only landmark gene expression measurement available. Hyper-parameter λ_{D_x}, λ_{D_y}, $\lambda_{D_{xy}}$, λ_{G_x}, λ_{G_y}, $\lambda_{G_{xy}}$, λ_{tra}, λ_{rec}, λ_{inv}, λ_{syn} and λ_{con}.

Initialize parameter θ_{D_x}, θ_{D_y} and $\theta_{D_{xy}}$ for discriminators, parameter θ_{G_x}, θ_{G_y} for generators, parameter θ_{I_x} for the inverse network I_x and parameter θ_F for the inference network F.

for number of training iterations **do**

  for t = 1,..., T **do**

    Randomly choose mini-batch $\Omega_l^t\subset\{1,\dots,n_l\}$ of size b and mini-batch $\Omega_u^t\subset\{1,\dots,n_i\}$ of size b.

    Update the parameters θ_{D_x}, θ_{D_y} and $\theta_{D_{xy}}$ by ascending along the stochastic gradient *w.r.t.* the following adversarial loss.

$$\begin{aligned}\max_{D_x,D_y,D_{xy}}\ &\frac{1}{b}\sum_{i=1}^{b}\lambda_{D_x}\log(D_x(\mathbf{x}_i^u))+\lambda_{G_x}\log(1-D_x(G_x(\mathbf{z}_i)))\\&+\lambda_{D_y}\log(D_y(\mathbf{y}_i^l))+\lambda_{G_y}\log(1-D_y(G_y(\mathbf{z}_i)))\\&+\lambda_{D_{xy}}\log(D_{xy}(\mathbf{x}_i^l,\mathbf{y}_i^l))\\&+\lambda_{G_{xy}}\log(1-D_{xy}(G_x(\mathbf{z}_i),G_y(\mathbf{z}_i)))\end{aligned}$$

    Update the parameters θ_{G_x} and θ_{G_y} by descending along the stochastic gradient *w.r.t.* the following loss.

$$\begin{aligned}\min_{G_x,G_y}\ &\frac{1}{b}\sum_{i=1}^{b}\lambda_{G_x}\log(1-D_x(G_x(\mathbf{z}_i)))+\lambda_{G_y}\log(1-D_y(G_y(\mathbf{z}_i)))\\&+\lambda_{G_{xy}}\log(1-D_{xy}(G_x(\mathbf{z}_i),G_y(\mathbf{z}_i)))\\&+\lambda_{rec}||G_x(I_x(\mathbf{x}_i^u))-\mathbf{x}_i^u||_1\\&+\lambda_{tra}||\mathbf{y}_i^l-G_y(I_x(\mathbf{x}_i^l))||_1\end{aligned}$$

    Update the parameters θ_{I_x} by descending along its stochastic gradient *w.r.t.* the following loss.

$$\begin{aligned}\min_{I_x}\ &\frac{1}{b}\sum_{i=1}^{b}\lambda_{rec}||G_x(I_x(\mathbf{x}_i^u))-\mathbf{x}_i^u||_1+\lambda_{tra}||\mathbf{y}_i^l-G_y(I_x(\mathbf{x}_i^l))||_1\\&+\lambda_{inv}\|I_x(G_x(\mathbf{z}))-\mathbf{z})\|_1\end{aligned}$$

    Update the parameters θ_F by descending along its stochastic gradient *w.r.t.* the following loss.

$$\begin{aligned}\min_{F}\ &\frac{1}{b}\sum_{i=1}^{b}||G_y(\mathbf{z}_i)-F(G_x(\mathbf{z}_i))||_1+\lambda_{syn}||\mathbf{y}_i^l-F(\mathbf{x}_i^l)||_1\\&+\lambda_{con}\|F(\mathbf{x}_i^u\oplus\mathbf{e})-F(\mathbf{x}_i^u\oplus\mathbf{e}')\|^2\end{aligned}$$

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. We download three different publicly available datasets from https://cbcl.ics.uci.edu/public_data/D-GEX/ for this analysis, which includes: the microarray-based GEO dataset, the RNA-Seq-based GTEx dataset data and the 1000 Genomes (1000G) RNA-Seq expression data.

