

Latent variable modelling and variational inference for scRNA-seq differential expression analysis

Joana Godinho
joana.godinho@tecnico.ulisboa.pt

October 2019

Abstract

Disease profiling, treatment development, and the identification of new cell populations are some of the most relevant applications relying on differentially expressed genes (DEG) analysis. In this context, three leading technologies emerged; namely, DNA microarrays, bulk RNA sequencing (RNA-seq), and single-cell RNA sequencing (scRNA-seq), the main focus of this work. Although scRNA-seq tends to offer more accurate data, it is still limited by many confounding factors. We introduce two novel approaches to assess DEG: extended Bayesian zero-inflated negative binomial factorization (ext-ZINBayes) and single-cell differential analysis (SIENA). In addition, we benchmark the proposed methods with known DEG analysis tools for single-cell and bulk RNA data, using two real public datasets. One contains house mouse cells of two different types, while the other gathers human peripheral blood mononuclear cells divided into four types. The results show that the two procedures can be competitive with existing methods (scVI, SCDE, MAST, and DEseq) in identifying relevant putative biomarkers. In terms of scalability and correctness, SIENA stands out from ext-ZINBayes and some of the existing methods. As single-cell datasets become increasingly larger, SIENA may emerge as a powerful tool to discover functional differences between two conditions. Both methods are publicly available (<https://github.com/JoanaGodinho>).

Keywords: differential expression, scRNA-seq, latent variable models, variational inference

1. Introduction

Gene expression is a fundamental biological process that affects how each living organism operates. As such, studying and understanding gene expression leads to a broader knowledge of how cells work and how they evolve. With this knowledge, groundbreaking advances can be achieved in the fields of genetics, molecular biology, and medicine.

One of the most relevant tasks performed through gene expression assessment is the identification of differentially expressed genes (DEG). DEG are genes that show different expression levels across different types of cells. With DEG identification, we can deepen our understanding on cell differentiation, study disease phenotypes, and assess how certain treatments perform [14].

Research has provided several computational methods aiming to carry out such task. Initially, differential expression (DE) analysis was only performed using gene expression obtained from DNA microarrays. Then, technological advances empowered the emergence of RNA sequencing (RNA-seq) protocols to profile gene expression. In a first approach, DE analysis over bulk RNA data was performed using packages, such as Limma [17], that were initially designed to account for microarray input. However, due to differences between microarray and RNA-seq data, new methods, such as DEseq [1] and edgeR [16], were developed specifically for the latter.

In more recent years, single-cell RNA sequencing (scRNA-seq) has stood out from the previous two. The appeal for this kind of data is the possibility to perform detailed analysis with high-resolution data, given that gene expression is described by mRNA counts in individual cells. Nonetheless, the data is still subject to the presence of noise, which unfolds as extra variation and false zero counts, caused by dropout events, batch effects, stochastic gene expression, or variations in sequencing depth (or library size).

In order to prevent wrong conclusions, one must seek to disentangle correct biological information from the noisy data. One suitable approach is to use a latent variable model. Methods such as SCDE [12], MAST [7], and scVI [13] take this approach to identify DEG. However, there is a need for new techniques, since scRNA-seq datasets are becoming increasingly larger, making some of the existent methods inefficient.

In this work, we propose two new methods to perform differential expression analysis (DEA): ext-ZINBayes (extended Bayesian zero-inflated negative binomial) and SIENA (SIngle-cELL differeNtial Analysis). Both rely on a latent variable model and variational inference (VI). ext-ZINBayes adopts an existing model developed for dimensionality reduction, ZINBayes [6]. SIENA operates under a new latent variable model defined based on existing models. We benchmark their performances with other methods, using two public datasets.

In the following section, we first review latent variable models and inference concepts; then, we detail the workings of our methods (Section 2). Finally, we conclude this work with a performance analysis (Section 3) and outline some final remarks (Section 4).

2. Methods

As we previously mentioned, to build a scRNA-seq analysis method, one must account for the presence of confounding factors. Using a latent variable model has shown to be a reliable approach to separate the additional variability added by such factors. In a latent variable model, variables are either observed or unobserved (latent). The latent variables (Z) are responsible for capturing and describing hidden factors that influence the observed variables (X). So, in the single-cell RNA context, the observed variables would be the counts, and the latent would describe the confounding factors.

If we take a Bayesian perspective, all latent variables, including unknown parameters, should be inferred using Bayesian inference. Under this framework, we first define probabilities that reflect a priori beliefs we may have about the latent factors. Then, these beliefs are updated using the observations, which in turn generate a posteriori assumptions. This process can be carried out using the Bayes theorem, where $p(Z|X)$ reflects the a posteriori beliefs and $p(Z)$ the a priori,

$$p(Z|X) = \frac{p(X|Z)p(Z)}{p(X)}. \quad (1)$$

Finally, the hidden variables are inferred using the posterior, $p(Z|X)$. One can set them as maximum a posteriori (MAP) estimates or as expected values of $Z|X$.

However, for complex models, it may be impossible to obtain the exact posterior, because the marginal likelihood, $p(X)$, can be intractable. In some of these cases, approximate inference techniques, such as variational inference (VI), are required.

The main idea behind VI [3] is to find a tractable distribution $q(Z)$ that best approximates the posterior. To do so, it assumes that $q(Z)$ belongs to a family of distributions, defined by parameters v . So, from a deeper perspective, VI aims to find the parameter values which make $q(Z)$ closest to $p(Z|X)$. To evaluate the dissimilarity between the distributions, VI relies on the Kullback-Leibler (KL) divergence, calculated as follows:

$$\mathbb{E}_{q(Z)}[\log q(Z) - \log p(Z|X)], \quad (2)$$

where $\mathbb{E}_{q(Z)}$ is the expected value with respect to $q(Z)$. In this setting, finding the optimal v amounts to finding v which minimize equation (2).

However, the KL divergence involves the unknown posterior, thus an alternative metric is required. This metric is known as the Evidence Lower BOund (ELBO) and is derived from the KL divergence. The ELBO is calculated as follows:

$$\mathbb{E}_{q(Z)}[\log p(Z, X) - \log q(Z)]. \quad (3)$$

In this case, to find v one maximizes the ELBO.

