A

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

by

# APPROVAL SHEET

This

is submitted in partial fulfillment of the requirements
for the degree of

Author:

Advisor:

Advisor:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

# Estimating Cell-type-Specific Fractions with Autoencoders

Xin Shu

(ABSTRACT)

In this work, we introduce autoencoder architectures and the evolution of these methods leading to disentangled representation learning. We apply three autoencoder architectures to single-cell RNA sequencing (scRNAseq) data. ScRNAseq characterizes cellular heterogeneity by measuring the expression profiles of individual cells. However, this measurement remains relatively expensive compared with bulk RNAseq, where expression profiles are averaged over many cells of various cell types and at different cell states, preventing us from capturing cellular heterogeneity. We propose a new bulk RNA-seq data deconvolution method, termed expDC, to estimate cell-type-specific proportions from bulk RNA-seq data. The latent codes of autoencoders offer additional interpretability to explore the grouping of cell types. To do this, we first learn reliable and denoised representations for each cell type given single-cell RNAseq as a reference, then we use these representations to deconvolute simulated bulk RNAseq data to infer cell-type-specific proportions. Our method estimates cell-type composition in a two-stage process. We evaluate our methods on three PBMC datasets and found that a shallow autoencoder architecture performs best in deconvolution.

# Acknowledgments

I would like to thank both my advisors, Professor Zhang and Professor Bekiranov, for being supportive throughout this journey. Their patience and kindness have enlightened me to understand the essence of academic research, and moreover, encouraged my passion for lifelong learning. I would not have become who I am today without your guidance. Last but not least, I am grateful to all my friends and family for their warm support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Autoencoder is an unsupervised learning approach capable of learning compressed representations in the latent space. More generally, an autoencoder is a neural network that prioritizes the efficient compression and reconstruction of input features at the output layer [1]. A compressed representation is learned in an autoencoder's latent space or Information Bottleneck (IB), and these representations are called latent codes, or latent representations. Working with representations of lower dimensions reduces computational costs compared with dealing with the full input expression matrix.

Variational autoencoder (VAE) [2] is an extension of the autoencoder architecture that adds a distribution constraint (usually Gaussian for its property of Central Limit Theorem [3]) on the formations of latent codes. Many innovations in autoencoders have added characteristic constraints on the latent codes such that these codes would satisfy certain properties. For instance, the Gaussian constraint on the variational bottleneck ensures that the latent codes have a spread of an isotropic Gaussian ball, this nice property has inspired an area of research called disentanglement, where a single perturbation on one latent factor leads to changes in quantifiable factors of variation [4]. With applications of VAE, scVI [5] is one of the well-known tools for scRNAseq that uses a variational autoencoder to learn probabilistic representations with reduced technical noise and bias. These representations are then used in many

downstream tasks, for instance, clustering, imputation, and differential expression analysis. scVI explicitly models both library size and batch effect, which are nuisance factors in scRNAseq data. In contrast, in Deep Count Autoencoder (DCA) [6], each cell is represented as a deterministic point or a single latent code in the latent space. Instead of using a distribution to constrain the latent codes, DCA uses a count-based distributional loss for reconstruction under the assumptions of data following a Zero-Inflated Negative Binomial (ZINB) distribution.

The success of VAEs in generation tasks motivated the emergence of controllable generation, where a single perturbation on one latent factor leads to changes in quantifiable factors of variation [4]. There have been discussions in the community debating on whether disentanglement benefits downstream tasks, using the latent codes as input to another method, and the role of variational inference in disentanglement [7, 8]. Still, we are interested in whether having a distributional constraint on the latent space improves our cell-type-specific proportion estimation.

Generative Adversarial Network (GAN) is another approach for generation, having a generator and a discriminator playing a zero-sum game. There is yet a principled way to combine the benefits of both GAN and VAE. These architectures have distinct objectives: VAE approximates the posterior distribution, $q(z|x)$, and maximizes evidence lower bound (ELBO); GAN maximizes the mutual information between a generator network and a discriminator network. VAE possesses more training stability compared with GAN, however, its reconstruction quality is inferior to GAN. Generally, VAE uses an isotropic Gaussian prior to regularizing its latent codes, which pushes Gaussian balls toward the center. Nevertheless, when the regularization is too strong, posterior collapse can occur. In the results section, we simulate the strength of the Gaussian prior to a posterior collapse using $\beta$VAE.

Gene expression profiling is one of the most popular approaches in characterizing cellular states or disease states [9]. Such profiling technique allows us to observe the expression pattern of a cell at the transcription level, for instance, tumor evolution and response to treatments [10]. Compared with bulk RNAseq, scRNAseq profiles individual cells thus capturing cellular heterogeneity within cell populations. Nevertheless, bulk RNAseq is used in large cohorts of clinical studies due to its inexpensiveness and experimental simplicity, relative to scRNAseq. For example, The Cancer Genome Atlas (TCGA) has characterized over 20,000 samples, including normal and primary cancer samples, across 33 cancer types using bulk RNAseq [11]. These data still hold substantial value in research areas such as cancer classification [12], risk stratification [13], transcriptome profiling [14], biomarker identification [15], etc.

Unfortunately, having bulk RNAseq data in abundance does not mitigate its flaw, namely an inability to account for cellular heterogeneity as bulk RNAseq expressions are averaged over cells of heterogeneous cell types. Blood and tumor samples are known for their heterogeneity [16], which posits a challenge for quantifying changes in gene expression levels. Still, it is possible to uncover the cell type composition of these samples to correlate with changes in their gene expression. This class of tools is termed deconvolution or decomposition methods.

Over the years, deconvolution methods have evolved from simple linear regression methods, such as non-negative least squares (NNLS) [17] and ordinary least squares (OLS), [18] to support vector regression (SVR), notably CIBERSORT [19] and CIBERSORTx [20], to capture existing non-linearity in gene expression profiles. Furthermore, [21] showed that simple linear regression methods without transformation perform poorly. Perhaps it is inevitable to redeem non-linearity since gene expression

data are essentially high-dimensional and sparse. Correspondingly, deep neural networks (DNNs) achieve state-of-the-art performance on classification and regression tasks as a universal function approximator. DNNs' capacity to model higher-order relationships within data renders them superior to classical linear regression methods. [22] developed an ensemble of DNNs called Scaden to estimate cell-type proportions. They also hypothesized that the hidden layers of a DNN yield non-linear combinations of optimal features for cell-type deconvolution.

# Chapter 2

# Literature Review

## 2.1 Autoencoder Architectures

We first introduce the simplest autoencoder. It is deterministic, meaning there is no stochastic sampling occurring in the latent space, nor are the latent codes regularized by any distributional constraints. Its loss function is reconstruction loss only, which is usually the Mean Squared Error (MSE):

$$\mathcal{L}(x, y) = \frac{1}{n} \sum (x - y)^2 \tag{2.1}$$

Our data input $x$, which is a vector or a tensor. Reconstructed output/target $y$ is usually of the same dimension as input $x$. Finally, we have our number of samples $n$.

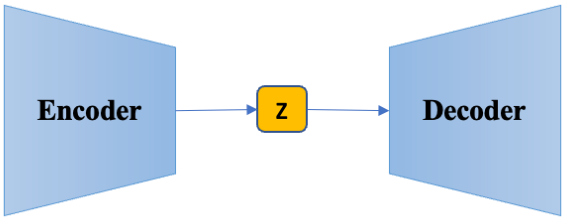


Figure 2.1: Autoencoder architecture

The main components of an autoencoder architecture are an encoder, latent layer(s), and a decoder. The encoder compresses the input to reduced dimensions, the latent

layer(s) to perform transformations that yield latent codes, and the decoder reconstructs the input given the latent codes. This operation is done after many training iterations until the reconstruction is as close to the input as possible. The benefit of this three-component framework is three-fold. Firstly, it denoises the data during its compression phase. Secondly, redundant features are removed in the reduction of dimensionality. Thus, the neural network's learning capacity, i.e., the number of hidden layers and the dimension of latent codes, becomes a hyperparameter that could be adjusted according to specific needs. Thirdly, one could exercise anomaly detection or out-of-distribution (OOD) detection with our models, which can be useful when attempting to transfer learned knowledge. That said, there is no one-size-fits-all choice of design for each problem, and we recommend a careful selection of design choices to optimize model performance.

Next, we have Variational Autoencoder (VAE). This architecture adds variational inference and reparameterization trick. This is achieved by adding a KL-divergence, forcing the latent codes to follow Gaussian distribution in the case of Gaussian VAE.

