scVAE: Variational auto-encoders for single-cell gene expression data

Christopher Heje Grønbech [1,2], Maximillian Fornitz Vording [2], Pascal Timshel [3,4], Casper Kaae Sønderby [5], Tune Hannes Pers [3,4], and Ole Winther [1,2,5]

[1]Centre for Genomic Medicine, Rigshospitalet, Copenhagen University Hospital.
[2]Section for Cognitive Systems, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
[3]The Novo Nordisk Foundation Center for Basic Metabolic Research, Faculty of Health and Medical Sciences, University of Copenhagen.
[4]Department of Epidemiology Research, Statens Serum Institut.
[5]Bioinformatics Centre, Department of Biology, University of Copenhagen.

Abstract

Motivation: Models for analysing and making relevant biological inferences from massive amounts of complex single-cell transcriptomic data typically require several individual data-processing steps, each with their own set of hyperparameter choices. With deep generative models one can work directly with count data, make likelihood-based model comparison, learn a latent representation of the cells and capture more of the variability in different cell populations.

Results: We propose a novel method based on variational auto-encoders (VAEs) for analysis of single-cell RNA sequencing (scRNA-seq) data. It avoids data preprocessing by using raw count data as input and can robustly estimate the expected gene expression levels and a latent representation for each cell. We tested several count likelihood functions and a variant of the VAE that has a priori clustering in the latent space. We show for several scRNA-seq data sets that our method outperforms recently proposed scRNA-seq methods in clustering cells and that the resulting clusters reflect cell types.

Availability and implementation: Our method, called scVAE, is implemented in Python using the TensorFlow machine-learning library, and it is freely available at `https://github.com/chgroenbech/scVAE`.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) enables measurement of gene expression levels of individual cells and thus provides a new framework to understand dysregulation of disease at the cell-type level (Regev *et al.*, 2017). To date, a number of methods[1] have been developed to process gene expression data to normalise the data (Haghverdi *et al.*, 2018) and cluster cells into putative cell types (Satija *et al.*, 2015; Kiselev *et al.*, 2017). Seurat (Satija *et al.*, 2015) is a popular method which is a multi-step process of normalisation, transformation, decomposition, embedding, and clustering of the gene expression levels. This can be cumbersome, and a more automated approach is desirable. Four recent methods, cellTree (duVerle *et al.*, 2016), DIMM-SC (Sun *et al.*, 2017), DCA (Eraslan *et al.*, 2018), and scVI (Lopez *et al.*, 2018), model the gene expression levels directly as counts using latent Dirichlet allocation, Dirichlet-mixture generative model, an auto-encoder, and a variational auto-encoder, respectively. The last two methods are described below.

Here, we show that expressive deep generative models, leveraging the recent progress in deep neural networks, provide a powerful framework for modelling the data distributions of raw count data. We show that these models can learn biologically plausible representations of scRNA-seq experiments using only the highly sparse raw count data as input entirely skipping the normalisation and transformation steps needed in previous methods. Our approach is based on the variational auto-encoder (VAE) framework presented in Kingma and Welling (2013) and Rezende *et al.* (2014). These models learn compressed latent representations of the input data without any supervisory signals, and they can denoise the input data using its encoder-decoder structure. We extend these models with likelihood (link) functions suitable for modelling (sparse) count data and perform extensive experiments on several data sets to analyse the strengths and weaknesses of the models and the likelihood functions. Variational auto-encoders have been examined and extended to use multiple latent vari-

[1]A list of software packages for single-cell data analysis is available at `https://github.com/seandavi/awesome-single-cell`.

ables for, e.g., semi-supervised learning (Kingma *et al.*, 2014), clustering (Dilokthanakul *et al.*, 2016; Maaløe *et al.*, 2017; Jiang *et al.*, 2017), and structured representations (Sønderby *et al.*, 2016; Johnson *et al.*, 2016; Lin *et al.*, 2018). They have also been used in variety of cases to, e.g., generate sentences (Bowman *et al.*, 2016), transfer artistic style of paintings (Gatys *et al.*, 2015), and create music (Roberts *et al.*, 2017). Recently, VAEs have also been used to study normalised bulk RNA-seq data (Way and Greene, 2017) as well as visualising normalised scRNA-seq data (Ding *et al.*, 2018).

Compared to most other auto-encoder methods, VAEs have two major advantages: (a) It is probabilistic so that performance can be quantified and compared in terms of the likelihood, and (b) the variational objective creates a natural trade-off between data reconstruction and model capacity. Other types of auto-encoders also learn a latent representation from data that is used to reconstruct the same as well as unseen data. The latent representation is tuned to have low-capacity, so the model is forced to only estimate the most important features of the data. Different auto-encoder models have previously been used to model normalised (or binarised) bulk gene expression levels: denoising auto-encoders (Tan *et al.*, 2014; Gupta *et al.*, 2015; Tan *et al.*, 2016), sparse auto-encoders (Chen *et al.*, 2016), and robust auto-encoders (Cui *et al.*, 2017). A bottleneck auto-encoder (Eraslan *et al.*, 2018) was also recently used to model single-cell transcript counts, and a generative adversarial network (Goodfellow *et al.*, 2014), which is a related model, was recently used to model normalised single-cell gene expression levels (Ghahramani *et al.*, 2018).

Our contributions are threefold: (a) We have developed new generative models based on the VAE framework for directly modelling raw counts from RNA-seq data; (b) we show that our models using either a Gaussian or a Gaussian-mixture latent variable prior learn biologically plausible groupings of scRNA-seq data of as validated on data sets with known cell types; and (c) we provide a publicly available framework for unsupervised modelling of count data from RNA-seq experiments.

2 Methods and materials

We have developed generative models for directly modelling the raw read counts from scRNA-seq data. In this section, we describe the models as well as the different data likelihood (link) functions for this task.