As stated by Blei and colleagues [3], the performance of VI techniques is greatly influenced by the choice of the family Q . The most commonly used is the mean-field variational family, which assumes independence between all latent variables. As such, each unobserved variable follows a separate variational distribution. Then, for a set of N latent variables, $q(Z)$ can be obtained through

$$q(Z) = \prod_{j=1}^N q(Z_j). \quad (4)$$

The following subsections present two methods we developed using the techniques formerly described, with the goal to perform DE analysis. Both were developed in Python and are freely available online at <https://github.com/JoanaGodinho>.

2.1. ext-ZINBayes

The first method we present is an extension of an existing model, ZINBayes [6], developed to perform dimensionality reduction. With an additional feature, we enable it to detect DEG.

When designing ZINBayes, the aim was to create an approach able to discover a true biological representation of the data, without the distortion caused by noise factors. Thus, ZINBayes takes into consideration batch effects, dropout events, and stochastic gene expression.

The model is built upon a Gamma-Poisson mixture so that each count follows a Negative Binomial (NB) distribution. As research has shown, the NB is highly adequate to describe RNA-seq data due to its ability to account for overdispersion. However, it may not be sufficient to account for the excessive amount of zeros caused by dropout events. Therefore, the authors added zero-inflation (ZI) to the generative process.

For a given set of G genes and N cells, the count of each gene g in cell i is defined by variable X_{ig} , where $g = 1 \dots G$ and $i = 1 \dots N$. X_{ig} is either governed by the NB component or, in case of a dropout, is modelled as a constant zero. These conditional assignment is devised as follows:

$$\begin{aligned} \lambda_{ig} &\sim \text{Gamma}\left(\theta_g, \frac{\theta_g}{\rho_{ig}L_i}\right) \\ Y_{ig} &\sim \text{Poisson}(\lambda_{ig}) \\ D_{ig} &\sim \text{Bernoulli}(\pi_{ig}) \\ X_{ig} &= \begin{cases} Y_{ig} & \text{if } D_{ig} = 0 \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

where Y_{ig} generates the count's magnitude if D_{ig} indicates that X_{ig} is not a dropout. λ_{ig} parameterizes Y_{ig} and thus, corresponds to the mean expression of g in i .

L_i is a scale factor linked to the library size of cell i , i.e., the total amount of transcripts detected in cell i , while θ_g corresponds to a dispersion factor related to gene g . Both are seen as latent random variables,

$$\begin{aligned} L_i &\sim \text{Lognormal}(\mu_i, \sigma_i) \\ \theta_g &\sim \text{Gamma}(2, 1). \end{aligned}$$

The formulations ρ_{ig} and π_{ig} correspond, respectively, to the percentage of transcripts of gene g in cell i and the probability of X_{ig} being a dropout. These yield both cell-specific and gene-specific features:

$$\begin{aligned} C_i &= [Z_i, s_i] \\ \rho_{ig} &= \frac{C_i W_{0,g}}{\sum_g C_i W_{0,g}} \\ \text{logit}(\pi_{ig}) &= C_i W_{1,g}. \end{aligned}$$

The cell-related features are the batch and the K -dimensional biological signature of the cell (s_i and Z_i). The gene-related are the factor loadings $W_{0,g}$ and $W_{1,g}$. While s_i is a B -sized one-hot representation, with B being the number of batches, Z_i , $W_{0,g}$, and $W_{1,g}$ are collections of random variables whose components are modelled as follows:

$$\begin{aligned} W_{0,gk'} &\sim \text{Gamma}(0.1, 0.3) \\ W_{1,gk'} &\sim \text{Normal}(0, 1) \\ Z_{ik} &\sim \text{Gamma}(2, 1), \end{aligned}$$

where $k = 0, \dots, K$ and $k' = 0, \dots, K + B$. See [6] for more details about the hyperparameters choice.

Several of the variables above are also used in scVI and have the same purpose. However, scVI authors use very different parameterizations for some of them.

Given the model's definition, exact inference can not be performed due to the intractability of the posteriors. In addition, the model is not conditionally conjugate, making it impossible to use coordinate ascent variational inference (CAVI). As a result, the authors resorted to reparameterization gradients (RG).

To identify DEG between two cell subpopulations, we adopted the procedure developed in [13]. For each gene g , we define two hypotheses given a pair of cells from different populations. Both cells are from the same batch and have counts x_1 and x_2 :

$$H_a^g = \rho_{1g} > \rho_{2g} \text{ and } H_b^g = \rho_{1g} \leq \rho_{2g}. \quad (5)$$

The first hypothesis states that the percentage of transcripts of gene g in cell 1 is higher than in cell 2, while the second hypothesis translates into the opposite. Then, a Bayes factor, B , is calculated as follows:

$$B = \frac{p(H_a^g | x_1, x_2) p(H_b^g)}{p(H_b^g | x_1, x_2) p(H_a^g)}. \quad (6)$$

Its value quantifies the ratio between the likelihood probabilities given each hypothesis. High factors reflect stronger beliefs over H_a^g , while factors closer to

zero reflect more support over H_b^g . To simplify the assessment of the probability ratio, we consider the factor's logarithm and not its raw value. If the logarithm is negative, it means H_b^g is more prone to be true; if it is positive, it means the opposite: H_a^g is more likely correct. This implies that higher positive values yield higher supports over H_a^g , whereas lower negative values yield higher supports over the alternative hypothesis. Given that H_a^g and H_b^g are mutually exclusive and have equal prior probabilities, i.e., $p(H_a^g) = p(H_b^g)$, $\log(B)$ is calculated as follows:

$$\begin{aligned} \log(B) &= \log \frac{p(H_a^g | x_1, x_2)}{1 - p(H_a^g | x_1, x_2)} \\ &= \log(p(H_a^g | x_1, x_2)) - \log(1 - p(H_a^g | x_1, x_2)). \end{aligned} \quad (7)$$

To compute the posterior $p(H_a^g | x_1, x_2)$, the probabilities of all ρ_{1g} and ρ_{2g} pairs, which make H_a^g true, need to be summed. Given that the ρ values depend on Z_1 , Z_2 , and $W_{0,g}$, we need to integrate all possible combinations of the three that yield H_a^g true,

$$\begin{aligned} p(H_a^g | x_1, x_2) &= \iiint_{(w_{0,g}, z_1, z_2)} \mathbb{I}[\rho_{1g} > \rho_{2g}] q(\cdot), \\ q(\cdot) &= q(z_1) q(z_2) q(w_{0,g}). \end{aligned} \quad (8)$$

In the equation above, $q(z_1)$ and $q(z_2)$ correspond to the probabilities of cell 1 having a z_1 representation and cell 2 having a z_2 representation, $q(w_{0,g})$ corresponds to the probability of gene g having $w_{0,g}$ has its loading factors. Each of these probabilities is obtained through the corresponding variational distribution shaped during inference.

