Probabilistic modelling of single-cell transcriptomics

Pedro Miguel Falé Ferreira
pedro.fale@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2018

Abstract

The gene expression profile of a cell dictates its function in molecular processes, and can be used to probe its health status. This represents a step forward in the deep characterization of diseases such as cancer and may lead to breakthroughs in their treatment. The technology used to measure the gene expression of isolated cells, single-cell RNA-seq (scRNA-seq), has emerged in the last decade as a key enabler of this progress. However, the use of existing methods for dimensionality reduction, clustering and differential expression is limited by the specificities of the data obtained from scRNA-seq experiments, where technical factors may confound analyses of the true biological signal and contribute to spurious results. To overcome this issue, a possible approach is designing probabilistic generative models of the data with hidden variables encoding different underlying processes. In this thesis we study the state-of-the-art probabilistic models of scRNA-seq and propose two novel methods which can be used for robust downstream analyses, mainly clustering of cell types. To ensure expressiveness and scalability to large data sets, we develop variational inference algorithms to approximate the posterior distributions of the hidden variables of both models. We show that the proposed methods are competitive with the state-of-the-art models for robust dimensionality reduction in modern data sets, and improve upon the current best Bayesian model for small numbers of cells. The results show that building probabilistic models of latent variables which encode domain knowledge and use variational inference is a promising approach to analysing scRNA-seq data at scale.
Keywords: scRNA-seq, probabilistic models, statistical inference, dimensionality reduction

1. Introduction

Single-cell RNA-sequencing (scRNA-seq) has emerged in the last decade as a key technology in using gene expression to study cell heterogeneity [1]. With the data obtained from these experiments, researchers can, for example, apply clustering algorithms to identify cell types and find genes which are differentially expressed between two conditions.

The expression of a gene in a cell is measured in scRNA-seq by counting the number of mRNA molecules generated from the gene's DNA sequence in a process called transcription. Formally, let N be the number of cells in a data set and P the number of genes. Then, the expression matrix $\mathbf{X}$ is of size $N \times P$, where each observation x_{np} contains the counts of mRNA molecules per gene p in cell n. The observations are assumed to be independent across both cells and genes.

Due to the large number of genes measured in a scRNA-seq experiment (never less than a few hundreds and occasionally up to tens of thousands, depending on the data acquisition protocol), the most common initial step in scRNA-seq data analysis is dimensionality reduction, i.e., reducing the observations from the original space $\mathbb{N}^P$ to a lower-dimensional one, obtaining the reduced data matrix $\mathbf{Z}$ of size $N \times K$, with $K < P$. This makes clustering analyses more tractable and, if $K = 2$ or $K = 3$, it allows for easy data visualization.

However, because mRNA molecules are counted within single cells, the total number of transcripts available may be very small and so they may not be detected at all. These undetected transcripts – often called "dropouts" – result in more zero counts than expected in the data matrix [2]. In practice, this means there are two types of zeroes in $\mathbf{X}$: some due to a true lack of expression and others due to dropouts. Although there are other confounding factors which deserve attention – capture efficiency and sequencing depth, batch effects and transcriptional noise –, this ambiguity makes dropouts the strongest source of noise in the data [3].

As the mentioned confounders hide the true biological variability, using traditional methods for dimensionality reduction, clustering or differential expression is not reliable. To overcome this, several probabilistic models for scRNA-seq data have been proposed along with corresponding inference mechanisms [4, 5, 6, 7]. This thesis considers the problem of performing scalable inference on useful scRNA-seq models to perform accurate downstream analyses, mainly clustering of cell types in big data sets.

We propose two novel probabilistic models for scRNA-seq data: modified probabilistic count matrix factorization (m-pCMF) and Bayesian zero-inflated

negative binomial factorization (ZINBayes). These build upon previous models in the literature while leveraging scalable Bayesian inference via variational methods. They are readily available online at [8] and [9]. We compare their performance with existing models on a set of benchmarking tasks. The code to reproduce all experiments is available at [10].

The remaining of this document is organized as follows. Section 2 formally introduces probabilistic modelling and inference. Section 3 provides an overview of the state-of-the-art probabilistic models for scRNA-seq data. Section 4 introduces the proposed models and Section 5 assesses their performance on real public data sets. In Section 6 conclusions are drawn.

2. Probabilistic modelling and variational inference

Formally, a latent variable model is defined by a joint probability distribution $p(\mathbf{x}, \mathbf{z})$ over hidden and observed variables: $\mathbf{z}$ and $\mathbf{x}$, respectively. This joint distribution describes how the observations and the hidden variables interact, and it can be factorized into the likelihood $p(\mathbf{x}|\mathbf{z})$ and the prior distribution $p(\mathbf{z})$:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}). \tag{1}$$

The joint distribution thus encodes the assumptions about the generative process from which the observations arise: a sample from the prior distribution over the hidden variables is drawn and used to sample an observation from the likelihood.

After observing data, the hidden quantities that describe that particular data set are uncovered via the posterior probability distribution, $p(\mathbf{z}|\mathbf{x})$. This is analytically obtained from Bayes' theorem:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}. \tag{2}$$

Here, the denominator is a normalizing constant that guarantees that $p(\mathbf{z}|\mathbf{x})$ integrates to 1. Hence, it is given by

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}. \tag{3}$$

This normalization constant (with respect to the latent variables) is often called the "evidence" of the model and can be used for model selection. When defining latent variable models, the choice of prior distributions for the parameters of the likelihood function has a strong effect on whether the posterior distribution can be obtained analytically due to the integral in Eq. (3). Often, for complex models, $p(\mathbf{x})$ is either not available in closed form or computationally intractable.

Although the posterior distribution may be intractable for exact inference, a wide variety of approximate inference algorithms are available. In this work we focus on variational inference (VI) [11]. VI is a deterministic method for approximating intractable posterior distributions via optimization. It posits an approximating distribution $q(\mathbf{z}; \nu)$ over the latent variables $\mathbf{z}$ with parameters ν. These parameters are optimized as to minimize the distance between $q(\mathbf{z}; \nu)$ and the true posterior $p(\mathbf{z}|\mathbf{x})$. In VI, this distance is measured via the Kullback-Leibler (KL) divergence. Thus, VI aims at solving the optimization problem

$$\nu^* = \underset{\nu}{\operatorname{argmin}} \operatorname{KL}(q(\mathbf{z}; \nu) \parallel p(\mathbf{z}|\mathbf{x})), \tag{4}$$

where ν^* are the variational parameters that yield the best approximation to the posterior.

