Generating Synthetic Single-Cell RNA-Sequencing Data from Small Pilot Studies using Deep Learning

Martin Treppner[1,2], Adrián Salas-Bastos[3], Moritz Hess[1,2], Stefan Lenz[1,2], Tanja Vogel[3,4], and Harald Binder[1,2]

[1] *Institute of Medical Biometry and Statistics*
Faculty of Medicine and Medical Center - University of Freiburg

[2] *Freiburg Center of Data Analysis and Modelling Mathematical Institute*
Faculty of Mathematics and Physics University of Freiburg

[3] *Institute of Anatomy and Cell Biology, Department of Molecular Embryology*
Medical Faculty, University Freiburg

[4] *Center for Basics in NeuroModulation (NeuroModul Basics)*
University of Freiburg

May 27, 2020

Abstract

Motivation: When designing experiments, it is advised to start with a small pilot study for determining the sample size of full-scale investigations. Deep learning techniques for single-cell RNA-sequencing data that can uncover low-dimensional representations of expression patterns within cells could be useful also with pilot data. Here, we examine the ability of these methods to learn the structure of data from a small pilot study and generate synthetic expression datasets useful for planning full-scale experiments.

Results: We investigate two deep generative modeling approaches. First, we consider single-cell variational inference (scVI) in two variants, either generating samples from the posterior distribution, which is the standard approach, or from the prior distribution. Second, we propose single-cell deep Boltzmann machines (scDBM), which might be particularly suitable for small datasets. When considering the similarity of clustering results on synthetic data to ground-truth clustering, we find that $scVI_{posterior}$ exhibits high variability. Expression patterns from $scVI_{prior}$ and scDBM perform better. All approaches show mixed results for cell types with different abundance by sometimes overrepresenting highly abundant cell types and missing less abundant cell types. Taking such tradeoffs in performance into account, we conclude that for making inference from a small pilot study to a larger experiment, it is advantageous to use $scVI_{prior}$ or scDBM, as $scVI_{posterior}$ produces signals that are not justified by the original data. The proposed scDBM seems to have an advantage for small pilot datasets. Overall, the results show that generative deep learning approaches might be valuable for supporting the design of scRNA-seq experiments.

1. Introduction

Single-cell RNA-sequencing (scRNA-seq) experiments result in data reflecting gene expressions for individual cells in tissues, leading to an improved understanding of cell-type composition. However, despite the increase in throughput of scRNA-seq experiments, dataset sizes have to be carefully chosen due to budget constraints (Ye *et al.*, 2019). In particular, the typical data analysis workflow of detecting cell types via dimensionality reduction and subsequent clustering raises the question of how many cells need to be assayed to identify cell types with high confidence (Svensson *et al.*, 2019). In the following, we are going to investigate generative deep learning techniques that might be useful for answering such questions.

Deep generative approaches, such as variational autoencoders (VAEs; Kingma and Welling, 2013), are increasingly used to investigate the underlying structure of scRNA-seq data by learning a low-dimensional latent representation of gene expression within cells. Often, the focus of these applications is to explore latent features of the data — representing cell types — after which they are used for clustering, imputation, or differential expression analysis (Lopez *et al.*, 2018; Eraslan *et al.*, 2019; Amodio *et al.*, 2019). Besides dimension reduction, another interesting property of these generative approaches is that they can provide synthetic data once they have been trained on some dataset. As experimental design of scRNA-seq studies is often based on simulations (Hafemeister, 2019; Zappia *et al.*, 2017; Zhang *et al.*, 2019; Svensson *et al.*, 2019; Marouf *et al.*, 2020), such synthetic data could be useful, e.g., when training a generative approach on some pilot data. Sampling from latent representations then allows for generating in-silico expression patterns, ideally reflecting the most important patterns from the pilot data, and can subsequently be utilized for planning experiments. More specifically, researchers would specify different numbers of cells to be simulated, then apply downstream analyses to the simulated data, after which they evaluate the number of cells that promises the desired statistical power of the downstream analysis to detect anticipated effects—or rare cell types.

