

A deep generative model for gene expression profiles from single-cell RNA sequencing

*Nir Yosef
Michael Jordan
Romain Lopez*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/Eecs-2018-21

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/Eecs-2018-21.html>

May 1, 2018



Copyright © 2018, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**A deep generative model for gene expression profiles from single-cell
RNA sequencing**

by Author Romain Lopez

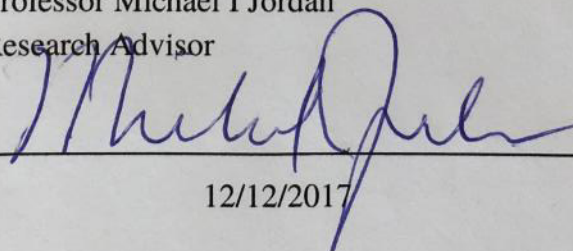
Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

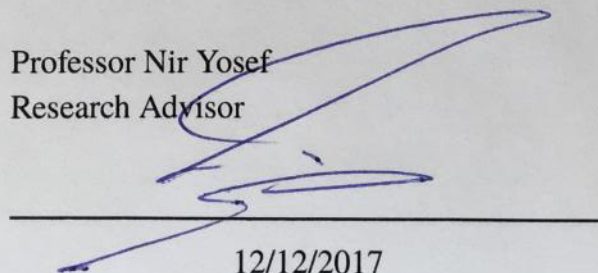
Committee:

Professor Michael I Jordan
Research Advisor



12/12/2017

Professor Nir Yosef
Research Advisor



12/12/2017

A deep generative model for gene expression profiles
from single-cell RNA sequencing

Romain Lopez

Submitted to the Department of Electrical Engineering and Computer
Sciences, University of California, Berkeley

December 12, 2017

Abstract

We propose a probabilistic model for interpreting gene expression levels that are observed through single-cell RNA sequencing. In the model, each cell has a low-dimensional latent representation. Additional latent variables account for technical effects that may erroneously set some observations of gene expression levels to zero. Conditional distributions are specified by neural networks, giving the proposed model enough flexibility to fit the data well. We use variational inference and stochastic optimization to approximate the posterior distribution. The inference procedure scales to over one million cells, whereas competing algorithms do not. Even for smaller datasets, for several tasks, the proposed procedure outperforms state-of-the-art methods like ZIFA and ZINB-WaVE. We also extend our framework to account for batch effects and other confounding factors, and propose a Bayesian hypothesis test for differential expression that outperforms DESeq2 and MAST.

Contents

1	Single-cell RNA sequencing: from experiments to data analysis	3
1.1	The technology and its ongoing breakthroughs	3
1.2	Statistical challenges	4
1.3	Opportunities	5
1.4	Regular workflow	5
2	Relevant work	7
2.1	Zero-Inflated Factor Analysis	7
2.2	Zero-Inflated Negative Binomial Wanted Variation Extraction	8
2.3	Two-way ANOVA	9
2.4	Model-based Analysis of Single-cell Transcriptomics	9
3	Probabilistic modeling for scRNA-seq data	10
3.1	Kinetic models for stochastic gene expression	10
3.2	Technical effects	11
3.3	scVI: a generative approach	11
3.4	Fast inference via stochastic optimization	12
3.5	Bayesian Differential Expression	15
4	Experiments	16
4.1	Software implementation	16
4.2	Datasets and preprocessing	16
4.3	Algorithms used for benchmarking	17
5	Results	18
5.1	Scaling up scRNA-seq data analyses	18
5.2	Fit on real data	18
5.3	Data imputation	21
5.4	scVI yields biologically meaningful clusters	22
5.5	Separation of biological information from technical noise	24
5.6	Selecting differentially expressed genes	27

Chapter 1

Single-cell RNA sequencing: from experiments to data analysis

Single-cell RNA sequencing (scRNA-Seq) is a revolutionary technology, which allows studying fundamental biological questions that were previously out of reach [1, 2]. It allows, for the first time, to reveal a cell's identity and characterize its molecular circuitry in an unbiased, data-driven way.

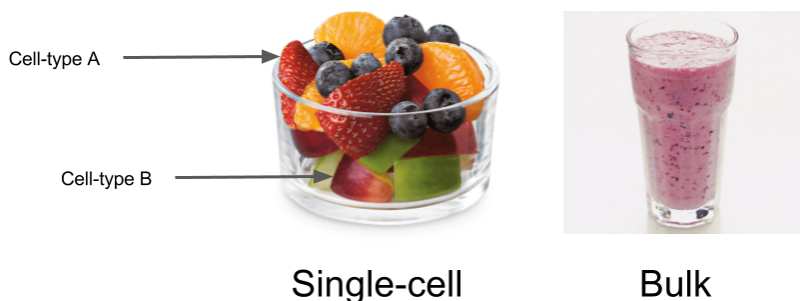


Fig. 1.1: Gene expression through the lens of single-cell RNA sequencing - Shalek Lab

1.1 The technology and its ongoing breakthroughs

After micro-arrays and bulk RNA sequencing, it has always been the dream of biologists to get gene expression profiling at the single-cell level. Seminal experiments in microfluidics and biology [3] allows one to take cells from a tissue and put each of them into a droplet of water. A droplet-specific barcode can be added to the mRNA to trace back which molecule came from which cell. Finally, the mRNA can be translated in cDNA and sequenced.

After alignment procedures, we get a matrix $X_{n,g}$ of counts for expression of gene g in cell n . For the first time, biologists are able to observe effects of mutations, cell types and diseases at the scale of a single-cell. With previous technologies, all cell types were mixed

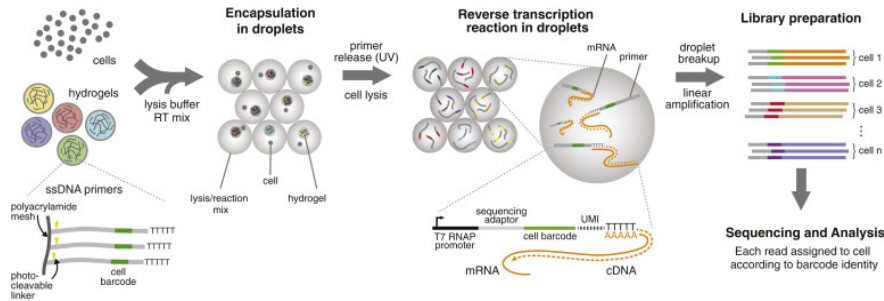


Fig. 1.2: A Platform for DNA Barcoding Thousands of Cells - Klein Lab [3]

together: it was technically impossible to decouple inter-sample variation (i.e individuals) with intra-sample variation (i.e cell-types) and researchers had to dig at the single-cell level to truly unravel relationships between individual genes and specific biological phenomena. Therefore, single-cell experiments took gene expression understanding to the next level with applications in immunology [4], oncology [5] and other subfields of biology [6].

This step of isolating cells was pretty hard at the early stage of single-cell. So first experiments would have pretty low sample-size (tens to a few hundred cells) and researchers designed ad-hoc algorithms to analyze the experimental data X . As the field evolved to take experiments to the next step and sequence millions of cells [7], the renowned biotech company 10X GENOMICS calls for methods to analyze it:

Our Million Cell Dataset defines a new standard for scaling up single cell analysis by orders of magnitude, opening up the possibility of tissue atlas studies that seek to comprehensively describe cellular subtypes and ultimately accelerate the characterization of all biological systems. To this end, we are making the dataset available for download without restrictions. 10X GENOMICS, 2017

1.2 Statistical challenges

As already underlined, careful computational analysis allows deriving from such data exciting insights in diverse biomedical fields [8, 9]. However, as one sequences more and more cells, technical limitations in the experimental protocol makes the gene expression matrix X more sparse than it should be in ideal (e.g low mRNA capture efficiency). While it is typical to observe thousands of gene products per cell, many transcripts are observed very infrequently, and for technical reasons related to the method of sequencing these are particularly prone to high variance.

Some of these zeros are believed to be part of the technical noise and not actually symptomatic of the gene not being expressed, this is an essential problem of single-cell datasets. We therefore say that entries of X are typically zero-inflated [10].

1.3 Opportunities

The biologist’s pain landscape could be described as:

- **Filtering** — How to check whether there is a cell in a droplet and not just noise ?
How to filter the genes to know which ones have biological signal ?
- **Clustering** — What kind of cell types is present in the experiment ?
- **Differential Expression** — What genes are particularly expressed in those cell types ?
- **Disentanglement** — How to separate the technical variance from the biological signal ?
- **Imputation** — How to establish whether a zero in the matrix is a technical or a biological zero ?
- **Multiple donor scenarios** — How to understand the heterogeneity of samples when we have multiple human donors ? Particularly when we have clinical phenotypes which we want to study through the lens of cellular heterogeneity ?