The objective in Eq. (4) is not readily available because it depends on the posterior distribution which we aim at approximating. However, we can re-write the KL divergence in terms of a lower bound of $p(\mathbf{x})$ which we call the Evidence Lower BOund (ELBO). Minimizing the KL divergence is now achieved by maximizing the ELBO:

$$\begin{aligned} \nu^* &= \underset{\nu}{\operatorname{argmax}} \, \mathbb{E}_q \left[\log p(\mathbf{x}, \mathbf{z})\right] - \mathbb{E}_q \left[\log q(\mathbf{z}; \nu)\right] \\ &= \underset{\nu}{\operatorname{argmax}} \, \operatorname{ELBO}(\nu) \end{aligned} \tag{5}$$

In VI, the most widely used form for the variational distribution is the mean-field family, where the latent variables are mutually independent and each one is governed by a distinct factor in the variational density [12], meaning $q(\mathbf{z})$ factorizes over all the M latent variables $\mathbf{z} = (z_1, ..., z_M)$:

$$q(\mathbf{z}; \nu) = \prod_{m=1}^{M} q(z_m; \nu_m), \tag{6}$$

where each $q(z_m; \nu_m)$ may assume a different parametric form, according to what is more suitable for each latent variable.

The most commonly used algorithm to find the ν that correspond to a (local) maximum of the ELBO is coordinate ascent, which we refer in the following sections as CAVI (Coordinate Ascent Variational Inference). CAVI algorithms can be easily derived for conditionally conjugate models. Additionally, for models with global latent variables, stochastic variational inference (SVI) can be employed by iteratively subsampling a mini-batch of data, updating the corresponding local variational parameters, and using only the mini-batch to update the global variational parameters, instead of using the whole data set. This provides scalability to massive data sets. More recently, ELBO optimization has been generalized into the wider class of non-conditionally conjugate models, effectively allowing the design of more expressive models. This is done, for example, using reparameterization gradients to estimate the gradient of the ELBO with respect to the variational parameters and applying a gradient method to update them [12].

3. Probabilistic models for scRNA-seq data

A number of probabilistic models which provide a centralized analysis of scRNA-seq by simultaneously normalizing and reducing the dimensionality of the data

have been proposed [4, 5, 6, 7]. These models aim at explicitly accounting for the technical factors of variation in order to separate them from the underlying cell biology, thus providing an unbiased analysis of the latter. Specifically, they consist of factorized probabilistic models of the data, in which each observation is assumed to have been generated by some probability distribution parameterized by some combination of unknown factors. These unknown factors encode, for example, library size variations, experimental batches, gene length, and some lower-dimensionality representation of the cells.

3.1. Zero-inflated factor analysis

Zero-inflated factor analysis (ZIFA) [4] was the first probabilistic dimensionality reduction method designed specifically for scRNA-seq data. It consists of a Factor Analysis (FA) model with an additional latent variable accounting for dropouts. The observations $\mathbf{X}$ are the logarithm of counts (plus one to prevent taking the logarithm of zero), which allows them to be well approximated by a Gaussian distribution with zero-inflation.

While ZIFA accounts for the typical zero-inflation observed in scRNA-seq data, it does so indirectly, via the log-counts, instead of modelling the raw counts. Additionally, while dropouts are important, they are not the only confounding factor present in scRNA-seq. ZIFA is thus limited in its power to separate the biological signal from the technical noise. Furthermore, the parametric form for the dropout probability assumed by ZIFA does not always provide a good fit [6]. In terms of computational complexity, the inference mechanism developed by the authors requires the full batch of data to run, which makes it unsuited to massive data sets which may not fit in memory.

3.2. Probabilistic count matrix factorization

Probabilistic count matrix factorization (pCMF) consists of a Bayesian matrix factorization method for count data. It extends the Gamma-Poisson (GaP) factor model introduced in [13] by considering dropouts and a sparse mapping from the lower-dimensional space to the observation space. The authors jointly infer the latent variables and estimate the hyperparameters of the model in a variational Expectation-Maximization scheme. By considering Gamma priors on the latent representations and the projection matrices, pCMF models the overdispersion of the count data. However, the inference scheme proposed by the authors requires the full batch of data, which means inference of pCMF is not ready for massive scRNA-seq data sets.

3.3. Single-cell variational inference

Single-cell variational inference (scVI) is a non-linear model for scRNA-seq which accounts for various factors of variation. It defines a latent variable model of the raw counts parameterized by neural networks in such a way that the marginal distribution of the observations is a ZINB. If available, the model can also include the batch annotation of each cell in order to subtract batch effects from the biological signal.

Additionally, scVI utilizes neural networks to specify non-linear transformations between latent variables. Notably, it associates the cells' latent representations with the probability of dropout occurrence by a neural network. Another neural network is used to map from the lower-dimensional space to the original-dimensional space. It contains latent variables encoding a per-cell normalized gene expression vector which can be used for differential expression. Inference of scVI is scalable with the number of cells, as it can be done by passing small batches of data at each iteration of the optimization procedure. The main disadvantage of scVI is that, while the use of neural networks allows for great model expressiveness, the typical large number of parameters to fit may render them inadequate for small sample sizes.

3.4. Zero-inflated negative binomial-based wanted variation extraction

Zero-inflated negative binomial-based wanted variation extraction (ZINB-WaVE) [6] considers all the latent factors of variation as fixed quantities (unlike ZIFA, pCMF and scVI, which considered them to be random variables). It models the observations – in this case, the raw counts – with a ZINB distribution with parameters given by linear regression of latent factors, which may be known or not. ZINB-WaVE leverages the fact that scRNA-seq data is well described by a ZINB distribution. Furthermore, it is expressive, accounting for a large number of factors which may confound the analysis of scRNA-seq such as library size variations, batches and amplification bias. It directly models the raw counts and the sample-level intercepts serve as normalization factors. However, it has two main drawbacks: it assumes linearity, which may prevent it from capturing complex, non-linear relationships between the assumed lower-dimensional representations and the observations. Plus, as in ZIFA and pCMF, its estimation procedure requires the full batch of data, thus not scaling to modern data sets generated by droplet-based experimental protocols.