2.1 Latent-variable models

We take a generative approach to modelling raw count data vectors $\mathbf{x}$, where $\mathbf{x}$ represents a single cell and its components x_n, which are also called *features*, correspond to the gene expression count for gene n. We assume that $\mathbf{x}$ is drawn from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}) = p(\mathbf{x} \,|\, \boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$. The number of features (genes), N, is very high, but we still want to model co-variation between the features. We achieve this by introducing a stochastic latent variable $\mathbf{z}$, with fewer dimensions than $\mathbf{x}$, and condition the data-generating process on this. The joint probability distribution of $\mathbf{x}$ and $\mathbf{z}$ is then

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z}) \; p_{\boldsymbol{\theta}}(\mathbf{z}), \tag{1}$$

where $p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})$ is the likelihood function and $p_{\boldsymbol{\theta}}(\mathbf{z})$ is the prior probability distribution of $\mathbf{z}$. Marginalising over $\mathbf{z}$ results in the marginal likelihood of $\boldsymbol{\theta}$ for one data point:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z}) \; p_{\boldsymbol{\theta}}(\mathbf{z}) \, \mathrm{d}\mathbf{z}. \tag{2}$$

The log-likelihood function for all data points (also called *examples*, which in this case are cells) will be the sum of the log-likelihoods for each data point: $\mathcal{F}(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log p_{\boldsymbol{\theta}}(\mathbf{x}_m)$, where M is the number of cells. We can then use maximum-likelihood estimation, $\arg\max_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta})$, to infer the value $\boldsymbol{\theta}$.

We use a deep neural network to map from $\mathbf{z}$ to the sufficient statistics of $\mathbf{x}$. However, as the marginalisation over the latent variables in Equation (2) is intractable for all but the simplest linear models, we have to resort to approximate methods for inference in the models. Here we use variational Bayesian optimisation for inference as introduced in the variational auto-encoder framework (Kingma and Welling, 2013; Rezende *et al.*, 2014). Details of this approach is covered in Section 2.4.

2.2 Deep generative models

The choice of likelihood function $p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})$ depends on the statistical properties of the data, e.g., continuous or discrete observations, sparsity, and computational tractability. Contrary to most other VAE-based articles considering either continuous or categorical input data, our goal is to model discrete count data directly, which is why Poisson or negative binomial likelihood functions are natural choices. This will be discussed in more detail below. The prior over the latent variables $p(\mathbf{z})$ is usually chosen to be an isotropic standard multivariate Gaussian distribution to get the following generative process (Fig. 1A):

$$p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z}) = f(\mathbf{x}; \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z})), \tag{3a}$$

$$p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \tag{3b}$$

where f is a discrete distribution such as the Poisson (P) distribution: $f_{\mathrm{P}}(x; \lambda) = \mathrm{e}^{-\lambda} \frac{\lambda^x}{x!}$ and $f(\mathbf{x}; \boldsymbol{\lambda}) = \prod_k f(x_k; \lambda_k)$. The Poisson rate parameters are functions of $\mathbf{z}$: $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z})$. We can for example parameterise it by a single-layer feed-forward neural network:

$$\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}) = h(\mathbf{W}\mathbf{z} + \mathbf{b}). \tag{4}$$

Here, $\mathbf{W}$ and $\mathbf{b}$ are the weights and bias of a linear model, $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b})$, and $h(\cdot)$ is an appropriate element-wise non-linear transformation to make the rate non-negative. The Poisson likelihood function can be substituted by other probability distributions with parameters of the distribution being non-linear functions of $\mathbf{z}$ in

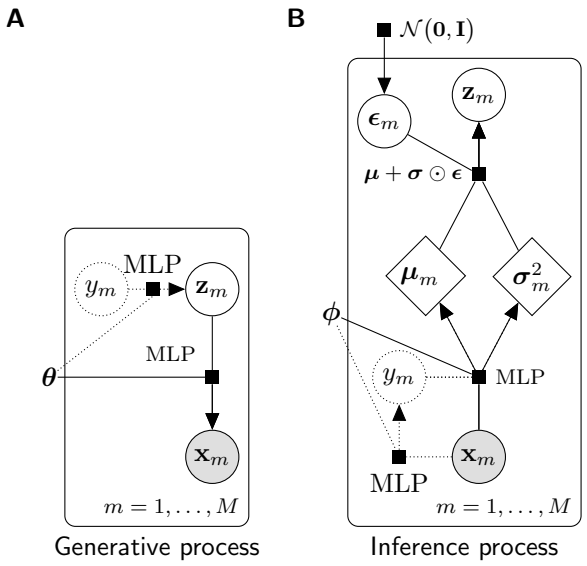

Fig. 1. Model graphs of (**A**) the generative process and (**B**) the inference process of the variational auto-encoder models. Dotted strokes designate nodes and edges for the Gaussian-mixture model only. Grey circles signify observable variables, white circles represent latent variables, and rhombi symbolise deterministic variables. The black squares denote the functions next to them with the variables connected by lines as input and the variable connected by an arrow as output.

the same fashion as above (see Section 2.3). In order to make the model more expressive, we can also replace the single-layer model with a deep model with one or more hidden layers. For L layers of adaptable weights we can write: $\mathbf{a}_l = h_l(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l)$ for $l = 1, \ldots, L$, $\mathbf{a}_0 = \mathbf{z}$, and $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{a}_L$ with $h_l(\cdot)$ denoting the activation function of the lth layer. For the hidden layers the rectifier function, $\mathrm{ReLU}(x) = \max(0, x)$, is often a good choice.