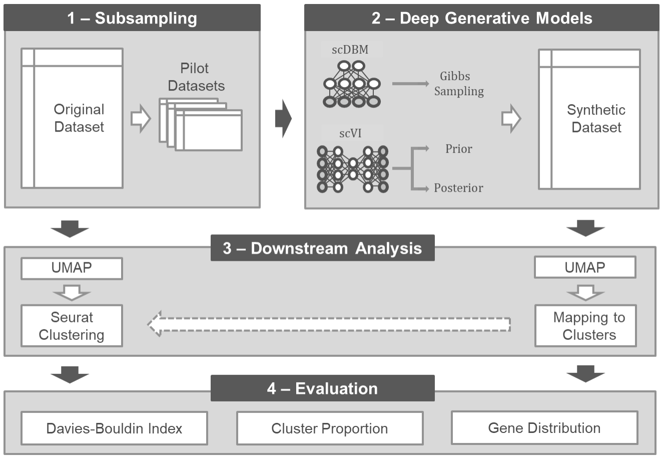

Figure 1: *Design for evaluating the performance of deep generative models with small pilot datasets: (1) Take a subsample from an original dataset (4182 cells) to obtain pilot data with a known ground truth. (2) Train the deep generative approaches on the subsampled pilot dataset and generate synthetic cell data in the size of the original data. (3) Apply the anticipated data analysis, in this case dimensionality reduction with UMAP and Seurat clustering, to the original data and map each synthetic observation to the closest observation from the original data, thus getting a cluster assignment. (4) Evaluate the quality of synthetic samples based on the Davies-Bouldin index, cluster proportions and distributions per gene. The whole analysis is performed for different sizes of pilot datasets (500, 1000, or 2000 cells) and repeated 30 times for each size.*

The right part of **Figure 1** indicates this workflow. In contrast, the left part illustrates the approach we are going to use in the following for evaluating the performance of deep generative approaches in this setting.

In particular, we will use a large ground truth dataset and small pilot datasets drawn from the large set to investigate whether deep learning approaches can deliver synthetic data similar to the original data when presented only with a small pilot dataset.

Besides VAEs, we also consider a second generative deep learning approach, namely deep Boltzmann machines (DBMs). While VAEs have already been proposed for scRNA-seq data, DBMs still need to be adapted, and we show how this can be achieved using them with a negative binomial distribution and incorporating a regularized Fisher scoring algorithm to estimate the inverse dispersion parameter. We chose DBMs because synthetic observations are generated by Gibbs sampling, which has theoretical properties that are potentially advantageous for working with smaller sample sizes compared to variational inference in VAEs (Blei *et al.*, 2017).

VAEs reconstruct their input through a bottleneck layer that corresponds to a low-dimensional latent representation. They offer two ways of generating samples from the latent representation. Most commonly, samples are generated from the posterior, which is the probability of the latent variables given the original data. In a pilot study setting, this will typically mean that multiple copies of the original observations have to be used to obtain a larger synthetic dataset. This might be problematic, as patterns or random fluctuations from single cells could be over-emphasized. In contrast, sampling from the prior might produce samples from a diverse region of the latent space. In our evaluation together with DBMs, we therefore not only investigate the performance of VAEs when feeding in the original data multiple times for obtaining a larger number of cells but also when sampling directly from the prior, which has —to our knowledge— not been considered in the scRNA-seq literature so far.

In the following, we are going to introduce a novel variant of DBMs, single-cell deep Boltzmann machines (scDBM), and give a brief overview of the other methods used throughout this work—single-cell variational inference (scVI) and the Davies-Bouldin index (DBI), which will be used as a performance criterion in the subsequent empirical investigation (see **Figure 1**). We close with a discussion and an outlook on how the investigated techniques could subsequently be used for sample size calculation.

2. Methods

2.1. Single-Cell Deep Boltzmann Machine

We adapted deep Boltzmann Machines (DBMs), an unsupervised neural network approach with multiple hidden layers (Salakhutdinov and Hinton, 2009), to the negative binomial distribution. Specifically, we employ the exponential family harmonium framework (Welling *et al.*, 2005) that allows restricted Boltzmann machines (RBMs), the single-hidden layer version of DBMs, to deal with any distribution from the exponential family as input. This framework was further extended and simplified by Li and Zhu (2018).

We use a parametrization of the negative binomial probability mass function that has been suggested by Risso *et al.* (2018):

$$p_{NB}(v;\mu,\theta) = \frac{\Gamma(v+\theta)}{\Gamma(v+1)\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{v}$$

The mean of the distribution is denoted as μ, the variance is given by $\mu + \mu^2/\theta$, and θ is the inverse dispersion. Γ denotes the gamma function.

For simplicity, we describe a three-layer DBM where the visible layer corresponds to an input of unique molecular identifier (UMI) counts for M genes, which can be modeled by a negative binomial distribution (Grün *et al.*, 2014). The first and second hidden layers are denoted as $h^{(1)}$ and $h^{(2)}$, respectively.

Following Li and Zhu (2018), we define the energy function of the state $\{x, h^{(1)}, h^{(2)}\}$ as:

$$E(x, h^{(1)}, h^{(2)}; \Theta) = -a^T x - \sum_{m=1}^{M} \log\left(\frac{(x_m + \theta_m - 1)!}{(\theta_m - 1)! x_m!}\right)$$
$$-b^{(1)T} h^{(1)} - b^{(2)T} h^{(2)} - x^T W^{(1)} h^{(1)} - h^{(1)T} W^{(2)} h^{(2)}$$