4. Proposed methods

4.1. m-pCMF

Our first proposal is a modified version of pCMF. We modify it in the following ways:

- Introduce cell-specific scaling factors, similarly to ZINB-WaVE and scVI. These are meant to account for capture efficiency and sequencing depth variations.

- Introduce batch index annotations, similarly to ZINB-WaVE and scVI. This allows for the correction of batch effects if batch annotations are available.

- Remove the sparsity-inducing prior on the loadings matrix. Instead, we introduce sparse loadings via a sparse Gamma prior on them.

Specifically, the model is defined via the following generative process, where we consider gene expression measurements to have been obtained from B different experimental batches:

1. For $p = 1, ..., P$ and $j = 1, ..., K + B$:
   (a) Sample a factor loading $w_{pj} \sim \text{Gamma}\,(\beta_1, \beta_2)$.
2. For $n = 1, ..., N$:
   (a) Sample a scaling factor $l_n \sim \text{Gamma}\,(\nu_1, \nu_2)$
3. For $n = 1, ..., N$ and $k = 1, ..., K$:
   (a) Sample a latent factor $z_{nk} \sim \text{Gamma}\,(\alpha_1, \alpha_2)$.
4. For $n = 1, ..., N$ and $p = 1, ..., P$:
   (a) Sample a raw count $y_{np} \sim \text{Poisson}\left(l_n \sum_j w_{pj} \left[\mathbf{z}_n, \mathbf{s}_n\right]_j\right)$.
   (b) Sample non-dropout event $d_{np} \sim \text{Bernoulli}\left(\pi_p^d\right)$.
   (c) If $d_{np} = 0$, $x_{np} = 0$. Else, $x_{np} = y_{np}$.

The latent variables are $\mathbf{z}$, $\mathbf{w}$, $\mathbf{d}$ and $\mathbf{l}$. The $\mathbf{z}$ represents the cells in a lower-dimensional space of size $K < P$, $\mathbf{w}$ is the map from $\mathbf{z}$ to the observation space, $\mathbf{d}$ models the occurrence of dropout in each observation and $\mathbf{l}$ is a cell-specific scaling factor which accounts for capture efficiency and sequencing depth variations.

The batch annotations are included in the model by concatenating them as one-hot encoded vectors $\mathbf{s}_n$ (a B-dimensional vector where entry $s_{nb} = 1$ if cell n comes from batch b, and is zero otherwise) to the latent space representations $\mathbf{z}_n$ and multiplying the resulting vector by the mapping matrix $\mathbf{w}$: $\sum_{k=1}^{K} w_{pk} z_{nk} + \sum_{b=1}^{B} w_{p,K+b} s_{nb}$. We represent this product as $\left[\mathbf{z}_n, \mathbf{s}_n\right] \mathbf{w}_p^T$. Introducing this additional flexibility in the model results in representations which are disentangled from the experimental batch, thus correcting for batch effects.

Regarding the parameters of the prior distributions – i.e., the model hyperparameters – we choose them so as to warp the space of solutions to the inference problem in a way that favours the structure we believe to describe the data. Namely, in general terms, we aim at obtaining lower-dimensional representations $\mathbf{z}_n$ in which different cell types are clustered together and different ones are well separated. The Normal distribution is usually the distribution of choice for tasks like these, but because we need the gamma distribution to ensure positiveness, we choose α_1, α_2 to obtain a somewhat similar shape as the normal distribution. In our experiments, we fix $\alpha = 16$ and $\alpha = 4$.

Regarding the factor loadings, we set the shape parameter to be less than 1. When the shape is less than 1, gamma distributions put most of their mass near zero; [14] call this type of distribution sparse gamma and note that it is akin to a soft spike-and-slab prior, which is often used for unsupervised feature selection. This is useful for m-pCMF because it encodes the fact that the latent representations, which should encode relevant biological variability, are only related to a small number of genes. In practice, we fix $\beta_1 = 0.1$ and $\beta_2 = 0.3$.

The prior probabilities π_p^d are chosen as in [5], where each π_p^d is set to the proportion of non-zeros present in gene p across all cells.

Finally, the prior over the scaling factors has a significant impact on the interpretation of m-pCMF. We consider two situations. First, similarly to scVI, we set the scalings to explicitly account for capture efficiency and sequencing depth variations, by making l_n follow the cells' library sizes. This is done by choosing ν_1 and ν_2 such that the mean and variance of l_n corresponds to the mean l_μ and variance l_{σ^2} of the library sizes of all cells, respectively. Specifically, this amounts to setting

$$\nu_1 = \frac{l_\mu^2}{l_{\sigma^2}}, \quad \nu_2 = \frac{l_\mu}{l_{\sigma^2}}. \tag{7}$$

The second situation leverages two (correlated) ideas:

- scRNA-seq data is well described by a zero-inflated negative binomial distribution due to overdispersion;
- The overdispersed counts are largely due to count depth variation.

Specifically, the marginal likelihood of m-pCMF given $\mathbf{z}, \mathbf{w}$ and $\mathbf{d}$ (integrating $\mathbf{l}$ out) is a zero-inflated negative binomial if each l_n is given by a gamma distribution with equal prior shape and rate. Formally (and considering only the NB term and one experimental batch for simplicity), this means that if

$$\begin{aligned} &x_{np} | l_n, \mathbf{z_n}, \mathbf{w_p} \sim \text{Poisson}(l_n \mathbf{z_n} \mathbf{w_p}^T), \\ &l_n \sim \text{Gamma}\,(\nu, \nu)\,, \end{aligned} \tag{8}$$

then $x_{np} | l_n, \mathbf{z_n}, \mathbf{w_p}$ is NB with mean $\mathbf{z_n} \mathbf{w_p}^T$ and dispersion ν. In this construction, l_n has mean of 1 and variance given by $1/\nu$, and lives in the positive reals. It can be interpreted as a cell size factor. Under this interpretation, we expect variations in l_n to correlate with variations in library sizes. We encode this expectation in the hyperparameter ν by setting $\nu = 1/l_{\sigma^2}$, which means that $\text{Var}\left[l_n\right] = l_{\sigma^2}$.