2.2.1 Gaussian-mixture variational auto-encoder

Using a Gaussian distribution as the prior probability distribution of $\mathbf{z}$ only allows for one mode in the latent representation. If there is an inherent clustering in the data, like for scRNA-seq data where the cells represent different cell types, it is desirable to have multiple modes, e.g., one for every cluster or class. This can be implemented by using a Gaussian-mixture model in place of the Gaussian distribution. Following the specification of the M2 model of Kingma *et al.* (2014), with inspiration from Dilokthanakul *et al.* (2016) and modifications by Rui Shu[2], a categorical latent random variable $y \in \{1, \ldots, K\}$ is added to the VAE model:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, y, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})\; p_{\boldsymbol{\theta}}(\mathbf{z} \,|\, y)\; p_{\boldsymbol{\theta}}(y). \tag{5}$$

where the likelihood term $p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})$ is same as for the standard VAE (Eq. 3a) and

$$p_{\boldsymbol{\theta}}(\mathbf{z} \,|\, y) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(y), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(y)\mathbf{I}), \tag{6a}$$

$$p_{\boldsymbol{\theta}}(y) = \mathrm{Cat}(y; \boldsymbol{\pi}). \tag{6b}$$

The generative process for the Gaussian-mixture variational auto-encoder (GMVAE) is illustrated in Figure 1A. Here, $\boldsymbol{\pi}$ is a K-dimensional probability vector, where K is the number of components in the Gaussian-mixture model. The component π_k of $\boldsymbol{\pi}$ is the mixing coefficient of the kth Gaussian distribution, quantifying how much this distribution contributes to the overall probability distribution. Without any prior knowledge, the categorical distribution is fixed to be uniform.

2.3 Modelling gene expression count data

Instead of normalising and transforming the gene expression data, the original transcript counts are modelled directly to take into account the total amount of genes expressed in each cell also called the *sequencing depth*. To model count data the likelihood function $p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})$ will need to be discrete and only have non-negative support. As described earlier, parameters for these probability distributions are modelled using deep neural networks that takes the latent vector $\mathbf{z}$ as input. We will consider a number of such distributions in the following and investigate which ones are best in term of likelihood on held-out data.

Our first approach to model the likelihood function uses either the Poisson or negative binomial (NB) distributions. The Poisson distribution provides the simplest distribution naturally handling count data, whereas the negative binomial distribution corresponds well with the over-dispersed nature of gene expression data (Oshlack *et al.*, 2010). To properly handle the sparsity of the scRNA-seq data (Vallejos *et al.*, 2017), we test two approaches: a zero-inflated distribution and modelling of low counts using a discrete distribution. A zero-inflated distribution adds an additional parameter, which controls the amount of excessive zeros added to an existing probability distribution. For the Poisson distribution, $f_{\mathrm{P}}(x; \lambda)$, the zero-inflated version (ZIP) is defined as

$$f_{\mathrm{ZIP}}(x; \rho, \lambda) = \begin{cases} \rho + (1-\rho) f_{\mathrm{P}}(x; \lambda), & x = 0, \\ (1-\rho) f_{\mathrm{P}}(x; \lambda), & x > 0, \end{cases} \tag{7}$$

where ρ is the probability of excessive zeros. The zero-inflated negative binomial (ZINB) distribution have an analogous expression.

The so-called piece-wise categorical distribution, which has been used for demand forecasting (Seeger *et al.*, 2016), uses a mixture of discrete probabilities for low counts and a shifted probability distribution for counts equal to or greater than K. The piece-wise categorical version of the Poisson distribution (PCP) is

$$f_{\mathrm{PCP}}(x; \boldsymbol{\tau}, \lambda) = \begin{cases} \tau_k, k = x, & 0 \le x < K, \\ \tau_K f_P(x - K; \lambda), & x \ge K\ , \end{cases} \tag{8}$$

where τ_k, $k = 0, \ldots, K-1$, is the probability of a count being equal to k and τ_K the probability that the count is K or above.

[2]Detailed in a blog post by Rui Shu: `http://ruishu.io/2016/12/25/gmvae/`.

Lastly, we also investigate the so-called constrained Poisson (CP) distribution (Salakhutdinov and Hinton, 2009) which reparameterises the Poisson to depend explicitly on the sequencing depth D_m of cell m. For the constrained Poisson distribution, the rate parameter λ_{mn} for gene n in cell m is

$$\lambda_{mn} = D_m p_{mn} \ , \tag{9}$$

where p_{mn} is constrained to be a probability distribution for each cell: $p_{mn} \geq 0$ and $\sum_n p_{mn} = 1$. This construction ensures the rates are set such that the expected number of counts are equal to the sequencing depth while we can still model each gene by a Poisson distribution.

Since we do not know the true conditioning structure of the genes, we make the simplifying assumption that they are independent for computational reasons and therefore use feed-forward neural networks.

2.4 Variational auto-encoders

In order to train our deep generative models, we will use the variational auto-encoder framework which amounts to replacing the intractable marginal likelihood with its variational lower bound and estimate the intractable integrals with low-variance Monte Carlo estimates. The lower bound is maximised for the training set and then evaluated on a test set in order to perform model comparison.

Since the likelihood function $p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z})$ is modelled using non-linear transformations, the posterior probability distribution $p_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x} \,|\, \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}) / p_{\boldsymbol{\theta}}(\mathbf{x})$ becomes intractable. Variational auto-encoders use a variational Bayesian approach where $p_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})$ is replaced by an approximate probability distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \,|\, \mathbf{x})$ modelled using non-linear transformations parameterised by $\boldsymbol{\phi}$. Thus, the marginal log-likelihood can be written as

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \,|\, \mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}). \tag{10}$$

Here, the first term is the Kullback–Leibler (KL) divergence between the true and the approximate posterior distributions. $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ can be rewritten as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] \\ - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{z})), \end{aligned} \tag{11}$$

where the first term is the *expected negative reconstruction error* quantifying how well the model can reconstruct $\mathbf{x}$, and the second term is the *relative KL divergence* between the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$ of $\mathbf{z}$.

Evaluating Equation (10) is intractable due to the appearance of the intractable true posterior distribution. However, as the KL divergence is non-negative, Equation (11) provides a lower bound on the marginal, e.g., $\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x})$. The integrals over $\mathbf{z}$ in Equation (11) are still analytically intractable but a low-variance estimator can be constructed using Monte Carlo integration as described below.