Here, a, $b^{(1)}$, and $b^{(2)}$ are the bias terms of the first, second, and third layer, respectively. Furthermore, $W^{(1)}$ and $W^{(2)}$ denote the weight matrices connecting the layers. Hence, $\Theta = (\theta, a, b^{(1)}, b^{(2)}, W^{(1)}, W^{(2)})$ are the model parameters. Therefore, the probability of the visible vector is defined as:

$$p(x; \Theta) = \frac{1}{Z(\Theta)} \sum_{h^{(1)}, h^{(2)}} \exp(-E(x, h^{(1)}, h^{(2)}; \Theta))$$

$Z(\Theta)$ is the partition function which is typically intractable (Salakhutdinov and Hinton, 2009). According to this, the conditional distributions over the visible and the two sets of hidden units are given as:

$$p(x|h^{(1)}) = \prod_{m=1}^{M} NB\Big(\hat{\mu}, \hat{\theta}\Big), \quad \hat{\mu} = \frac{\hat{\theta}_m e^{\hat{a}_m}}{(1 - e^{\hat{a}_m})}$$

$$p(h^{(1)}|x, h^{(2)}) = \prod_{k=1}^{K} Bern\Big(\sigma(\hat{b}_k^{(1)})\Big)$$

$$p(h^{(2)}|h^{(1)}) = \prod_{l=1}^{L} Bern\Big(\sigma(\hat{b}_l^{(2)})\Big)$$

Where $\hat{a}_m = a_m + \sum_{k=1}^{K} W_{mk}^{(1)} h_k^{(1)}$ represents the estimate for the visible bias of UMI counts per gene m ($m = 1, ..., M$) and the bias of the first and second hidden layer correspond to $\hat{b}_k^{(1)} = b_k^{(1)} + \sum_{m=1}^{M} W_{mk}^{(1)} x_m + \sum_{l=1}^{L} W_{kl}^{(2)} h_k^{(2)}$ and $\hat{b}_l^{(2)} = b_l^{(2)} + \sum_{k=1}^{K} W_{kl}^{(2)} h_k^{(1)}$, where $k = 1, \ldots, K$ and $l = 1, \ldots, L$ indicate the hidden nodes in the first and second hidden layer, respectively. The sigmoid activation function is denoted as σ.

Training of the scDBMs via stochastic gradient descent can be performed just as for standard DBMs. For a detailed description, see Salakhutdinov and Hinton (2009) and Hinton and Salakhutdinov (2012).

After training, synthetic observations can be generated by Gibbs sampling. It can be shown that Gibbs sampling produces asymptotically exact samples, which leads to more accurate results as compared to VAEs (Robert and Casella, 2013; Blei *et al.*, 2017). This comes at the cost of a higher computational burden, which might be acceptable in small sample scenarios. In contrast, scVI (described in more detail below) uses variational inference, which scales to scenarios with millions of observations but does not have the advantage of generating exact samples (Blei *et al.*, 2017).

2.1.1 Estimating the Dispersion Parameter

For the negative binomial distribution, we also need to determine values for the inverse dispersion parameter of each gene which is notoriously difficult (Love *et al.*, 2014).

We use a regularized Fisher scoring algorithm (Jennrich and Sampson, 1976) to estimate the inverse dispersion parameter θ_m for each gene m. For this, we use the log-likelihood function of the negative binomial probability mass function indicated above. The Fisher scoring algorithm can be derived using a two-term Taylor expansion of the score function, the first derivative of the log-likelihood, at the initial choice of the inverse dispersion θ_m^0 (Hilbe, 2011). To stabilize estimates of the inverse dispersion parameters, we add $\frac{\lambda}{\theta^2}$ as a regularization term to the log-likelihood, which results in the following scoring algorithm:

$$\theta_{m,k+1} = \theta_{m,k} + \frac{V(\theta_{m,k}) + \lambda\frac{2}{\theta^3_{m,k}}}{\mathcal{I}(\theta_{m,k}) + \lambda\frac{6}{\theta^4_{m,k}}}$$

Here, $V(\cdot)$ is the score function, $\mathcal{I}(\cdot)$ denotes the Fisher information matrix, λ is the regularization parameter, and k is the current iteration step.

The inverse dispersion parameter θ_m corresponds to the amount of heterogeneity between cells, where a smaller θ_m indicates more heterogeneity. Recall that the negative binomial variance is defined as $\mu + \mu^2/\theta$. Due to the regularization term in our model, smaller θ_m are subject to larger regularization. This ensures that we learn the baseline variability between cells, without deflating the estimates of the inverse dispersion due to, e.g., differences between clusters of cells or excess zeros.

2.1.2 scDBM Training

By combining scDBMs with Fisher scoring, we can estimate all model parameters $\Theta = (\theta, a, b^{(1)}, b^{(2)}, W^{(1)}, W^{(2)})$. First, we initialize all parameters at some reasonable values and learn only a subset of Θ, namely, $(a, b^{(1)}, b^{(2)}, W^{(1)}, W^{(2)})$. Hence, the inverse dispersion is fixed. After a predefined number of epochs, say five, we use the regularized Fisher scoring algorithm to estimate the inverse dispersion parameter $\hat{\theta}_m$ and plug the new estimate into the scDBM. Accordingly, all parameters of the scDBM are refined after a fixed time, e.g., every five epochs.