Since calculating the exact value of the integral is very computationally demanding, we used Monte-Carlo approximation. Thus, $p(H_a^g | x_1, x_2)$ is an empirical average of $\rho_{1g} > \rho_{2g}$ over a random set of triplets $(z_1, z_2, w_{0,g})$ sampled from the variational distributions:

$$p(H_a^g | x_1, x_2) \approx \frac{1}{|S|} \sum_{(w_{0,g}, z_1, z_2)} \mathbb{I}[\rho_{1g} > \rho_{2g}], \quad (9)$$

where $|S|$ is the total number of samples assessed.

This process is performed over all possible cell pairs that contain one cell from each of the two subpopulations under study. However, the ρ values are affected by s , which is responsible for specifying the batches. Thus, the process is only viable if all cells come from the same batch. When the counts come from two or more batches, each cell must be paired with another cell from the same batch but with different types/populations. If inter-batch pairs were allowed, the differences between the cells' ρ could be biased by batch effects, leading to erroneous Bayes factors.

After calculating the factor's logarithm of each pair, the obtained values are averaged, and the resulting mean

is used as a score of differential expression. If the absolute value of the average is higher than a certain threshold, gene g is classified as a DEG. The threshold used is customizable, but we recommend setting between 2 and 3 [11] since it translates into having one of the hypotheses approximately 7 to 20 times more probable than the opposite one.

To scale this procedure to very large datasets, the method enables the use of a subset of cell pairs. To do so, the user needs to instruct the number of pairs to be selected. If the dataset contains cells from only one batch, we simply randomly peek the specified number of pairs. On the other hand, if the dataset gathers multiple batches, the proportion of pairs from each batch in the subset is equal to the proportion of each batch in the original dataset. For instance, a given dataset contains a thousand cells where 300 are from batch 1 and 700 are from batch 2. If the user specifies a subset of 100 pairs, this means that 30 of those pairs are from batch 1, whereas the other 70 are from batch 2, thus keeping the original proportions.

2.2. SIENA

For our second proposed method, we designed a new latent variable model, where each count follows a zero-inflated NB (ZINB) distribution. As we mentioned before, with a ZINB distribution, one can depict the overdispersion and the excess of zero entries, typical of scRNA data. Like in ZINBayes, the NB is built through a Gamma-Poisson mixture.

We decided to adopt several variables used both in ZINBayes and in scVI, making our model able to account for noise factors such as different library sizes, dropouts, and stochastic gene expression. The major difference is the removal of indicators s and variables Z , which specify the batches and the low dimensional representations of each cell's biological features. Below we present the model, where X_{ig} reports the number of reads mapped to gene g in cell i :

$$\begin{aligned} L_i &\sim \text{Lognormal}(\mu_i, \sigma_i) \\ \beta_{ig} &\sim \text{Gamma}\left(\frac{1}{3}, 1\right) \\ \rho_{ig} &= \frac{\beta_{ig}}{\sum_g \beta_{ig}} \\ \lambda_{ig} &\sim \text{Gamma}(\theta_g L_i \rho_{ig}, \theta_g) \\ Y_{ig} &\sim \text{Poisson}(\lambda_{ig}) \\ D_{ig} &\sim \text{Bernoulli}(\pi_{ig}) \\ X_{ig} &= \begin{cases} Y_{ig} & \text{if } D_{ig} = 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

On one hand, variables L_i , ρ_{ig} , D_{ig} , and λ_{ig} encode the same as in ZINBayes. L_i is a scaling factor, ρ_{ig} is the percentage of gene g transcripts in cell i , λ_{ig} is the expression mean, and D_{ig} indicates if count X_{ig} is a dropout.

On the other hand, θ_g , the gene's dispersion factor, is not seen as a random variable, but as a non-random model parameter, and π_{ig} , the probability of a dropout event, is defined as a hyperparameter. Nonetheless, these are not the only differences between this model and ZINBayes's.

Similarly to what was done in [13], L_i is drawn from a log-normal where the mean and variance of the underlying normal, μ_i and σ_i , are set, respectively, as the mean and the variance of the log scaled sequencing depths/library sizes considering only cells from the same batch as cell i . In ZINBayes, μ_i and σ_i are the mean and variance of the log library sizes considering all cells. The choice to model L_i as a log-normal is to restrict its domain to be positive since it's a scaling factor. Note that L_i encodes a factor proportionally related to the sequencing depth; it is not the actual sequencing depth, as pointed out in [13].

As an alternative, we also tested L_i as a Gamma, where its mean and variance are equal to the mean and variance of the library sizes in i 's batch.

Regarding ρ_{ig} , they are set as the ratio between a factor related to gene g and cell i , β_{ig} , and the sum of cell i factors with each gene. We take this formulation to not only restrict ρ values to be between 0 and 1 but also to constraint the sum of all ρ_{ig} of a given cell to be 1, i.e., $\sum_g \rho_{ig} = 1$. Both of these conditions need to be imposed because ρ_{ig} reflects a percentage, which translates into a relative frequency. An alternative approach would be to model ρ as a Beta distribution. However, using a Beta does not fit ρ properly since it only complies with the domain constraint. Moreover, given that no biological representation is defined for each cell, the biological variability is implicitly described by variable ρ . Notwithstanding, each ρ may be affected by the cell's batch since no batch-specific variable is modelled.

For the latent factors β_{ig} , we chose to posit a Gamma with $\alpha = \frac{1}{3}$ and $\beta = 1$ because it leads to a distribution where most of its probability density is placed near zero, yet its expected value is $\frac{1}{3}$. Due to its tail, this Gamma generates, in each cell, very low factors for most genes, but higher factors for a restricted set. In theory, this set is composed by cell i highly expressed genes.

