Bayesian modelling to assess differential expression from single-cell data

Joana Godinho[1,*], Alexandra M. Carvalho[1] and Susana Vinga[2]

[1]Instituto de Telecomunicações and [2]INESC-ID
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[*]joana.godinho@tecnico.ulisboa.pt

August 2019

Abstract

In the field of molecular biology, genomic signature analysis is a powerful mean to unravel obscured cellular aspects. One of the main tasks in gene expression analysis is the exploration of transcriptomic data, which enables the recognition of genes that are differentially expressed (DEG). Disease profiling, treatment development, and the identification of new cell populations are some of the most relevant applications that rely on DEG identification. In this context, three leading technologies emerged; namely, DNA microarrays, bulk RNA sequencing (RNA-seq), and single-cell RNA sequencing (scRNA-seq), being the last the main focus of this work. Although it tends to offer more accurate data than the other two, it is still limited by many confounding factors. We introduce two novel approaches to perform DEG identification over single-cell data: extended Bayesian zero-inflated negative binomial factorization (ext-ZINBayes) and single-cell differential analysis (SIENA). Both techniques rely on latent variable models to account for the misleading factors present in the data and resort to variational inference mechanisms to ascertain model components. In addition, we benchmark the proposed methods with known DEG analysis tools for single-cell and bulk RNA data. To do so, we used two real public datasets that have been used in past assessment tasks. One contains house mouse cells of two different types, while the other gathers human peripheral blood mononuclear cells divided in four types. The results show that the two procedures can be very competitive with existing methods in identifying relevant putative biomarkers. In terms of scalability and correctness, SIENA stands out from ext-ZINBayes and some of the existing methods. Thus, as single-cell datasets become increasingly larger, SIENA may emerge as a powerful tool to discover functional differences between two conditions.
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 broaden knowledge on how cells work and how they evolve. With this knowledge, ground breaking advances can be achieve in the fields of genetics, molecular biology and even 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 [13] and assess how certain treatments perform.

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 [16], 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 [15], 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 (a.k.a. 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 [10], MAST [6] and scVI [12] 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 as we will see further.

With 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 [5]. 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 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 are responsible for capturing and describing hidden factors that influence the observed variables. 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, i.e., if we assume that each latent variable follows a given probabilistic distribution, they can be inferred using Bayesian inference.

In this technique, we first define probabilities that reflect à priori beliefs we may have about the latent factors. Then, these beliefs are updated using the observations which in turn generate à posteriori assumptions.

In statistical terms, this process can be translated into the Bayes theorem where $p(Z|X)$ reflects the à posteriori beliefs and p(z) the à priori beliefs.

$$p(Z|X) = \frac{p(X|Z)p(Z)}{p(X)}, \tag{1}$$

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

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

The main idea behind VI [2] is to find a distribution $q(Z)$ that best approximates the posterior. To do so, it assumes that $q(Z)$ belongs to a family of distributions, Q, defined by parameters v. So, in a deeper perspective, VI aims to find the parameters v which make $q(Z)$ closest to $p(Z|X)$. To measure the dissimilarity between the two distributions, VI relies on the Kullback-Leibler (KL) divergence, calculated as follows:

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

where $\mathbb{E}_{q(Z)}$ is the expected value with regards 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 using the equation below:

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

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

As stated in [2], 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, given a set of N latent variables, $q(Z)$ can be obtained through,

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

The following subsections, present two methods we developed using the techniques formerly described, with the goal to perform DE analysis.

2.1. ext-ZINBayes

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

When designing ZINBayes, the authors aimed 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 build 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 to the generative process.

Below we detail the ext-ZINBayes model:

$$
\begin{aligned}
\theta_g &\sim \text{Gamma}(2,1) \\
W_{0,gk'} &\sim \text{Gamma}(0.1, 0.3) \\
W_{1,gk'} &\sim \text{Normal}(0,1) \\
Z_{ik} &\sim \text{Gamma}(2,1) \\
L_i &\sim \text{LogNormal}(\mu_i, \sigma_i) \\
C_i &= [Z_i, S_i] \\
\rho_{ig} &= \frac{C_i W_{0,g}}{\sum_g C_i W_{0,g}} \\
\lambda_{ig} &\sim \text{Gamma}\left(\theta_g, \frac{\theta_g}{\rho_{ig} L_i}\right) \\
Y_{ig} &\sim \text{Poisson}(\lambda_{ig}) \\
logit(\pi_{ig}) &= C_i W_{1,g} \\
D_{ig} &\sim \text{Bernoulli}(\pi_{ig}) \\
X_{ig} &= \begin{cases} Y_{ig} & \text{if } D_{ig} = 0 \\ 0 & \text{otherwise} \end{cases},
\end{aligned}
$$

where $k = 0, \ldots, K$ and $k' = 0, \ldots, K + B$.