The original GEO dataset consists of 129158 gene expression profiles corresponding to 22268 probes (978 landmark genes and 21290 target genes) that are collected from the Affymetrix microarray platform. The original GTEx dataset is composed of 2921 profiles from the Illumina RNA-Seq platform in the format of Reads Per Kilobase per Million (RPKM). While the original 1000G dataset includes 2921 profiles from the Illumina RNA-Seq platform in the format of RPKM.

We follow the pre-processing steps in [5] for duplicate samples removal, joint quantile normalization and cross-platform data matching. In the joint quantile normalization, we map the expression values in the GTEx and 1000G datasets according to the quantile computed in the GEO data. The expression value has been quantile normalized to the range between 4.11 and 14.97. Finally, the expression value of each gene has been normalized to zero mean and unit variance. After pre-processing, there are a total of 111009 profiles in the GEO dataset, 2921 profiles in the GTEx dataset while 462 profiles in the 1000G dataset. All the profiles correspond to 10463 genes (943 landmark genes and 9520 target genes).

4.1.2 Evaluation Criterion. In the experiments, we use two different evaluation metrics, including mean absolute error (MAE) and concordance correlation (CC). Given a set of testing data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, for a certain model we denote the predicted expression set as $\{\hat{\mathbf{y}}_i\}_{i=1}^n$. The definition of MAE is:

$$MAE_t = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_{jt} - y_{jt}|,$$

where y_{it} represents the expression value for the t-th target gene in the i-th testing profile, and $\hat{y}_{it}$ indicates the corresponding predicted value. MAE_t is the MAE value for the t-th target gene.

The following equation shows the definition of CC:

$$CC_t = \frac{2\rho\sigma_{\mathbf{y}_t}\sigma_{\hat{\mathbf{y}}_t}}{\sigma^2_{\mathbf{y}_t} + \sigma^2_{\hat{\mathbf{y}}_t} + (\mu_{\mathbf{y}_t} - \mu_{\hat{\mathbf{y}}_t})^2},$$

where CC_t indicates the concordance correlation for the t-th target gene. ρ is the Pearson correlation, while $\mu_{\mathbf{y}_t}$, $\mu_{\hat{\mathbf{y}}_t}$, and $\sigma_{\mathbf{y}_t}$, $\sigma_{\hat{\mathbf{y}}_t}$ are the mean and standard deviation of $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ respectively.

4.1.3 Baseline Methods. In the LINCS program, the gene expression inference is based on the least square regression (LSR) model:

$$\min_{W,\mathbf{b}} \sum_{i=1}^{n_l} ||W^T\mathbf{x}_i^l + \mathbf{b} - \mathbf{y}_i^l||^2$$

where W is the weight matrix and $\mathbf{b}$ is the bias term. The learning is based on the labeled profiles $\Omega_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{n_l}$. The LSR model is prone to overfit the training model, and therefore has limited prediction power. To deal with the overfitting problem, we also consider two other linear regression models in the comparison, which are ridge regression, *i.e.,* LSR with ℓ_2-norm regularization (LSR-L2) and LASSO regression, *i.e.,* LSR with ℓ_1-norm regularization (LSR-L1).

Besides the linear regression models, we also compare with the k nearest neighbor (KNN) method for regression, where the prediction of a given profile is formulated as the average of its k nearest profiles. Moreover, we compare with a deep learning method for gene expression inference (D-GEX) [5] to validate the performance of our SemiGAN model. The D-GEX model use a fully connected multi-layer perceptron for regression. To the best of our knowledge, D-GEX is the only model that apply deep learning frameworks to the gene expression inference problem.

Following the experimental settings in [5], we evaluate the methods under two different circumstances. Firstly, we use 80% of the GEO data for training, 10% of the GEO data for validation while the other 10% of the GEO data for testing. Secondly, we use the same 80% of the GEO data for training, the 1000G data for validation while the GTEx data for testing. Among the training data, we set the portion of labeled profiles to be $\{1\%, 3\%, 5\%, 10\%, 20\%\}$ respectively and leave the remaining as unlabeled. In the second scenario, the training, validation and testing comes from different platforms, which is designed to validate if comparing methods are capable of capturing the information for cross-platform prediction. We use the training data to construct the predictive model, validation data for model selection and parameter setting, while the testing data to conduct the evaluation. For LSR-L1 and LSR-L2 model, we tune the hyperparameter λ in the range of $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$ according to the performance on the validation data. For each method, we follow the experimental protocol in [5] and report the average performance and standard deviation over all target genes on the testing data.