As mentioned before, the NB is attained through a Gamma-Poisson mixture determined by variables λ_{ig} and Y_{ig} , according to the following:

$$\begin{aligned} &\text{If } X \sim \text{Poisson}(\lambda) \\ &\text{and } \lambda \sim \text{Gamma}(r, \frac{1-p}{p}) \\ &\text{then } X \sim \text{NB}(r, p). \end{aligned} \tag{10}$$

In this formulation, the NB output is defined as the number of successes until r failures occur, given a p probability of success. As a result, its expected value is $\frac{rp}{1-p}$. This is the NB formulation taken in our model. When deciding the parameters of λ_{ig} 's Gamma, we aimed to

fix the NB expected value as $L_i \rho_{ig}$. By defining the shape as $\theta_g L_i \rho_{ig}$ and rate as θ_g we achieve that.

Finally, zero-inflation is employed by variable D_{ig} , which determines if X_{ig} is necessarily zero. D_{ig} is drawn from a Bernoulli distribution, since D_{ig} only needs to take two values, one indicating dropout occurrence and another one stating non-occurrence. The probability of the Bernoulli, π_{ig} , is set as the proportion of zero entries of gene g over all cells from the same type and batch as cell i . For instance, if 60% of gene g counts in type A and batch 1 cells are zero, then π_{ig} is set as 0.6 for all type A and batch 1 cells.

Regarding inference, we use reparameterization gradients. We resort to VI because the counts' marginal likelihood is intractable, so exact inference can not be applied. In addition, the model is not conditionally conjugate, so CAVI can not be implemented. However, to use RG, variable D_{ig} needs to be discarded since it is not differentiable. As such, instead of defining X_{ig} with a conditional assignment, we set it as a mixture of two components: one is the NB while the other models the zero-inflation, generating only zeros. The ZI part is determined by a Deterministic distribution, which takes only the value zero, so all its density is placed over that value. Given that in a ZI model, zeros can be generated from the two components, the likelihood of X_{ig} is calculated as follows:

$$p(X_{ig}|\beta_i, L_i) = \pi_{ig} \times p_{\text{Det}}(X_{ig}) + (1 - \pi_{ig}) \times p_{\text{NB}}(X_{ig}|\beta_i, L_i) \quad (11)$$

As we can see from the equation above, we manage to also integrate out λ_{ig} and Y_{ig} , due to Equation (10). As such, the RG mechanism merely has to find a distribution q , which approximates $p(\beta_i, L_i|X_i)$. The variational distribution $q(\beta_i, L_i)$ is considered mean-field, and thus can be factorized in $q(\beta_i)$ and $q(L_i)$; $q(\beta_i)$ can be further factorized into $\prod_{i=1}^G q(\beta_{ig})$. Both $q(L_i)$ and $q(\beta_{ig})$ distributions are assumed to be log-normal since β_{ig} and L_i are positive variables, and RG performs better when it has to optimize normal distributions.

We build two inference networks: one outputs $q(\beta_i)$ parameters, i.e., the mean and variance of each of its $q(\beta_{ig})$; the other generates the mean and variance for $q(L_i)$. With this approach, we are able to scale inference to very large datasets, since optimization is carried only over global variables, the weights, instead of local variables, the means and variances.

Each network has one hidden layer with 128 nodes, and its output layer has two heads, one for the mean(s) and another one for the variance(s). A softplus transformation is applied over the variance head, to restrict it to be positive. In the hidden layer, a batch normalization step is employed before activation. In addition to the neural networks, memory-wise scalability is improved via batch training where, in each iteration, we break the full dataset into several subsets with equal size and use each one to do an update step.

Regarding θ_g optimization, we iteratively set it as a Maximum Likelihood Estimation (MLE), after one update step over the networks' weights. Therefore, after inference, θ_g will have a value that maximizes the likelihood of the counts given the obtained optimal variational parameters.

To assess if a given gene g is a DEG, we apply the same procedure as the one used in ext-ZINBayes. Given a cell pair, we define two exclusive hypotheses like the ones in Equation (5). Then, log scaled Bayes factors are calculated for each cell pair, and the absolute value of their average is used as a metric to classify g as a DEG or a non-DEG. The difference from the ext-ZINBayes procedure is the calculation of $p(H_a^g|x_1, x_2)$, since, in this approach, ρ_{ig} only depends on β_i . Consequently, it is only necessary to integrate all possible combinations of β_1 and β_2 that make H_a^g true:

$$p(H_a^g|x_1, x_2) = \iint_{(\beta_1, \beta_2)} \mathbb{I}[\rho_{1g} > \rho_{2g}] q(\beta_1) q(\beta_2). \quad (12)$$

This integral is also approximated through Monte Carlo, where the samples are drawn from β_1 and β_2 variational distributions.

Given that, in our model, we do not specify any variable identifying each cell's batch, the ρ values will be tampered by batch effects. To overcome this, we only pair cells that come from the same batch, just like in ext-ZINBayes. This way, the differential expression analysis is more truthful to biological differences. Furthermore, we scale the Bayes factor calculation by providing the optional use of a subset of pairs.

3. Results

To assess the performance of ZINBayes and SIENA, we used two known real scRNA-seq datasets: Islam and PBMC (Peripheral Blood Mononuclear Cells). Since none of the datasets has the genes identified as being DEG or not, we considered as ground truth the ones detected in the corresponding microarray dataset using Limma. This is a similar procedure to the one applied in [4, 12]. We also assessed their performances over synthetic data. However, we only present the resulting analysis in the complete document of the thesis.

The Islam dataset was gathered in the study [9] and contains expression counts of 92 embryonic cells of the house mouse: 48 Embryonic stem (ES) cells and 44 Embryonic fibroblast (MEF) cells. The PBMC is a droplet-based dataset that contains count data of human peripheral blood mononuclear cells, which were sequenced in two different batches. The cells are divided into four different types, where 4996 are CD4+ T cells, 1448 are CD8+ T cells, 1621 are B cells, and 339 are Dendritic cells, which amount to a total of 8404 cells.

Regarding the gold standard results, we used the microarray dataset Moliner [15] for the comparison between ES and MEF cells and two microarray datasets

of PBMC, one for the CD4+T vs. CD8+T analysis and the other for the B vs. Dendritic analysis. To obtain the Islam and the two PBMC microarray datasets (CD4+T vs. CD8+T and B vs. Dendritic), we used the GEO database [5] using the codes GSE29087, GSE8835, and GSE29618, respectively. The single-cell PBMC dataset is a subset of the one used in [13]. For Moliner, we extracted the data from the .CEL files¹ used in [4].