During training, biases and weights of the network have to be constrained, where $a_m = min\{a_m, -\epsilon\}$ with $\epsilon = 10^{-10}$ and $w_{m,k} = min\{w_{m,k}, 0\}$. This is done because we use the natural form of the exponential family and hence a_m is used in logarithmic scale (Li and Zhu, 2018).

2.2. Single-Cell Variational Inference

Lopez *et al.* (2018) proposed a method called single-cell variational inference (scVI), which utilizes the structure of VAEs to encode the transcriptome onto a lower-dimensional representation from which the input is reconstructed. Just as the scDBM, scVI is also based on the (zero-inflated) negative binomial distribution (Lopez *et al.*, 2018). The model comprises two components, the encoder and the decoder parts of the network. Lopez *et al.* (2018) use four neural networks for encoding the size factors and the latent variables using the variational distribution $q(z_n, l_n|x_n, s_n)$ as an approximation to the posterior $p(z_n, l_n|x_n, s_n)$, where z_n is a low-dimensional vector of Gaussians, l_n is a one-dimensional Gaussian encoding technological differences in capture efficiency and sequencing depth, x_n is the vector of observed expressions of all genes of cell n, and s_n describes the batch annotation for each cell (Lopez *et al.*, 2018). The variational distribution can be written as:

$$q(z_n, l_n|x_n, s_n) = q(z_n|x_n, s_n)q(l_n|x_n, s_n)$$

Therefore, the variational lower bound is:

$$\log p(x|s) \geq E_{q(z,l|x,s)} \log p(x|z, l, s)$$
$$-D_{KL}(q(z|x, s)||p(z)) - D_{KL}(q(l|x, s)||p(l))$$

The probabilistic model of scVI is based on a gamma-Poisson mixture. It starts by sampling from the latent space, a standard multivariate normal distribution, which is then fed into a neural network—together with the batch annotation. The neural network then learns the mean proportion of transcripts expressed across all genes. The output is used to sample from a gamma distribution together with the inverse dispersion θ_m. The model accounts for technical effects by incorporating a library size scaling factor which, in combination with the gamma-distributed samples, is used to sample from a Poisson distribution. This mixture of the gamma and Poisson distribution is equivalent to the negative binomial distribution (Lopez *et al.*, 2018). scVI additionally learns a neural network to account for technical dropouts.

Observations are generated from the scVI approach by using original data as input and then sampling from the posterior distribution $p(z|x)$. A straightforward approach for generating more samples than were used during training is to create (multiple) copies of the original data. For example, for scVI trained on 500 cells, we sampled from the model seven times and stacked the resulting samples together to make inference about a larger number of cells. As an alternative, we adapted scVI to sampling from the prior distribution $p(z)$ instead of the more common sampling from the posterior $p(z|x)$. To do that, we changed the inference procedure to sample latent z from $Normal(0, 1)$ and library sizes from $Normal(l_\mu, 1)$.

2.3. Evaluation of Synthetic Data Quality

The overall approach taken here for evaluating the quality of generated synthetic observations is illustrated in **Figure 1**. Specifically, a relatively large original dataset is used as ground truth data, and deep generative approaches are tasked with generating synthetic data based on pilot datasets drawn from the original data. We consider Seurat clustering (Butler *et al.*, 2018;Stuart *et al.*, 2019) on the UMAP representations (Becht *et al.*, 2019) of the original data as a typical data analysis workflow, which provides ground truth cluster labels for the original data. When subsequently assessing synthetic data, each generated observation is assigned the cluster label of the nearest original observation, as determined by Euclidean distance. If a generative approach can provide synthetic observations very close to the original observations, these cluster labels will correspond to a reasonable clustering solution also in the synthetic data. Thus, we can evaluate the quality of the synthetic data by calculating summary statistics for the clusters in the synthetic data, and compare them to cluster statistics from the original data.

Specifically, we use the Davies-Bouldin index (DBI)

$$DBI(C_K) = \frac{1}{K}\sum_{i=1}^{K} D_i,$$

where

$$D_i = max_{j \neq i} R_{ij}$$

with between-cluster similarity

$$R_{ij} = \frac{S_i - S_j}{M_{ij}} \quad , \quad i, j = 1, \ldots, K$$

the distance between cluster centroids

$$M_{ij} = \left\|\bar{x}_i - \bar{x}_j\right\|_p$$

and within-cluster dispersions

$$S_k = \left(\frac{1}{n_k}\sum_{c(i)=k}\left\|x_i - \bar{x}_k\right\|_2^q\right)^{\frac{1}{q}}$$

where we set $p = q = 2$.

Consequently, a small DBI indicates homogeneous and well-separated clusters (Davies and Bouldin, 1979 ; Hennig *et al.*, 2015).