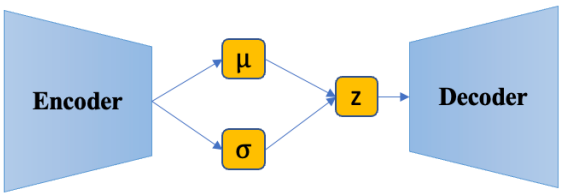


Figure 2.2: Variational Autoencoder architecture

The first term on the RHS is the Kullback-Leibler divergence, and the second term on the RHS is the likelihood of data at reconstruction. More specifically, $q(z|x)$ is the posterior, $p(z)$ is the prior, and $p(x|z)$ is the likelihood of data. Being a maximum likelihood estimation originally, negating the terms yields:

$$\mathcal{L}(x) = D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{2.2}$$

For experiments on image data, we use MSE as reconstruction loss, because our decoder is continuous Gaussian (greyscale or RGB values). If the data follow a discrete distribution rather than continuous, one could consider count distribution as reconstruction (e.g. Poisson, Binomial, etc.)

The next question is, what happens when we weigh the KL term differently? $\beta$-VAE[23] does so by adding more penalty on the KL term, strengthening its factorial prior. The effect of this re-weighting led to disentanglement. Briefly, automatically isolating factors of variations. For example, a latent factor could be a mixture of color, shape, orientation, etc. A successful disentangling of one factor is when we traverse one latent factor, and the rest of the factors remain unchanged.

$$\mathcal{L}(x) = \beta D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{2.3}$$

Then we have $\beta$-TCVAE[24], the authors focused on the KL term of the loss, splitting the original objective to index-code mutual information, total correlation, and dimension-wise KL. The authors found that the most useful term for disentanglement is the total correlation term. Prior work[8, 25] suggests learning a disentangled representation, the two important quantities are 1) mutual information between latent variables and data. 2) independence amongst the latent variables, which is most likely already addressed by having a Gaussian factorial prior in VAE-like architectures.

The loss function of $\beta$-TCVAE is formulated as follows:

$$\begin{aligned}\mathcal{L}(x) = -\mathbb{E}_{x\tilde{q}(z,x)}[\log p(x|z)] + \alpha \mathbf{I}_q(z;n) + \beta D_{KL}(q(z)||\prod_j q(z_j)) \\ +\gamma \sum_j D_{KL}(q(z_j)||p(z_j))\end{aligned} \tag{2.4}$$

The first term is the likelihood of data, the second term is the Index-Code Mutual Information, the third term is the Total Correlation, and the last term is the Dimension-wise KL divergence.

Burgess et al. [26] adds a constant C to $\beta$-VAE. This improves reconstruction quality and information storage. They applied monotonic scheduling of C.

$$\mathcal{L}(x) = \gamma|D_{KL}(q(z|x)||p(z)) - C| - \mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{2.5}$$

## 2.2 Posterior Collapse

Now that we have seen the most representative model architectures, let us dive into another big challenge in training variational autoencoders. This problem is often referred to as posterior collapse, or KL-vanishing, as one can observe the KL term degenerate to 0 during training. Not to be confused with mode collapse, which occurs in generative adversarial networks. Extending from the toy dataset, MNIST, to single-cell gene expression has brought to light identifiability issues in the latent space. While training VAEs, we observed the posterior quickly collapsed to a single isotropic Gaussian ball. This led to poor downstream performance in estimating cell type proportions. To understand the problem of posterior collapse, we showcase the

ongoing debate in the community and our tools to mitigate such a problem.

There are three major schools of idea.

Firstly, [27, 28, 29] finds the strength of the Kullback-Leibler(KL) regularization being too strong. The regularization in conjunction with the expected log-likelihood conditioned on the latent codes forms the Evidence Lower Bound. Thus, heuristically, one would think the most influential term leading to posterior collapse should be the KL regularization, which pushes $q(z|x)$ towards prior $p(z)$. Following these ideas, one trick to mitigate posterior collapse is to warm up, or anneal, the weight in front of the KL term from 0 to 1. [30] introduced a cyclical annealing scheme that instead of monotonic annealing, we repeatedly anneal in cycles from 0 to 1, or over a specified interval. Experiment results show lower reconstruction error and higher KL divergence.

Secondly, [31] argues posterior collapse is a result of optimization issues, i.e. the model being stuck at a bad local minimum. Interestingly, when an AE model is stuck at a bad local minimum, having deeper architecture yields a larger reconstruction error. In this context, adding a KL regularization term only worsens reconstruction. [32] analyzed the loss landscapes of probabilistic PCA (pPCA) and linear variational autoencoder. They found that in pPCA, the stability of the stationary points of the log marginal likelihood is attributed to $\sigma^2$. If $\sigma^2$ increases, we lose our ability to learn more components. As $\sigma^2$ decreases, the likelihood at these stationary points increases. In deep nonlinear VAEs, when $\sigma^2$ initialization is large, the posterior tends to collapse towards the prior. Therefore, it is likely that posterior collapse, in some cases, is an optimization issue.

Thirdly, [33] proposes latent variable non-identifiability, meaning posterior collapse

occurs when the latent variables could take on two or more values. Consequently, when we marginalize $z$, the conditional likelihood becomes the marginal likelihood of the dataset. This result implies that the posterior collapse is intrinsic to the model and data, instead of optimization or inference.

## 2.3 Introduction to scRNAseq analysis

Before we introduce applications of generative models in single-cell RNAseq (scRNAseq). Let us revisit the intricacies of scRNAseq. There are two major classes of scRNAseq technologies [34]. One of them is high-throughput, low-depth, usually the 10x Chromium platform. The other one is low-throughput, high-depth, isolating cells into wells followed by SmartSeq2. We then acquire a gene expression matrix with each entry showing the number of transcripts for each gene per cell. Because the majority of the genes are only expressed in a small subset of cell types, most of the expression matrices are zeros. However, some zeros are genes not detected due to low sequencing depth and insufficient starting material, this is often called dropout events. Silverman et al. [35] discussed dropout events in detail, where they tested various zero count models and found disagreement in finding the most differentially expressed sequences. We brief their guidelines as follows: 1) we could simply use sampling zeros in the model to account for biological zeros. 2) we should avoid zero-inflated models, as they tend to produce spurious conclusions that ignored differentially expressed sequences. Additionally, they found that zero-inflated models add bias when sample-specific complete technical processes are absent. In this regard, they recommend simple models such as Poisson, Negative Binomial to account for zeros.

Another noteworthy problem in scRNAseq is batch correction. As described in [34] technical variations such as time of experiment, technicians doing the experiments, reagent differences could be present in data. For scRNAseq, we could consider mnnCorrect or canonical correlation analysis in Seurat. Nevertheless, there is a possibility of removing biological in the process of batch correction.

It is worth noting the difference between bulk RNAseq and scRANseq. Firstly, bulk RNAseq measures the average expression level for each gene in a population of cells. This is useful for comparing samples of the same tissue from different species. However, bulk RNAseq is insufficient for studying heterogeneity at the cell level. scRNAseq solves this problem by measuring the expression levels of individual cells. In this way, we are able to investigate cell-specific changes. For example, cell type identification [36], cell responses [37], cell state variations [38], gene regulatory network inferences [39].

## 2.4 Applications of Generative Models in scRNAseq

Yu and Welch [40] combined the merits of both VAE and GAN. The training process of VAEs possesses more stability compared with GANs. Moreover, VAEs tend to produce more semantically meaningful latent representations yet generate blurry images. GANs tend to produce sharper images compared with VAEs, but it is notorious for their training instability. Because single-cell RNAseq data are high-dimensional, deep generative methods have great potential to extract useful representations, which could be used for uncovering cellular identity and predicting unseen cell states. This work is built upon scGen [41], which uses mouse data to predict human cell responses to perturbation. MichiGAN improves on ScGen's disentanglement and data gener-

ation. As for evaluation, they used Spearman correlation, Mutual Information Gap (MIG), and the one factor of variation disentanglement metric used in FactorVAE [42]. MIG is the average gap between the largest and the second largest normalized mutual information, where we look for two latent variables with the largest mutual information against a given ground truth variable. If the gap is large, then it is very likely that the ground truth variable is captured by a single latent (largest) variable.

Empirically, PCA has the largest Spearman correlation gap, whereas $\beta$-TCVAE has the best performance in both FactorVAE and MIG metrics. $\beta$-TCVAE's superior performance in the disentanglement of single-cell RNAseq data is consistent with the superior disentanglement performance on image data. cscGAN [43] introduced random forest error to assess the quality of reconstruction. The higher the error, the better the reconstruction, being realistic enough to fool the random forest classifier. Despite having the best disentanglement performance, $\beta$-TCVAE has the worst generation/reconstruction performance. In addition, WGAN-GP [44] performed best at generation/reconstruction.

MichiGAN attempts to train VAE and GAN in conjunction sacrificed both training stability and generation. This paper trains a VAE first, and then GAN. GAN can find complex, multimodal distributions by minimizing the Wasserstein distance between two distributions, generated vs. true. Use VAE to learn latent codes for GAN, then use either posterior means or random samples from the posterior as the condition for GANs. However, the lack of an inference network in GAN prevents us from measuring the mutual information for the generators. This work could be used for 1) predicting high-dimensional data from combinations of latent variables and 2) drug profile expression: First estimate mean profile, then add effects. Interestingly, for biological data, two hidden layers achieve the best training stability.