The first question can be answered with standard noise model from statistics [3]. However, each of the subsequent questions is essentially a hard machine learning problem. From that ensemble of problems, the biologist can then answer biological questions like identifying new cell types [11] or finding new regulators of autoimmunity [12].

This thesis will essentially propose a generative model whose inference is scalable to modern dataset sizes and can be used for the downstream analyses mentioned above. We will name the algorithm *single-cell Variational Inference* (scVI) after the method used for fast and approximate inference. We will comment on the performance of scVI through the lens of quantitative benchmarking for all the tasks mentioned above.

1.4 Regular workflow

While there is often little prior knowledge of single-cell heterogeneity generating X , a reasonably general assumption is that X has been generated from a low-dimensional manifold of cellular states [1]. Therefore, even though some algorithms are trying to address one particular question at a time (e.g clustering, imputation, removal of unwanted variation), dimensionality reduction remains the major step of the problem. Computational biologists are provided with a natural workflow when analyzing scRNA-Seq data and one description would be:

1. Filter the cells by total number of transcripts
2. Filter the genes by variance or inverse-dispersion parameter

3. Normalize the data (e.g apply a transformation by cell and by gene)
4. Apply a dimensionality reduction algorithm
5. Feed the output into a clustering and a visualization algorithm
6. Look at which genes are differentially expressed between pairs of clusters
7. Identify cell-subpopulations by pooling data from a database.

Our algorithm will assume the data is already filtered and that we have a procedure to biologically name given clusters (first and last task of the pipeline).

Chapter 2

Relevant work

Numerous dimensionality reduction techniques have been proposed for interpreting X (e.g., to facilitate clustering, visualization, and data imputation). Each technique has shortcomings, however. Most are based on linear models of the data [10, 13, 14] though there is no basis for assuming linearity. Most are optimized with batch algorithms, preventing them from scaling beyond thousands of cells [10, 13, 15]. However, sequencing millions of cells is becoming possible [7]. The best performing method to date [13] is particularly complicated to train, involving numerous subroutines for alternating minimization. Recent articles apply neural networks, but without an architecture based on biology [16, 17].

We present here in detail ZIFA [10] and ZINB-WaVE [13], state-of-the-art methods for dimensionality reduction of single-cell data. We will also refer to them during the benchmarking part. We do not present Principal Component Analysis and Factor Analysis but note that by default it remains a key candidate for analyzing large datasets. We will also present some common tools used in scRNA-seq data analysis for removal of unwanted variation and differential expression.

2.1 Zero-Inflated Factor Analysis

First, let us note that for all the methods with an underlying Gaussian assumption, it is common to apply the model after applying the transformation $x \mapsto \log(1+x)$ that conserves the zero but distort the counts to have a better fit with Gaussian conditionals. We now turn to the description of the first generative model specially tailored for single-cell data [10], in Fig 2.1.

We see that our count-matrix cannot be negative, even after the log-transformation so the Gaussian conditional might not be suitable. Also, the parametric assumption of the zero rate seems to be verified in practice but is not really flexible given our poor understanding of what these zeros really are. This parametric assumption for the dropout is however central to the inference since it allows us to derive EM updates in closed form. As we will see, this is not enough to yield an expressive and scalable model.

Require: constant for technical dropout λ

Require: fitted diagonal matrix W , dense matrix A and mean μ

- 1: **for** cell n in batch B **do**
- 2: Choose a low-dimensional vector $z_n \sim \mathcal{N}(0, I)$ describing the cell
- 3: Choose a gene expression vector $y_n \sim \mathcal{N}(Az_n + \mu, W)$
- 4: **for** gene g in gene set G **do**
- 5: Choose a dropout event with $h_{ng} \sim \text{Bernoulli}(e^{-\lambda y_{ng}^2})$
- 6: Apply dropout to expression level y_{ng} and output x_{ng}
- 7: **end for**
- 8: **end for**

Fig. 2.1: Generative model for ZIFA

2.2 Zero-Inflated Negative Binomial Wanted Variation Extraction

This recent research [13] does not involve a generative model. However, the assumption for the what the conditional would have been under a generative model are really appealing. Let N be the number of cells and K be the chosen dimension for the latent space.

Parameters : $W \in \mathcal{M}_{N,K}$, $\alpha_\mu \in \mathcal{M}_{K,V}$, $\alpha_\pi \in \mathcal{M}_{K,V}$, $O_\mu \in \mathbb{R}^N$, $O_\pi \in \mathbb{R}^N$, $\zeta \in \mathbb{R}^V$
Regression:

$$\begin{cases} \log(\mu) = X\beta_\mu + (V\gamma_\mu)^T + W\alpha_\mu + O_\mu \\ \text{logit}(\pi) = X\beta_\mu + (V\gamma_\mu)^T + W\alpha_\pi + O_\pi \\ \log(\theta_{i,j}) = \zeta_j \\ x_i \sim \text{ZINB}(\pi_i, \mu_i, \theta_i) \end{cases}$$

where $\text{ZINB}(\pi, \mu, \theta)$ is a mixture with rate μ between a zero distribution and a negative binomial with mean μ and dispersion θ parametrized by:

$$\forall y \in \mathbb{N}, \mathbb{P}(y; \mu, \theta) = \frac{\Gamma(y + \theta)}{\Gamma(y + 1)\Gamma(\theta)} \left(\frac{\theta}{\theta + \mu} \right)^\theta \left(\frac{\mu}{\mu + \theta} \right)^y$$

we then penalize the likelihood of the data X given *deterministic variables* W . Here the matrix $W = \text{rows}(z_i)$ could be a point estimate for the latent variable z_i . Since there is no latent variables, ZINB-WaVE can be seen as a matrix factorization algorithm where the loss is specifically tailored for scRNA-seq experiments. It is now clear that ZINB-WaVE is formulated as a regression problem with no prior on the latent variables.

Let us remark that ZINB-WaVE is also performing a general linear version of the two way ANOVA where gene-level and sample-level confounding factors can be regressed out.

Inference is made with alternating minimization, which is a standard technique for solving matrix completion problem. Each minimization is made with approximate second-order methods on the full batch of data. That is the main reason explaining why ZINB-WaVE does not scale well for modern size datasets.

2.3 Two-way ANOVA

scRNA-seq analysis literature often uses this type of method to remove unwanted variation. Take a data set for which an observed variable x is dependent on two others which are therefore potential sources of variation. The first source of variation can be the biology z (e.g cell-types) and the other one a unwanted variation r (e.g some quality metrics or batch identifier).

A *two-way ANOVA*-type approach would use the following linear model:

$$\begin{cases} x_i | \mu_i, \sigma^2 \sim \mathcal{N}(\mu_i, \sigma^2) \\ \mu_i = \mu + \alpha^T z_i + \beta^T r_i \end{cases}$$

It is then easy to adapt this type of model to use more suitable conditional probabilities (ZINB-WaVE) or enrich it with empirical Bayes as in Combat [18]. Let us note that when the biology z is a latent variable to be estimated, putting a prior on the two-way ANOVA is crucial in order to prevent the model from overfitting and removing biological information.

Let us remark that this approach is inherently discriminative, like ZINB-WaVE. Therefore, it is not clear at this point how to make it generative.

2.4 Model-based Analysis of Single-cell Transcriptomics

In general, any Bayesian model with a tractable posterior will be able to perform hypothesis testing. When looking at empirical Bayes models, building a statistics from a point estimate of the model will also be possible. We describe here MAST [19], state-of-the-art model for identifying differentially expressed genes and specifically tailored for scRNA-seq.

Define the indicator h_{ng} for gene g to be expressed in cell n , the log-normalized counts x_{ng} and a cell-specific scaling factor y_n .

$$\begin{cases} h_{ng} \sim \text{Bernoulli}(\text{sigmoid}(y_n \alpha_g)) \\ x_{ng} | h_{ng} = 1 \sim \mathcal{N}(y_n \beta_g, \sigma_g^2) \end{cases}$$

After putting suitable prior on σ_g , it is possible to derive a statistical test for differential expression by using a Z test on the coefficients α_g and β_g learned on two distinct sub-populations. Let us remark that their method is not designed for counts, takes into account zero-inflation and perform some normalization.

Chapter 3

Probabilistic modeling for scRNA-seq data

The task of building a generative model for scRNA-seq data might be deemed confusing at first sight. There are lots of different experimental protocols, each with sensibly different steps involved (UMIs, amplification etc...).