4.1.4 Implementation Details. We use networks with similar architecture for the both datasets, train the networks only using the training data, tune the hyper-parameters via the validation samples, and report the results on the test sets. For the inference network, we utilize a DenseNet [18] architecture with three hidden layers, each one containing $3,000$ hidden units. For the generators and discriminators, we use fully connected networks with three and one hidden layers respectively, where all the hidden layers include $3,000$ hidden units. The similar architecture to the generator is considered for the inverse network. We consider leaky rectified linear unit (LReLU) [24] with leakiness ratio 0.2 as the activation function of all layers except the last layer of generator network, which has linear function due to the mean-zero and unit-variance data normalization. Moreover, we set the maximum and minimum learning rates to 5×10^{-4} and 1×10^{-5} respectively, and linearly decrease it during training till the maximum epoch 500. Adam algorithm [21] is adopted as our optimization method with the default hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. The batch size is set to 200. We also utilize weight normalization [34] as layer normalization to speed up the convergence of training process. The parameters of all layers are all initialized by Xavier approach [13]. We use Theano toolbox for writing our code, and run the algorithm in a machine with one Titan X pascal GPU.

Table 1: MAE comparison on the prediction of GEO data when using different portion of labeled data. Better results correspond to lower MAE value. The best result is marked in bold.

Methods	1%	3%	5%	10%	20%	100%
LSR	1.6789±0.4747	0.4939±0.1100	0.4435±0.0979	0.4080±0.0906	0.3914±0.0872	0.3763±0.0844
LSR-L1	0.4507±0.0924	0.4181±0.0837	0.4119±0.0820	0.4072±0.0809	0.4051±0.0805	0.3756±0.0841
LSR-L2	0.4363±0.0813	0.4072±0.0840	0.3992±0.0849	0.3912±0.0855	0.3849±0.0854	0.3758±0.0842
KNN	0.5299±0.0886	0.4847±0.0898	0.4659±0.0901	0.4407±0.0906	0.4173±0.0918	0.3708±0.0958
D-GEX	0.4542±0.0916	0.4077±0.0822	0.3891±0.0858	0.3735±0.0862	0.3514±0.0862	0.3204±0.0879
SemiGAN	**0.4202±0.0876**	**0.3818±0.0883**	**0.3651±0.0878**	**0.3432±0.0873**	**0.3245±0.0871**	**0.2997±0.0869**

Table 2: CC comparison on the prediction of GEO data when using different portion of labeled data. Better results correspond to higher CC value. The best result is marked in bold.

Methods	1%	3%	5%	10%	20%	100%
LSR	0.2429±0.1208	0.7409±0.1209	0.7774±0.1110	0.8008±0.1035	0.8121±0.0996	0.8227±0.0956
LSR-L1	0.7460±0.1211	0.7737±0.1102	0.7778±0.1089	0.7811±0.1077	0.7817±0.1078	0.8221±0.0960
LSR-L2	0.7403±0.1197	0.7838±0.1091	0.7948±0.1062	0.8058±0.1026	0.8131±0.0998	0.8223±0.0959
KNN	0.6409±0.1352	0.7097±0.1190	0.7314±0.1144	0.7586±0.1098	0.7818±0.1063	0.8218±0.1001
D-GEX	0.7504±0.1202	0.7892±0.1094	0.8012±0.1072	0.8188±0.1028	0.8316±0.0992	0.8514±0.0908
SemiGAN	**0.7606±0.1187**	**0.8013±0.1096**	**0.8155±0.1069**	**0.8346±0.1026**	**0.8503±0.0988**	**0.8702±0.0927**

Table 3: MAE comparison on the prediction of GTEx data when using different portion of labeled data. Better results correspond to lower MAE value. The best result is marked in bold.