As a preprocessing step, we filtered out the genes in the single-cell datasets that were not in the corresponding microarray datasets and vice-versa. In addition, genes for which there was no information about their length were also removed, since MAST, one of the benchmarking methods, implements a TPM (Transcripts per Million) normalization, that requires the length. As such, DE analysis between types ES and MEF was carried out over 6757 genes, while for the CD4+T vs. CD8+T and B vs. Dendritic analyses, only 3346 genes were evaluated.

In the following subsections, we first assess the effects of different settings of SIENA, and ext-ZINBayes, then we benchmark their performances with existing methods: SCDE, MAST, scVI, and DEseq. The first three were designed specifically for scRNA data, whereas DEseq is used for both bulk and single-cell data. To run MAST and DEseq, we used the corresponding R packages available on the Bioconductor project. For SCDE, we used the R implementation² provided by the authors, and for scVI, we used the Python release 0.3.0³. Finally, we compare the biological conclusions drawn from each methods' rank through a gene set enrichment analysis (GSEA), where we compare the Gene Ontology [2] (GO) and KEGG [10] (Kyoto Encyclopedia of Genes and Genomes) pathway enrichments.

Configurations Inspired by what the authors in [18] concluded, we decided to evaluate how the zero-inflation affected SIENA and ext-ZINBayes. They state that, to model droplet scRNA data, the NB is sufficient. As such, we first discuss the performance of both methods with the Islam dataset, using a simple NB or the zero-inflated version. Simultaneously, we test if the use of the gene dispersion factor improves the results. Figure 1 summarizes this analysis.

The plots show the average area under the ROC curve (AUC) for each method's configuration: ZINB with dispersion, NB with dispersion, ZINB without dispersion, and NB without dispersion. For each configuration, we conducted 30 runs of 1000 epochs and averaged the resulting AUC scores. For SIENA, the no zero-inflation plus gene dispersion combination yields the best average AUC; however, it has the highest variance. The configurations ZI plus dispersion and ZI plus no dispersion have the lowest average. For ext-ZINBayes, em-

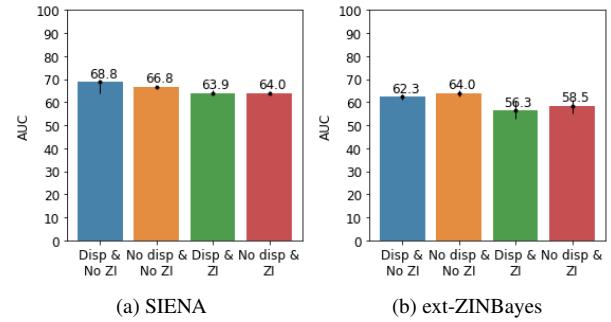


Figure 1: Average AUC values for each configuration with the Islam dataset. Black bars indicate the maximum and minimum AUC achieved.

ploying a simple NB combined with no dispersion leads to a higher average AUC, and, like for SIENA, the ZI plus dispersion and ZI plus no dispersion configurations prompt the worst AUC. Yet, in ext-ZINBayes, the difference between these two configurations is more accentuated, with the former standing out as the worst. We also checked how each combination behaved with the B vs. Dendritic test from the PBMC dataset and verified what we partially concluded from Figure 1: SIENA achieves a better mean AUC with the no ZI plus dispersion combination, while ext-ZINBayes has a better mean without both ZI and dispersion. Apart from the mean difference between SIENA's ZI configurations, all the contrasts between the plotted mean AUC are statistically significant (Welch's t-tests with p -values < 0.01). Therefore, one can conclude that for real datasets, both methods achieve better AUC without zero-inflation. However, SIENA requires the use of the gene dispersion factor, whereas ext-ZINBayes does not.

For the SIENA configurations assessed, we adopted a log-normal distribution to model library scalings. However, as we mentioned in Section 2, we employed two alternatives for the library scalings, one using a log-normal distribution and another using a Gamma. From Figure 2, we see that the log-normal alternative is slightly more robust, hence our choice.

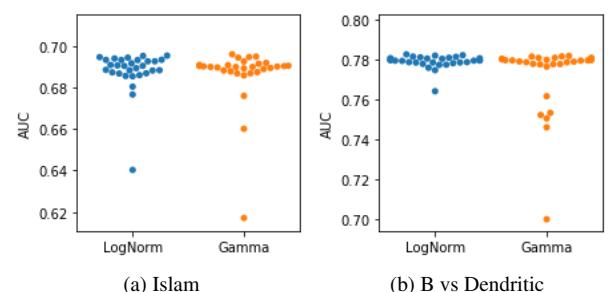


Figure 2: AUC values for each SIENA library size alternative. Each dot corresponds to one run.

¹http://carlosibanezlab.se/Data/Moliner_CELfiles.zip

²<https://hms-dbmi.github.io/scde/package.html>

³<https://github.com/YosefLab/scVI/releases>

The plots summarize the AUC values of 30 runs with each SIENA alternative, leveraging the NB plus gene dispersion configuration. Both options have similar AUC results with the Islam dataset, while in the B vs. Dendritic analysis, the log-normal leads to less dispersed AUC scores. In fact, the difference between the average AUC in the B vs. Dendritic comparison is statistically significant (Welch's t-test with p -value = 0.048). However, it does not seem to be considerably high, since the resulting confidence interval at 95% is between 0.016% and 1.3%.

Given that SIENA resorts to batch training, we also used the PBMC dataset to assess how the mini-batch size affects the performance. Although a larger number of batches translates into less data used in each update step, the detection accuracy is practically unaffected by such variation.

Benchmark To start our comparison analysis, we first contrast our methods' and the four mentioned DE procedures' ability to identify DEG using the Islam dataset. Out of those four, only MAST and DEseq are deterministic. Similarly to what was done in the previous section, we use as a benchmark measure the average AUC. The results are shown in Figure 3.