To examine whether the models learn to adequately represent frequencies of different cell types, we also compare the number of cells per cluster in the original data and the synthetic observations.

It should be noted that an in-depth evaluation of samples, instead of comparing model fit based on the log-likelihood, is indispensable because it was shown that comparing deep generative models based only on the log-likelihood can be misleading. In particular, even when log-likelihood is low, the quality generated samples can be good and vice-versa (Theis *et al.*, 2015). In contrast, we focus on properties, such as cluster quality, which are important for experimental design.

2.4. Data Description and Processing

We evaluate the performance of the two scVI variants and the scDBM approach on three typical datasets. First, a 10x Genomics dataset containing peripheral blood mononuclear cells from a healthy donor (Zheng *et al.*, 2017) is considered. We preprocessed the data following Amezquita *et al.* (2019), after which 4182 cells and 1352 most highly variable genes were left for downstream analysis. We refer to this dataset as $PBMC4k$ throughout this work.

Second, analyses are performed on a data set of neuronal subtypes in the mouse cortex and hippocampus, where Zeisel *et al.* (2015) sequenced 3005 cells from male and female juvenile mice. We specifically consider data from 2816 cells and 1816 highly variable genes which were left after preprocessing (Amezquita *et al.*, 2019). We refer to this dataset as $Zeisel$ throughout this work.

Additionally, we demonstrate the performance on a currently unpublished scRNA-seq dataset from the hippocampus of three embryonic (E16.5) mice processed with the CEL-Seq2 protocol (Hashimshony *et al.*, 2016; Sagar *et al.*, 2018). The unnormalized count matrix contained 3808 cells, and we selected the 1500 most highly variable genes for downstream analysis. We used scran and scater (Lun *et al.*, 2016; McCarthy *et al.*, 2017) for pre-processing. We refer to this dataset as $Hippocampus4k$ throughout this work. The results for $Zeisel$ and $Hippocampus4k$ can be found in the supplementary material.

2.5. Implementation

The scDBM implementation is based on the Julia package ‘BoltzmannMachines.jl’ (Lenz *et al.*, 2019) and extends the packages’ scope to scRNA-seq data. Furthermore, we used the Python implementation of scVI (https://github.com/YosefLab/scVI), which we adapted to be able to sample from the prior distribution.

3. Results

We evaluated how well scDBM and scVI perform in learning the distribution of pilot data. To mimic a situation where we want to plan an experiment using a pilot study with a small number of cells, we investigated the impact of varying amounts of cells and generative approaches on the clustering performance, measured by the Davies-Bouldin index (DBI). To do this, we took 30 subsamples of 500, 1000, and 2000 cells of the original dataset, trained the scDBM and scVI on these subsamples, and generated synthetic data. More precisely, we sampled from the scDBM using Gibbs sampling and from scVI using the prior and posterior distribution, respectively. We then applied UMAP and acquired the cluster labels by mapping to the original data (**Figure 1**). For hyperparameters of scDBM and scVI, see Supplementary Table 1.

As seen in **Figure 2**, the scDBM results are rather close to the original data, particularly for small sample sizes, which indicates good performance for small experiments. The performance of $scVI_{posterior}$ is decidedly worse for small sample sizes, with large variability, but improves for larger pilot data. The $scVI_{prior}$ approach exhibits little variability, similar to scDBM, but does not surpass the performance of the latter even for large pilot data. We observed similar patterns for the $Zeisel$ and $Hippocampus4k$ datasets (see Supplementary Data, **Figure S1, S2**).

To uncover heterogeneity and subpopulation frequencies, we inspected whether the models accurately estimated the proportions of cells per cluster. As seen in **Figure 3**, scDBM and $scVI_{posterior}$ tend to underestimate the number of cells per cluster. The scDBM encountered difficulties with detecting the smaller clusters where especially cluster 9 was not identified. In contrast, $scVI_{posterior}$ consistently overestimated the size of clusters 7 and 9. We observed a similar pattern for $scVI_{prior}$ (not shown here).

We also inspected the marginal distributions of two exemplary genes, specifically RPS6 and FTL, in samples from scDBM and scVI and compared them with the distributions in the original data (**Figure 4A**). We observed that the synthetic data generated from the scDBM trained on 500 cells match the true distribution of RPS6 rather well, but miss the bimodality of FTL. In contrast, $scVI_{prior}$ and $scVI_{posterior}$