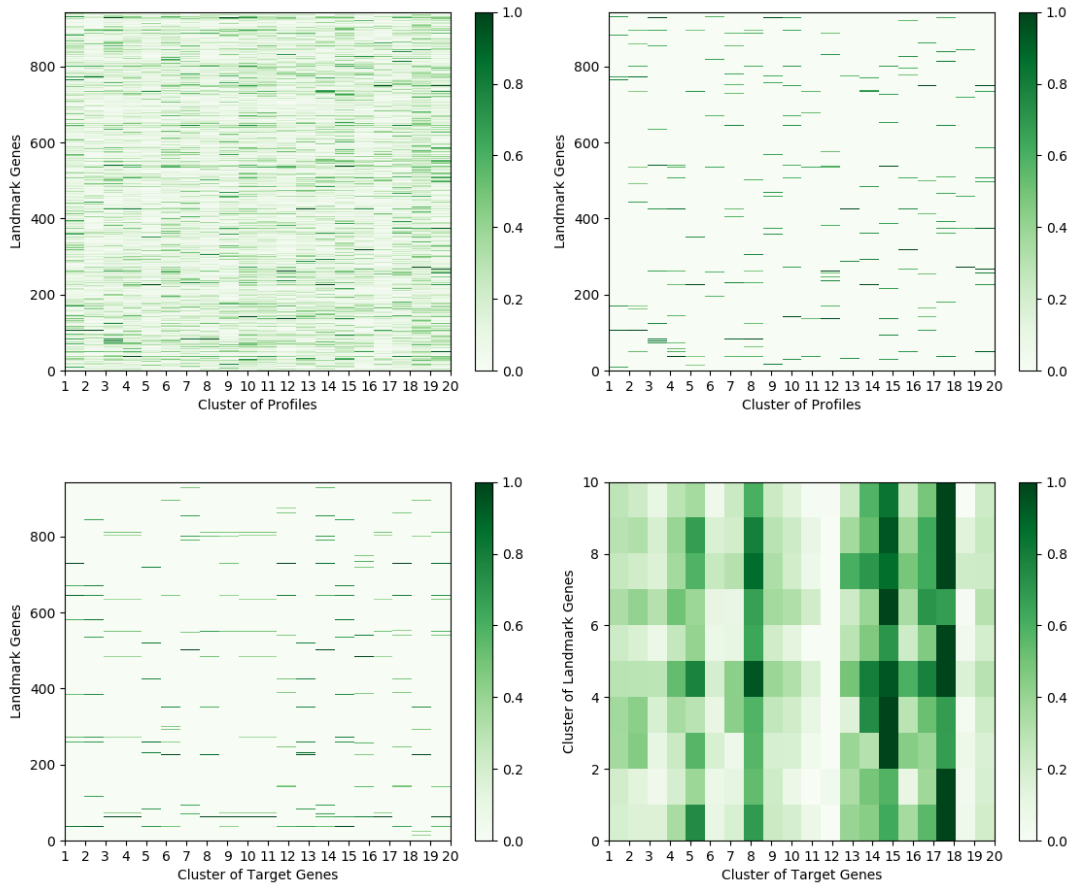
Methods	1%	3%	5%	10%	20%	100%
LSR	2.1908±0.6561	0.6307±0.1463	0.5630±0.1338	0.5170±0.1277	0.4936±0.1254	0.4704±0.1235
LSR-L1	0.5431±0.1319	0.4970±0.1269	0.4910±0.1265	0.4844±0.1265	0.4815±0.1267	0.4669±0.1274
LSR-L2	0.5190±0.1183	0.4901±0.1206	0.4868±0.1214	0.4818±0.1227	0.4775±0.1234	0.4682±0.1233
KNN	0.6758±0.1367	0.6530±0.1467	0.6502±0.1454	0.6375±0.1468	0.6324±0.1469	0.6225±0.1469
D-GEX	0.5385±0.1244	0.4847±0.1212	0.4922±0.1224	0.4656±0.1262	0.4505±0.1252	0.4393±0.1239
SemiGAN	**0.5105±0.1202**	**0.4748±0.1233**	**0.4643±0.1232**	**0.4470±0.1239**	**0.4341±0.1250**	**0.4223±0.1266**

Table 4: CC comparison on the prediction of GTEx data when using different portion of labeled data. Better results correspond to higher CC value. The best result is marked in bold.