We have thus defined two ways of encoding library size variations into the model. They both scale the expected gene expression by some expected size factor

but, unlike the first one, the second construction ensures that the distribution $p(x_{np}|\mathbf{z}_n, \mathbf{w}_p, d_{np})$ is ZINB.

Because the cell-specific scalings are gamma-distributed, the model is, like pCMF, conditionally conjugate and we can derive a CAVI algorithm to update the variational parameters. The batch annotations do not introduce new variables, instead they just increase the dimensions of the loadings matrix. The CAVI algorithm for m-pCMF is thus an extension of the one derived in [5], where we also make use of auxiliary variables $u_{npj} \sim \text{Poisson}(l_n c_{nj} w_{pj})$. The variational approximation is mean-field and the variational factors are

$$\begin{aligned} q(z_{nk}; \mathbf{a}_{nk}) &= \text{Gamma}\left(z_{nk}|a_{1,nk}, a_{2,nk}\right), \\ q(w_{pj}; \mathbf{b}_{pj}) &= \text{Gamma}\left(w_{pj}|b_{1,pj}, b_{2,pj}\right), \\ q(l_n; \mathbf{v}_n) &= \text{Gamma}\left(l_n|v_{1,n}, v_{2,n}\right), \\ q(d_{np}; p^d_{np}) &= \text{Bernoulli}\left(d_{np}|p^d_{np}\right), \\ q(\mathbf{u}_{np}; \mathbf{r}_{np}) &= \text{Multinomial}\left(\mathbf{u}_{np}|x_{np}, \mathbf{r}_{np}\right). \end{aligned} \tag{9}$$

The variational parameters are updated as

$$\begin{aligned} a_{1,nk} &= \alpha_1 + \sum_{p=1}^{P} p^d_{np}\, x_{np}\, r_{npk}, \\ a_{2,nk} &= \alpha_2 + \frac{v_{1,n}}{v_{2,n}} \sum_{p=1}^{P} p^d_{np} \frac{b_{1,pk}}{b_{2,pk}}, \\ b_{1,pj} &= \beta_1 + \sum_{n=1}^{N} p^d_{np}\, x_{np}\, r_{npk}, \\ b_{2,pj} &= \beta_2 + \sum_{n=1}^{N} p^d_{np} \frac{v_{1,n}}{v_{2,n}} \mathbb{E}_q\left[c_{nj}\right], \\ v_{1,n} &= \nu_1 + \sum_{p=1}^{P} p^d_{np}\, x_{np}, \\ v_{2,n} &= \nu_2 + \sum_{p=1}^{P} p^d_{np} \sum_{j=1}^{J} \frac{b_{1,pj}}{b_{2,pj}} \mathbb{E}_q\left[c_{nj}\right], \\ r_{npj} &\propto \left[\exp\left(\Psi(a_{1,n}) - \log a_{2,n}\right), s_n\right]_j \\ &\quad \times \exp\left(\Psi(b_{1,pj}) - \log b_{2,pj}\right), \\ \text{logit}(p^d_{np}) &= \text{logit}(\pi^d_p) - \frac{v_{1,n}}{v_{1,n}} \sum_{j=1}^{J} \mathbb{E}_q\left[c_{nj}\right] \frac{b_{1,pj}}{b_{2,pj}}, \end{aligned} \tag{10}$$

where $\mathbb{E}_q\left[c_{nj}\right] = \frac{a_{1,nj}}{a_{2,nj}}$ if $j \leq K$ and $\mathbb{E}_q\left[c_{nj}\right] = s_{n,j-K}$ if $j > K$.

Additionally, in order to scale m-pCMF up to large data sets, we turn the CAVI algorithm into an SVI algorithm, following the general procedure described in [15]. At each iteration we: (1) sample a minibatch of M samples from the data set, (2) update the local parameters for each of the samples in the minibatch and (3) update the global variational parameters $\mathbf{b}$ using only the local variational parameters for the sampled minibatch. The update of the global parameters starts by computing intermediate global parameters as if each $\mathbf{x}_m$ was observed N times, and scaling by the mini-batch size M. Then, we update $\mathbf{b}$ at iteration t by taking a gradient step.

4.2. ZINBayes

We develop a more complex model using the fact that, if

$$\begin{aligned} X|\lambda &\sim \text{Poisson}(\lambda), \\ \lambda &\sim \text{Gamma}\left(r, \frac{r}{\mu}\right), \end{aligned} \tag{11}$$

then X is NB with mean μ and dispersion r. We leverage this construction to account for the fact that scRNA-seq data is well described by a ZINB distribution. This model is similar to scVI but it dismisses the use of neural networks and instead performs Bayesian inference over linear mappings from the lower-dimensional space to the count space. This allows for the mappings to be sparse, which can help interpreting what genes are more relevant to what latent component, and to what cell type, consequently.

The following generative process defines the model:

1. For $p = 1, ..., P$:
   (a) Sample a dispersion parameter $r_p \sim \text{Gamma}\left(\theta_1, \theta_2\right)$.
2. For $p = 1, ..., P$ and $j = 1, ..., K + B$:
   (a) Sample a negative-binomial factor loading $w_{0,pj} \sim \text{Gamma}\left(\beta_1, \beta_2\right)$.
   (b) Sample a dropout factor loading $w_{1,pj} \sim \text{Normal}\left(0, 1\right)$.
3. For $n = 1, ..., N$:
   (a) Sample a scaling factor $l_n \sim \text{LogNormal}\left(l_\mu, l_{\sigma^2}\right)$
4. For $n = 1, ..., N$ and $k = 1, ..., K$:
   (a) Sample a latent factor $z_{nk} \sim \text{Gamma}\left(\alpha_1, \alpha_2\right)$.
5. Concatenate latent factors $\mathbf{z}_n$ with one-hot encoded batch annotation s_n: $c_{nj} = \left[\mathbf{z}_n, s_n\right]_j$.
6. For $n = 1, ..., N$ and $p = 1, ..., P$:
   (a) Compute a normalized mean gene expression: $\rho_{np} = \frac{\mathbf{c}_n \mathbf{w}_{0,p}}{\sum_p \mathbf{c}_n \mathbf{w}_{0,p}}$.
   (b) Sample a mean gene expression $\lambda_{np} \sim \text{Gamma}\left(r_p, \frac{r_p}{\rho_{np} l_n}\right)$
   (c) Sample a raw count $y_{np} \sim \text{Poisson}\left(\lambda_{np}\right)$.