Lastly, despite all the benefits of using this model, one big limitation is that the latent arithmetic assumes average cell type differences are homogeneous across various treatments. This assumption may not hold where there exists a strong interaction effect between cell type and drug treatment.

In the autoencoder survey paper [45], the authors claim multi-omics is the key to understanding the associated phenotype. Multiomics usually consists of genome, proteome, transcriptome, epigenome, metabolome, etc. By combining multiple datasets, we are compensating for missing data and errors occurring in a single dataset. This should more reliably capture the complexity of Rare Diseases, which are diseases impacting a small percentage of the population and are often undiagnosed.

They provided insights for the following autoencoder architectures/methods. AE's dimensionality reduction and reconstruction retain the information that captures the most variability in the bottleneck layer, leaving out the information with less variability. SAE [46] has a bottleneck layer enforced by a sparsity loss without reduction of neurons in the hidden layers. To improve robustness, one could try ensembling with different random seeds. VASC [47] and scVI [48] used zero-inflated negative binomial (ZINB) to model scRNA noise. By modeling the noise distribution, ZINB accounts for RNAseq count distribution, overdispersion, and sparsity. scGEN [41] is capable of performing vector arithmetic. It learns cell-type and species-specific information, separating responding and non-responding cells. scVAE [49] uses Gaussian or Gaussian-mixture latent prior. Rand index measuring between cluster similarity. scMM [50] uses statistical multi-omics analysis, learning joint representations that are interpretable across modalities. DeepDR [51] is a combination of AE and network, i.e. encoder-decoder and a drug response predictor network. It also explored drug repositioning by converting the topological structure of the network into vector

representations. The representation is then used to create a pointwise mutual information matrix, which is then input to a multimodal deep autoencoder. DeepProfile [52] combined a pre-trained VAE and a linear model for drug response prediction. Dr.VAE [53] is a semi-supervised, learning latent representation of gene expression data. The representation is then fed into a logistic regression classifier. Gene superset autoencoder (GSAE) [54] has predefined gene sets as a priori, finding the functional or clinical relevance of learned gene supersets. Gene superset is an unbiased combination of genesets as nodes the latent, along with weights from training. Multiview Factorization Autoencoder (MAE) [55] is capable of incorporating domain knowledge.

Autoencoders are also used in correcting batch effect, which is introduced from the technical variations in experiments, where groups are separated due to non-biological variations. [56] uses disentangled representation learning to remove batch effect. Inspired by domain adaptation, this work minimizes the distances among distributions of various batches, as a single batch or one scRNAseq dataset is analogous to a single domain. They approach this problem by reserving the mixture of shared cell types and the independence of distinct cell types by adopting a parameterized gradient reversal strategy in the discriminator. In addition to the gradient reversal, they also have an auxiliary classifier to facilitate the integration of shared cell types. The architecture consists of an autoencoder, a discriminator trained along the encoder for minimizing distributional differences between different batch distributions, a noise classifier for predicting batch effect, and an auxiliary classifier to encourage the target batch to be close to the source batch. The key components to achieving disentanglement are the noise classifier and the discriminator. They optimize two tasks jointly: 1) batch effect prediction and 2) reducing the distance between batch distributions. This is achieved by minimizing reconstruction loss, minimizing classification loss of both noise classifier

and auxiliary classifier, and maximizing classification loss of the discriminator.

## 2.5 Disentangled Representations

Do and Tran [57] states that the goal of disentangled learning is to find a set of independent factors that led to the observation. Unsupervised disentangled representation learning typically encourages independence among factors/latent variables, yet more independence may lead to poor reconstruction. Supervised methods may be more adequate in matching human interpretation with the latent variables. Current challenges are 1) lacking formal definitions of disentangled representations and 2) inadequate robust evaluation metrics. They defined disentangled representation learning as "a process of decorrelating information in the data into separate informative representations, each of which corresponds to a concept defined by humans." If the latent codes capture the data distribution well, the mutual information between $x$ and $z$ should be large, and the posterior $q(z|x)$ should have low variance. To achieve full interpretability, we need $I(z_i, y_k) = H(z_i) = H(y_k)$; for partially interpretable (to generalize beyond $y_k$), we need $I(z_i, y_k) = H(y_k)$ or $H(y_k|z_i) = 0$. They believe factors should be separated and learned one at a time. There exists a trade-off between informativeness, independence, and the number of latent variables. A robust metric for disentanglement should: 1) support both supervised/unsupervised model 2) applicable to real data beyond toy datasets 3) does not require additional training effort and hence computationally straightforward 4) consistency 5) agrees with visual evaluation. The authors introduced various evaluation metrics to assess disentanglement. MISJED addresses the flaw in the original mutual information definition by introducing joint entropy. This changes the objective from noisy and independent to

informative and independent. WSEPIN and WINDIN concern the residual information not captured by a single latent code. Both use $I(x, z_i|z_{\neq i})$, which measures how disentangled a latent code is when the ground truth factors are absent. This quantity should be close to 0 if the latent code is all noise, and large if $z_i$ are disentangled in terms of informativeness and separability.

Later on, Locatello et al. [7] questioned assumptions of disentanglement. Firstly, the dimensions of the aggregated posterior are not guaranteed to be independent. Secondly, unsupervised disentangled representation learning is more contingent on random seeds and hyperparameters. Thirdly, there is a lack of evidence that disentanglement helps with downstream tasks.

The main conclusion of this paper is that it is impossible to learn disentangled representations in an unsupervised manner without adding inductive biases on both models and data. Inductive biases are extra assumptions such that the model can generalize to unseen data. These biases can be inherent to model architectures. For example, one of the most popular architectures in computer vision is a convolution neural network, and one of its inductive biases is the locality assumption, i.e. neighboring pixels are relevant in the receptive field. Similar to the situation in section 2.2, identifiability is again brought to our attention. Given a set of observations x, we can find infinitely many generation models with the same marginal distribution, and the true generation model becomes unidentifiable.

To explain the identifiability issue more formally, let us look at the proof of Theorem 1 in the paper.

We also note that the general sampling process is $z \sim P(Z)$ and $x \sim P(X|Z)$.

Overall, the idea of this proof is to find a set of transformations for functions that

| Notation | Meaning |
|---|---|
| $P(Z)$ | Prior distribution of the latent space |
| $P(X\|Z)$ | Generative distribution |
| $P(Z\|X)$ | True posterior |
| $Q(Z\|X)$ | Varitational distribution approximating the true posterior |
| $r(X)$ | Learned representation, usually the mean of $Q(Z\|X)$ |

Table 2.1: Helpful notations for proof of Theorem 1

are invertible with nonzero Jacobian and have the same distribution marginal distribution $P(X)$. To begin with, we take the densities $p(z) = \Pi_i p(z_i)$, where $i$ refers to the $i$th dimension, transform to cdf $g_i(v) = P(z_i \leq v_i)$ such that $g(z) \sim \mathcal{U}$. We then transform $g$ with $h_i(v)$, which is the inverse of Guassian cdf $h_i(v) = \psi^{-1}(v_i)$ so that $h(g(z)) \sim \mathcal{N}(0, I)$. The next step is critical, which is finding a set of orthogonal matrices $A$ with nonzero entries. To do this, we use Householder transformation to obtain $A$, and our operations so far are $Ah(g(z))$. Finally, we undo our transformation with $h^{-1}, g^{-1}$. Finally, we name this sequence of operation $f(u) = g^{-1}(h^{-1}(Ah(g(u))))$, where $u \in supp(z)$.

In summary, $z$ and $f(z)$ are entangled because the Jacobian $\frac{f_i(u)}{u_j} \neq 0$, meaning changing one dimension in $z$ leads to changes in all dimensions in $f(z)$. Even worse, $z$ and $f(z)$ have the same marginal distribution, hence it is impossible to identify the disentangled model.

Alemi et al. [58] also proposed ways to mitigate uninformative latent code issues. They addressed that most loss functions only depend on $p(x|\theta)$ instead of $p(x, z|\theta)$, thus not considering the quality of learned latent representations. More specifically, if we have a powerful decoder, it is possible to acquire a high marginal $p(x|\theta)$ without using $z$. Therefore, a good marginal likelihood or a good evidence lower bound does not necessarily mean we learned good latent representations. The authors postulated

that by measuring the mutual information between observed variable $x$ and latent variable $z$. Because this problem is intractable, there is a need for deriving variational lower and upper bounds. Moreover, there exists a trade-off between data compression and information retention. Such a trade-off is represented by a Rate-Distortion (RD) curve in information theory. Rate (R) measures the average KL divergence between encoder distribution and learned marginal distribution. Distortion (D) measures the reconstruction. Echoing Locatello et al [7], having $ELBO = -(D + R)$ in the RD plane, there is no difference between models that depend on informative latent representations and models that do not depend on latent representations at all. The authors also proposed a solution to posterior collapse, which is simply using $\beta < 1$.