Each read count is represented by X_{ig}, where i specifies the cell and g the gene. The latent variable 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. W_0 and W_1 are factor loadings which combined with Z_i, a K-dimensional biological representation of cell i, determine ρ_{ig}, the percentage of transcripts of gene g present in i and π_i, the probability of X_{ig} being a dropout event. Furthermore, λ_{ig} is the mean expression of g in i, S_i is a one-hot representation with size B of cell's i batch and D_{ig} indicates if X_{ig} is a dropout event or not. Finally, θ_g illustrates the dispersion associated with gene g Here we do not explain 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 models definition, exact inference can not be performed due to the intractability of the posteriors. In addition, the model is not conditionally conjugated, making it impossible to use CAVI. As a result, the authors resorted to Automatic Differentiation Variational Inference (ADVI) [11] to implement variational inference.

To identify DEG between two cell subpopulations we adopted the procedure developed in [12]. For each gene g, we define two hypotheses for a given cell pair. Each cell is from a different subpolulation. Furthermore, both cells are from the same batch and have counts x_1 and x_2:

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

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:

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

Its absolute value quantifies the difference between the hypotheses probabilities. Therefore, high magnitude factors reflect strong beliefs that g has a high expression difference between the two cells. Given that H_a^g nad H_b^g are mutually exclusive and have equal prior probabilities, i.e., $p(H_a^g) = p(H_b^g)$, B can be reformulated as:

$$B = \frac{p(H_a^g|x_1, x_2)}{1 - p(H_a^g|x_1, x_2)}, \tag{7}$$

where

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

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 latent 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) sampled from the variational distributions:

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

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. Given that ρ is affected

by the batch of cell i, each cell must be paired with another cell from the same batch but with different type. If inter-batch pairs were allowed the differences between the cells ρ could be biased by batch effects.

After calculating B for each pair, the factors logarithms are averaged. If the absolute value of the average is higher than a certain threshold, gene g is classified as a DEG. The threshold used is defined by the user, but we recommend setting as 2 or 3 [9] since it translates into having one of the hypotheses approximately 7 or 20 times more probable than the opposite one.

To scale this procedure to very large datasets, the method ena

2.2. SIENA

For our second proposed method we designed a new latent variable model, where each count follows a zero-inflated NB 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 different library sizes and stochastic gene expression. The major difference is the removal of variables s and Z, which specify the batches and the low dimensional representations of each cells biological features. Below we present the model, where X_{ig} reports the number of reads mapped to gene g in cell i:

$$
\begin{aligned}
L_i &\sim \text{LogNormal}(\mu_i, \sigma_i^2) \\
\beta_{ig} &\sim \text{Gamma}(\frac{1}{3}, 1) \\
\rho_{ig} &= \frac{\beta_{ig}}{\sum_g \beta_{ig}} \\
\lambda_{ig} &\sim \text{Gamma}(\theta_g L_i \rho_{ig}, \theta_g) \\
Y_{ig} &\sim \text{Poisson}(\lambda_{ig}) \\
D_{ig} &\sim \text{Bernoulli}(\pi_{ig}) \\
X_{ig} &= \begin{cases} Y_{ig} & \text{if } D_{ig} = 0 \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}
$$

On one hand random variables L_i, ρ_{ig}, D_{ig} and λ_{ig}, encode the same as in ZINBayes. L_i encodes 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, π_{ig}, the probability of a dropout event and θ_g, the gene's dispersion are not seen as random variables; π_{ig} is a hyperparameter and θ_g is a model parameter. Nonetheless, these are not the only differences between this model and ZINBayes.

Similarly to what was done in [12], L_i is drawn from a logNormal where the mean and variance of the underlying Normal, μ_i and σ_i^2, are set respectively as the mean and the variance of the log scaled library sizes considering only cells from the same batch as cell i. In zinbayes, μ_i and σ_i^2 are the mean and variance of the log scaled library sizes considering all cells. The choice to model L_i as a logNormal is to restrict its domain to be positive since it's a scaling factor. Note that L_i encodes a scaling factor related to the library size, it is not the actual library size, as pointed out in [12].

As an alternative, we also tested L_i as a Gamma, where its mean and variance were the mean and variance of the library sizes in is batch. In this case, L_i encodes the actual library size.