Therefore, we would not take down the road of analyzing one precise experiment and trying to come up with an exact model. Instead, we analyze a simple model of stochastic gene expression and come up with an adequate graphical model.

3.1 Kinetic models for stochastic gene expression

A simple model proposed in [20] propose to model the gene expression by a Markov jump process with a latent variable that indicating whether the gene is promoted. What we observe through the experiment is a draw from the time-collapsed probability distribution.

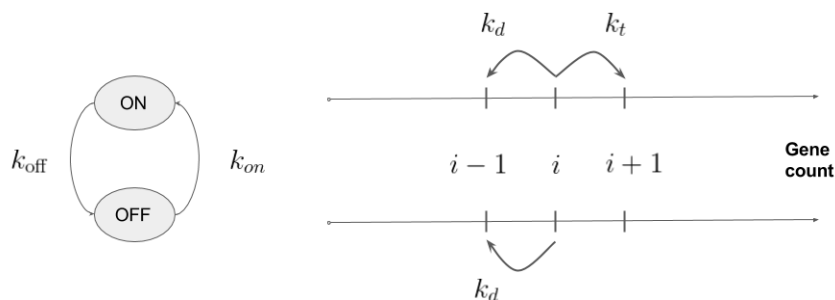


Fig. 3.1: A continuous time Markov Jump Process to model the kinetics of stochastic gene expression. Letters indicate exponential waiting times.

It turns out that this time-integrated probability distribution is not writable in closed form for continuous parameters. However, we can write it as a scaled-compound between

a Beta and a Poisson. Most of the time however, we assume that $k_{\text{off}} \ll k_{\text{d}}$ can approximate the hyper-geometric distribution by a negative binomial, which is a Gamma Poisson compound [20].

We will then start with the statement that some conditional in the graphical model should be a negative binomial. We will further refine that statement by introducing the graphical model.

3.2 Technical effects

Due to the low transcript efficiency, we expect the mean of expression to be very small. We partially try to correct the data from created side effects.

Dropout Also, there is a bias in the sampling (gene length, GC content, cell efficiency...) that can be modeled as introducing additional zeros in the data. These events, called "dropout" in computational biology, are completely different from the dropout regularization used in neural networks. We will then add an additional point mass at zero that will be treated as technical effects.

Library size We enforce the latent variable to encode directly the proportion of the mean gene expression over the whole gene set. We can then decouple the number of transcripts captured, called *library-size* (that is mainly technical for some applications) from the proportion.

3.3 scVI: a generative approach

We present here our generative model, *scVI* that benefits from adequate probabilistic assumptions and the flexibility of neural-nets. Our model explicitly models library-size, dropout and can remove unwanted variations as well as batch effects. We recapitulates those features and compare to other algorithms in Figure 3.2.

Figure 3.3 represents the probabilistic model graphically. The generative process is defined in Figure 3.4.

l_{μ}, l_{σ} are set to be the empirical mean of log-library size. Constant γ_n are optional covariates that can be passed to f_w that account for confounding effects (eg. sample batch and quality [21, 22]), to remove unwanted variation from the latent representation.

Neural network f_w is constrained during the inference to encode the mean proportion of transcripts expressed across all genes by using a softmax activation at the last layer. Neural network f_h encodes whether a particular entry has been "zeroed out" due to technical effects [10, 13]. All neural networks use dropout regularization and batch normalization. Batch normalization parameters can be batch-specific to remove batch-effects. Each network has 1, 2, or 3 fully connected-layers, with 128 or 256 nodes each. The activation functions are all ReLU, exponential, or linear. Weights for some layers are shared between f_w and f_h .

	Count	Dropout	Cell Scaling	Metadata	DE	Scalability
FA	-	-	-	-	-	+
ZIFA	-	+	-	-	-	-
ZINB-WaVE	+	+	+	+	-	-
BASICS	+	-	+	-	-	-
BISCUIT	-	-	+	-	-	-
DESEQ2	+	-	+	+	+	-
MAST	-	+	+	-	+	-
scVI	+	+	+	+	+	+

Fig. 3.2: Algorithms in scRNA-seq data analysis and their functionalities. Metadata indicates whether the algorithm can handle metadata. DE refers to differential expression.

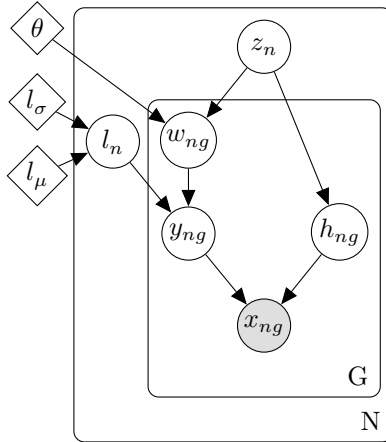


Fig. 3.3: The scVI graphical model

scVI is designed to explicitly remove library size and confounding effects while keeping the conditional distribution $p(x_{ng}|z_n, l_n)$ to a zero-inflated negative binomial—a distribution known to effectively model the kinetics of stochastic gene expression with some entries replaced by zeros [20]. In our experiments, we will focus on UMI-based data. This means we can use the negative binomial distribution more confidently since we have a low amplification bias.

3.4 Fast inference via stochastic optimization

The posterior distribution combines the prior knowledge with information acquired from the data X . We cannot directly apply Bayes rule to determine the posterior because the denominator (the marginal distribution) $p(x_n)$ is intractable.

Making inference over the whole graphical model is not needed. We can integrate out the latent variables w_{ng} , h_{ng} and y_{ng} by making sure the conditional $p(x_{ng}|z_n, l_n)$ has a closed-form density.

Require: constant prior for cell-specific scaling l_μ, l_σ
Require: optional covariates γ_n modeled as constant
Require: fitted gene-specific inverse dispersion parameter θ
Require: fitted neural networks f_w, f_h

- 1: **for** cell n **do**
- 2: Choose a low-dimensional vector $z_n \sim \mathcal{N}(0, I)$ describing the cell
- 3: Choose a cell-scaling factor $l_n \sim \text{Log}\mathcal{N}(l_\mu, l_\sigma^2)$
- 4: **for** gene g in gene set G **do**
- 5: Choose a normalized expression mean $w_{ng} \sim \text{Gamma}(f_w(z_n, \gamma_n), \theta)$.
- 6: Choose an expression level $y_{ng} \sim \text{Poisson}(l_n w_{ng})$.
- 7: Choose a dropout event with $h_{ng} \sim \text{Bernoulli}(f_h(z_n, \gamma_n))$
- 8: Apply dropout to expression level y_{ng} and output x_{ng}
- 9: **end for**
- 10: **end for**

Fig. 3.4: Generative model for scVI

First, take r to be the gene-specific shape parameter of a Gamma variable w , $\frac{p}{1-p}$ to be its scale parameter, and use a scalar $\lambda \in \mathbb{R}^+$ then the count variable $y|w \sim \text{Poisson}(\lambda w)$ has a Negative Binomial marginal distribution with mean $r\lambda\frac{p}{1-p}$

$$\begin{aligned}
p(y) &= \int p(y|w)p(w)dw \\
&= \int \frac{w^{r-1}e^{-w(\frac{1}{p}-1)}(1-p)^r}{p^r\Gamma(r)} \frac{e^{-\lambda w}\lambda^y w^y}{\Gamma(y+1)} dw \\
&= \frac{\Gamma(y+r)}{\Gamma(y+1)\Gamma(r)} \left(\frac{1-p}{1-p+\lambda p}\right)^r \left(\frac{p\lambda}{1-p+\lambda p}\right)^y
\end{aligned} \tag{3.1}$$

Second, multiplication by zero to y_{ng} can be formally encoded as a mixture between a point-mass at zero and the original distribution of y_{ng} .

Consequently, our conditional $p(x_{ng}|z_n, l_n)$ is a zero-inflated Negative Binomial with probability mass function:

$$\begin{cases} p(x_j = 0|z, l) = f_h(z)_j + (1 - f_h(z)_j) \left(\frac{\theta_j}{\theta + lf_w(z)_j}\right)^\theta \\ p(x_j = y|z, l) = (1 - f_h(z)_j) \frac{\Gamma(y + \theta_j)}{\Gamma(y + 1)\Gamma(\theta_j)} \left(\frac{\theta_j}{\theta + lf_w(z)_j}\right)^\theta \left(\frac{f_w(z)_j}{\theta + lf_w(z)_j}\right)^y, \forall y \in \mathbb{N}^* \end{cases}$$

when $f_h(z, \gamma)$ is encoding the zero probability of h and $f_w(z, \gamma)$ the mean of w .