Methods	1%	3%	5%	10%	20%	100%
LSR	0.1669±0.1409	0.6265±0.2123	0.6647±0.2112	0.6923±0.2101	0.7050±0.2097	0.7184±0.2072
LSR-L1	0.6259±0.2237	0.6666±0.2154	0.6673±0.2163	0.6703±0.2175	0.6693±0.2188	0.7163±0.2188
LSR-L2	0.6131±0.2203	0.6766±0.2121	0.6867±0.2115	0.6997±0.2110	0.7070±0.2105	0.7181±0.2076
KNN	0.4617±0.2139	0.5210±0.2125	0.5286±0.2111	0.5509±0.2089	0.5597±0.2078	0.5748±0.2052
D-GEX	0.6288±0.2115	0.6818±0.2128	0.6823±0.2120	0.7016±0.2112	0.7189±0.2115	0.7304±0.2072
SemiGAN	**0.6389±0.2188**	**0.6928±0.2135**	**0.7026±0.2131**	**0.7205±0.2118**	**0.7317±0.2113**	**0.7443±0.2087**

4.2 Comparison on the GEO Data

In this subsection, we evaluate the methods on the prediction of target gene expression in GEO data. From the summarization in Table 1 and 2, we can notice apparent improvement of our model over the counterparts. Firstly, we can find deep learning models (D-GEX and SemiGAN) consistently outperform all linear models (LSR, LSR-L1 and LSR-L2), since the deep neural network is capable of interpreting the non-linear association among gene expression patterns. Deep learning models indicate remarkable representation power to estimate the latent data distribution thus make better prediction for the expression of target genes. Besides, KNN shows worse results than the comparing methods, which is because of the inconsistence between the nearest neighbors in the training and testing data. Moreover, our SemiGAN model presents consistent advantage over the comparing deep model, D-GEX, due to the following two reasons: 1) the semi-supervised framework in our model enables the integration of unlabeled profiles in the learning, which strengthens the estimation of the data distribution and also



(c) Relevance score of landmark genes *w.r.t.* cluster of target genes (d) Relevance score of landmark gene clusters *w.r.t.* target genes clusters

Figure 2: Visualization of the relevance score calculated for each landmark gene in the prediction of GEO data. (a) We divide the gene expression profiles into 20 clusters using K-means and plot the contribution of each landmark gene to different profile clusters. (b) For each profile cluster, only the top 20 landmark genes in (a) are kept for a clear illustration. (c) The 9520 target genes are grouped into 20 clusters via K-means and the cleaned version of landmark gene contribution is presented. (d) The landmark genes are clustered into 10 groups and the contribution to the prediction of different target gene clusters is plotted.

introduces more data to train the inference network; 2) the estimation of both conditional distribution $p(\mathbf{y}|\mathbf{x})$ and joint distribution $p(\mathbf{x}, \mathbf{y})$ provides guidance for each other, such that the training of both generator and inference framework can be improved.

Moreover, we can notice that the superiority of SemiGAN model is more obvious with the labeled portion being 10% and 20%. When the labeled portion is too small, all methods are influenced by the limited number of labeled profiles. However, with just 10% labeled profiles available, the generators in our model can approximately estimate the joint distribution $p(\mathbf{x}, \mathbf{y})$ and produce reliable profiles to improve the learning of inference network. Conversely, the construction of the inference network also guide the generators to produce realistic gene expression profiles. This result validates that the SemiGAN can make good prediction of the target gene expression given very limited number of labeled profiles, which provides an accurate and cost-effective strategy for reliable genome-wide expression profiling.

4.3 Comparison on the GTEx Data

Furthermore, we evaluate the comparing methods on the cross-platform prediction, where we use GEO data for training, 1000G data for validation while GTEx data for testing. This cross-platform setting is used to test if the methods can capture appropriate information for predicting target gene expression from a different platform. As we can notice from the comparison results in Table 3 and 4, our SemiGAN model maintains significant advantage over the counterparts. Since the training and testing data come from different platform (*i.e.,* different data distribution), the performance on the GTEx data is not as good as the one for GEO data prediction. In the cross-platform prediction, SemiGAN still performs better, which validates that our model can take advantage of the learning of both conditional distribution and joint distribution to strengthen the cross-platform learning.