To generate the bar plot, we ran and calculated the AUC of MAST and DEseq only one time, whereas, for the other methods, we repeated the process 50 times and averaged the AUC values. Both scVI and SIENA were run with gene dispersion and without ZI since this configuration leverages better results, and each run had 1000 epochs. ext-ZINBayes was also operated without ZI and with 1000 epochs per run, but unlike for SIENA and scVI, no dispersion factor was adopted.

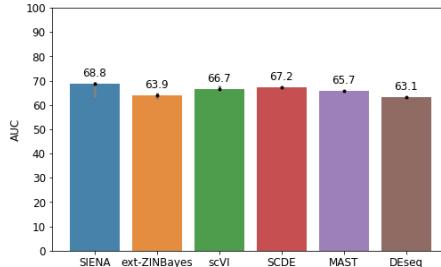


Figure 3: Average AUC values for each method with the Islam dataset (ES vs. MEF analysis). Grey bars indicate the maximum and the minimum AUC obtained.

As seen in Figure 3, with the Islam dataset, SIENA yields better results showing an average AUC close to 69%, while DEseq has the lowest average out of all the methods. SCDE has the second best AUC score, followed by scVI, MAST, and ext-ZINBayes. Nonetheless, SIENA presents a higher variation (around 5%), given that two runs generated an AUC of approximately 64%. The differences between the average AUC are actually statistically significant since all Welch's t-tests be-

tween two methods mean AUC show p -values smaller than 0.01. Note that in this assessment, we did not consider a t-test between MAST and DEseq since these two methods are deterministic.

With the average AUC, we can test classification accuracy and robustness, yet it is also pivotal to assess how each method scores the genes, i.e., how certain they are that a given gene is a DEG. As such, in Figure 4, we compare for each gene in the Islam dataset, the DE metrics of each method with the p -values obtained by Limma. Note that the p -values are adjusted to the false discovery rate (FDR). For SIENA, ext-ZINBayes, scVI, and SCDE, we plot the metrics median of each gene considering the 50 runs conducted for the previous analysis. So, for SCDE, we show the absolute Z -score's median, while for the other three, we outline the median of the Bayes factor's logarithm. For MAST and DEseq, we only consider FDR adjusted p -values of one run.

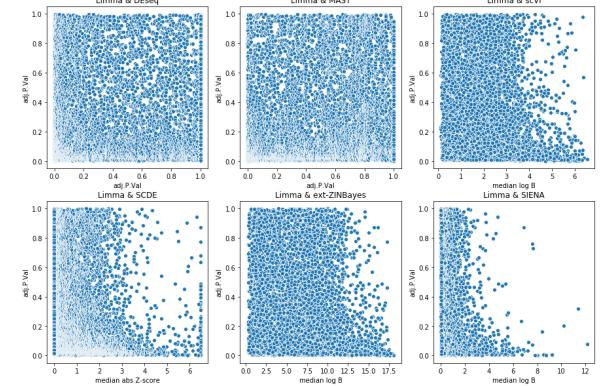


Figure 4: Comparison between each method's DE metrics and the adjusted p -values of Limma. Each blue dot corresponds to a gene.

In this comparison, MAST and DEseq present the worst results since there is no visible relation between their p -values and those obtained by Limma. Ideally, there should be a positive correlation. The other four methods have a more distinct correlation with Limma since, at a certain threshold, the p -values tend to be lower as the absolute Z -scores or log Bayes factors increase. For instance, in SCDE, genes with absolute Z -scores higher than ≈ 2.5 tend to show lower p -values as the score increases. The same happens for scVI, ext-ZINBayes, and SIENA for genes with log Bayes factors higher than ≈ 3.5 , ≈ 12 , and ≈ 2 , respectively. Nevertheless, SIENA seems to show a better correlation with Limma than the others.

We also used the PBMC dataset to evaluate the methods average AUC, applying the same methodology. We also employed the same configurations for SIENA, ext-ZINBayes, and scVI, but we set only 500 epochs per run for SIENA and scVI. We used less iterations because the PBMC dataset has more entries (cells) than the Islam dataset, making optimization of global parameters

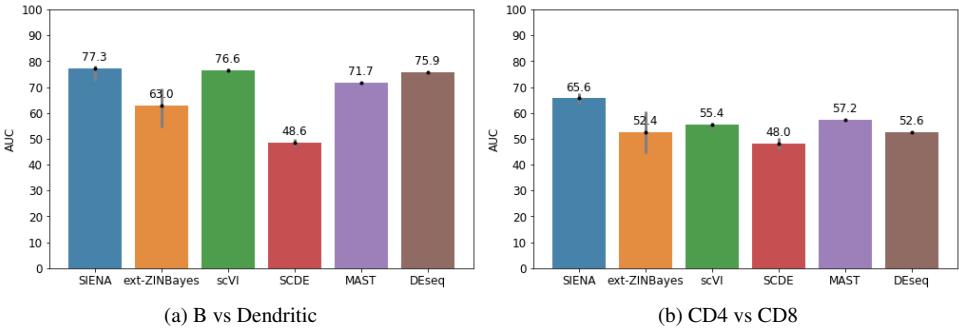


Figure 5: Average AUC values for each method with the PBMC dataset. Grey bars indicate the maximum and the minimum AUC achieved.

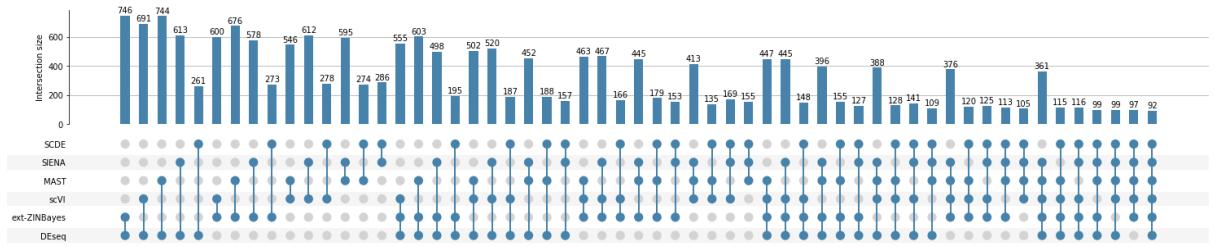


Figure 6: Intersections of the top 1000 DEG regarding the CD4 vs. CD8 analysis. Matrix dots specify the method combinations, and the bars encode the number of DEG in common of the corresponding combination.

require less update steps.