Having simplified our model, we use variational inference [23] to approximate the posterior $p(z_n, l_n|x_n)$. Our variational distribution $q(z_n, l_n|x_n)$ is mean-field:

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

The variational distribution $q(z_n | x_n)$ is chosen to be Gaussian with a diagonal covariance matrix, mean and covariance are given by an encoder network applied to x_n , as in [24]. The encoder network may, optionally, be given the constant covariates γ_n (along with x_n) if we wish to discourage z_n from encoding batch effects and other unwanted variations. The variational distribution $q(l_n | x_n)$ is chosen to be log-Normal with scalar mean and variance also given by an encoder network applied to x_n .

The variational lower bound is

$$\log p(x) \geq \mathbb{E}_{q(z,l|x)} \log p(x|z, l) - KL(q(z|x)||p(z)) - KL(q(l|x)||p(l)) \quad (3.2)$$

To optimize the lower bound, we use the analytic expression for $p(x|z, l)$ and use analytic expressions for the Kullback–Leibler divergences. We use the reparametrization trick to compute low-variance Monte-Carlo estimates of the expectations’ gradients. Now, our objective function is continuous and end-to-end differentiable, which allows us to use automatic differentiation operators.

Since our model assumes cells are identically independently distributed, we can also benefit from stochastic optimization from sampling the training set. We then have an online optimization procedure that can handle massive datasets.

Negative binomial PMF parametrization A choice of parametrization is crucial for optimization consideration. We could follow [13] by using a mean μ and an inverse-dispersion θ parameter:

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

We also keep in mind a more gentle parametrization with nicer form even though still non-convex:

$$p_{NB}(n; p, r) = \frac{\Gamma(n + r)}{\Gamma(n + 1)\Gamma(r)} p^n (1 - p)^r$$

with $(p, r) = (\frac{\mu}{\theta + \mu}, \theta)$ or $(\mu, \theta) = (\frac{rp}{1-p}, r)$

Because the first parametrization has a better behavior when scaling the Poisson mean as we do with library size normalization, this is the one we retain.

Equivalence of parametrization Assume now one wants to simulate what would have been the rate of the latent corresponding Poisson variable, one has to sample from a Gamma of shape r and scale $\frac{p}{1-p}$ or rate $\frac{1-p}{p}$

Numerical considerations We transformed the expression to incorporate logits and use Tensorflow numerically stable functions. Instead of writing explicitly a sigmoid non-linearity, the probability of zero in the mixture is given by:

$$f_\pi(z) = \frac{1}{1 + e^{-F_\pi^z}}$$

where F_π^z is the output of the neural network without non-linearity. We then write the log-likelihood as a function of F_π^z .

r that can either be parametrized by a neural net or constant for each gene will be kept noted r for simplicity. \mathcal{S} denotes the softplus function $x \mapsto \log(1 + e^x)$.

$$\begin{aligned} \log p(y|z) = & \mathbb{1}_{y=0} \left[\mathcal{S}(-F_\pi^z + f_\theta^z \log \frac{f_\theta^z}{f_\theta^z + f_\mu^z}) - \mathcal{S}(-F_\pi^z) \right] \\ & + \mathbb{1}_{y>0} \left[-F_\pi^z - \mathcal{S}(-F_\pi^z) + f_\theta^z \log \frac{f_\theta^z}{f_\theta^z + f_\mu^z} + y \log \frac{f_\mu^z}{f_\theta^z + f_\mu^z} + \log \frac{\Gamma(y + \theta)}{\Gamma(\theta)\Gamma(y + 1)} \right] \end{aligned}$$

3.5 Bayesian Differential Expression

Let A and B be two set of cells and g a fixed gene. Now take $(a, b) \in A \times B$ and say we want to test the following:

$$\mathcal{H}_0^g : \rho_{ag} < \rho_{bg} \quad \text{vs.} \quad \mathcal{H}_1^g : \rho_{ag} \geq \rho_{bg}$$

where $\rho = f_w(z_n, \gamma_n)$ is the mean of the gene expression conditioned on a non-dropout event. The posterior of these hypotheses can be approximated via the variational distribution:

$$p(\mathcal{H}_0^g|x) \approx \iint_{z_a, z_b, w_{ag}, w_{bg}} p(\rho_{ag} < \rho_{bg}) dq(z_a|x_a) dq(z_b|x_b)$$

where all the measures are low-dimensional so we can use naive monte-carlo to compute these integrals. We can then use a Bayes factor for the test.

$$\text{Reject when } \log \frac{p(\mathcal{H}_0^g|x)}{p(\mathcal{H}_1^g|x)} \text{ is large}$$

Our model assumes cells are i.i.d sampled from the generative model so simple arithmetic shows we can average the Bayes Factors when we have clusters of cells.

Let us remark that our model would in theory allow to ask even more questions about the data, pretty much anything that can have a tractable posterior. For instance, one could think about *robust differential expression* where we would reject if only the mean is more significant by a certain fraction or over a certain amount of the clusters population. We leave this as future work in this manuscript.

Chapter 4

Experiments

4.1 Software implementation

Our model is implemented in Python and TensorFlow. A functional code can be found at <https://github.com/YosefLab/scVI>

4.2 Datasets and preprocessing

Mouse Cortex Cells The dataset from [11] contains 3005 mouse cortex cells and gold-standard labels for seven distinct cell types. Each cell type corresponds to a cluster to recover. We sample top 558 genes ordered by variance as in [15].

PBMCs We extract 12039 Peripheral blood mononuclear cells (PBMCs) from [25] with 10310 sampled genes and get biologically meaningful clusters with the software Seurat [26]. We first filter genes that we could not match with the bulk data used for differential expression to be left with $g = 3346$. This is the dataset we will use for the differential expression analysis. For the clustering, imputation, likelihood and normalization analysis, we further filter the genes so that the other algorithms can be run (ZIFA and ZINB-WaVE did not complete after 3 hours for $g = 3346$). We therefore keep only the top 800 genes by variance.

Brain cells We also use a dataset that contains 1.3 million brain cells from 10X GENOMICS [7]. We randomly shuffle the data to get a 1M subset of cells and order genes by variance to retain 720 sampled variable genes.

Bipolar cells of mouse retina We use a dataset of bipolar cells from [27] and follow their recommended pipeline for genes and cells filtering. We obtain 27,499 cells and 13,166 genes coming from two batches. We also use their DE verified clusters as labels. We also extract their normalized data with Combat and use it for benchmarking.

4.3 Algorithms used for benchmarking

Factor Analysis We used the Factor Analysis method from the scikit-learn python package. FA is always applied to log-data.

ZIFA We used the zero-inflated factor analysis method (ZIFA) from <https://github.com/epierson9/ZIFA> with default parameters. We always apply ZIFA to log-data.

ZINB-WaVE We applied the ZINB-WaVE procedure from the R package zinbwave with the gene-level covariate to be a column of one and the cell-level covariate to be a column of ones. We always apply ZINB-WaVE to count-data.

PCA We used the Principal Component Analysis method from the scikit-learn python package. We always apply PCA to log-data.

Normalization We used the package SCONE to normalize the data. In particular, we relied on the package to perform QC matrix removal and rank hundreds of normalization strategies on the PBMCs dataset.

MAST We used the R package MAST on log-counts to provide our differential expression analysis.

DESeq2 We used the R package DESeq2 on raw counts to provide our differential expression analysis.

Chapter 5

Results

5.1 Scaling up scRNA-seq data analyses

Only scVI and Factor Analysis scale to the larger benchmark datasets—a key advantage relative to ZIFA and ZINB-WaVE. ZIFA and ZINB-WaVE are based on batch¹ optimization algorithms. Their runtimes for *each* iteration of their numerical optimization routines scale linearly in the number of samples and linearly in the number of genes—both potentially very large. scVI relies on stochastic optimization and its complexity per iteration is not dependent on the dataset size. Furthermore, it is a low memory footprint algorithm compared to ZINB-WaVE which — having its number of parameters linear in the number of cells — quickly runs out of memory or could overfit.

For reference, we investigate running time on our sample of the 1.3 million cells dataset. For 10,000 cells, each of these methods requires more than 20 minutes of computation. For 5,000 cells, both methods run out of memory on a machine with 32 GB RAM. scVI trains on the entire 1.3 million cell dataset in less than two hours on a single GPU, using off-the-shelf neural network software. More precise results are mentioned in Figure 5.1.

5.2 Fit on real data

For real datasets, we provide a multi-variate metric of goodness of fit on the data. We fit the desired algorithm using a training set and evaluate the log-likelihood on a held-out set. This common method of evaluation for generative models is robust to over-fitting because it evaluates the model on data it has never seen and is also detached from clustering.