(d) Sample dropout event $d_{np} \sim \text{Bernoulli}\left(\mathbf{c}_n \mathbf{w}_{1,p}\right)$ (logit parameterization).

(e) If $d_{np} = 1$, $x_{np} = 0$. Else, $x_{np} = y_{np}$.

There are three main differences between this model and m-pCMF:

1. The negative binomial's dispersion parameter is independent from the cell-specific scalings, which encode cell size factors. This allows the model to capture other gene-specific dispersion sources besides the count depth variation between cells;

2. The dropout probabilities are related to the latent representations $\mathbf{z}$, instead of being entirely separated from the NB part of the generative process;

3. We define the expected relative gene expression frequencies for each cell in ρ, which can, in principle, be used for non-biased differential expression. Its scaled version, $\rho \mathbf{l}$ gives the NB mean.

We choose the hyperparameters $\alpha_{1,2}$ and $\beta_{1,2}$ the same way as described for the corresponding hyperparameters in m-pCMF in Section 4.1, in order to facilitate clustering in $\mathbf{z}$ and sparse loadings $\mathbf{w}_0$. The prior over the gene-specific dispersion parameters $\mathbf{r}$ is chosen to be a generic gamma distribution, as we have no prior expectations of what they should be. However, we do not expect them to be very large (e.g. $r_p \gg 10$), so we choose $\theta_1 = 2, \theta_2 = 1$.

This model is not conditionally conjugate, due to dropout structure and the parameterization of λ, so we can not derive a CAVI algorithm for the variational updates based on the assumptions described in [15]. Instead, we use generic methods for inference of all the latent variables. Namely, we use reparameterization gradients-based variational inference to fit the model to data using the Edward probabilistic programming language [16].

5. Results

In this section we apply the proposed models to real data sets. We evaluate the models in a series of settings: held-out data log-likelihood, dropout imputation, cluster separability in the latent space, separation of biological from technical signals and batch effect correction. As a baseline for a scRNA-seq data-agnostic model, we consider a Factor Analysis (FA).

For FA, we use the scikit-learn Python package [17]. For ZIFA, we use the Python implementation provided by the authors and the scikit-learn wrapper provided by [7]. For ZINB-WaVE, we use the R package provided by the authors with the additional code provided in [7] to compute the held-out data log-likelihood. For scVI, we use the Python implementation provided by the authors. For pCMF we use the R package provided by the authors and the scikit-learn wrapper we developed.

In this study we consider data sets with different characteristics in terms of size and protocol used. We are particularly interested in data sets which have been used in previous studies, and whose cells have been previously and reliably annotated. Table 1 summarizes the data sets we consider in this section.

Table 1: Brief description of the data sets considered in this report. The "UMI" column indicates the use of Unique Molecular Identifiers for the experimental expression quantification. The "Droplet" column indicates whether the data set was generated by a droplet-based experimental protocol. The LARGE data set does not contain annotations.

ID	N cells	P genes	# groups	UMI	Droplet
POLLEN [18]	130	6,982	11	No	No
ZEISEL [19]	3,005	558	7	Yes	No
LARGE [20]	1,3 M	10,000	–	Yes	Yes

To facilitate comparison with other methods, we utilize the data sets with the same pre-processing used in previous studies. Namely, for POLLEN, we use the data available from the *scRNAseq* R package [21] and retain only the 1000 genes with highest standard deviation as per [6]. For ZEISEL, we retain only the 558 genes with highest standard deviation, as in [7]. For LARGE, we use the count data organized by [7].

5.1. NB improves m-pCMF's fit

In Section 4.1 we highlighted the fact that setting the hyperparameters of the cell-specific scalings' Gamma prior as $\nu_1 = \nu_2$ yielded a negative-binomial marginal likelihood. In fact, this yields a increase in model fitness, as Fig. 1 illustrates. Specifically, we subsampled 5,000 cells and considered only the 500 most variable genes from the LARGE data set and ran CAVI on m-pCMF five times independently with $\nu_1 = \nu_2 = \nu$, with ν equal to the variance of the library sizes, and another five times with ν_1 and ν_2 chosen to achieve a prior mean equal to the mean library size and a prior variance equal to the library size variance, as detailed in Section 4.1.

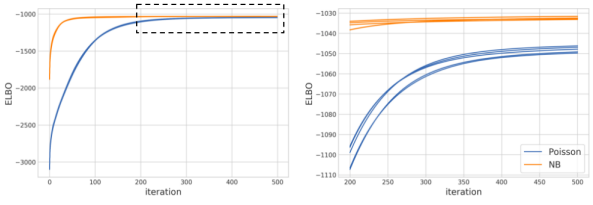


Figure 1: ELBO per iteration of m-pCMF without (Poisson) and with NB structure for 5 runs of 500 iterations. On the left we show the full convergence, and on the right we detail the last 300 iterations.

Because the CAVI algorithm starts with a random initialization of the variational parameters and the ELBO is not convex, each run achieves a different local min-

imum (this is highlighted in the zoom of the last 300 iterations in the figure). However, the behaviour for the two different m-pCMF settings is consistently different across runs: the NB construction always achieves better values of the objective function than the less expressive Poisson construction. As such, in the subsequent experiments with m-pCMF we always use $\nu_1 = \nu_2 = \nu$, with ν equal to the variance of the library sizes.

5.2. Effect of the number of Monte Carlo samples for gradient estimation in ZINBayes

Because ZINBayes' inference is based on (unbiased) estimates of gradients of the objective function, and these estimates are computed via Monte Carlo (MC) sampling, the choice of the number of MC samples has an effect on the variance of these gradient estimates, which in turn impacts the convergence of the inference algorithm. Using the same data as in Section 5.1, we briefly illustrate this impact in Fig. 2.