Mita et al [59] proposed a prior with ground truth factor information encoded in an auxiliary observed variable. With the learned network approximating the posterior, one can reuse the approximate posterior as a prior for another generative model. CausalVAE uses a Causal Layer, converting independent exogenous factors into causal endogenous factors.

# Chapter 3

# Methodology

Our autoencoders have two main components Fig. 3.1 (A, B), an autoencoder model and a proportion estimation network. The first component has two variations, autoencoder, and variational autoencoder. More specifically, we designed a shallow autoencoder (SHAE), a non-linear autoencoder (NLAE), and a variational autoencoder as a form of ablation study. Both variations reduce the dimensions of the input gene expression matrix via complex non-linear transformation. The second component Fig. 3.1 (C) estimates cellular proportions with a separate neural network connected to the output layer of the encoder. The use of the autoencoder's latent representation reduces computational cost because we are not directly using the full gene expression matrix as input to our proportion estimation network. Moreover, the encoder-decoder framework reduces noise in the latent representation through compression.

## 3.1 Generating Latent Representations

By design, we reduce the dimensions as well as preserve maximum information in the bottleneck layer imposed by reconstruction loss. The preservation of information in the information bottleneck (IB), where latent codes reside, is enforced by a negative log-likelihood loss with the Poisson distribution of the target, which accounts for sparsity and the very nature of count data.

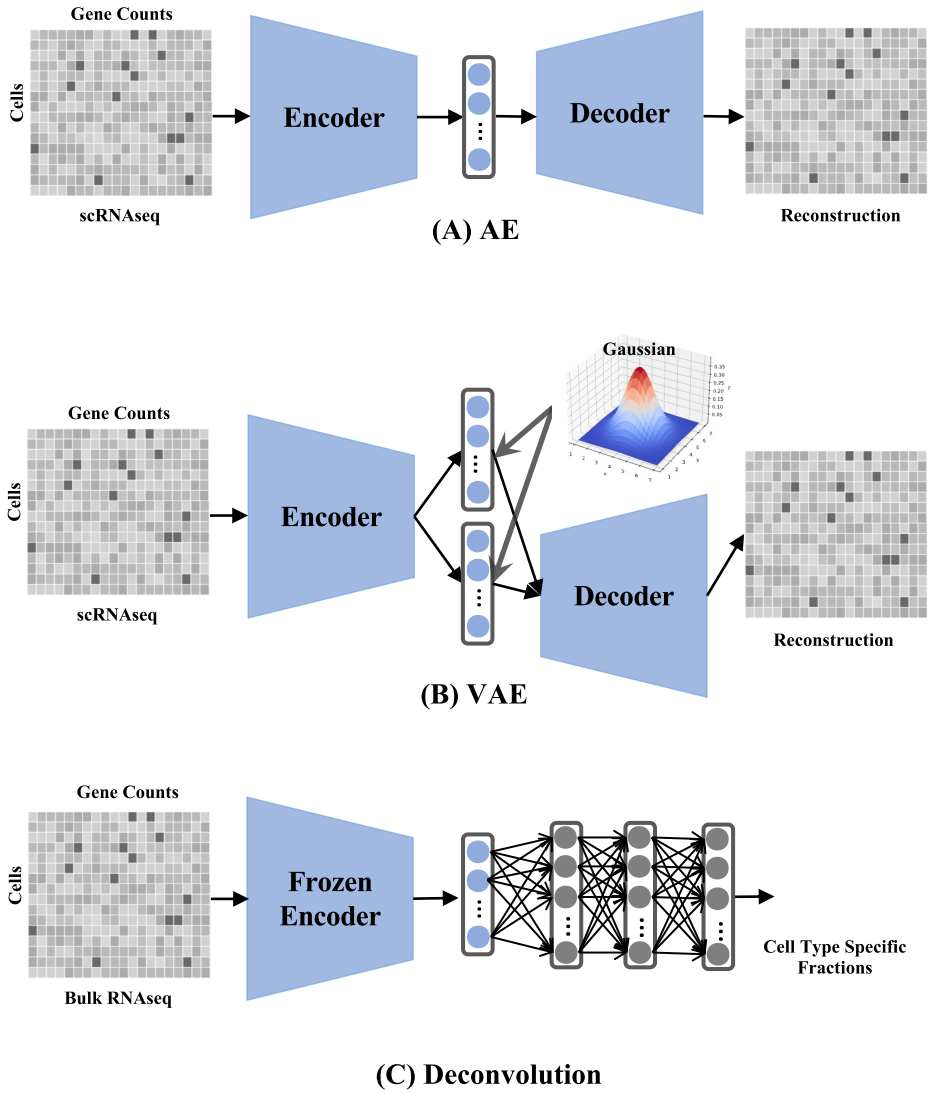


Figure 3.1: (A) General model architecture for autoencoder. Its input layer takes the full raw count gene expression data $x$. Followed by an encoder $f(x)$ that compresses input into latent codes. Then, a decoder $g(f(x))$ attempts to reconstruct the full gene expression matrix. (B) Similar to AE, VAE adds an additional distributional constraint on the latent codes, forcing them to conform to isotropic Gaussian distribution (C) is our deconvolution network, the encoder is frozen to retain learned parameters from training on scRNAseq data. In addition, the frozen encoder of the deconvolution network takes simulated bulk RNAseq as input, and output cell type fractions, given a number of cell types.

Our autoencoder is deterministic, meaning its latent codes are not regularized by any distributional constraints. This gives the latent space more freedom to encode compressed information. To model the raw counts directly, our reconstruction loss is based on the Poisson negative likelihood:

$$\mathcal{L}_{\text{AE}}(x, y) = f_d(f_e(x)) - y \times \log(f_d(f_e(x))) + \log(y!) \tag{3.1}$$

where $y$ is the gene expression input, $f_e$ is the encoder, $f_d$ is the decoder, and $f_d(f_e(\hat{x}))$ is the reconstructed output.

VAE adds variational inference and reparameterization to produce a more versatile latent space. The use of Gaussian distribution allows the latent codes to spread smoothly, forcing the latent codes to follow Gaussian distribution, on top of the original reconstruction loss of the AE.

$$\mathcal{L}_{\text{VAE}}(x, y) = -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{3.2}$$

where $\mathbb{E}_{q(z|x)}[\log p(x|z)] = \mathcal{L}(x, y) = x - y \times \log(x) + \log(y!)$. The first term on the RHS is the Kullback-Leibler divergence, and the second term on the RHS is the likelihood of data at reconstruction.

## 3.2 Assigning Cell Type Labels

In practice, cell type assignment is done after clustering analysis given a set of marker genes for each cell type. Such a procedure is manually laborious and time-consuming in selecting marker genes, plus adding subjectivity to the assignment. Therefore, instead of annotating cell type manually, we use a recently established fully automated

cell type annotation pipeline, scType [36], to label our cells. Aside from its efficient cell type identification, scType offers a marker gene database as a reference. Moreover, scType is capable of differentiating between healthy and malignant cell populations.