All models relying on a probabilistic model can be mapped with a likelihood score that can be compared across models to perform model selection. We then add some code to ZIFA and ZINB-WaVE to make them perform these operations. Finally, we classically compare lower bounds on likelihood provided by EM algorithms, variational inference or pure optimization.

¹Here batch is referring to the optimization terminology. a batch optimization algorithm uses the whole dataset at once for the procedure and cannot handle new data coming in. It has always a high memory footprint compared to an online algorithm.

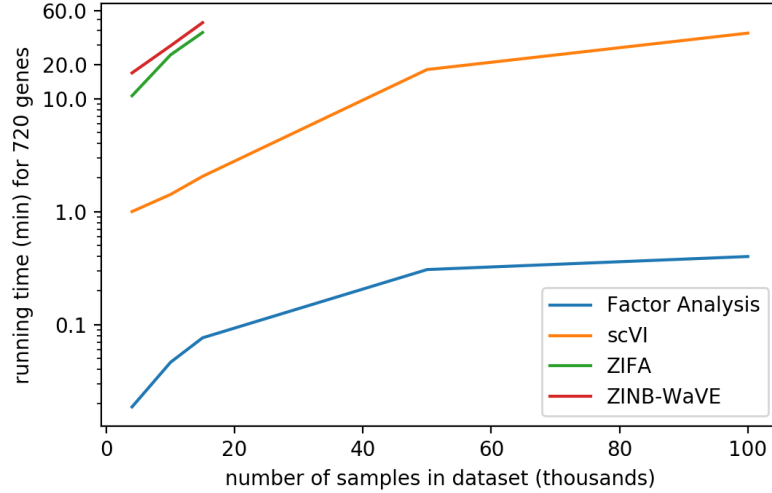


Fig. 5.1: Running time of scVI compared to other main choices for dimensionality reduction in scRNA-seq

5.2.1 Comparing likelihoods for log-data and non-log data

Since some models are meant to be fitted on log-data and other non non-log data, we take into account this by looking at the densities. Let X be a positive random variable and let us note $Y = \log(1 + X)$ and suppose we have a model for Y written \mathbb{P}_Y . The likelihood score on the raw data is given by evaluating the density \mathbb{P}_X which is:

$$\forall x = (x_1, \dots, x_d) \in \mathbb{R}^d, d\mathbb{P}_X(x) = d\mathbb{P}_Y(\log(1 + x)) \prod_{i=1}^d \frac{1}{1 + x_i}$$

so this yield for the likelihood scores:

$$\log \mathbb{P}_X(X = x) = \log \mathbb{P}_Y(Y = \log(1 + x)) - \sum_{i=1}^d \log(1 + x_i)$$

5.2.2 Log-likelihood for ZINB-WAVE

The function to be optimized for ZINB-WAVE is essentially penalized likelihood. One can thus run once the full optimization function on a training set as follow:

$$\max_{\beta, \gamma, W, \alpha, \zeta} \mathcal{L}_{\text{train}}(\beta, \gamma, W, \alpha, \zeta) - \text{Pen}(\beta, \gamma, W, \alpha, \zeta)$$

This optimization is performed by alternating minimization. By fixing the variables β, α, ζ learned from the training set, we can compute a likelihood on a validation set by

performing inference over the latent variables γ, W which is a simple Ridge that can be solved in parallel by a simple modification of their code.

$$\max_{\gamma, W} \mathcal{L}_{\text{val}}(\beta^*, \gamma, W, \alpha^*, \zeta^*) - \text{Pen}(\beta^*, \gamma, W, \alpha^*, \zeta^*)$$

5.2.3 Log-likelihood for ZIFA

The EM algorithm naively provide a lower bound on the log-likelihood:

$$\log p(Y|\Theta) \geq \mathbb{E}_{p(Z, X, H|Y, \Theta)} \log p(Z, X, H, Y|\Theta)$$

The complete log-likelihood has a simple expression:

$$\begin{aligned} \log p(z_i, x_i, h_i, y_i|\Theta) &= -\frac{1}{2} z_i^T z_i - \sum_j \log(\sigma_j) \\ &+ \sum_{j|y_{i,j}=0} -\frac{(x_{i,j} - (Az_i)_j - \mu_j)^2}{w\sigma_j^2} - \lambda_j x_{i,j}^2 \\ &+ \sum_{j|y_{i,j}>0} -\frac{(y_{i,j} - (Az_i)_j - \mu_j)^2}{w\sigma_j^2} + \log(1 - e^{-\lambda_j y_{i,j}^2}) \end{aligned}$$

and the prior distribution is close to Gaussian so we can modify ZIFA code and use a E step to compute the desired value. E-step gives us the following values:

$$\begin{aligned} \mathbb{E}(x_i \odot x_i) &\triangleq \text{EX}^2 \\ \mathbb{E}(z_i z_i^T) &\triangleq \text{EZZ}^T \\ \mathbb{E}(x_i) &\triangleq \text{EX} \\ \mathbb{E}(z_i) &\triangleq \text{EZ} \end{aligned} \tag{5.1}$$

Then we have:

$$\begin{aligned} LL &= -\frac{1}{2} \text{tr}(\text{EZZ}^T) - \frac{d}{2} \log(2\pi) - \sum_j \left[\frac{\log(2\pi\sigma_j^2)}{2} + \frac{\mu_j^2}{2\sigma_j^2} + \frac{(\text{AEZZ}^T A^T)_{j,j}}{2\sigma_j^2} + \frac{(\text{AEZ} \odot \mu)_j}{\sigma_j^2} \right] \\ &+ \sum_{j|y_{i,j}=0} \frac{1}{2\sigma_j^2} [-\text{EX}_j^2 + 2(\text{EXZA}^T)_{j,j} + 2(\text{EX} \odot \mu)_j] - \lambda \text{EX}_j^2 \\ &+ \sum_{j|y_{i,j}>0} \frac{1}{2\sigma_j^2} [-y_{i,j}^2 + 2(y_i \odot \text{AEZ})_{j,j} + 2(y_i \odot \mu)_j] + \log(1 - e^{-\lambda y_{i,j}^2}) \end{aligned}$$

where \odot denotes the Hadamard product.

5.2.4 Log-likelihood for scVI

Our variational inference procedure provides us with a lower bound on the log-likelihood of held-out data:

$$\log p(x) \geq \mathbb{E}_{q(z,l|x)} \log p(x|z, l) - KL(q(z|x)||p(z)) - KL(q(l|x)||p(l)) \quad (5.2)$$

To compare fairly with ZINB-WaVE who is using a deterministic design and therefore still optimizing at test-time, we relax the Gaussian prior and allows to optimize our inference network at test-time. That is essentially equivalent to assess the marginal likelihood of held-out data, conditioned on a latent representation learned for the held-out data.

5.2.5 Results

For each method, we learn a mapping from the 10-dimensional latent space to a reconstruction of training set X . Table 5.1 shows that scVI best compresses the held-out data, even for our smallest dataset. scVI’s lead over the other methods grows as the dataset size grows.

cells	4k	10k	15k	50k	100k
FA	-1175.36	-1177.35	-1177.27	-1171.93	-1169.86
ZIFA	-1250.44	-1250.77	-1250.59	NA	NA
ZINB-WaVE	-1166.39	-1163.91	-1163.39	NA	NA
scVI	-1150.96	-1146.59	-1144.88	-1136.57	-1133.94

Table 5.1: Marginal log likelihood for a held-out subset of the brain cells dataset. NA means we could not run the given algorithm for this sample size.

5.3 Data imputation

We fit the ZIFA model to each dataset to assess a parametric model of dropout (ie $p_{ij} \sim e^{-\lambda y_{ij}^2}$). Based on that model, we generate a corrupted training set by masking out non-zero entries by probability p_{ij} . Because we have introduced these zeros synthetically, we know 1) each entry’s true value, and 2) that each entry is zero because of a technical effect, not because the true expression level is nearly zero. We then fit this perturbed dataset with the desired algorithm and we evaluate it by looking how the zero are imputed. We also compare for this task to a state-of-the-art method MAGIC [28] based on diffusion in the cell k-nearest neighbors graph.

We use two metrics for that task that we detail here.

5.3.1 Accuracy of modeling zeros

For each class of models we want to compare, we use the fitted model to output a probability of zero and compute a binary cross-entropy:

$$-\frac{1}{nJ} \sum_{ij} \mathbb{1}_{Y_{ij}=0} \log p_{ij} + \mathbb{1}_{Y_{ij}>0} \log(1 - p_{ij})$$

where the zeros probability is defined according to the model:

- For ZIFA, take a E-step on the corrupted data and use $\mathbb{E}(X)$ as an approximation of x in the dropout probability $p_{ij} = e^{-\lambda x_{ij}^2}$ that will be exactly the zero probability.
- For ZINB-Wave or scVI, the zero probability is given by $p_{ij} = \pi_{ij} + (1 - \pi_{ij}) \left(\frac{\theta_j}{\theta_j + \mu_{ij}} \right)^{\theta_j}$.