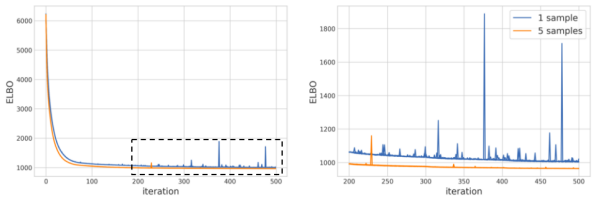


Figure 2: ELBO per iteration of ZINBayes with 1 and 5 Monte Carlo samples for gradient estimation. On the left we show the full convergence, and on the right we detail the last 300 iterations.

In this case, increasing the number of MC samples from 1 to 5 led not only to a better value of the objective function in the fixed number of iterations, but also did so with a considerably lower variance. However, although not shown here, we note that increasing the number of MC samples slows down the algorithm, so there is a trade-off between speed and variance. In practice, we always use 5 MC samples in subsequent experiments.

5.3. m-pCMF and ZINBayes disentangle technical factors of variation

Because m-pCMF and ZINBayes account for differences in cell capture efficiency, which lead to differences in library sizes, they can separate these factors of variation from the underlying biological signal. In both models the prior distribution of these scalings are related to the observed library sizes and are thus expected to correlate with them. In Fig. 3 we plot the log of the inferred scalings against the observed log library sizes of the ZEISEL data set. Clearly, the expected correlation is found in these two data sets. Notice that ZINBayes' scaling factors have the same scale as the observed library sizes, which is expected by their definition. On the other hand, m-pCMF's scalings, due to the NB construction, vary closer to 1, due to the "exposure" variable interpretation presented in Section 4.1.

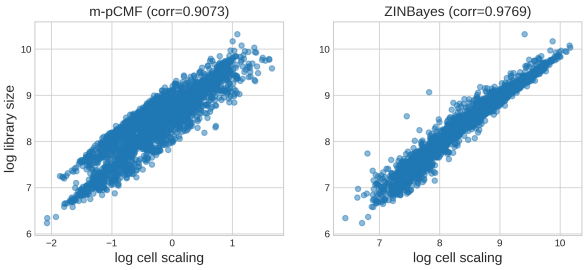
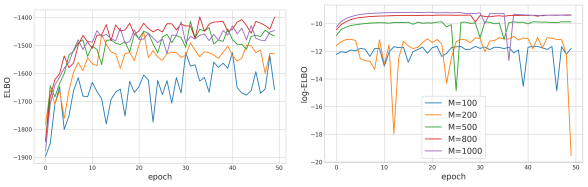


Figure 3: Scatter plots of the log of the estimated cell scaling factors by ZINBayes, m-pCMF and scVI against cell's log library sizes of the ZEISEL data set. For each scatter plot we compute the corresponding Pearson correlation coefficient. ZINBayes's scalings correlate very strongly with library sizes.

5.4. m-pcmf and ZINBayes are ready for large-scale experiments

Finally, we consider the same LARGE subsample considered in Section 5.1 to note the scalability of m-pCMF and ZINBayes to large data sets. Because these models are ammenable to stochastic variational inference, they can, in principle, be applied to extremely large scRNA-seq data sets with massive numbers of cells. Because these data sets may not fit into memory of the practicioner's machine, being able to fit an scRNA-seq data model by feeding it mini-batches of data instead of the whole batch of data at once is a very valuable feature. While we do not apply our methods to such data sets and always consider, in the subsequent experiments, batch algorithms, we illustrate the capability of m-pCMF and ZINBayes to do so.

Specifically, in Fig. 4 we consider different mini-batch sizes and confirm that, for both models, increasing the mini-batch size leads to higher values of the ELBO, i.e., better model fitness. Importantly, we notice that



(a) m-pCMF (b) ZINBayes

Figure 4: Convergence curves for m-pCMF and ZINBayes with mini-batch stochastic variational inference mechanisms for varying mini-batch sizes, M. Each "epoch" consists of a full pass of the data set.

while there is a significant difference in the achieved ELBO values from $M = 100$ to $M = 500$ in the 5,000 cells data set, both for m-pCMF and ZINBayes, the gain decreases significantly as the mini-batch sizes increase from there to $M = 1000$. This means that we may be able to achieve sufficiently good performances at low

memory requirements.

5.5. Held-out data log-likelihood

We compare the goodness-of-fit and generalization ability of each model by evaluating the per-cell marginal log-likelihood they assign to data unseen during inference. To do this, we perform inference on a large subsample of the data set and then compute the log-likelihood in the rest in a 5-fold cross-validation procedure. We do not consider pCMF as the implementation provided by the authors does not allow for a straightforward way of evaluating the log-likelihood on held-out data.

FWe draw a random subsample of 500 cells from the LARGE data set (containing only the 700 most variable genes) and perform 5-fold cross-validation. The results are shown in Fig. 5. m-pCMF performs worse than all methods except scVI, which yields the worst fit. ZINBayes, while competitive, is unable to beat ZINB-WaVE, ZIFA and FA. Focusing on the test data log-likelihood, scVI is significantly worse than competing methods, despite having been run for a large number of epochs (1000), as recommended by the authors in the case of small data sets. This is related to the fact that there are more local latent variables than global variables in the model, making the use of mini-batch optimization and amortized inference inefficient.

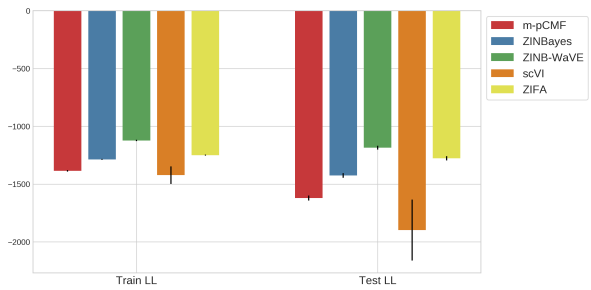


Figure 5: Mean train and test data log-likelihoods of each model with error bars over 5 cross-validation folds in 500 cells from the LARGE data set.