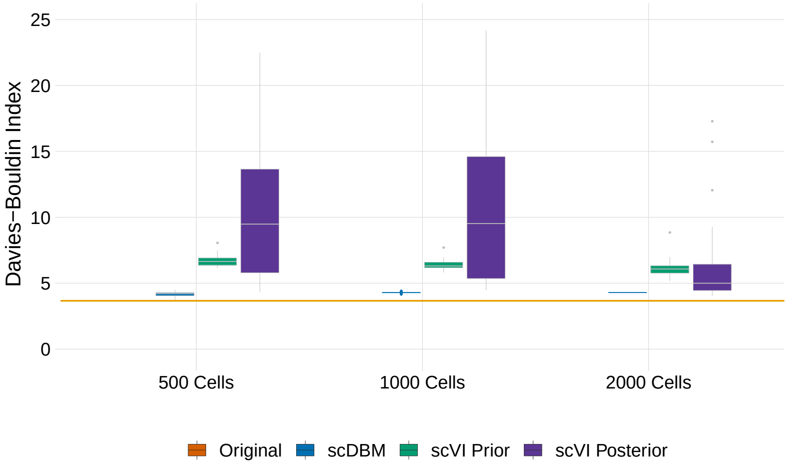

Figure 2: *Davies-Bouldin index (DBI), indicating the quality of synthetic data generated by scDBM and scVI (prior and posterior sampling) from pilot data of different sizes. The orange line indicates the reference DBI for the Seurat clustering on the original data ($PBMC4k$) with 4182 cells.*

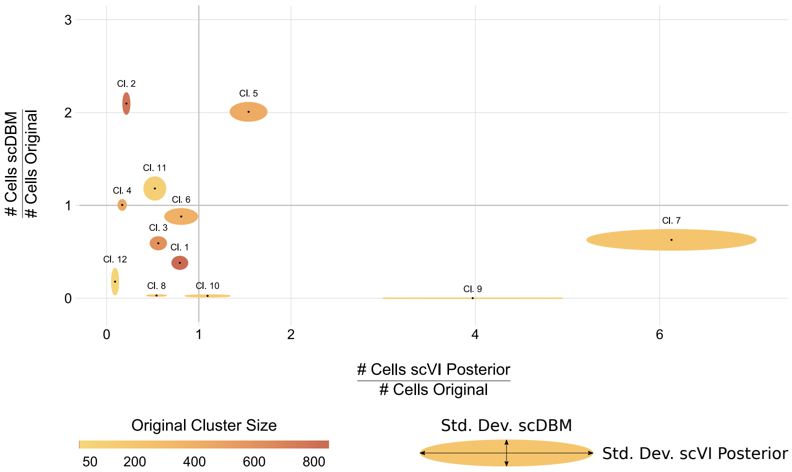

Figure 3: *Performance with respect to recovering cell-type abundances. Each ellipse represents the number of cells in a specific cluster from one of the generative models divided by the number of cells in the original cluster. scDBM and scVI were trained on a subsample of 500 cells after which 4182 cells were generated. The y-axis shows the number of cells per cluster generated by the scDBM divided by the number of cells per cluster in the original data. Hence, if the proportion is higher than one, scDBM overestimated the amount of cells in that particular cluster. The x-axis exhibits the same for samples from $scVI_{posterior}$. The color coding represents the number of cells per cluster in the original dataset. Additionally, the width of ellipses shows the standard deviations of cluster proportions for the 30 subsamples (500 cells each).*

tend to underestimate the expression of FTL, while $scVI_{posterior}$ correctly estimated the mean of RPS6, but exhibits high dispersion.

To get a better insight to what extent relations between genes are recovered, we also consider bivariate scatterplots, shown in **Figure 4B**. Samples from the scDBM adequately reflect the correlation between RPS6 and FTL, whereas $scVI_{posterior}$ tends to overestimate correlation and exhibits a much higher variability.

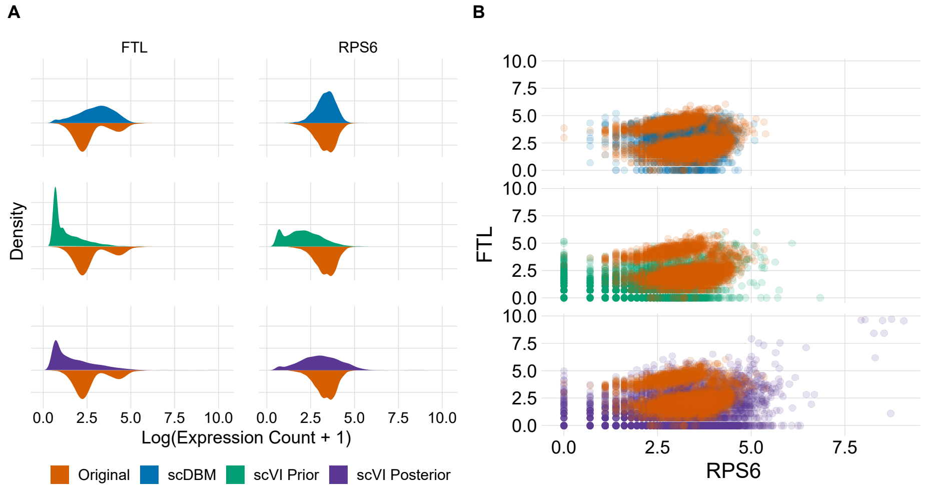

Figure 4: *Univariate (Panel A) and bivariate distribution (Panel B) of expression values for exemplary genes, as generated by scDBM and scVI when trained on 500 cell pilot data, compared to the original data.*

$scVI_{prior}$ performs better in capturing the relation between genes but also shows higher variability compared to the scDBM approach. We also checked pairwise correlations between further genes (results not shown) and found similar patterns.