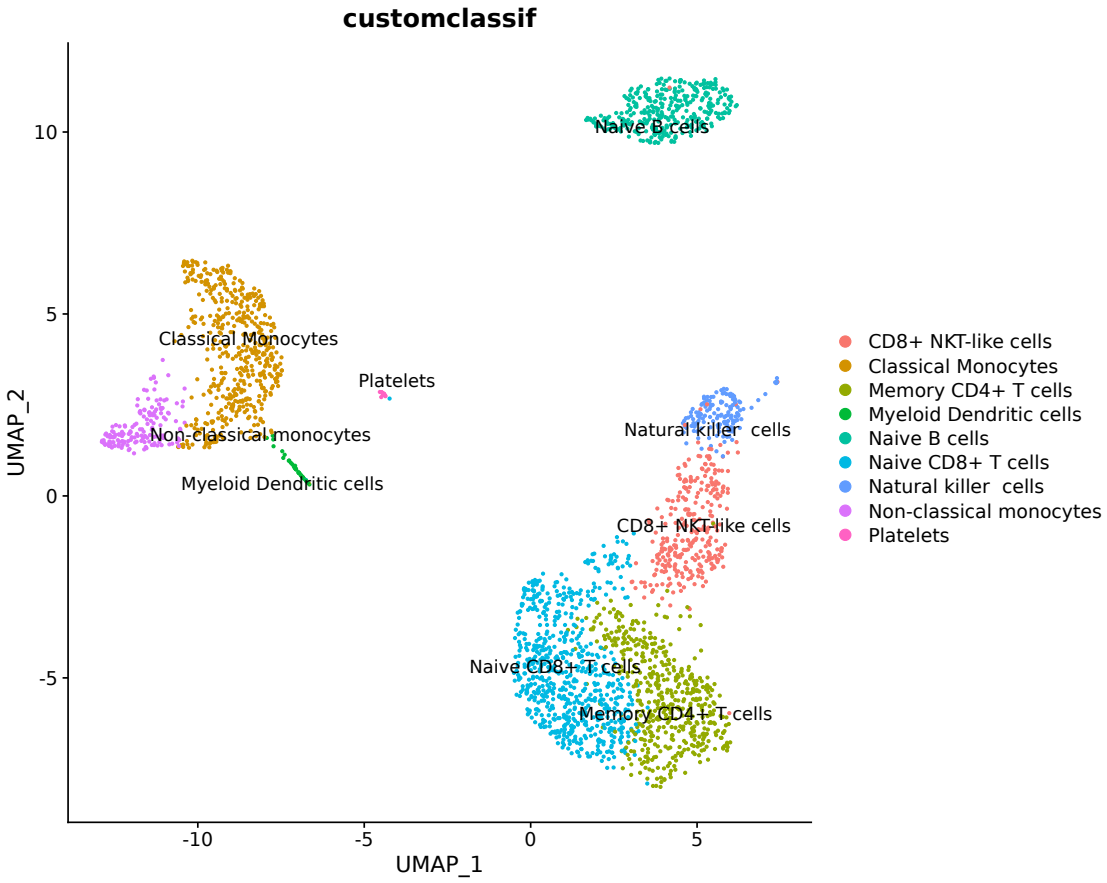


Figure 3.2: PBMC3k Cell Type Annotation by scType. scType uses UMAP to visualize a reduced dimension of cell representation. The colors show cell-type assignments.

Since we have two datasets in the absence of cell type assignments, we run scType for both PBMC3k and PBMC20k datasets. In Fig. 3.2, we see that T cells are closely clumped together, whereas monocytes, platelets, and dendritic cells are grouped together. Naive B cells are further away from the two groups. In Fig. 3.3, we can see the same pattern, but with more diverse cell types. Notably, we see an unknown collection of cells between natural killer cells and groups of T cells. Overall, the quality

of cell type annotation is satisfactory. After executing scType annotation pipeline, cell type annotations are saved as labels for simulating bulk samples.

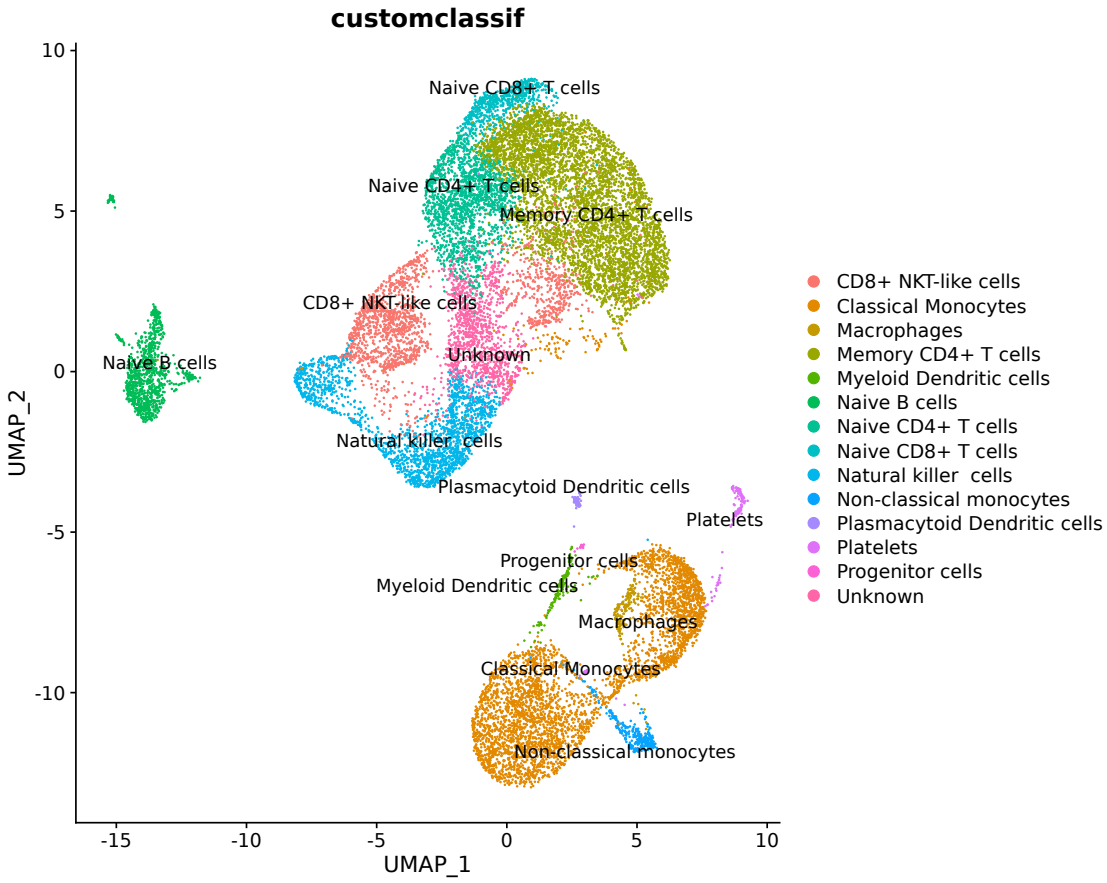


Figure 3.3: PBMC20k Cell Type Annotation by scType.

## 3.3 Deconvolution

Our deconvolution task figure. 3.1(C) begins with generating simulated bulk RNA-seq data [22].

In order to simulate bulk RNAseq data, we first generate cell-type-specific fractions $r_{cell}$ from $\mathcal{U}(0,1)$. Then, we scale the cell fractions such that all cell-specific fractions

sum to one:

$$f_c = \frac{r_c}{\sum_c r_c^i}. \tag{3.3}$$

The number of cells for each cell type, $N_c$, to aggregate is simply $N_c = f_c \times N_{total}$, where $N_{total}$ is the total number of single-cell samples within a simulated bulk sample. In this work, we set $N_{total} = 500$. Finally, we randomly sample $N_c$ single-cell samples from scRNAseq data for each cell type. Finally, a single simulated bulk sample is $\sum_c N_c$. We simulated 100,000 samples for our deconvolution task.

We treat the simulated bulk samples $x_b$ as input to our pre-trained encoder $f_e(x_b)$. The model parameters of the encoder are frozen, which means the weights $w_i$ and bias $b_i$ remain constant. In this way, we are reusing the encoding process learned from scRNAseq data. Model parameter updates only occur at the three layers after the latent code layer. Doing so prevents the downstream neural net from learning shortcuts [60] that are considered spurious correlation, which sometimes drive better model performance, yet do not necessarily reflect the desirable patterns underlying observations. For example, a deep neural net may learn to use the background 'grass' to classify 'cows', while not learning the recognition of cows, hindering model generalization outside of grassy backgrounds [61]. In the context of medical applications, a machine learning model trained on X-ray scans to detect pneumonia was revealed to classify scans based on the hospital tokens instead of the lungs themselves. This implies that a deep neural net may learn undesirable characteristics of input and use those to output unreliable predictions.

We conjectured the latent codes from the autoencoder to perform better at deconvolution compared with its variational counterpart. An autoencoder is distribution-free at the bottleneck layer which yields latent codes $z = f_e(x_b)$, meaning it is deterministic and is free to model latent codes with fewer constraints.

However, this leads to worse generalization capabilities as the latent codes are less regularized and more likely to spread out, leaving holes in the latent space like Swiss cheese. For example, suppose we have a datapoint $\hat{x}$ that is not a simulated bulk sample, its latent code $\hat{z} = f_e(\hat{x})$ should deviate from the latent codes of simulated bulk samples, and one can observe an increase in the reconstruction error $\mathcal{L}_{\text{AE}} = y - f_d(f_e(\hat{x}))$, where $y$ is the input of the deviant, and $f_d(f_e(\hat{x}))$ is the reconstructed output of the deviant if we use L1 as loss function. The property gave birth to an area called anomaly detection, which utilizes a classifier to separate the normal dataset from the anomalous dataset [62]. In short, we find an anomaly score $\epsilon$ such that if a sample exceeded $\epsilon$, i.e. $\mathcal{L}_{\text{AE}} = y - f_d(f_e(\hat{x})) > \epsilon$, it is considered anomalous. Surely, when our dataset has fewer outliers, holes are of minor concern.

Variational autoencoder has more control over the spread of its latent codes constrained by an isotropic Gaussian prior. Gaussian distribution is one of the most popular choices due to its symmetry and closed-form solution. In particular, its encoder has a posterior distribution $p_\theta(z|x)$. We see that each latent code $z$ is conditioned upon the new datapoint, and each latent code is sampled from a multivariate Gaussian distribution given $\mu, \sigma$, which are learned by the VAE. Visually, this constraint gathers the latent codes into isotropic Gaussian balls, where similar latent codes are grouped within the same ball. Formally, it means our Gaussian distribution has a covariance matrix $\Sigma = \sigma^2 I$, with constant variance in all directions, plus such a simplified form saves computational cost drastically.