For the standard VAE, the approximate posterior distribution is chosen to be a multivariate Gaussian distribution with a diagonal covariance matrix:

$$q_{\boldsymbol{\phi}}(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})\mathbf{I}), \tag{12}$$

where $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})$ are non-linear transformations of $\mathbf{x}$ parameterised by a neural network with parameters $\boldsymbol{\phi}$ in a similar fashion as Equation (4) is used to parameterise the generative model. Equation (12) is called the inference model, and it is illustrated in Figure 1B together with the inference model for the GMVAE, which is described in Supplementary Section S1.

The parameters of the generative and inference models are optimised simultaneously using a stochastic gradient descent algorithm. The reparameterisation trick for sampling from $q_{\boldsymbol{\phi}}(\mathbf{z} \,|\, \mathbf{x})$ allows back-propagation of the gradients end-to-end (Kingma and Welling, 2013). We use a *warm-up* scheme during optimisation as described in the Supplementary Section S2, and we report the marginal log-likelihood lower bound averaged over all examples. The VAE *reconstruction* of $\mathbf{x}$ is defined as the mean of the expression values using a stochastic encoding and decoding step: $\tilde{\mathbf{x}} = \int \sum_{\mathbf{x}'} \mathbf{x}' p(\mathbf{x}' \,|\, \mathbf{z}) q(\mathbf{z} \,|\, \mathbf{x}) d\mathbf{z}$. The variance of the reconstruction can also be computed in a similar fashion.

2.5 Clustering and visualisation of the latent space

The aim of using a VAE is to capture the joint distribution of multivariate data such as gene expression profiles. Since our method is likelihood-based, model comparison can be performed by evaluating the marginal log-likelihood (lower bound) on a test set for different models. For model comparison with non-likelihood-based methods or methods that preprocess the data, we need other metrics such as a clustering quality. For latent variable methods, clustering in the latent space is the most obvious approach. Therefore, we do this as well.

The categorical latent variable used in the GMVAE model gives a built-in clustering by using as assignment the mixing coefficient with the highest *responsibility* $q_{\boldsymbol{\phi}}(y|\mathbf{x})$. The responsibilities are computed as part of the inference network as described in Supplementary Section 1. The standard VAE model does not have this feature built in, so instead k-means clustering (kM) is used to cluster the latent representations of the cells for this model. We also test k-means clustering for the GMVAE model.

We use two clustering metrics to measure the similarity between a clustering found by one of our models and the clustering given in the data set. These are the adjusted Rand index (Hubert and Arabie, 1985), R_{adj}, and adjusted mutual information (AMI; Vinh *et al.* (2009)). For both metrics, two identical clusterings have a value of 1, while the expected value of the metrics is 0 (they are not bounded below).

To visualise the latent space in which the latent representations reside, the mean values of the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are plotted instead of the

Table 1. Overview of gene expression data sets.

Data set	Examples	Features	Classes	Sparsity
PBMC	92 043	32 738	9[a]	0.980
MBC	1 306 127	27 998	—	0.928
TCGA	10 830	58 581	29	0.525
GTEx	11 688	54 271	53	0.472

[a] CD19+ B cells, CD34+ cells, CD4+ helper T cells, CD4+/CD25+ regulatory T cells, CD4+/CD45RA+/CD25- naïve T cells, CD4+/CD45RO+ memory T cells, CD56+ natural killer cells, CD8+ cytotoxic T cells, CD8+/CD45RA+ naïve cytotoxic T cells.

samples. This is done to better get a representation of the probability distribution for the latent representation of each cell. For the standard VAE, this is just $\bar{\mathbf{z}} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})$, and for the GMVAE, $\bar{\mathbf{z}} = \mathrm{E}_y\big[\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}, y)\big]$. For latent space with only two dimensions, we plot these directly, but to visualise higher dimensional latent spaces, we project to two dimensions using t-distributed stochastic neighbour embeddings (t-SNE; van der Maaten and Hinton, 2008).

2.6 RNA-seq data sets

We model four different RNA-seq data sets as summarised in Table 1. The first data set is of single-cell gene expression counts of peripheral blood mononuclear cells (PBMC) used for generation of reference transcriptome profiles (Zheng *et al.*, 2017). It is combined from nine separate data sets of different purified cell types and published by 10x Genomics[3], and we use the filtered gene–cell matrices for each data set. Another data set of purified monocytes is also available, but since another cell type was discovered in this data set (Zheng *et al.*, 2017), and since no separation of the two is available in the data set, we only use the other nine data sets, which are listed in Table 1. This table also shows the high degree of sparsity of the data set.

The second data set is a bulk RNA-seq data set made publicly available by TCGA (Weinstein *et al.*, 2013).[4] This data set consists of RSEM (Li and Dewey, 2011) expected gene expression counts for samples of human cancer cells from 29 tissue sites. Gene IDs are used in this data set, so the available mapping from TCGA is used to map the IDs to gene names. The difference in number of genes and sparsity from the original data set is not large.

The two remaining data sets are described, modelled, and analysed in Supplementary Sections 3–4.

[3] Available online at `https://support.10xgenomics.com/single-cell-gene-expression/datasets/` (under "Single Cell 3′ Paper: Zheng et al. 2017").

[4] Available online at `https://xenabrowser.net/datapages/?dataset=tcga_gene_expected_count&host=https://toil.xenahubs.net`.

2.7 Experiment setup

Each data set is divided into training, validation, and test sets using a 81 %–9 %–10 % split with uniformly random selection. The training sets are used to train the models, the validation sets are used for hyperparameter selection during training, and the test sets are used to evaluate the models after training.

For the deep neural networks, we examine different network architectures to find the optimal one for each data set. We test deep neural networks of both one and two hidden layers with 100, 250, 500, or 1000 units each. We also experiment with a latent space of both 10, 25, 50, and 100 dimensions. A standard VAE with the negative binomial distribution as the likelihood function (a VAE-NB model) is used for these experiments. Using the optimal architecture, we test the link functions introduced in Section 2.3 as likelihood function for both the standard VAE as well as the GMVAE. We train and evaluate each model three times for each likelihood function.