In regards to ρ_{ig}, we set it as the ratio between a factor related to gene g and cell i, β_{ig}, and the sum of cell's i factors with each gene. We take this formulation to not only restrict ρs domain to be between 0 and 1, but also to constraint the sum of all rho values in a cell to be one, 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 rho as a Beta distribution. However, using a Beta doesnt fit rho properly since it only complies with the domain constraint.

Given that we don't define any biological representation of each cell, the biological variability is implicitly described by variable ρ. However, as we will explain further, each ρ is also affected by the cells batch.

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 do 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 highly expressed genes.

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

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

In this formulation, the NB output is defined as the number of success 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 the λ_{ig}'s Gamma, we aimed to fix the NB expected value as $L_i \rho_{ig}$. By defining the Gamma's 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 as the proportion of zero entries of gene g over all cells from the same type and batch as cells i. Regarding inference, we use ADVI. 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 conjugated, so CAVI can not be implemented.

Given equation (10), we manage to integrate out variables λ_{ig} and Y_{ig}, thus, ADVI merely has to find a distribution q which approximates $p(\beta_{ig}, L_i|X_{ig})$. The variational distribution $q(\beta_{ig}, L_i)$ is considered mean-field, as such it can be factorized in $q(\beta_{ig})$ and $q(L_i)$. Both variational distributions are assumed to be LogNormal, since β_{ig} and L_i are positive variables and ADVI performs exceedingly better when it has to optimize Normal distributions, due to the reparameterization trick.

For each LogNormal we build a neural network, responsible for outputting its mean and variance. This type of neural networks are typically called inference networks. With this approach we are able to scale inference to very large datasets, since ADVI only needs to optimize 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 and another one for the variance. A sotfplus 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, we improve memory-wise scalability by implementing batch training.

Regarding the optimization of θ_{ig}, we iteratively set it as a Maximum Likelihood Estimation (MLE), after updating the networks weights. Therefore, after training, θ_{ig} will have a value that maximizes the counts likelihood, 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) and evaluate them with a Bayes factor logarithm. We calculate the log Bayes factor for each cell pair and use their average absolute value as a metric to classify the gene as DEG or not DEG.

The difference from the ZINBayes is the calculation of $p(H_a^g|x_1, x_2)$, since in this approach ρ_{ig} only depends on βig. Consequently, we only need to account for βig's variational distribution:

$$p(H_a^g|x_1, x_2) = \iint_{(\beta_1,\beta_2)} \mathbb{I}[\rho_{1g} > \rho_{2g}] \cdot q(\beta_1)\, q(\beta_2) \quad (11)$$

Given that in our model we dont specify any variable identifying each cells 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 we do in ext-ZINBayes. This way, the differential expression analysis is more truthful to biological differences.

3. Results

To assess the performance of ZINBayes and SIENA we used to known public scRNA-seq datasets: Islam and PBMC. Since none of the datasets has the genes identified as being DEG or not, we considered has ground truth the top 1000 DEG detected in the corresponding microarray dataset using limma. This was the same procedure applied in [3, 8, 10].

The Islam dataset was gathered in the study [7] and contains expression counts of 92 embryonic cells of the house mouse: 48 Embryonic stem cells (ESC) and 44 Embryonic fibroblast (MEF) cells. The PBMC dataset here used contains count data of human Peripheral Blood Mononuclear Cells, hence the name, that were sequenced in two different batches. The cells are divided in 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 [14] for the comparision between ESC and MEF cells and two different 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 [4] using the codes GSE29087, GSE8835 and GSE29618, respectively. The single cell PBMC dataset that we work with is a subset of the one used in [12]. For Moliner we extracted the data from the .CEL files[1] used in [3].

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 normalization. As such, DE analysis between types ESC 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.

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

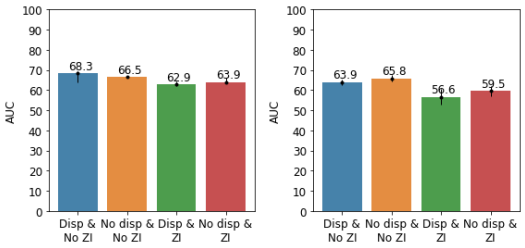
In the following sections, we first assess the effects of using different settings of SIENA, and ext-ZINBayes, then we benchmark their performances with existing DE analysis methods previously mentioned: SCDE, MAST, scVI and DEseq. The first three were designed specifically for scRNA data whereas DEseq is used both for bulk and single cell RNA data.

To run MAST and DEseq we used the respective R packages available on the Bioconductor project. For SCDE we used the R implementation[2] provided by the authors and for scVI we used the Python release 0.3.0[3]

3.1. Configurations assessment

Inspired by what the authors in [17] concluded, we decided to evaluate how the zero-inflation affected SIENA and ext-ZINBayes performances. 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 gene-specific dispersion improves the results. Figure X 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.