In short, an autoencoder benefits from having more flexibility in the latent space, whereas a variational autoencoder is advantageous for the compactness of its latent codes.

Because our simulated bulk samples are aggregated scRNAseq samples, the latent

codes at the output layer of the frozen encoder are no longer within the same space as the latent code produced by training on scRNAseq data. To clarify, we now have a different set of samples $x_b = \sum_c N_c$, where $c$ denotes cell types. Thus, we need another neural net $h(\cdot)$ to learn a mapping from $\mathbb{R}^{\text{latent dim.}} \rightarrow \mathbb{R}^n$ such that the learned proportions $h(f_e(x_b)) \in \mathbb{R}^n$ are close to that of ground truth fractions $y$. The proportion estimator has three layers, each layer having matching dimensions with the latent codes. The output layer is of the same dimension as the number of cell types. We hypothesize that our deconvolution performance is contingent on the quality and/or consistency of cell-type annotations.

Our loss function for the proportion estimation network is the Mean Squared Error (MSE), also known as the L2 loss, between predicted proportions $x \in \mathbb{R}^n$ and ground truth proportions $y \in \mathbb{R}^n$, where $n$ is the total number of cell types.

$$\mathcal{L}_{\text{Prop.Est.}}(h(f_e(x_b)), y) = \|h(f_e(x_b)) - y\|_2^2 \tag{5}$$

In addition, we used $LeakyReLU(\cdot)$ as our activation function to enable faster gradient updates, followed by Softmax activation:

$$\text{Softmax}(h(f_e(x_b^i))) = \frac{e^{h(f_e(x_b^i))}}{\sum_j e^{h(f_e(x_b^j))}} \tag{6}$$

to output cell-type-specific fractions. Multiple gradient updates per step are facilitated by mini-batching of size 10. It is also worth mentioning that our autoencoders did not see all of the single-cell training examples due to an 80/20 train/test split, i.e. 20% of the scRNAseq data is unseen for the encoder. We also used the same 80/20 splitting ratio for training our proportion estimator. Furthermore, when we simulated bulk RNAseq data, we randomly sampled from the entire scRNAseq data

before aggregation.

The distinction between our method and a black-box feedforward neural network is that we utilize the latent representation as input instead of the full expression matrix. This reduces computational load, and the quality of compression in the latent representation is ensured by the autoencoder architecture. Moreover, our proportion estimation network can be interpreted as a regression task, having output proportion estimates as response variables.

# Chapter 4

# Results

## 4.1 Datasets

We choose a peripheral blood mononuclear cells (PBMC) dataset of eight Purified cell types, from [63], containing filtered raw counts and cell type annotations. We did not All datasets used in this work are publicly available[1].

| Name | # samples | # cells | # cell types |
|---|---|---|---|
| Zheng et al. 2017 | 29 | 80,830 | 8 |
| 3k Human PBMCs | 1 | 2,700 | 9 |
| 20k Human PBMCs | 1 | 18,470 | 14 |

Table 4.1: PBMC datasets used in this work.

There are lymphocytes, monocytes, and dendritic cells in PBMC data, and their fractions in the cell population vary. Lymphocytes are expected to be in the range of 70-90%, monocytes from 10 to 20%, and 1-2% of dendritic cells [64]. Still, the proportions of the cell population are contingent on the donor's physiological status, hence a need for multiple donors that allows the model to generalize. The Purified PBMC dataset we used has 29 samples, with 80830 cells in total and 32738 genes per cell. Our input is the raw data and we do not perform any additional data

[1]Data available online: 1. Zheng et al. 2017 [63]; 2. PBMC3k from Seurat, Guided Clustering Tutorial; 3. 20k Human PBMCs, 5' HT v2.0., from 10x Genomics

preprocessing steps, as [63] is already filtered. The other two datasets, however, do not have multiple samples, but they are from a healthy donor.

## 4.2 Evaluating Latent Representations

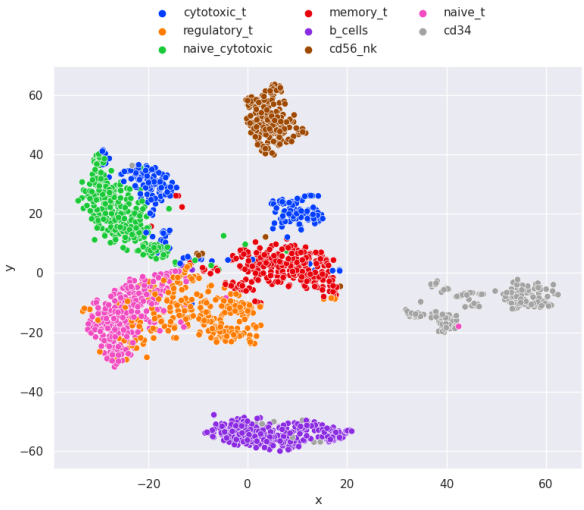


Figure 4.1: t-SNE visualization of latent codes encoded by Shallow Autoencoder (SHAE)

### 4.2.1 Qualitative Evaluation

Our latent representation is illustrated as follows, each generated latent code has dimension 100, compressed by the encoder of an AE. The latent visualizations are produced by running t-SNE and cell type annotations are given [63].

We see that the latent codes are grouped and well-separated from the shallow autoencoder Fig. 4.1. The deeper autoencoder achieved similar separation Fig. 4.2. As expected, the latent codes of the VAE is more spread out 2.2, yet T cells are blended more tightly.

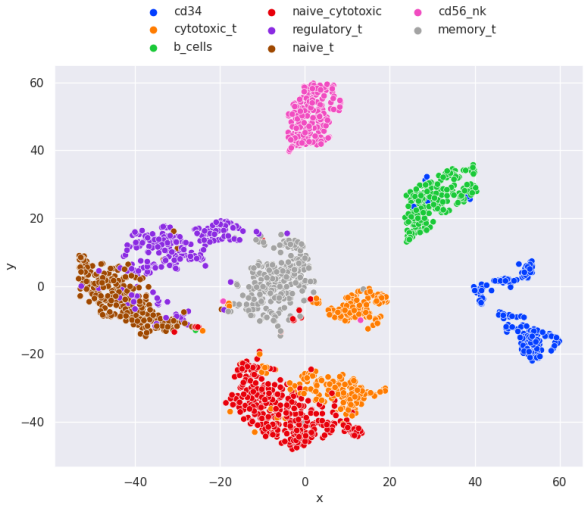


Figure 4.2: t-SNE visualization of latent codes encoded by Deep Autoencoder (NLAE) Adding non-linear activations to the model architecture Fig. 4.2, we modeled non-linear combinations of features with a non-linear autoencoder. We see a clearer separation between cell types, though the shapes of these clustering are more irregular compared with the shallow autoencoder. Moreover, cytotoxic T and naive cytotoxic grouped together while the cytotoxic T cluster is further apart.

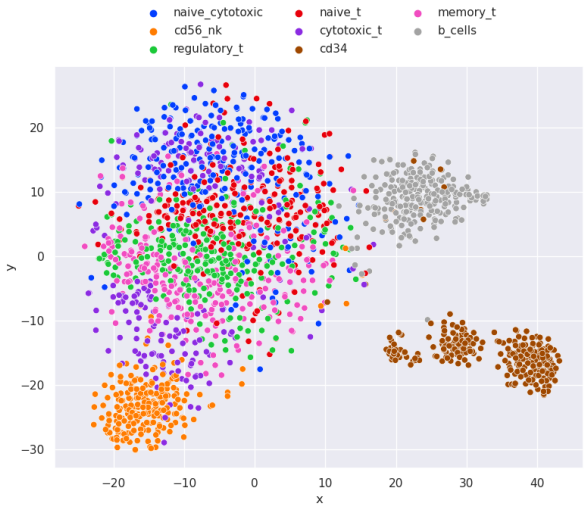


Figure 4.3: t-SNE embedding visualization of latent representations produced by the encoder of Variational Autoencoder (VAE).

### 4.2.2 Quantitative Evaluation

We use the Silhouette score as part of our criteria for stopping the training of autoencoders. Together with qualitative inspection of the latent codes, we double-check the status of autoencoders to prevent overfitting. The table of Silhouette score is listed as follows, see Table. 4.2. We took the peak of the score over training epochs as our second stopping criterion.

| Architecture | Max. Silhouette score |
|---|---|
| SHAE | 0.4220 |
| NLAE | 0.4763 |
| VAE | 0.1852 |

Table 4.2: Silhouette score of Purified PBMC t-SNE embedding.

Interestingly, VAE maintained a decent deconvolution performance despite its lower Silhouette score.

## 4.3 Evaluating Proportion Estimator

We compare our method with Scaden [22]. The shallow autoencoder is the top performer of the three autoencoder architectures we tested. Among the three datasets, our autoencoder architectures are more accurate on the Purified PBMC dataset. We see a more pronounced performance drop in estimating the proportions of simulated bulk from PBMC 20k, possibly due to confounding cell types. A sample deconvolution output is shown in Fig. 4.4.