5.3.2 Accuracy of imputing missing data

As imputation tantamount to replace missing data by its mean conditioned on being observed, we use the median \mathbb{L}_1 distance between the original dataset and the mean of the generative distribution (conditioned on a non-zero event) for corrupted entries only.

5.3.3 Results

We report on the PBMCs, the brain cells and the mouse cortex cells in Figure 5.2.

5.4 scVI yields biologically meaningful clusters

To further assess the models, we compare how each clusters cells of known types (e.g., muscle cells, blood cells) in latent space. For this task, we make a slight modification to our model: we treat each z_n as an unknown parameter to estimate rather than a latent variable with a distribution. This way, our procedure maximizes mutual information between z_n and x_n [29]. That measure will prevent the clusters to be well separated since the KL regularization term would tend to have the clusters sticked together.

We will use several clustering metrics throughout the paper.

Silhouette width The silhouette width requires either a similarity matrix or a latent space. We can define a silhouette score for each sample i with:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance of i to all data points in the same cluster c_i . $b(i)$ is the lowest average distance of i to all data points in the same cluster c among all clusters c .

The following metrics requires a clustering and not simply a similarity matrix. For these ones, we will use a k-means clustering on the given latent space and report the best score in $T = 10$ runs.

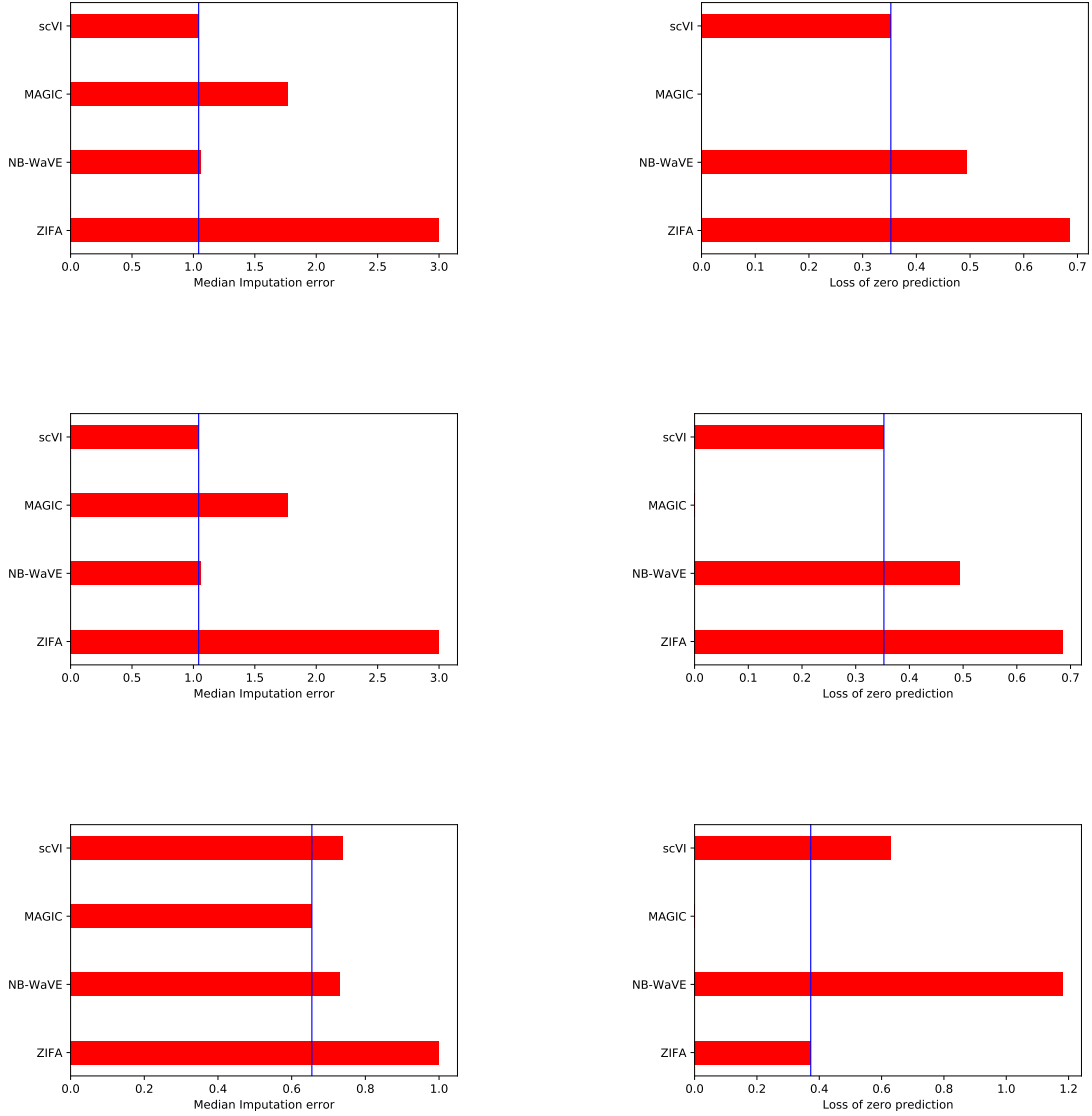


Fig. 5.2: Absolute errors for imputing zeroed entries (column 1), mean cross entropy for predicting which entries were zeroed-out entries (column 2). Mouse cortex cells (row 1), Brain cells (row 2), PBMCs (row 3). MAGIC does not predict dropout probabilities.

Adjusted Rand Index This index requires a clustering. Most

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

where n_{ij}, a_i, b_j are values from the contingency table.

Normalized Mutual Information

$$NMI = \frac{I(P; T)}{\sqrt{\mathbb{H}(P)\mathbb{H}(T)}}$$

where P, T designates empirical categorical distributions for the predicted and real clustering. I is the mutual entropy and \mathbb{H} is the Shannon entropy.

5.4.1 Results

We will later investigate more flavors of clustering situations. For now, we can focus on the mouse cortex dataset whose labels are trusted and compare scVI to its concurrents. Results are reported in Figure 5.3.

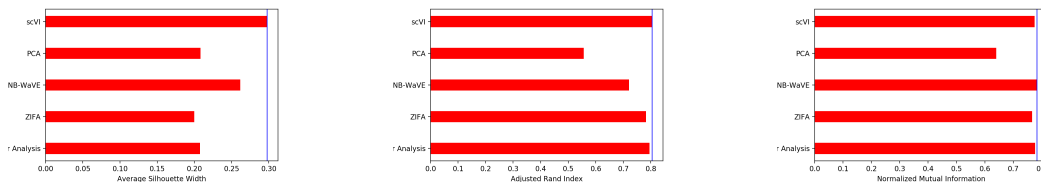


Fig. 5.3: Silhouette score (column 1), adjusted rand index (column 2), normalized mutual information (column 3) on the mouse cortex dataset.

5.5 Separation of biological information from technical noise

We will now investigate more nuanced flavors of clustering by a study case on quality control metrics for the PBMCs data and removal of batch-effect.

5.5.1 Removal of QC matrix on PBMCs data

Any flavor of variational auto-encoding Bayes — even the simplest like a Factor analysis or PCA — has a limitation of using the data to explicitly compute the latent space. This causes the latent variables to be correlated with the data and probably also by technical noise. On

the opposite, by optimizing the pseudo-latent space, ZINB-WaVE circumvents the risk of embedding technical noise at the price of computations.

We use this problem as a study case for the PBMCs data. We use SCONE [30] to select most important factors of unwanted variation to be incorporated into downstream models.

In addition to batch and biological condition, SCONE scores the extent to which covariation across “positive” or “negative” control genes is preserved by normalization. Positive controls (n=202) were derived from a list of the 500 most common genes from the C7 collection of the msigDB [31]. Negative control lists (n=202) were matched in average expression and were derived from a list of housekeeping genes expressed across a compendium of 47 human tissues [32]. Nine quality control metrics were defined from molecular level read information:

1. Number of UMIs per cell
2. Number of reads per cell
3. Mean read per UMI per gene
4. Standard deviation of reads per UMI per gene
5. Number of mapped reads
6. Number of mapped reads
7. Number of genomic reads
8. Number of unmapped reads
9. Number of corrected UMIs
10. Number of corrected cell barcodes

These metrics were similarly probed for their association with the normalized latent space.