4.4 Analysis of Landmark Genes in the Prediction

In this subsection, we look into the roles of landmark genes in the prediction of target gene expression. We use the Layer-wise Relevance Propagation (LRP) [2] method to calculate the importance of each landmark gene and plot the illustration figure in Fig. 2. The LRP method calculates the relevance score for each landmark gene, where higher relevance score shows more contribution in the overall prediction of the target gene expression. Firstly, we analyze the contribution of each landmark gene for different profiles. Since there are a large number of profiles, we divide them into 20 groups and show the accumulated relevance score pattern for each profile group in Fig. 2 (a) and (b). We can notice that the landmark gene expression patterns vary for different profile groups, which replicates the previous findings in cancer clustering analysis that different group of cancer samples usually exhibit different expression patterns. The breast cancer subtype discovery study indicates different expression-based prognostic signatures for different subtypes [11]. And cancer landscape study also identified that cross-tissue cancer clusters can be characterized by different gene expression patterns [37]. Afterwards, we analyze the relationship between landmark genes and target genes. We cluster the target genes into 20 groups and calculate the relevance score for each target gene cluster, where we plot the overall contribution of each landmark gene across all profiles. To make a clear illustration, we group the landmark genes into 10 clusters and display the association between landmark gene clusters and target gene clusters in Fig. 2 (d). Similar to the results between profiles and landmark genes, apparent difference in the relevance patterns can also be observed for different target gene clusters. This finding has also been validated by previous gene cluster analysis [25], where gene cluster information is related to the structure of biosynthetic pathways and metabolites.

5 CONCLUSION

In this paper, we put forward a novel deep generative model (SemiGAN). We formulated our model in a semi-supervised learning approach, in which the learning of our inference network involved not only the labeled profiles but also the generated profiles with no ground truth label available. The prediction results on both GEO and GTEx data validated the performance of our SemiGAN model in gene expression inference. Moreover, by visualizing the role of different landmark genes in the prediction, we revealed interesting relationship among gene expression patterns that have been validated in previous literature. It is notable that our SemiGAN is an effective model in semi-supervised regression tasks, where the input data in labeled and unlabeled sections share similar distribution. The semi-supervised learning setting introduced more reliable profiles to strengthen the training of the inference network. In addition, the construction of the inference network also improved the estimation of the joint distribution between the input data and label.

REFERENCES

[1] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* 33, 8 (2015), 831.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140.

[3] Mostapha Benhenda. 2017. ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity? *arXiv preprint arXiv:1708.08227* (2017).

[4] Alvis Brazma, Helen Parkinson, Ugis Sarkans, Mohammadreza Shojatalab, Jaak Vilo, Niran Abeygunawardena, Ele Holloway, Misha Kapushesky, Patrick Kemmeren, Gonzalo Garcia Lara, et al. 2003. ArrayExpressâĂŤa public repository for microarray gene expression data at the EBI. *Nucleic acids research* 31, 1 (2003), 68–71.

[5] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. 2016. Gene expression inference with deep learning. *Bioinformatics* 32, 12 (2016), 1832–1839.

[6] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. 2017. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems*. 4091–4101.

[7] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning (ICML)*. ACM, 160–167.

[8] Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in neural information processing systems (NIPS)*. 1486–1494.

[9] Pietro Di Lena, Ken Nagata, and Pierre Baldi. 2012. Deep architectures for protein contact map prediction. *Bioinformatics* 28, 19 (2012), 2449–2457.

[10] Ron Edgar, Michael Domrachev, and Alex E Lash. 2002. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic acids research* 30, 1 (2002), 207–210.

[11] Greg Finak, Nicholas Bertos, Francois Pepin, Svetlana Sadekova, Margarita Souleimanova, Hong Zhao, Haiying Chen, Gulbeyaz Omeroglu, Sarkis Meterissian, Atilla Omeroglu, et al. 2008. Stromal gene expression predicts clinical outcome in breast cancer. *Nature medicine* 14, 5 (2008), 518.

[12] Hongchang Gao, Xiaoqian Wang, and Heng Huang. 2016. New Robust Clustering Model for Identifying Cancer Genome Landscapes. *IEEE International Conference on Data Mining (ICDM 2016)* (2016), 151–160.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 249–256.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*. 2672–2680.

[15] Xiaobo Guo, Ye Zhang, Wenhao Hu, Haizhu Tan, and Xueqin Wang. 2014. Inferring nonlinear gene regulatory networks from gene expression data based on distance correlation. *PloS one* 9, 2 (2014), e87446.

[16] Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. 2016. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems* 2, 4 (2016), 239–250.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.

[18] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (2017).