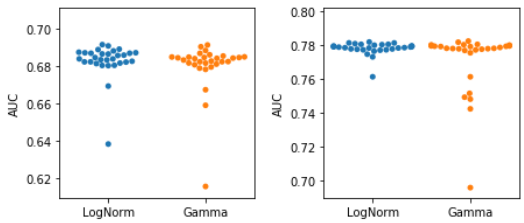


(a) SIENA (b) ext-ZINBayes

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

For SIENA the no zero inflation (ZI) plus gene dispersion combination yields the best average AUC, however it has the highest variance. The configuration ZI plus dispersion has the lowest average. For ext-ZINBayes, employing a simple NB combined with no dispersion leads to a higher average AUC and like for SIENA, the ZI plus dispersion configuration prompts the worst AUC. We also checked how each combination behaved with the B vs Dendritic cells test from the PBMC dataset and verified what we partially concluded from Figure 1: both methods perform better without zero inflation, however SIENA requires the use of gene dispersion whereas ext-ZINBayes doesn't.

For the SIENA configurations assessed, we adopted a log-Normal distribution to model library sizes. However, as we mentioned in section 2, we employed two alternatives for the library sizes, one using a log-Normal distribution and another using a Gamma. From Figure 6 we see that the log-Normal alternative is slightly more robust, hence our choice. 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. Looking at the 6b graph, the difference doesn't seem to be considerable, however the t-test between the AUC means translates into a p-value a little over 5% , so there actually is a significance between the results.



(a) islam (b) B vs Dendritic

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

Given that SIENA resorts to batch training, we also used the PBMC dataset to assess how the mini-batch size affects the performance. As such, we gathered the average, minimum and maximum AUC values obtained when setting the number of mini-batches as 1 (no mini-batches), 8, 18 and 44, which corresponds to defining mini samples of sizes 8404, 1051, 467 and 191, respectively. For each parameterization we ran SIENA 30 times. The results are shown in Figure 3.

[2]https://hms-dbmi.github.io/scde/package.html
[3]https://github.com/YosefLab/scVI/releases

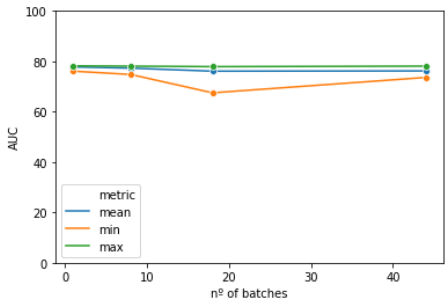


Figure 3: Effect of the number of batches on SIENA's average, minimum and maximum AUC. The dots encode the metrics obtained when using 1, 8, 18 and 44 batches. Results refer to the B vs Dendritic analysis.

3.2. 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. Similarly to what was done in the previous section, we use as a benchmark measure the average AUC. Note that out of those four only MAST and DEseq are deterministic. The results are shown in Figure 4.

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 zero-inflation, since this configuration leverages better results, and each run had 1000 epochs. ext-ZINBayes was also operated without zero-inflation and with 1000 epochs per run, but unlike for SIENA and scVI, no dispersion was adopted.

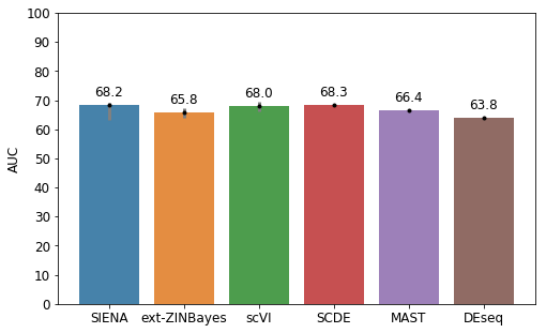


Figure 4: Average AUC values for each method with the Islam dataset. Grey bars indicate the maximum and the minimum AUC obtained.

As seen in Figure 4, SIENA, scVI and SCDE have similar performances with the Islam dataset, showing an average AUC of approximately 68%. ext-ZINBayes and MAST have slightly lower values while DEseq has the lowest average AUC out of all the methods. Nonetheless, SIENA presents a higher variation (around 5%), given that two runs generated an AUC of around 63%.

With the average AUC we can test classification accuracy and robustness, however 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 5, 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 the FDR adjusted p-values of one run.