Additionally, looking at the specific cell type fractions shown in Table 4.3 and Table 4.4, we can see our shallow autoencoder performed better in estimating cell fractions of T cells, approaching $\pm 0.002$ from groundtruth fractions.

| Model | B cells | CD34 | CD56 nk | Cytotoxic T |
|---|---|---|---|---|
| Scaden | 0.158 | 0.064 | 0.175 | 0.157 |
| SHAE (Ours) | 0.169 | 0.068 | 0.163 | 0.175 |
| NLAE (Ours) | 0.154 | 0.064 | 0.166 | 0.180 |
| VAE (Ours) | 0.156 | 0.067 | 0.161 | 0.178 |
| Ground Truth | 0.163 | 0.061 | 0.175 | 0.176 |

Table 4.3: Specific proportion estimates and groundtruth proportions of Purified PBMC, corresponding to the same sample in Figure 4.4.

| Model | Memory T | Naive Cytotoxic | Naive T | regulatory T |
|---|---|---|---|---|
| Scaden | 0.065 | 0.097 | 0.137 | 0.146 |
| SHAE | 0.063 | 0.090 | 0.140 | 0.134 |
| NLAE | 0.072 | 0.092 | 0.112 | 0.160 |
| VAE | 0.059 | 0.087 | 0.135 | 0.158 |
| Truth | 0.065 | 0.085 | 0.139 | 0.136 |

Table 4.4: Table. 4.3 continued.

| Dataset | Method | Pearson's r | RMSE | R squared |
|---|---|---|---|---|
| Purified PBMC | Scaden | 0.988 | 0.010 | 0.972 |
| | SHAE | 0.980 | 0.013 | 0.955 |
| | NLAE | 0.962 | 0.018 | 0.918 |
| | VAE | 0.953 | 0.020 | 0.904 |
| PBMC 3k | Scaden | 0.998 | 0.004 | 0.996 |
| | SHAE | 0.978 | 0.014 | 0.952 |
| | NLAE | 0.985 | 0.013 | 0.963 |
| | VAE | 0.919 | 0.023 | 0.846 |
| PBMC 20k | Scaden | 0.960 | 0.009 | 0.922 |
| | SHAE | 0.929 | 0.014 | 0.848 |
| | NLAE | 0.887 | 0.020 | 0.774 |
| | VAE | 0.919 | 0.016 | 0.827 |

Table 4.5: Evaluation on all three PBMC datasets with Pearson's r, Root Mean Squared Error (RMSE) and R squared, averaged over random test samples.

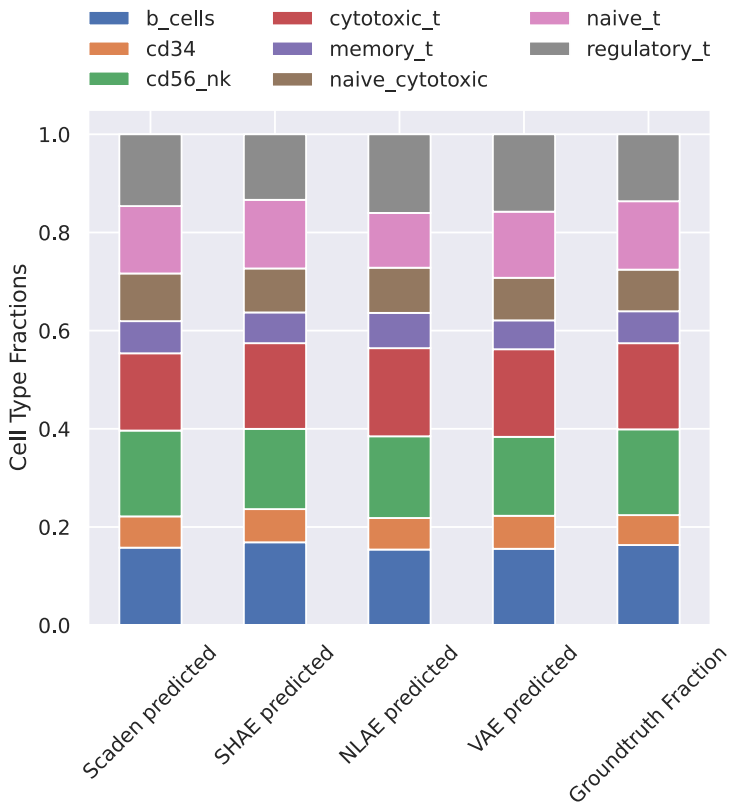


Figure 4.4: Deconvolution output of cell-type-specific fractions estimated from a random sample.

Our benchmarking results, Table. 4.5, show that Scaden remains the top performer. One explanation for this compromise in accuracy is that we have a denoising step before the deconvolution net and it may have removed some of the signals beneficial to the deconvolution. Despite that, it is easier for us to detect out-of-distribution samples with our two-step process by either looking at the latent visualization or the behaviors of loss functions. In addition, our deconvolution allows us to inspect the composition of the latent layer, thereby having a heuristic opportunity to prevent the upstream autoencoder from overfitting or over-regularizing by stopping the training early.

The shallow autoencoder (SHAE) performs best among our three autoencoder archi-

tectures. This is likely because cell types were annotated with PCA involved such that a shallow linear autoencoder is akin to a linear matrix factorization, hence having an encoder architecture close to PCA is potentially beneficial for deconvolution.

# Chapter 5

# Discussion

The flexibility of deep learning allows us to investigate from fine-grained cell states to coarser-grained cell types [65]. For tasks that are linear in nature, for instance, deconvolution of cell types, it is sufficient to use a shallow autoencoder followed by a deep neural net for deconvolution. In fact, shallow autoencoders are rather suitable for sparse data [66]. For tasks that are more sophisticated, such as modeling cell states, that are non-linear and require incorporation of biological structure, we need a more complex model architecture to capture such structure. Existing works such as density-tree biased autoencoder [67] show differentiated cell states and biological structures. In fact, our visualizations of the latent codes presented similar hierarchical structures, more so in a nested manner, showing more diversity amongst T cells. After all, as stated in [67], autoencoders alone do not automatically uncover meaningful hierarchical properties, and the trick is to choose a reasonable prior.

It is noteworthy that we encountered challenges in training the autoencoder, especially VAE. Due to the sparsity and high dimensionality of our single-cell data, the posterior quickly collapsed towards our Gaussian prior. To solve this problem, we tried many tricks ranging from warming-up and cyclical annealing to early stopping. We found that early stopping is the most useful. In deciding when to stop training, we use the peak of the Silhouette Score over training epochs as well as visual inspection of the latent codes.

We also note that our model is dependent on the quality of cell-type annotation given by datasets or annotation pipelines, to fully streamline an analysis that involves cell-type annotation, it is better to fully integrate an annotation pipeline into the neural network, possibly in the forms of classification modules, clustering-based loss functions, etc.

# Chapter 6

# Conclusion

Among three candidate architectures in the context of estimating cell-type-specific fractions, shallow autoencoder, deep non-linear autoencoder, and variational autoencoder, we found that shallow autoencoder performs best, and is closest to state-of-the-art. There has been an ongoing controversy [68, 69] regarding whether disentanglement improves downstream tasks.

In our empirical analysis, our VAE performed worse than an autoencoder. Conclusively, modeling single-cell data with autoencoders appears to be a promising direction as the cost of single-cell sequencing technology decreases. With more data, we could achieve better reconstruction and obtain informative latent codes to further advance disease classification and drug re-purposing.

# Bibliography

[1] Ruoqi Wei, Cesar Garcia, Ahmed El-Sayed, Viyaleta Peterson, and Ausif Mahmood. Variations in variational autoencoders-a comparative evaluation. *Ieee Access*, 8:153651–153670, 2020.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013.

[3] Charles Boncelet. Image noise models. *The Essential Guide to Image Processing*, pages 143–167, 1 2009.

[4] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[5] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods 2018 15:12*, 15:1053–1058, 11 2018.

[6] Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature Communications 2019 10:1*, 10:1–14, 1 2019.

[7] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions

in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[8] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[9] Daphne Tsoucas, Rui Dong, Haide Chen, Qian Zhu, Guoji Guo, and Guo-Cheng Yuan. Accurate estimation of cell-type composition from gene expression data. *Nature communications*, 10(1):1–9, 2019.

[10] Yating Lin, Haojun Li, Xu Xiao, Lei Zhang, Kejia Wang, Jingbo Zhao, Minshu Wang, Frank Zheng, Minwei Zhang, Wenxian Yang, et al. Daism-dnnxmbd: Highly accurate cell type proportion estimation with in silico data augmentation and deep neural networks. *Patterns*, 3(3):100440, 2022.