The normalization model that performed best by SCONE metrics involves a full-quantile normalization, followed by an RUV-like normalization accounting for the first PC of QC variation. The normalized matrix was defined as the residual of the regression of log-expression on all this factor.

We also normalized the data to perform the RUV-like normalization for 3 PCs of QC and 8 PCs of QC. Because the normalization operation is not Bayesian, adding more QC might make the regression overfitting and thus remove biological information. We explicitly show this trade-off in Table 5.2 by reporting a correlation score (coefficient of determination) and clustering metrics.

In the case of scVI, we can disentangle these two sources of variation in a generative way by modifying slightly our generative model to add a latent variable p which will influence partly the expression level w but will also be forced to reconstruct the QC, treated as an observed variable. For the inference model, we force z to depend only on the expression data

	silhouette	ARI	NMI	QC correlation
QC1	0.37	0.69	0.78	0.24
QC3	0.31	0.63	0.73	0.11
QC8	0.27	0.62	0.72	0.10

Table 5.2: The effect of over-removal of quality metrics. Each row designates a normalization scheme with full quantile normalization and qc removal by regression. The number of qc principal components removed out is mentioned as row index.

x and p to depend only on the QC. Something less principled we also have to do is to — as in PCA — input in the variational distribution $q(z|x)$ the residual of a regression of the QC on x^2 .

We report results on Figure 5.4

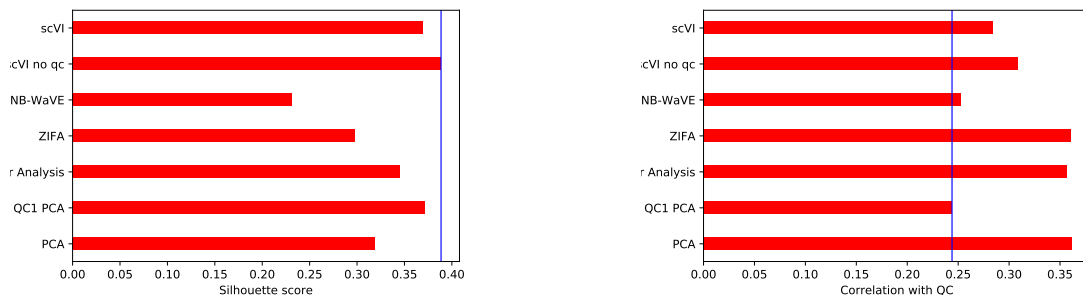


Fig. 5.4: Unwanted variation metric on the PBMCs dataset.

5.5.2 Removal of batch effects on bipolar cells

We now turn to a discrete model of unwanted variation: batch effects. The crucial choice is whether we want to think about batches as random or not. In the regimes of low number of batches (≤ 10), it makes sense to treat them as "stamps" written on the data that we want to remove via a normalization style technique or more generally via domain-adaptation. In a whole different regime where we see hundred of batches (which could happen when integrating data from large databases such as GEO), we might want to think about them as random and model their contribution.

In the following, we describe how to deal with the first situation and illustrate it with the bipolar dataset. As for the QC discussion, there might be a trade-off between removing information and clustering. This depends much on the composition of the experiment across batches. If all the batches are the same proportion of cell-types, we could technically perform a calibration to align perfectly the datasets. If not, there a fundamental trade-off. We will

²Without that step, we do not have clearly better results for now. We are still making some work on this.

for the following put ourselves in the situation where batches are biological replicates and should share significant properties.

We then investigate our batch effect removal strategy for scVI with state-of-the art method Combat on the bipolar data. Combat relies on empirical Bayes to regularize the two-way ANOVA described earlier and is claimed to do a better job than a one-way ANOVA (removing mean and scaling by genes) since the second one might distort biological information.

Removing batch effects with scVI Our way of removing batch-effect is a simple location-scaling strategy that happens in the hidden layers of the inference network. scVI uses batch-normalization between hidden layers to improve the performance of the generative model. This statistics normalization is learned with minibatches of data, during the training and is a simple location-scaling in reasonable dimension. We therefore perform this batch-normalization uniquely for each biological replicates to add only 128 parameters per batch to the inference network. That decreases the risk of overfitting.

We also feed the generative model network with the batch "stamp" so that it can regenerate well the data with the bath effects. Then, since the model learned to reproduce all the different batches, we could regenerate the data for only one batch and perform unbiased differential expression.

Entropy of batch mixing Fix a similarity matrix for the cells and take U to be a uniform random variable on the population of cells. Take B_U the empirical frequencies for the 100 neighrest neighbors of cell U being a in batch b . Report the entropy of this categorical variable and average over $T = 100$ values of U .

We evaluate the entropy of batch mixing as well as clustering performance and report the results of our experiment in Figure 5.5.

5.6 Selecting differentially expressed genes

A significant application of our generative model and of main interest in the field is to go from a clustering to a procedure for identifying gene differentially expressed between two cell-types. Our model relies on Bayesian statistics and can thus benefit from uncertainty evaluation to provide a hypothesis testing framework for differential expression.

We use again the PBMC dataset from [25] and the Seurat-based cell classification to understand how differential expression is captured by our testing method compared to traditional DESeq2 [33] and MAST. We defined a reference from a publicly-available bulk array expression profiling data for human B cells (n=10) and Dendritic cells (n=10) at baseline of vaccination GSE29618 which we use to test the association of each gene's expression with biological class, defining a 2-sided t-test p-value per gene. We also look at the same expression data for CD4 T cells (n=12) and CD8 T cells (n=12) GSE8835.

On the $g = 3346$ genes that we could match from these files, we choose for each case the top 300 by p-values and report an area under the ROC curve for the ranking provided by

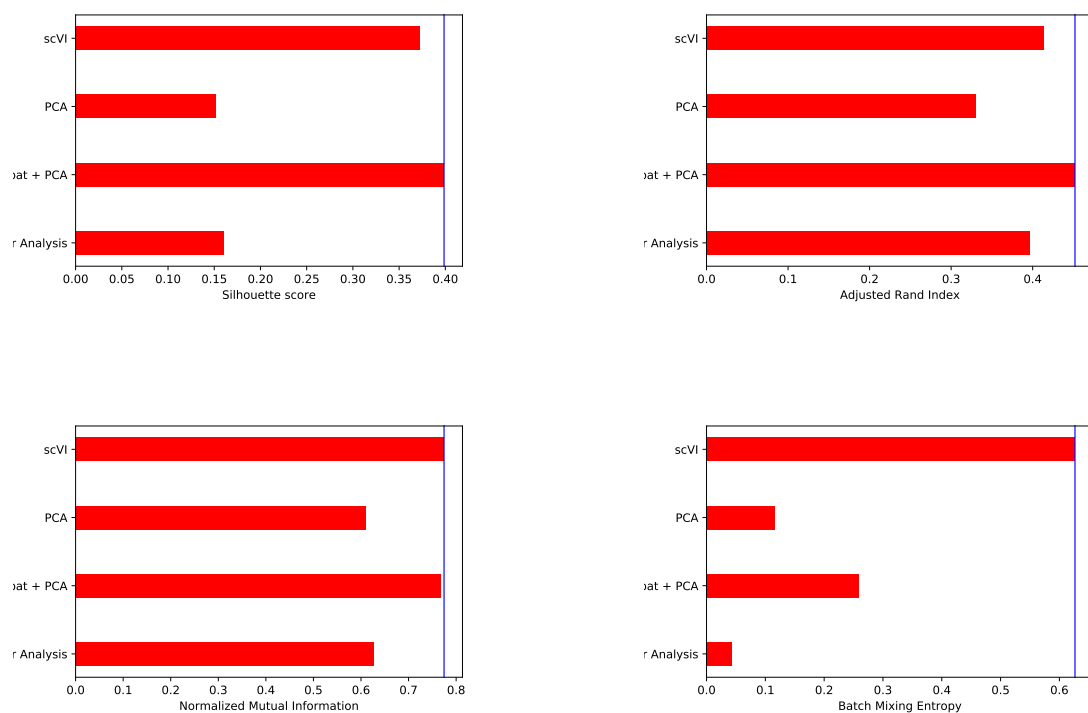


Fig. 5.5: Unwanted variation metric on the bipolar dataset.

the differential expression on the single-cell data. Because defining a threshold is ambiguous we also look at reproducibility between the microarray experiment and the family of tests used on the scRNA-Seq sequencing experiment. To quantify this, we model the relationship between significance ranks using the Irreproducible Discovery Rate model for matched rank lists [34]. It fits a copula mixture model to understand which p-values are reproducible across experiments and which one are not. The mixture weight then quantifies the proportion of genes whose rank is reproducible. We report this mixture weight of the reproducible components as well as the AUC in Figure 5.6. We used for each point 100 cells from each cluster. In scVI, we draw 200 samples from the variational posterior.