4. Discussion and Conclusion

In this paper, we investigated how well deep generative models can generate realistic synthetic scRNA-seq data from pilot studies. In particular, we evaluated the quality of the respective samples by training the models on small numbers of cells, after which we generated the number of cells corresponding to a large-scale study. For this, we generated samples using a single-cell deep Boltzmann machine (scDBM), and single-cell variational inference (scVI) approaches, where samples from the latter were either drawn from the prior or the posterior distributions, respectively. We could show that scDBMs outperform scVI in settings with small sample sizes and could, therefore, be more suitable for the design of scRNA-seq experiments — particularly for determining the appropriate amount of cells to be sequenced. Besides this, we examined differences between prior and posterior sampling, where we conclude that posterior sampling in scVI leads to increased variability in clustering results when inferring from a small to a larger population of cells.

Our investigation focused on the number of cells, while sequencing depth is a second critical component for deciding on the trade-off that biologists face in the design of scRNA-seq experiments (Zhang *et al.*, 2018; Svensson *et al.*, 2019). Taking this into account in future work could give even more detailed information on the ability of deep generative models to learn the structure of scRNA-seq data and, consequently, on the quality of generated cells by different methods and sampling approaches. Motivated by the results so far, which already indicate reasonable performance of generative approaches with pilot data, our future work will focus on a fully developed approach for determining the appropriate sample size for scRNA-seq experiments, based on cluster stability and statistical power for identifying clusters.

We conclude that deep generative models are promising for sample size determination as they learn important parts of the correlation structure from a small pilot dataset and subsequently generate synthetic data from varying numbers of cells for evaluation of cluster stability in the envisioned data analysis workflow. The corresponding improvement of experimental design could also advance the replicability of scRNA-seq experiments and might thus support translation to medical applications.

Funding

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 322977937/GRK2344

References

Amezquita, R. A., Lun, A. T., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., Martini, F., Rue-Albrecht, K., Risso, D., Soneson, C., *et al.* (2019). Orchestrating single-cell analysis with bioconductor. *Nature Methods*, pages 1–9.

Amodio, M., Van Dijk, D., Srinivasan, K., Chen, W. S., Mohsen, H., Moon, K. R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., Desai, A., Ravi, V., Kumar, P., Montgomery, R., Wolf, G., and Krishnaswamy, S. (2019). Exploring single-cell data with deep multitasking neural networks. *BioRxiv*, page 237065.

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F., and Newell, E. W. (2019). Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, **37**(1), 38.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, **112**(518), 859–877.

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, **36**(5), 411.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, **.**(2), 224–227.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. (2019). Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, **10**(1), 390.

Grün, D., Kester, L., and Van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. *Nature methods*, **11**(6), 637.

Hafemeister, C. (2019). How many cells? *Webtool*. Accessed: 2019-11-26.

Hashimshony, T., Senderovich, N., Avital, G., Klochendler, A., de Leeuw, Y., Anavy, L., Gennert, D., Li, S., Livak, K. J., Rozenblatt-Rosen, O., *et al.* (2016). Cel-seq2: sensitive highly-multiplexed single-cell rna-seq. *Genome biology*, **17**(1), 77.

Hennig, C., Meila, M., Murtagh, F., and Rocci, R. (2015). *Handbook of cluster analysis*. CRC Press.

Hilbe, J. M. (2011). *Negative binomial regression*. Cambridge University Press.

Hinton, G. and Salakhutdinov, R. (2012). An efficient learning procedure for deep boltzmann machines. *Neural Computation*, **24**(8), 1967–2006.

Jennrich, R. I. and Sampson, P. (1976). Newton-raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, **18**(1), 11–17.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lenz, S., Hess, M., and Binder, H. (2019). Unsupervised deep learning on biomedical data with boltzmannmachines. jl. *bioRxiv*.

Li, Y. and Zhu, X. (2018). Exponential family restricted boltzmann machines and annealed importance sampling. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature methods*, **15**(12), 1053.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, **15**(12), 550.

Lun, A. T. L., McCarthy, D. J., and Marioni, J. C. (2016). A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Res.*, **5**, 2122.

Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D. S., Krebs, C. F., and Bonn, S. (2020). Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nature Communications*, **11**(1), 1–12.

McCarthy, D. J., Campbell, K. R., Lun, A. T., and Wills, Q. F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics*, **33**(8), 1179–1186.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, **9**(1), 284.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Sagar, S., Hermann, J. S., Pospisilik, J. A., and Grün, D. (2018). High-throughput single-cell rna sequencing and data analaysis. *Methods in Molecular Biology*, **1766**, 257–283.