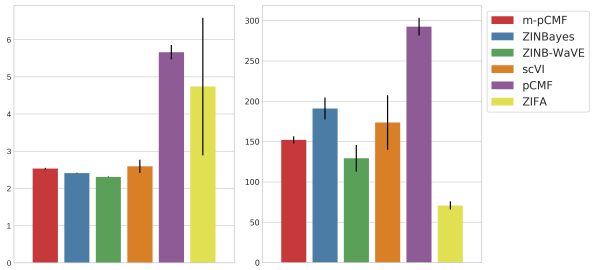
Additionally, we apply the considered scRNA-seq models to different-sized subsets of the LARGE data set and show the log-likelihoods for a held-out test set of 10K cells in Table 2. Here we do not consider a 5-fold CV scheme as ZINB-WaVE and ZIFA become very slow as the number of cells increases. Nevertheless, the obtained values are illustrative of the behaviour of the models with varying number of cells. Namely, scVI has the worst performance for 1K cells and the best performance among all methods for 15K cells. m-pCMF's performance appears to decrease with the number of cells and ZINBayes' tendency is to improve.

5.6. Imputation

As noted in [7], models which attribute a large likelihood to data sets which are dominated by zeros are not necessarily useful for our purpose, as we assume that some percentage of the observed zeros are due to nuisance factors of variation – this was not the case for the data considered in the previous section. As such, to assess model fitness, the authors generated a corrupted training sets by setting 10% uniformly chosen non-zero entries to zero. Then, they fit the perturbed dataset with each of the benchmark methods and evaluate them by comparing the inferred mean values to the original ones, via the median absolute error. We apply this evaluation procedure to the ZEISEL and POLLEN data sets and show the results in Fig. 6.

Table 2: Log-likelihood attributed by each model to a held-out set of 10K cells after performing inference on different random subsets of different sizes of the LARGE data set, with 720 genes.

	1K	5K	10K	15K
m-pCMF	-1098.78	-1224.10	-1205.44	-1462.68
ZINBayes	-1291.83	-1268.26	-1251.48	-1259.26
ZINB-WaVE	-1190.98	-1173.66	-1178.52	-1177.71
scVI	-1509.90	-1302.81	-1186.75	-1193.54
ZIFA	-1276.22	-1260.26	-1253.98	-1272.46
FA	-1211.61	-1199.72	-1193.32	-1196.03



(a) ZEISEL data set (b) POLLEN data set

Figure 6: Median imputation error of each model for 5 corruptions of the the ZEISEL and POLLEN data sets.

The results show that pCMF is always the worst performing method. m-pCMF achieves a lower error than scVI on the POLLEN data set, thus corroborating that scVI does not fare well on small data sets. ZINBayes is competitive for the ZEISEL data set, but has a large variance across runs for the POLLEN data.

5.7. Latent space structure

Finally, we compare the quality of the structure inferred by each model in the corresponding lower-dimensional space. Similarly to [7], we consider three metrics for assessing the clustering of cells in the latent space according to the existing biological annotations: Average Silhouette Width (ASW) [22], Adjusted Rand Index (ARI) [23] and Normalized Mutual Information (NMI) [24]. For the ARI and NMI metrics, a clustering is required. In this case, we perform K-means clustering on the latent space with K equal to the true num-

ber of clusters. The results for a 5-fold cross-validation for each method in ZEISEL and POLLEN are shown in Fig. 7. For the POLLEN data set, we use the batch annotations to assess whether each model is capable of providing cell-type separability which is independent of the experimental batch of each cell, i.e., to yield batch-independent lower-dimensional projections. We do this by evaluating the ASW metric with respect to the batch annotations – in this case, the lower, the better.

Finally, we consider the POLLEN data set to assess the correlation of the learned lower-dimensional projections with technical factors. To this end, we infer 2-dimensional projections of the data and evaluate the correlation of each dimension of each model's latent space with the library size and detection rates of the cells. The results are shown in Fig. 8. We first note that the ASW of ZINBayes's latent space does not have a signficant change from 10D (Fig. 7(b)) to 2D. Furthermore, ZINBayes is the method that achieves better clustering of different cell types while maintaining a very low correlation with library sizes and a median correlation with detection rates. Notably, FA, ZIFA and pCMF's latent factors are not only unable to cluster cell types, but the latent structure is also extremely correlated with library sizes.

5.8. Discussion

In terms of model fitness, the results show that ZINBayes outperforms scVI for the 500, 1K and 5K cell-subsamples of the LARGE data set, the ZEISEL data set (both in terms of marginal log-likelihood and imputation). m-pCMF outperforms scVI in the 500, 1K and 5k cell-subsamples of the LARGE data set and the POLLEN data set (imputation). All models, including ours, yield a better fit than pCMF. ZINB-WaVE is always the best-fitting model, except for the POLLEN data, for which ZIFA yields the lowest imputation error.

The fact that ZINB-WaVE is based on a similar parameterization to scVI withouth the non-linearities but still provides a better overall fit suggests that either the non-linearities are not necessary, its fitting mechanism is able to achieve better solutions of the parameters, or scVI's prior hyperparameters are inadequate. In this sense, the fact that ZINBayes also dismisses the use non-linearities points towards either the second or third hypothesis. To further investigate this, we should analyse the estimated values of ZINB-WaVE's parameters to see if they can not be captured by the prior distributions in ZINBayes.

In terms of cell type separability in the reduced spaces, m-pCMF and ZINBayes yield higher ASW scores and moderately lower ARI and NMI scores than all competing methods, except pCMF, in the ZEISEL data set. For the POLLEN data set, ZINBayes provided a better biological structure than all the other models according to every metric. This suggests that ZINBayes is a better suited model for smaller data sets than competing methods. Typically, smaller data sets exhibit a much larger gene-specific dispersion than droplet-based cohorts, and ZINBayes' prior on the over-dispersion is generic enough to allow for a wide range of values for this parameter while still constraining the parameter space, which facilitates optimization. This characteristic may also explain why m-pCMF performs very poorly on the POLLEN data set, being unable to find any biological structure, as it is unable to model gene-specific dispersion.

In terms of correction of batch effects, we observed for the POLLEN data set that only ZINBayes' and ZINB-WaVE's latent structures were uncounfounded by the cells' experimental batches. While for ZINBayes we explicitly included the batch annotations in the model, ZINB-WaVE was able to achieve this good performance without any batch information. We showed that these were the two models whose latent components were less correlated with library size variation, which was the main counfounding factor between experimental batches in the POLLEN data set.