As seen in Figure 5a, all methods, except SCDE and ext-ZINBayes, present a higher average AUC when conducting DE analysis between B and Dendritic cells than between ES and MEF cells. SCDE is the only that shows a great decrease in performance, having an average AUC lower than 50%, whereas SIENA stands out as the best with an average AUC of 77.3%. scVI and DEseq also perform well, showing a mean AUC of 76.6% and 75.9%, respectively. Unlike in the ES vs. MEF test, ext-ZINBayes shows the highest variance. Regarding the CD8 vs. CD4 comparison (Figure 5b), SIENA obtains the best mean AUC (65%), while all the other methods perform considerably worst, having an average AUC lower than 60%. Once again, SCDE shows the worst AUC. Following SIENA, MAST and scVI show, respectively, the second and third best average AUC, while ext-ZINBayes and DEseq come in fourth with an average AUC of approximately 52%. Nonetheless, like in the B vs. Dendritic test, ext-ZINBayes shows the highest variance in the results. Note that for this comparison, for both SIENA and ext-ZINBayes, the log Bayes factors were calculated using a subset of valid cell pairs. More specifically, 7.5×10^5 pairs were used, and, for each of those pairs, 100 samples of ρ values were computed. Furthermore, some SCDE runs had to be re-executed because sometimes the method could not fit a model for a specific cell.

In both PBMC tests, the obtained AUC are more divergent than the ones gathered in the ES vs. MEF test.

While in the latter, the difference between the average AUC of the best method and the worst is slightly lower than 6%, in the B vs. Dendritic and CD4 vs. CD8 tests, the difference is around 30% and 20%, respectively.

Almost all mean AUC differences are statistically significant for both PBMC comparisons, yielding Welch's t-tests with p -values lower than 0.01. The only difference that has no support is between ext-ZINBayes and DEseq in the CD4 vs. CD8 test (p -value = 0.075).

Given the poor results under the CD4 vs. CD8 test, we deepen our comparative analysis over the test with an intersection graph in Figure 6, without considering the microarray ground truth, i.e., the results from Limma. To generate the plot, we considered the 50 runs of SIENA, SCDE, scVI, and ext-ZINbayes conducted for the AUC analysis, and for each method, we calculated the median DE score of each gene. Then, we gathered the top 1000 genes with the highest medians. For MAST and DEseq, we gathered the top 1000 genes with the lowest FDR adjusted p -values of only one run.

From the plot, we can see that ext-ZINBayes and DEseq have a lot of genes in common, almost 750, which was expected given that the two methods have essentially the same average AUC. In fact, the pair has the largest intersection set out of all duos. DEseq has also more genes in common with SIENA than any of the other methods. This is curious given that MAST and scVI show average AUC closer to SIENA's. Furthermore, all two method combinations considering SCDE have the lowest number of genes in common, when

compared with the other two methods combinations. The same happens for three, four, and five method combinations. Moreover, if we consider all methods except SCDE, the number of genes in common goes from 92 to 361; it increases almost four times, whereas if one of the other methods is not considered, it only increases to values between 97 and 116. The only method that comes close to identify the same DEG as SCDE is SIENA, yet the number of genes in common is just a bit over 25%.

Time Beyond detection accuracy, it is also important to evaluate how the methods perform in terms of time usage, and how that usage scales as the number of cells increases. DEseq stands out as the fastest method in the real data analysis, taking less than 30 seconds. Of the two proposed methods, SIENA is the one that takes less time, yet it is still outrun by MAST and DEseq. Nonetheless, if we disable the use of gene dispersion, SIENA becomes faster since it has one less set of parameters to optimize. In fact, if the dataset contains more cells than genes, SIENA shows better times, since it has less dispersions to optimize.

Like scVI, SIENA and ext-ZINBayes are suitable to operate under GPUs, since both are compatible with the tensorflow-gpu library. This helps to increase their speed on machines with less processing power, allowing the methods to reach good performances in consumer-grade machines.

GSEA After gathering a DE ranking, the next step in any DE analysis is to perform enrichment analysis to extract biological meaning from the list. As such, it is important to compare the biological features outlined by each method’s list. To do so, we used the STRING [19] platform available online⁴, to compare the gene ontologies and KEGG pathways enriched by the top DEG of each method under the B vs. Dendritic analysis. For SIENA, ext-ZINBayes, scVI, and SCDE, we calculated the median DE score of each gene over the 50 runs considered for the plot in Figure 5a. Then, we used the medians to rank the genes in descending order. For MAST and DEseq, we used the rank list of one run.

For the ground truth (Limma’s) rank, 18 GO terms were considered significantly enriched, i.e., had an enrichment FDR corrected p -value lower or equal to 0.01. SIENA’s rank led to 73 enriched terms, ext-ZINBayes to 62, DEseq to 56, MAST to 41, and scVI to 31. Figure 7 illustrates for each method, the enrichment score of a set of GO terms. The outlined set corresponds to the union of the 10 most significantly enriched terms by each method’s ranking. From the heatmap, we can see that the top 10 GO terms for SIENA and ext-ZINBayes are the same as for the ground truth list. However, the scores related to ext-ZINBayes are greatly higher. Even though DEseq has one different enriched term, it has a closer

score signature to the ground truth than ext-ZINBayes. Out of all methods, scVI shows the most divergent GO pattern. Notwithstanding, all methods seem to detect DEG connected to biological terms such as myeloid leukocyte activation and leukocyte/myeloid cell activation involved in immune response, which means that the differences between B and Dendritic cells are probably related with such processes.

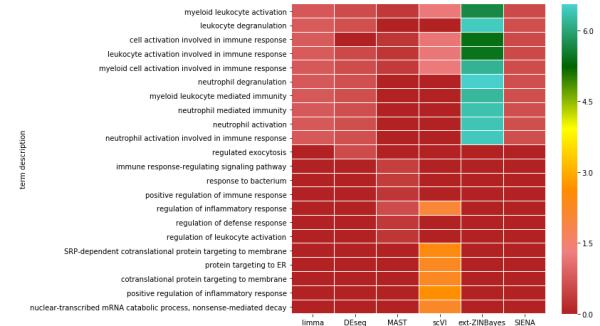


Figure 7: Gene Ontology enrichment analysis for each method under the B vs. Dendritic test. Each term considered is one of the 10 most significantly enriched terms of at least one method.