Salakhutdinov, R. and Hinton, G. (2009). Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*.

Svensson, V., da Veiga Beltrame, E., and Pachter, L. (2019). Quantifying the tradeoff between sequencing depth and cell number in single-cell rna-seq. *BioRxiv*, page 762773.

Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Welling, M., Rosen-Zvi, M., and Hinton, G. E. (2005). Exponential family harmoniums with an application to information retrieval. In *Advances in neural information processing systems*, pages 1481–1488.

Ye, P., Ye, W., Ye, C., Li, S., Ye, L., Ji, G., and Wu, X. (2019). schinter: imputing dropout events for single-cell rna-seq data with limited sample size. *Bioinformatics*.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell rna sequencing data. *Genome biology*, **18**(1), 174.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., *et al.* (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, **347**(6226), 1138–1142.

Zhang, M. J., Ntranos, V., and Tse, D. (2018). One read per cell per gene is optimal for single-cell rna-seq. *bioRxiv*.

Zhang, X., Xu, C., and Yosef, N. (2019). Simulating multiple faceted variability in single cell rna sequencing. *Nature communications*, **10**(1), 2611.

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., *et al.* (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications*, **8**, 14049.

Supplement

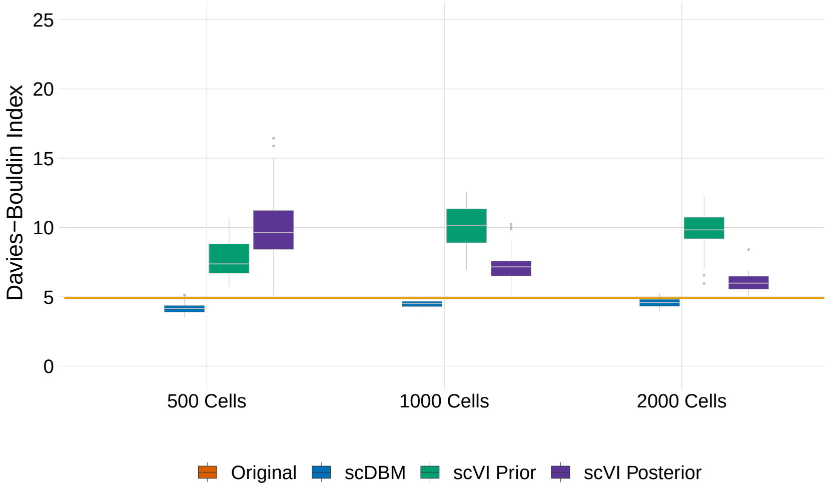

Figure S1: *Davies-Bouldin index (DBI), indicating the quality of synthetic data generated by scDBM and scVI (prior and posterior sampling) from pilot data of different sizes. The orange line indicates the reference DBI for the Seurat clustering on the original data (Zeisel) with 2816 cells.*

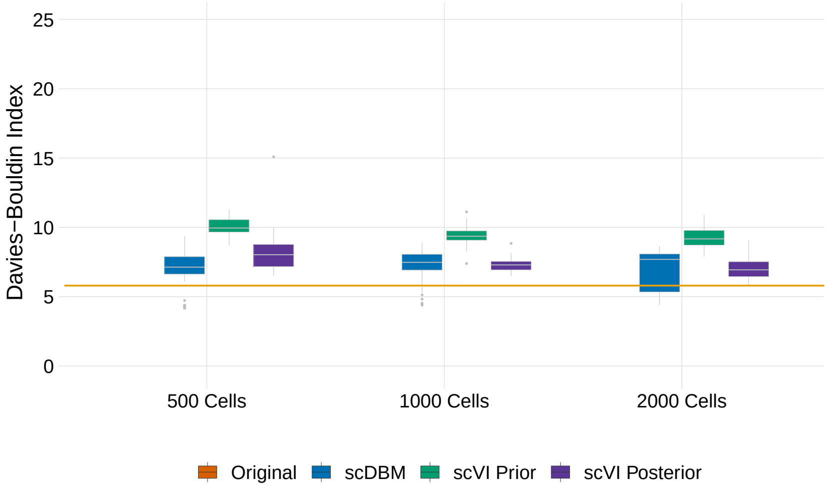

Figure S2: *Davies-Bouldin index (DBI), indicating the quality of synthetic data generated by scDBM and scVI (prior and posterior sampling) from pilot data of different sizes. The orange line indicates the reference DBI for the Seurat clustering on the original data (Hippocampus4k) with 3808 cells.*

Table 1: *Hyperparameters*

	PBMC4k		Zeisel		Hippocampus4k	
Model	scDBM	scVI	scDBM	scVI	scDBM	scVI
Learningrate	0.00005	0.001	0.00001	0.001	0.00001	0.001
Hidden layers	2	2	2	2	2	2
Epochs	270	100	200	100	2000	100