The hidden units in the deep neural networks use the rectifier function as their non-linear transformation. For real, positive parameters (λ for the Poisson distribution, r for the negative binomial distribution, σ in the Gaussian distribution), we model the natural logarithm of the parameter. For the standard deviation σ in the Gaussian-mixture model, however, we use the softplus function, $\log(1 + \mathrm{e}^x)$, to constrain the possible covariance matrices to be only positive-definite. The units modelling the probability parameters in the negative binomial distribution, p, and the zero-inflated distributions, π_k, use the sigmoid function, while for the categorical distributions in the constrained Poisson distribution, the piece-wise categorical distributions, as well as for the Gaussian-mixture model, the probabilities are given as logits with linear functions, which can be evaluated as probabilities using the softmax normalisation. Additionally for the piece-wise categorical distributions, we choose the cut-off K to be 4, since this strikes a good balance between the number of low and high counts for all examined data sets (see Supplementary Fig. S1 for an example of this). For the k-means clustering and the GMVAE model, the number of clusters is chosen to be equal to the number of classes, if cell types were provided.

The models are trained using one Monte Carlo sample for each example and using the Adam optimisation algorithm (Kingma and Ba, 2014) with a mini-batch size of 100 and a learning rate of 10^{-4}. Additionally, we use batch normalisation (Ioffe and Szegedy, 2015) to improve convergence speed of the optimisation. We train all models for 500 epochs and use early stopping with the validation marginal log-likelihood lower bound to select parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. We also use the warm-up optimisation scheme for 200 epochs for VAE and GMVAE models exhibiting overregularisation. As a baseline model to test whether the non-linearities in the neural network contributes to a better performance, we try out

models with the same link functions, but without the hidden layers in the generative process. This corresponds to factor analysis (FA) with a generalised linear model link function. We keep the inference model the same. The optimal network architecture for the inference process is found in the same way as for the VAE models. As with the VAE model, we cluster the latent space using k-means clustering. We also use this method to cluster the original data as another baseline.

We compare our best-performing models with Seurat and scvis for the PBMC data set. Seurat have in turn been compared to several other single-cell clustering methods (Duò *et al.*, 2018). Since Seurat uses the full data set, to make a fair comparison, we also train our best-performing models on the full data set. Because scvis is a visualisation method, we compare our models to it visually. In addition, we have trained our best-performing models using only two latent variables to visualise the latent space without using t-SNE.

2.8 Software implementation

The models described above have been implemented in Python using the machine-learning software library TensorFlow in an open-source and platform-independent software tool called scVAE (single-cell variational auto-encoders), and the source code is freely available online[5] along with a user guide and examples.

The RNA-seq data sets presented in Section 2.6 along with others are easily accessed using our tool. It also has extensive support for sparse data sets and can thus be used on the largest data sets currently available as demonstrated in Supplementary Section S3.

3 Results

Both the standard VAE and the GMVAE have been used to model both single-cell and bulk RNA-seq data sets (see Supplementary Sections S3–S4 for more). The performance of different likelihood functions have been investigated, and the latent spaces of the models have been examined.

3.1 scRNA-seq data of purified immune cells

We first modelled the scRNA-seq data set of gene expression counts for peripheral blood mononuclear cells (PBMC; Zheng *et al.*, 2017). To assess the optimal network architectures for these data sets, we first trained VAE-NB models with different network architecture as mentioned in Section 2.7 on the data set with all genes included. This was carried out as a grid search of number of hidden units and latent dimensions (Supplementary Fig. S2). We found that using two smaller hidden layers (of 100 units each) in the generative and inference processes yielded better results than only using one hidden layer, and a high-dimensional latent space of 100 dimensions resulted in the highest test marginal log-likelihood lower bound. A similar network architecture grid search were carried out using FA-NB models, and the optimal architecture was two large hidden networks of 1000 units each with a latent space of only 25 dimensions.

With these network architectures, different discrete likelihood functions were examined for the VAE, GMVAE, and FA models. The mean marginal log-likelihood lower bound and the mean adjusted Rand index for each model and likelihood function evaluated on the test set are plotted against each other in Figure 2. From this figure, it is clear that using a GMVAE model with the negative binomial distribution yielded the highest lower bound as well as the highest Rand index, with the zero-inflated Poisson distribution as the next best. For each likelihood function, the GMVAE model yielded the highest lower bound, and in most cases, the distribution-based clustering of the GMVAE model outperforms k-means clustering in the latent space of the same model or of the other models on average. This shows that complex non-linear models really makes a difference for the marginal log-likelihood of the data. It should be noted, however, that there does not seem to a high positive correlation between Rand index and marginal likelihood lower bound. Similar conclusions are obtained using adjusted mutual information. The latent space of the GMVAE-NB model for the test set of the PBMC data set can be seen in Figure 3 (see Supplementary Fig. S3 for a larger version), which shows different clusters corresponding to distinct cell types, while more similar cell types are clustered close to each other or mixed together. Furthermore, cells for several cell types seem to be split into subtypes which could both be a biological and a batch effect. There is more overlap of the cell types in the latent space for the VAE-NB and FA-NB models (Supplementary Fig. S4–5).

The PBMC data set has also been analysed using Seurat at different resolutions (see Supplementary Table S1), and the highest adjusted Rand index obtained was $R_{\text{adj}} = 0.634$ at a resolution of 0.2. This yields nine clusters, which is also the number of cell types in the data set. This performance of Seurat have been included in Figure 2, and even though Seurat is trained on the full data set, two of the GMVAE models (using withheld data) perform better on average. A GMVAE-NB model was also trained on the full data set. This resulted in a lower bound of $\mathcal{L} = -2046.0 \pm 0.4$ and a Rand index of $R_{\text{adj}} = 0.656 \pm 0.039$ (see Supplementary Fig. S6 for its latent space), which is better than the best Rand index obtained using Seurat on average. In the latent space, the cells are also almost separated according to cell type with two notable exceptions: The helper T cells are mixed in with both the regulatory and the naïve T cells (because they are helper cells, they can express genes for either cell type), and the cytotoxic T cells are split into two major groupings. It also looks like some cells are clustered together in subtypes. We also trained a scvis model with standard parameters as well as an GMVAE-NB model with a 2-d latent variable on the PBMC data set (see Supplementary Fig. S7 for

[5] `https://github.com/chgroenbech/scVAE`.