Discussion In both SIENA and ext-ZINBayes, configurations without zero-inflation lead to better results in the two real datasets. This contrasts with what research has assumed throughout the years. Nonetheless, as we stated before, the authors in [18] disproved this assumption for droplet-based data, thus supporting our findings regarding the PBMC dataset. Even though the authors affirm that in the case of plate-based counts zero-inflation mechanisms are necessary, our results with the Islam counts may refute such conclusions, since that dataset has probably a plate-based origin due to its small number of cells (< 100).

Comparing to existing methods, SIENA was able to detect more accurately the DEG in both PBMC and Islam analysis. In addition, SIENA exhibited the most consistent behaviour over the three real data tests, showing average AUC ranging from 65% to 78%. All the other methods presented more fluctuating performances when dealing with different types of datasets. This means that SIENA is more adequate to deal with both small and large datasets than some state-of-the-art methods. Moreover, SIENA is able to scale its memory usage during both inference and DE test computation without decreasing its overall accuracy. This is important given that, in the past years, single-cell datasets have exponentially grown in size.

Contrary to SIENA, ext-ZINBayes is unable to surpass the benchmarking methods over the real data tests. In fact, in all three tests, it yields the second worst mean AUC. Out of all methods assessed, ext-ZINBayes is the most unstable, reaching the highest AUC variations

⁴<https://string-db.org>

(around 15%) in both B vs. Dendritic and CD4 vs. CD8 tests. Given the non-convexity of the loss function (ELBO), these high variations may be due to the methods inability to escape local minima when dealing with large datasets, making the methods variational parameters converge to different results in each run. Nonetheless, with the Islam dataset, ext-ZINBayes shows a consistent behavior, whereas SIENA reaches its highest variation, which may also be due to local minima. Regarding the optimal configuration, ext-ZINBayes performs better without considering the gene dispersion factor (θ_g). However, in both scVI and SIENA, the dispersion factor improves the results. As such, we believe that a different prior or even a different formulation over θ_g can boost ext-ZINBayes results.

In terms of time, only SIENA is able to outrun current methods in certain settings, while ext-ZINBayes is consistently one of the slowest two. This confirms what we stated before that the use of inference networks speeds up the inference process since only global variables are optimized. Nonetheless, in the context of scRNA-seq analysis, this gain is not sufficient to make VI-based methods competitive with some alternative probabilistic procedures, when it comes to time consumption.

Of the two proposed methods, SIENNA shows overall rankings more correlated with the ground truth rankings. This is not only supported by the enrichment analysis but also by the score correlations obtained with the Islam dataset (Figure 4). Actually, comparing to other methods, SIENA shows DE scores more in line with Limma’s p -values. Moreover, in pair with DEseq, SIENA leads to biological conclusions closer to the ones drawn from the ground truth list.

Taking all this into account, we can conclude that only SIENA is able to compete with state-of-the-art procedures, managing to assemble more truthful differential expression scores in a more feasible amount of time.

4. Conclusion

We proposed two new Bayesian probabilistic procedures to assess differential expression. Both are built upon latent variable models and variational inference mechanisms. ext-ZINBayes adopts an existing probabilistic model (ZINBayes) designed for dimensionality reduction. It performs differential analysis using some of the model’s latent variables. SIENA devises a novel model, leveraging certain assumptions taken in state-of-the-art methods. Of the two procedures, SIENA yields the best results both in terms of correctness and resource consumption. In fact, SIENA is very competitive with existing differential analysis approaches.

Both methods could benefit from time improvements. For instance, ext-ZINBayes can be upgraded with the use of inference networks, whereas a new gene dispersion optimization mechanism may speed SIENA’s inference. Another potential future work would be to integrate SIENA with some batch removal method de-

signed specifically for scRNA data, in order to compute the Bayes factors without constraining the cell pairs by batch. One option is to employ batch correction by matching mutual nearest neighbors [8]. Finally, both models can, in principle, be used to devise some fold-change metric, which can, in turn, be combined with the Bayes factor, possibly generating more accurate differential expression scores.

Acknowledgements

This work was partially supported by national funds through FCT through projects PTDC/EMS-SIS/0642/2014, PTDC/CCI-CIF/29877/2017, PTDC/EEL-SII/1937/2014, UID/EEA/50008/2013, UID/CEC/50021/2019, UID/EMS/50022/2019.

References

- [1] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, 2010.
- [2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000.
- [3] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, feb 2017.
- [4] A. Dal Molin, G. Baruzzo, and B. Di Camillo. Single-Cell RNA-Sequencing: Assessment of Differential Expression Analysis Methods. *Frontiers in Genetics*, 8, May 23 2017.
- [5] R. Edgar. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, Jan. 2002.
- [6] P. F. Ferreira, A. M. Carvalho, and S. Vinga. Scalable probabilistic matrix factorization for single-cell RNA-seq analysis. dec 2018.
- [7] G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. K. Shalek, C. K. Slichter, H. W. Miller, M. J. McElrath, M. Prlic, P. S. Linsley, and R. Gottardo. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(1), dec 2015.
- [8] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421, 2018.
- [9] S. Islam, U. Kjallquist, A. Moliner, P. Zajac, J.-B. Fan, P. Lonnerberg, and S. Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research*, 21(7):1160–1167, may 2011.
- [10] M. Kanehisa and S. Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.
- [11] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.
- [12] P. V. Kharchenko, L. Silberstein, and D. T. Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, may 2014.
- [13] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053, 2018.
- [14] J. C. Mar, N. A. Matigian, A. Mackay-Sim, G. D. Mellick, C. M. Sue, P. A. Silburn, J. J. McGrath, J. Quackenbush, and C. A. Wells. Variance of gene expression identifies altered network constraints in neurological disease. *PLoS genetics*, 7(8):e1002207, 2011.
- [15] A. Moliner, P. Ernfors, C. F. Ibáñez, and M. Andäng. Mouse embryonic stem cell-derived spheres with distinct neurogenic potentials. *Stem Cells and Development*, 17(2):233–243, apr 2008.
- [16] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, nov 2009.
- [17] G. K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1–25, jan 2004.
- [18] V. Svensson. Droplet scRNA-seq is not zero-inflated. *bioRxiv*, 2019.
- [19] D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2018.