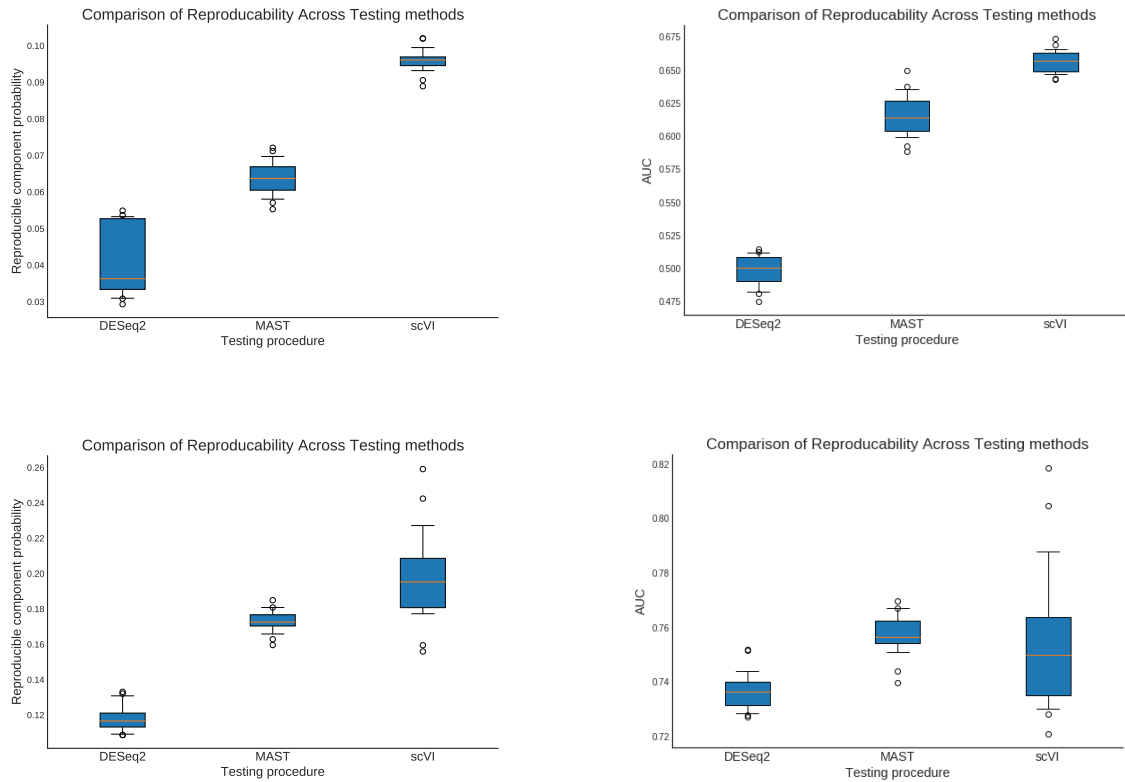


Fig. 5.6: Results on the Differential expression task. (a) Mixture weight of the reproducible components on CD4 against CD8 cells. (b) Area under the curve on CD4 against CD8 cells. (c) Mixture weight of the reproducible components on B cells against DC cells. (d) Area under the curve on B cells against DC cells.

Discussion

We have presented scVI, a complex and extensive framework for modeling single-cell RNA sequencing experimental data.

This project is a milestone towards the real objective whose question would be: how can we integrate knowledge about individual observations of cells to build understanding of global biological phenomena across a heterogeneous population of individuals ?

Bibliography

- [1] Allon Wagner, Aviv Regev, and Nir Yosef. Revealing the vectors of cellular identity with single-cell genomics. *Nature Biotechnology*, 34(11):1145–1160, 2016.
- [2] Amos Tanay and Aviv Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, 2017.
- [3] Allon M Klein, David A Weitz, and Marc W Kirschner Correspondence. Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell*, 161:1187–1201, 2015.
- [4] Alex K. Shalek, Rahul Satija, Xian Adiconis, Rona S. Gertner, Jellert T. Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, John J. Trombetta, Dave Gennert, Andreas Gnirke, Alon Goren, Nir Hacohen, Joshua Z. Levin, Hongkun Park, and Aviv Regev. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–240, Jun 2013. Letter.
- [5] Nicholas Navin, Jude Kendall, Jennifer Troge, Peter Andrews, Linda Rodgers, Jeanne McIndoo, Kerry Cook, Asya Stepansky, Dan Levy, Diane Esposito, Lakshmi Muthuswamy, Alex Krasnitz, W. Richard McCombie, James Hicks, and Michael Wigler. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, Apr 2011.
- [6] Dominic Grun, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251–255, Sep 2015. Letter.
- [7] 10x genomics, 1.3 million brain cells from e18 mice, 2017.
- [8] Hongkui Zeng and Joshua R Sanes. Neuronal cell-type classification: challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18, 2017.
- [9] Michael J T Stubbington, Orit Rozenblatt-Rosen, Aviv Regev, and Sarah A Teichmann. Single-cell transcriptomics to explore the immune system in health and disease. *Science*, 358(6359):58–63, oct 2017.
- [10] Emma Pierson and Christopher Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1):241, 2015.

- [11] Amit Zeisel, Ana B. Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.
- [12] Jellert Gaublomme, Nir Yosef, Youjin Lee, Rona Gertner, Li?V Yang, Chuan Wu, Pier?Paolo Pandolfi, Tak Mak, Rahul Satija, Alex Shalek, Vijay Kuchroo, Hongkun Park, and Aviv Regev. Single-cell genomics unveils critical regulators of th17 cell pathogenicity. *Cell*, 163(6):1400–1412, 2017.
- [13] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and JP Vert. ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data. *bioRxiv*, 2017.
- [14] David Detomaso and Nir Yosef. FastProject: A tool for low-dimensional analysis of single-cell RNA-Seq data. *DeTomaso BMC Bioinformatics*, 17, 2016.
- [15] Sandhya Prabhakaran, Elham Azizi, and Dana Pe’er. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. *Proceedings of The 33rd International Conference on Machine Learning*, 48:1070–1079, 2016.
- [16] Jiarui Ding, Anne E. Condon, and Sohrab P Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *bioRxiv*, 2017.
- [17] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Research*, 2017.
- [18] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.
- [19] Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K Shalek, Chloe K Slichter, Hannah W Miller, M. Juliana McElrath, Martin Prlic, Peter S Linsley, and Raphael Gottardo. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(1):278, 2015.
- [20] Dominic Grun, Lennart Kester, and Alexander van Oudenaarden. Validation of noise models for single-cell transcriptomics. *Nature Methods*, 11(6):637–640, 2014.
- [21] 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):565–571, 2017.
- [22] Davide Risso, John Ngai, Terence P Speed, and Sandrine Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, 32(9):896–902, 2014.

- [23] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.
- [24] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *The International Conference on Learning Representations*, 2014.
- [25] Grace X Y Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T Gregory, Joe Shuga, Luz Montesclaros, Jason G Underwood, Donald A Masquelier, Stefanie Y Nishimura, Michael Schnall-Levin, Paul W Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D Ness, Lan W Beppu, H Joachim Deeg, Christopher McFarland, Keith R Loeb, William J Valente, Nolan G Ericson, Emily A Stevens, Jerald P Radich, Tarjei S Mikkelsen, Benjamin J Hindson, and Jason H Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049, 2017.
- [26] Evan Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison Bialas, Nolan Kamitaki, Emily Martersteck, John Trombetta, David Weitz, Joshua Sanes, Alex Shalek, Aviv Regev, and Steven McCarroll. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2017.
- [27] Karthik Shekhar, Sylvain W. Lapan, Irene E. Whitney, Nicholas M. Tran, Evan Z. Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z. Levin, James Nemesh, Melissa Goldman, Steven A. McCarroll, Constance L. Cepko, Aviv Regev, and Joshua R. Sanes. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323.e30, 2017/11/28 XXXX.
- [28] David van Dijk, Juozas Nainys, et al. MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *bioRxiv*, page 111591, 2017.
- [29] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.
- [30] M. Cole and D. Risso. Single cell overview of normalized expression data, r package version 0.99.6, 2016.
- [31] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.
- [32] Eli Eisenberg and Erez Y Levanon. Human housekeeping genes are compact. *Trends in Genetics*, 19(7):362–365, 2003.

- [33] Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biol*, 15(12):550, 2014.
- [34] Qunhua Li, James B Brown, Haiyan Huang, and Peter J Bickel. Measuring reproducibility of high-throughput experiments. *Annals of Applied Statistics*, 5(3):1752–1779, 2011.