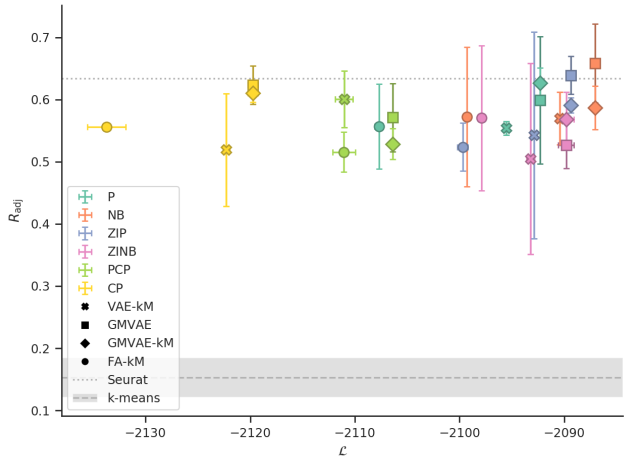

Fig. 2. Comparison of VAE, GMVAE, and FA models with different likelihood functions trained and evaluated on the PBMC data set. For each combination the mean and the standard deviation of the adjusted Rand index is plotted against the marginal log-likelihood lower bound. The mean and the standard deviation of the adjusted Rand index is also plotted for k-means clustering and for Seurat (trained and evaluated on the full data set).

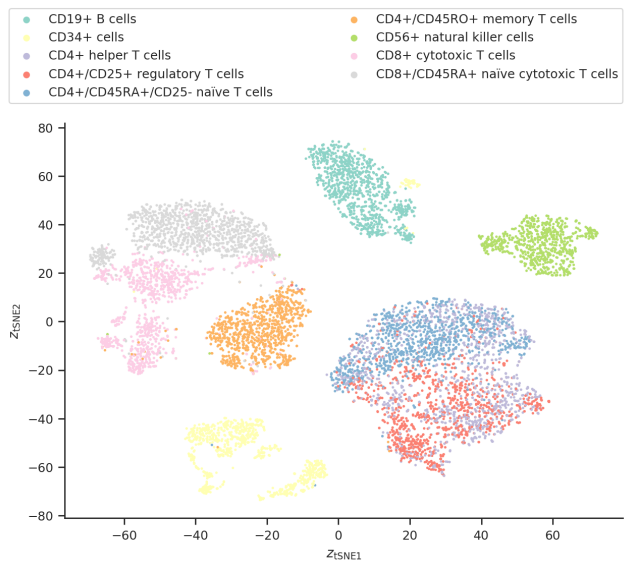

Fig. 3. Latent space of the median-performing GMVAE-NB model trained and evaluated on the PBMC data set. The encoding of the cells in the latent space has been embedded in two dimensions using *t*-SNE and are colour-coded using their cell types. Clear separation corresponding to different cell types can be seen, but some similar cell types are also clustered close together or mixing together.

their latent spaces). Comparing the latent space of the two models, both models succeed in separating out cells with dissimilar cell types, while struggling with cells of similar cell type, but the scvis model handles it better. This 2-d-latent-variable GMVAE-NB also has both a worse lower bound of $\mathcal{L} = -2103.7 \pm 1.3$ and a worse Rand index of $R_{\text{adj}} = 0.420 \pm 0.014$ compared to the GMVAE-NB using the optimal architecture. The *t*-SNE of the latent space of this model also has a much better separation of the cells than the latent space of the scvis model.

3.2 Bulk RNA-seq data of human cancer cells

The bulk RNA-seq data set of RSEM expected gene expression levels for human cancer cell samples from TCGA (Weinstein *et al.*, 2013) was also modelled. On a GPU with 12 GB of memory, we cannot model the data set using all genes with the optimal network architecture. Since the majority of the variance is captured in the 5000–10 000 most varying genes, we choose to limit the number of genes to 5000. As with the previous data set, different network architectures (Supplementary Fig. S8) and likelihood functions (Fig. 4) were investigated for the VAE, GMVAE, and FA models. There is a clear difference in marginal log-likelihood lower bound between using either of the negative binomial distributions from any of the Poisson distributions[6] as can be seen in Figure 4. For each likelihood function, the GMVAE model yielded the highest lower bound (see Supplementary Fig. S9 for a closer look at the NB models). *k*-means clustering on the VAE and GMVAE latent spaces for all likelihood functions resulted in the best adjusted Rand indices in general with the best one being for the GMVAE-kM-P model, whereas the distribution-based clustering of the GMVAE models performs a bit worse using the negative binomial distributions and much worse using the Poisson distributions. The latent space of the GMVAE-NB model for the test set of the TCGA data set can be seen in Figure 5 (see Supplementary Fig. S10 for a larger version), where samples belonging to different tissue sites are clearly separated. The latent space for the VAE-NB and FA-NB models look similar despite the difference in Rand index (Supplementary Fig. S11–12).

4 Discussion

We show that VAEs can be used to model single-cell and bulk RNA-seq data. The GMVAE model achieves higher marginal log-likelihood lower bounds on all data sets as well as a higher Rand indices for the PBMC data set compared to corresponding VAE and FA models, and the latent spaces for the GMVAE also show clear separation according to cell type. Its built-in clustering helps the GMVAE achieve this. The correlation between lower bounds and Rand indices, however, was in general weak

[6] The constrained Poisson distribution was not able to model the TCGA data set at all.