[11] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[12] Shikha Roy, Rakesh Kumar, Vaibhav Mittal, and Dinesh Gupta. Classification models for invasive ductal carcinoma progression, based on gene expression data-trained supervised machine learning. *Scientific reports*, 10(1):1–15, 2020.

[13] Kevin M Boehm, Emily A Aherne, Lora Ellenson, Ines Nikolovski, Mohammed Alghamdi, Ignacio Vázquez-García, Dmitriy Zamarin, Kara Long Roche, Ying Liu, Druv Patel, et al. Multimodal data integration using machine learning

improves risk stratification of high-grade serous ovarian cancer. *Nature cancer*, 3(6):723–733, 2022.

[14] Neha Shree Maurya, Sandeep Kushwaha, Aakash Chawade, and Ashutosh Mani. Transcriptome profiling by combined machine learning and statistical r analysis identifies tmem236 as a potential novel diagnostic biomarker for colorectal cancer. *Scientific reports*, 11(1):1–11, 2021.

[15] Xinlei Mi, Baiming Zou, Fei Zou, and Jianhua Hu. Permutation-based identification of important biomarkers for complex diseases via machine learning models. *Nature communications*, 12(1):3008, 2021.

[16] Dirk Repsilber, Sabine Kern, Anna Telaar, Gerhard Walzl, Gillian F Black, Joachim Selbig, Shreemanta K Parida, Stefan HE Kaufmann, and Marc Jacobsen. Biomarker discovery in heterogeneous tissue samples-taking the in-silico deconfounding approach. *BMC bioinformatics*, 11(1):1–15, 2010.

[17] Alexander R Abbas, Kristen Wolslegel, Dhaya Seshasayee, Zora Modrusan, and Hilary F Clark. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. *PloS one*, 4(7):e6098, 2009.

[18] Francisco Avila Cobos, José Alquicira-Hernandez, Joseph E Powell, Pieter Mestdagh, and Katleen De Preter. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nature communications*, 11(1):1–14, 2020.

[19] Aaron M Newman, Chih Long Liu, Michael R Green, Andrew J Gentles, Weiguo Feng, Yue Xu, Chuong D Hoang, Maximilian Diehn, and Ash A Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. *Nature methods*, 12(5):453–457, 2015.

[20] Aaron M Newman, Chloé B Steen, Chih Long Liu, Andrew J Gentles, Aadel A Chaudhuri, Florian Scherer, Michael S Khodadoust, Mohammad S Esfahani, Bogdan A Luca, David Steiner, et al. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature biotechnology*, 37(7):773–782, 2019.

[21] Ali Karimnezhad. More accurate estimation of cell composition in bulk expression through robust integration of single-cell information. *Bioinformatics Advances*, 2(1), 2022.

[22] Kevin Menden, Mohamed Marouf, Sergio Oller, Anupriya Dalmia, Daniel Sumner Magruder, Karin Kloiber, Peter Heutink, and Stefan Bonn. Deep learning–based cell composition analysis from tissue expression profiles. *Science advances*, 6(30):eaba2619, 2020.

[23] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[24] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[25] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[26] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in $\beta$-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[27] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[28] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[29] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

[30] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

[31] Bin Dai, Ziyu Wang, and David Wipf. The usual suspects? reassessing blame for vae posterior collapse. In *International conference on machine learning*, pages 2313–2322. PMLR, 2020.

[32] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019.

[33] Yixin Wang, David Blei, and John P Cunningham. Posterior collapse and latent

variable non-identifiability. *Advances in Neural Information Processing Systems*, 34:5443–5455, 2021.

[34] Tallulah S Andrews, Vladimir Yu Kiselev, Davis McCarthy, and Martin Hemberg. Tutorial: guidelines for the computational analysis of single-cell rna sequencing data. *Nature protocols*, 16(1):1–9, 2021.

[35] Justin D Silverman, Kimberly Roche, Sayan Mukherjee, and Lawrence A David. Naught all zeros in sequence count data are the same. *Computational and structural biotechnology journal*, 18:2789–2798, 2020.

[36] Aleksandr Ianevski, Anil K Giri, and Tero Aittokallio. Fully-automated and ultra-fast cell-type identification using specific marker combinations from single-cell transcriptomic data. *Nature communications*, 13(1):1–10, 2022.

[37] Hugh D Mitchell, Lye Meng Markillie, William B Chrisler, Matthew J Gaffrey, Dehong Hu, Craig J Szymanski, Yumei Xie, Eric S Melby, Alice Dohnalkova, Ronald C Taylor, et al. Cells respond to distinct nanoparticle properties with multiple strategies as revealed by single-cell rna-seq. *ACS nano*, 10(11):10173–10185, 2016.

[38] Miguel Juliá, Amalio Telenti, and Antonio Rausell. Sincell: an r/bioconductor package for statistical assessment of cell-state hierarchies from single-cell rna-seq. *Bioinformatics*, 31(20):3380–3382, 2015.

[39] Hung Nguyen, Duc Tran, Bang Tran, Bahadir Pehlivan, and Tin Nguyen. A comprehensive survey of regulatory network inference methods using single cell rna sequencing data. *Briefings in bioinformatics*, 22(3):bbaa190, 2021.

[40] Hengshi Yu and Joshua D Welch. Michigan: sampling from disentangled representations of single-cell data using generative adversarial networks. *Genome biology*, 22(1):1–26, 2021.

[41] Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

[42] Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4468–4476, 2022.

[43] Yuanyuan Liu, Wenbin Wang, Fang Fang, Lin Zhou, Chenxing Sun, Ying Zheng, and Zhanlong Chen. Cscgan: Conditional scale-consistent generation network for multi-level remote sensing image to map translation. *Remote Sensing*, 13(10):1936, 2021.

[44] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[45] David Pratella, Samira Ait-El-Mkadem Saadi, Sylvie Bannwarth, Véronique Paquis-Fluckinger, and Silvia Bottini. A survey of autoencoder algorithms to pave the diagnosis of rare diseases. *International journal of molecular sciences*, 22(19):10891, 2021.

[46] Wenjun Sun, Siyu Shao, Rui Zhao, Ruqiang Yan, Xingwu Zhang, and Xuefeng Chen. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, 89:171–178, 2016.

[47] Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018.

[48] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[49] Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther. scvae: variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16):4415–4422, 2020.

[50] Kodai Minoura, Ko Abe, Hyunha Nam, Hiroyoshi Nishikawa, and Teppei Shimamura. scmm: Mixture-of-experts multimodal deep generative model for single-cell multiomics data analysis. *bioRxiv*, pages 2021–02, 2021.

[51] Xiangxiang Zeng, Siyi Zhu, Xiangrong Liu, Yadi Zhou, Ruth Nussinov, and Feixiong Cheng. deepdr: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics*, 35(24):5191–5198, 2019.

[52] Ayse Dincer, Safiye Celik, Naozumi Hiranuma, and Su-In Lee. Deepprofile: deep learning of patient molecular profiles for precision medicine in acute myeloid leukemia. *BioRxiv*, page 278739, 2018.

[53] Ladislav Rampášek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr. vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, 35(19):3743–3751, 2019.

[54] Hung-I Harry Chen, Yu-Chiao Chiu, Tinghe Zhang, Songyao Zhang, Yufei

Huang, and Yidong Chen. Gsae: an autoencoder with embedded gene-set nodes for genomics functional characterization. *BMC systems biology*, 12(8):45–57, 2018.

[55] Tianle Ma and Aidong Zhang. Multi-view factorization autoencoder with network constraints for multi-omic integrative analysis. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 702–707. IEEE, 2018.

[56] Tiantian Guo, Yang Chen, Minglei Shi, Xiangyu Li, and Michael Q Zhang. Integration of single cell data by disentangled representation learning. *Nucleic Acids Research*, 50(2):e8–e8, 2022.

[57] Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. *arXiv preprint arXiv:1908.09961*, 2019.

[58] Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International conference on machine learning*, pages 159–168. PMLR, 2018.

[59] Graziano Mita, Maurizio Filippone, and Pietro Michiardi. An identifiable double vae for disentangled representations. In *International Conference on Machine Learning*, pages 7769–7779. PMLR, 2021.

[60] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[61] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita.

In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[62] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pages 1–5. IEEE, 2018.

[63] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

[64] Charlotte R Kleiveland. Peripheral blood mononuclear cells. *The impact of food bioactives on health*, pages 161–167, 2015.

[65] Guillermo García-Pérez, Marián Boguñá, and M Serrano. Multiscale unfolding of real networks by geometric renormalization. *Nature Physics*, 14(6):583–589, 2018.

[66] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pages 3251–3257, 2019.

[67] Quentin Garrido, Sebastian Damrich, Alexander Jäger, Dario Cerletti, Manfred Claassen, Laurent Najman, and Fred A Hamprecht. Visualizing hierarchies in scrna-seq data using a density tree-biased autoencoder. *Bioinformatics*, 38(Supplement_1):i316–i324, 2022.

[68] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124*, 2020.

[69] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems*, 32, 2019.