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

[20] Alexandra B Keenan, Sherry L Jenkins, Kathleen M Jagodnik, Simon Koplev, Edward He, Denis Torre, Zichen Wang, Anders B Dohlman, Moshe C Silverstein, Alexander Lachmann, et al. 2017. The library of integrated network-based cellular signatures NIH program: system-level cataloging of human cells response to perturbations. *Cell systems* (2017).

[21] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. 1097–1105.

[23] Michael KK Leung, Hui Yuan Xiong, Leo J Lee, and Brendan J Frey. 2014. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, 12 (2014), i121–i129.

[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, Vol. 30.

[25] Marnix H Medema, Renzo Kottmann, Pelin Yilmaz, Matthew Cummings, John B Biggins, Kai Blin, Irene De Bruijn, Yit Heng Chooi, Jan Claesen, R Cameron Coates, et al. 2015. Minimum information about a biosynthetic gene cluster. *Nature chemical biology* 11, 9 (2015), 625.

[26] Bradlee D Nelms, Levi Waldron, Luis A Barrera, Andrew W Weflen, Jeremy A Goettel, Guoji Guo, Robert K Montgomery, Marian R Neutra, David T Breault, Scott B Snapper, et al. 2016. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. *Genome biology* 17, 1 (2016), 201.

[27] Vasilis Ntranos, Govinda M Kamath, Jesse M Zhang, Lior Pachter, and N Tse David. 2016. Fast and accurate single-cell RNA-seq analysis by clustering of transcript-compatibility counts. *Genome biology* 17, 1 (2016), 112.

[28] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. 2017. SEGAN: Speech Enhancement Generative Adversarial Network. *arXiv preprint arXiv:1703.09452* (2017).

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2536–2544.

[30] Matthew G Rees, Brinton Seashore-Ludlow, Jaime H Cheah, Drew J Adams, Edmund V Price, Shubhroz Gill, Sarah Javaid, Matthew E Coletti, Victor L Jones, Nicole E Bodycombe, et al. 2016. Correlating chemical sensitivity and basal gene expression reveals mechanism of action. *Nature chemical biology* 12, 2 (2016), 109.

[31] Jonas Richiardi, Andre Altmann, Anna-Clare Milazzo, Catie Chang, M Mallar Chakravarty, Tobias Banaschewski, Gareth J Barker, Arun LW Bokde, Uli Bromberg, Christian Büchel, et al. 2015. Correlated gene expression supports synchronous activity in brain networks. *Science* 348, 6240 (2015), 1241–1244.

[32] Adriana Romero, Pierre Luc Carrier, Akram Erraqabi, Tristan Sylvain, Alex Auvolat, Etienne Dejoie, Marc-André Legault, Marie-Pierre Dubé, Julie G Hussin, and Yoshua Bengio. 2016. Diet Networks: Thin Parameters for Fat Genomic. *arXiv preprint arXiv:1611.09340* (2016).

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*. 2234–2242.

[34] Tim Salimans and Diederik P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. 901–909.

[35] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. 2016. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron* 92, 2 (2016), 342–357.

[36] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. 2016. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* 32, 17 (2016), i639–i648.

[37] Nora K Speicher and Nico Pfeifer. 2015. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics* 31, 12 (2015), i268–i275.

[38] Matt Spencer, Jesse Eickholt, and Jianlin Cheng. 2015. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM transactions on computational biology and bioinformatics* 12, 1 (2015), 103–112.

[39] Philip J Stephens, Patrick S Tarpey, Helen Davies, Peter Van Loo, Chris Greenman, David C Wedge, Serena Nik-Zainal, Sancha Martin, Ignacio Varela, Graham R Bignell, et al. 2012. The landscape of cancer genes and mutational processes in breast cancer. *Nature* 486, 7403 (2012), 400.

[40] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin Van Der Kooy, Matthew J Marton, Anke T Witteveen, et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *nature* 415, 6871 (2002), 530.

[41] Larry Wasserman and John D Lafferty. 2008. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems*. 801–808.

[42] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. 2017. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5485–5493.

[43] Muhammed A Yıldırım, Kwang-Il Goh, Michael E Cusick, Albert-László Barabási, and Marc Vidal. 2007. Drug-target network. *Nature biotechnology* 25, 10 (2007), 1119–1126.

[44] Jian Zhou and Olga G Troyanskaya. 2015. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods* 12, 10 (2015), 931.

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* (2017).