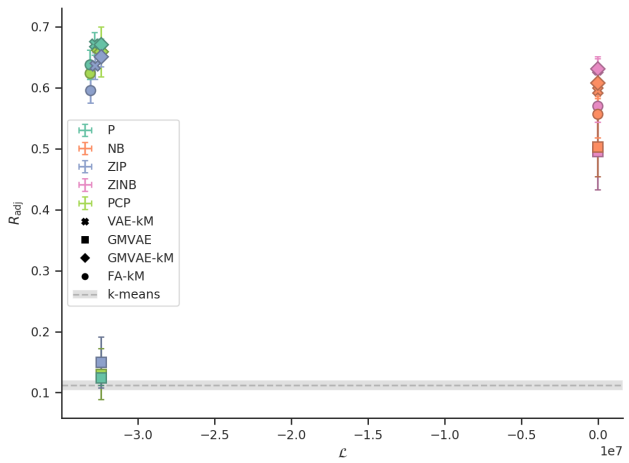

Fig. 4. Comparison of VAE, GMVAE, and FA models with different likelihood functions trained and evaluated on the gene name-mapped TCGA data set limited to the 5000 most varying genes. For each combination the mean and the standard deviation of the adjusted Rand index are plotted against the marginal log-likelihood lower bound. The mean and the standard deviation of the adjusted Rand index are also plotted for k-means clustering.

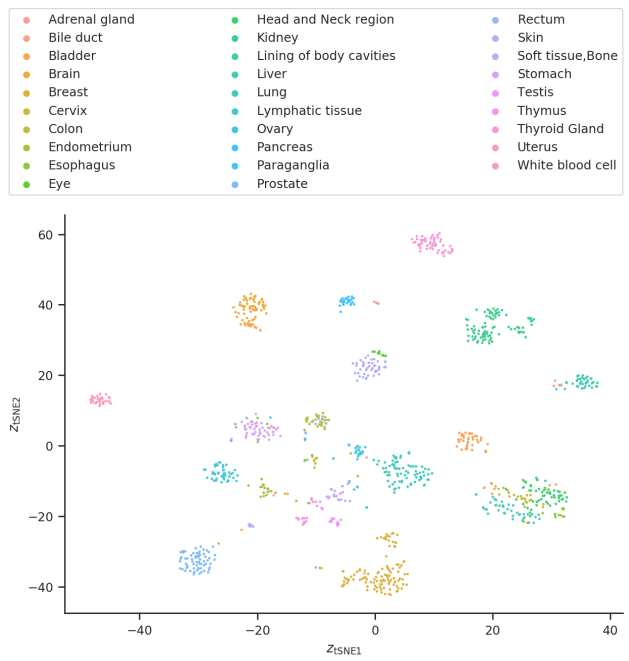

Fig. 5. Latent space of the median-performing GMVAE-NB trained and evaluated on the gene name-mapped TCGA data set limited to the 5000 most varying genes. The encoding of the cells in the latent space has been embedded in two dimensions using t-SNE and are colour-coded using the tissue sites to which they belong. Many of the clusters are clearly separated and they correspond to different tissue sites.

for all data sets. We have also shown that using a simpler linear-factor model for the generative process lead to significantly worse lower bounds, demonstrating that non-linear transformations can more easily express more subtle co-variation patterns in the data sets. We found that the negative binomial distribution modelled most data sets for most models being a close second to its zero-inflated version in the remaining cases. The distribution-based clustering of the GMVAE model, however, struggles somewhat with bulk RNA-seq data, but this might be explained by the diversity within each data sample.

In modelling four different data sets, we have found the following guidelines help in achieving a higher marginal log-likelihood lower bound score on a data set. The network architecture should adapt to the data set: Use a large network when there are more cells and/or more cell types in the data set and when the data is less sparse. The models for all data sets also benefited from deeper network architectures. If the optimal network architecture is too large for computation, one can limit the number of genes to the most varying genes.[7] For the likelihood function, try using the negative binomial distribution (or its zero-inflated version) first. In addition, the warm-up optimisation scheme proved valuable in avoiding overregularisation when modelling the scRNA-seq data sets.

Compared to Seurat (Satija *et al.*, 2015) the GMVAE-NB model achieved a higher adjusted Rand index on average for the PBMC data set, and this was done working directly on the transcript counts with no preprocessing of the data and using distribution-based clustering. With the GMVAE being a generative model, it can also model unseen data whereas Seurat cannot. By modelling the cells with the GMVAE model using the optimal configuration and then visualising the latent space using t-SNE, we also achieve a better visual representation of cells than the scvis model (Ding *et al.*, 2018). In Supplementary Section 3, we also showed that our method can scale up to very large data sets like the data set of 1.3 million mouse brain cells from 10x Genomics.

5 Summary and conclusion

We show that two variations of the variational auto-encoder are able to model gene expression counts using appropriate discrete probability distribution as likelihood (link) functions and provide a software implementation. These models are probabilistic and put a lower bound on the marginal likelihood of the data sets, enabling us to perform likelihood-based model comparison. We have applied both models successfully to both single-cell and bulk RNA-seq data sets, and the GMVAE model achieves better clustering and representation of the cells than Seurat and the scvis model. Building clustering into the Gaussian-mixture variational auto-encoder, we have a model that can cluster cells into cell populations without the need of a pipeline with several parameter

[7] We did this for the bulk RNA-seq data set, and the optimal network architecture stayed the same.

selection steps. Having both an inference process and a generative process, makes it possible to project new data onto an existing latent space, or even simulate new data from samples in the latent space. This means that new cells can be introduced to an already trained model, and it could enable combining the latent representations of two cells to generate a cell and the transitional states in-between. In the scVI model (Lopez *et al.*, 2018), a zero-inflated negative binomial distribution is used. We observe that using a negative binomial distribution overall is a better model.