6. Conclusions

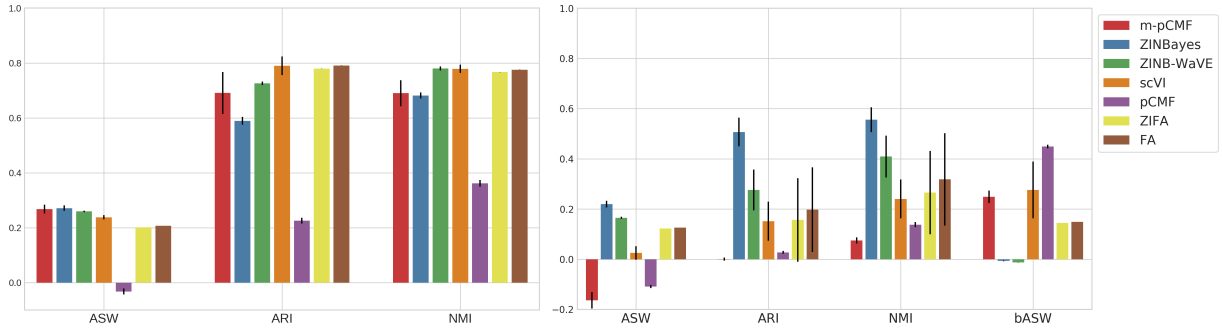
Two novel generative models for dimensionality reduction of scRNA-seq data were proposed. ZINBayes shares some structure with current state-of-the-art models, which may explain the overall better performance compared to m-pCMF, which, for example, does not consider gene-specific over-dispersion. While none of the proposed models provides an overall improvement in performance over the state-of-the-art methods ZINB-WaVE and scVI, we note that:

1. ZINBayes' performance on data sets with for which the number of cells is not large compared to the number of genes does not decrease, contrary to scVI;

2. m-pCMF and ZINBayes do not need the full batch of data to perform inference, contrary to ZINB-WaVE.

Future work would first consist of performing more extensive experiments on m-pCMF and ZINBayes. For example, we did not analyse the factor loadings on which sparsity was imposed through the sparse Gamma prior. These loadings should provide information on the contribution of genes to the lower-dimensional representations and could then be related with cell types. Furthermore, while we built Bayesian models, we did not assess the posterior uncertainties. These could be useful, for example, in differential expression analysis.

We argue that these scalable generative models will be useful in unlocking the many potential applications of scRNA-seq by providing a detailed look into the cell-to-cell heterogeneities in terms of gene expression. Characterizing cells in these terms can be used in describing diseases in terms of, for example, the presence of a certain kind of cells [25], which is a step forward towards personalized medicine. Another potential area of



(a) ZEISEL data set

(b) POLLEN data set

Figure 7: Clustering metrics on the latent space for 5 runs of each model on the whole ZEISEL and POLLEN data sets. bASW is the ASW score with respect to batch annotations.

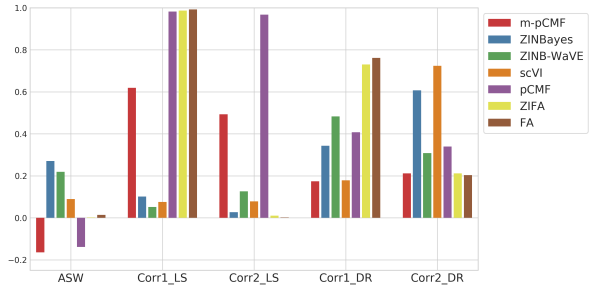


Figure 8: ASW of the 2-dimensional latent representations with respect to cell types of the POLLEN data set and absolute Pearson correlation coefficients of each dimension with library sizes (Corr1_LS, Corr2_LS) and detection rates (Corr1_DR, Corr2_DR).

application is using the RNA-level structure inferred by these models to improve the estimation of cancer stages provided by DNA data.

Acknowledgements

This work was partly supported by the European Union Horizon 2020 research and innovation program (grant No. 633974 – SOUND project).

References

[1] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell RNA sequencing. *Molecular Cell*, 58(4):610–620, 2015.

[2] Alessandra Dal Molin, Giacomo Baruzzo, and Barbara Di Camillo. Single-cell RNA-sequencing: assessment of differential expression analysis methods. *Frontiers in Genetics*, 8:62, 2017.

[3] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14(6):565, 2017.

[4] Emma Pierson and Christopher Yau. Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1):241, 2015.

[5] Ghislain Durif, Laurent Modolo, JE Mold, and Sophie Lambert-Lacroix. Probabilistic count matrix factorization for single cell expression data analysis. In *Research in Computational Molecular Biology*, page 254. Springer, 2018.

[6] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9(1):284, 2018.

[7] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael Jordan, and Nir Yosef. Bayesian inference for a generative model of transcriptome profiles from single-cell RNA sequencing. *bioRxiv*, 2018.

[8] Pedro F. Ferreira. m-pCMF implementation. `https://www.github.com/pedrofale/mpcmf`.

[9] Pedro F. Ferreira. ZINBayes implementation. `https://www.github.com/pedrofale/zinbayes`.

[10] Pedro F. Ferreira. Benchmarking implementations. `https://www.github.com/pedrofale/scrna_seq_benchmarking`.

[11] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

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

[13] John Canny. GaP: a factor model for discrete data. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 122–129. ACM, 2004.

[14] Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep exponential families. In *Artificial Intelligence and Statistics*, pages 762–771, 2015.

[15] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[16] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations*, 2017.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 32(10):1053, 2014.

[19] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015.

[20] 10X Genomics. `https://support.10xgenomics.com/single-cell-gene-expression`. Accessed: 2018-06-10.

[21] Cole M Risso D. *scRNAseq: A Collection of Public Single-Cell RNA-Seq Datasets*, 2016. R package version 1.6.0.

[22] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[23] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[24] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[25] Eirini Arvaniti and Manfred Claassen. Sensitive detection of rare disease-associated cell subsets via representation learning. *Nature Communications*, 8:14825, 2017.