While preprocessing the transcript counts can correct for batch effects, cell-cycle effects, and other confounders, we would rather build these corrections into the model in the future, e.g., as done by Lopez *et al.* (2018). We would also like to make the models more flexible by adding more latent variable layers (Sønderby *et al.*, 2016), making the models learn the number of clusters (Rasmussen, 2000) rather than setting it a priori, and learn the mixture coefficients of the prior distribution in the GMVAE rather keeping them uniform. We could also use semi-supervised learning and active learning to better classify cells and identify cell populations. This would also help with transfer learning enabling modelling multiple data sets with the same model. As mentioned above, it should also be possible to combine encodings of the cells in the latent space and produce in-between cells like Lotfollahi *et al.* (2018). We note that it is possible to apply these models to data sets with multiple modalities such as RNA-seq and exome sequencing (Brouwer and Lió, 2017).

Acknowledgements

This work was supported by the Lundbeck Foundation to THP; and the Novo Nordisk Foundation to THP. We would like to thank Lars Maaløe for discussions about multimodal VAEs and technical help for implementing these. We also want to thank Aki Vehtari for discussions about discrete probability distributions and model comparison. Lastly, we gratefully acknowledge the support of NVIDIA Corporation with the donation of Tesla Titan Xp GPUs used for this research.

References

Bowman, S.R. *et al.* (2016). Generating sentences from a continuous space. *arXiv preprint*.

Brouwer, T. and Lió, P. (2017). Bayesian hybrid matrix factorisation for data integration. In: Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 557–566. PMLR.

Chen, L. *et al.* (2016). Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17(S1), S9.

Cui, H. *et al.* (2017). Boosting gene expression clustering with system-wide biological information: A robust autoencoder approach. *bioRxiv*.

Dilokthanakul, N. *et al.* (2016). Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint*.

Ding, J. *et al.* (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.*, 9(1), 2002.

duVerle, D.A. *et al.* (2016). CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics*, 17(1), 363.

Duò, A. *et al.* (2018). A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7, 1141.

Eraslan, G. *et al.* (2018). Single cell RNA-seq denoising using a deep count autoencoder. *bioRxiv*.

Gatys, L.A. *et al.* (2015). A neural algorithm of artistic style. *arXiv preprint*.

Ghahramani, A. *et al.* (2018). Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. *bioRxiv*.

Goodfellow, I.J. *et al.* (2014). Generative adversarial nets. *arXiv preprint*.

Gupta, A. *et al.* (2015). Learning structure in gene expression data using deep architectures, with an application to gene clustering. *bioRxiv*.

Haghverdi, L. *et al.* (2018). Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, 36, 421.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classif.*, 2(1), 193–218.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*.

Jiang, Z. *et al.* (2017). Variational deep embedding: An unsupervised and generative approach to clustering. In: Sierra, C. (ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 1965–1972. IJCAI Organization.

Johnson, M. *et al.* (2016). Composing graphical models with neural networks for structured representations and fast inference. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2946–2954. NeurIPS.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*.

Kingma, D.P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint*.

Kingma, D.P. *et al.* (2014). Semi-supervised learning with deep generative models. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3581–3589. NeurIPS.

Kiselev, V.Y. *et al.* (2017). SC3: consensus clustering of single-cell rna-seq data. *Nat. Methods*, 14, 483.

Li, B. and Dewey, C.N. (2011). RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics*, 12(1), 323.

Lin, W. *et al.* (2018). Variational message passing with structured inference networks. In: *International Conference on Learning Representations*.

Lopez, R. *et al.* (2018). Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12), 1053–1058.

Lotfollahi, M. *et al.* (2018). Generative modeling and latent space arithmetics predict single-cell perturbation response across cell types, studies and species. *bioRxiv*.

Maaløe, L. *et al.* (2017). Semi-supervised generation with cluster-aware generative models. *arXiv preprint*.

Oshlack, A. *et al.* (2010). From RNA-seq reads to differential expression results. *Genome Biol.*, 11(12), 220.

Rasmussen, C.E. (2000). The infinite gaussian mixture model. In: Solla, S.A., Leen, T.K., and Müller, K.R. (eds.), *Advances in Neural Information Processing Systems 12*, pp. 554–560, Cambridge, MA, USA. MIT Press.

Regev, A. *et al.* (2017). The human cell atlas. *eLife*, 6.

Rezende, D.J. *et al.* (2014). Stochastic backpropagation and approximate inference in deep generative models. In: Xing, E.P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286. PMLR.

Roberts, A. *et al.* (eds.) (2017). *Hierarchical Variational Autoencoders for Music*.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *Int. J. Approx. Reason.*, 50(7), 969–978.

Satija, R. *et al.* (2015). Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, 33(5), 495–502.

Seeger, M.W. *et al.* (2016). Bayesian intermittent demand forecasting for large inventories. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4646–4654. NeurIPS.

Sun, Z. *et al.* (2017). DIMM-SC: a Dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics*, 34(1), 139–146.

Sønderby, C.K. *et al.* (2016). Ladder variational autoencoders. *arXiv preprint*.

Tan, J. *et al.* (2014). Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. In: *Biocomputing 2015*, pp. 132–143, Singapore. World Scientific.

Tan, J. *et al.* (2016). ADAGE-based integration of publicly available *Pseudomonas aeruginosa* gene expression data with denoising autoencoders illuminates microbe-host interactions. *mSystems*, 1(1), e00025–15.

Vallejos, C.A. *et al.* (2017). Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods*, 14(6), 565–571.

van der Maaten, L.J. and Hinton, G. (2008). Visualizing high-dimensional data using *t*-sne. *J. Mach. Learn. Res.*, 9, 545.

Vinh, N.X. *et al.* (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In: *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1073–1080, New York, NY, USA. ACM.

Way, G.P. and Greene, C.S. (2017). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In: *Biocomputing 2018*, Singapore. World Scientific.

Weinstein, J.N. *et al.* (2013). The cancer genome atlas pan-cancer analysis project. *Nat. Genet.*, 45, 1113–1120.

Zheng, G.X.Y. *et al.* (2017). Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, 8, 14049.