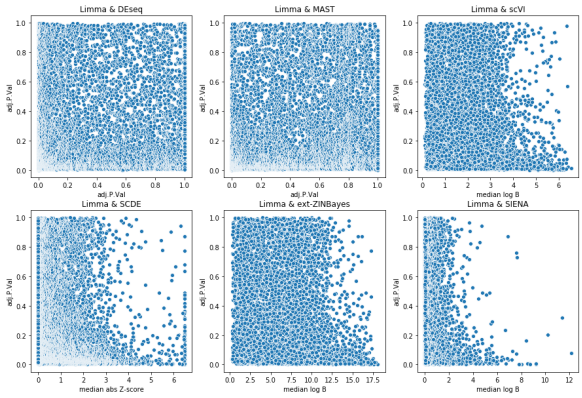
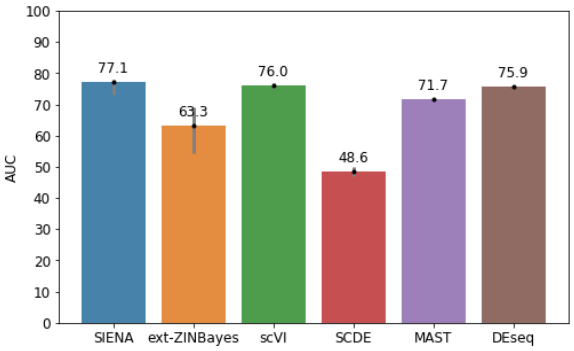


Figure 5: Comparison between each methods DE metrics and the adjusted p-values of limma. Each blue dot corresponds to a gene.

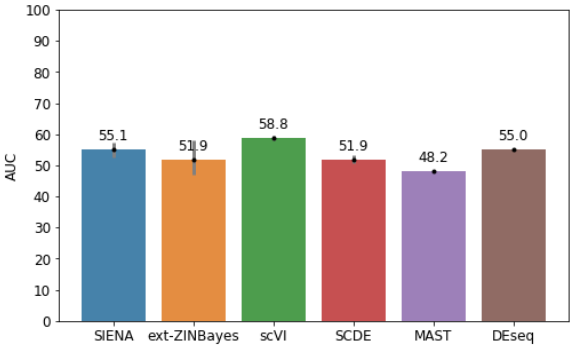
In this comparison, MAST and DEseq present the worst results since there is no visible relation between the p-values determined by both with the limma p-values. Ideally, there should be a linear correlation. The other four methods have a more distinct correlation with limma, since at a certain threshold the p-values tend to be lower as the absolute Z-scores or 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. Same happens for scVI, ext-ZINBayes and SIENA for genes with log Bayes factors higher than ≈ 3.5, ≈ 12.5 and ≈ 2, respectively. Nevertheless, SIENA seems to show a better correlation with limma than the others.

Regarding the PBMC dataset, we also used it to evaluate the methods average AUC, applying the same methodology as previously described. We also employed the same configurations for SIENA, ext-ZINBayes and scVI, but we set only 500 epochs

per run. We used less train iterations, because the PBMC dataset has a lot more data entries than the Islam dataset.



(a) B vs Dendritic



(b) CD4 vs CD8

Figure 6: Average AUC values for each method with the PBMC dataset. Grey bars indicate the maximum and the minimum AUC achieved.

As seen in Figure 6a, all methods, except SCDE and ext-ZINBayes, present a higher average AUC, when conducting DE analysis between B and Dendritic cells, than between ESC and MEF cells. SCDE is the only that shows a great decrease in performance in the B vs Dendritic test, having an AUC lower than 50%. Regarding the CD8 vs CD4 comparison, all methods perform considerably worst, having an average AUC lower than 60%. Nonetheless, unlike what happen in the ESC vs MEF analysis, the methods AUC are more divergent. The difference between the average AUC of the best method (scVI) and the worst (MAST) is around 10%, while in the Islam dataset is slightly lower than 5%.

4. Conclusions

Conclusions, future work and some final remarks...

Acknowledgements

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) through projects PERSEIDS (PTDC/EMS-SIS/0642/2014), PREDICT (PTDC/CCI-CIF/29877/2017) and NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014), and under contracts IT (UID/EEA/50008/2013), INESC-ID (UID/CEC/50021/2019) and LAETA (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] 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.

[3] 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.

[4] R. Edgar. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, Jan. 2002.

[5] P. F. Ferreira, A. M. Carvalho, and S. Vinga. Scalable probabilistic matrix factorization for single-cell RNA-seq analysis. dec 2018.

[6] 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.

[7] 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.

[8] M. K. Jaakkola, F. Seyednasrollah, A. Mehmood, and L. L. Elo. Comparison of methods to detect differentially expressed genes between single-cell populations. *Briefings in Bioinformatics*, page bbw057, jul 2016.

[9] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[10] 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.

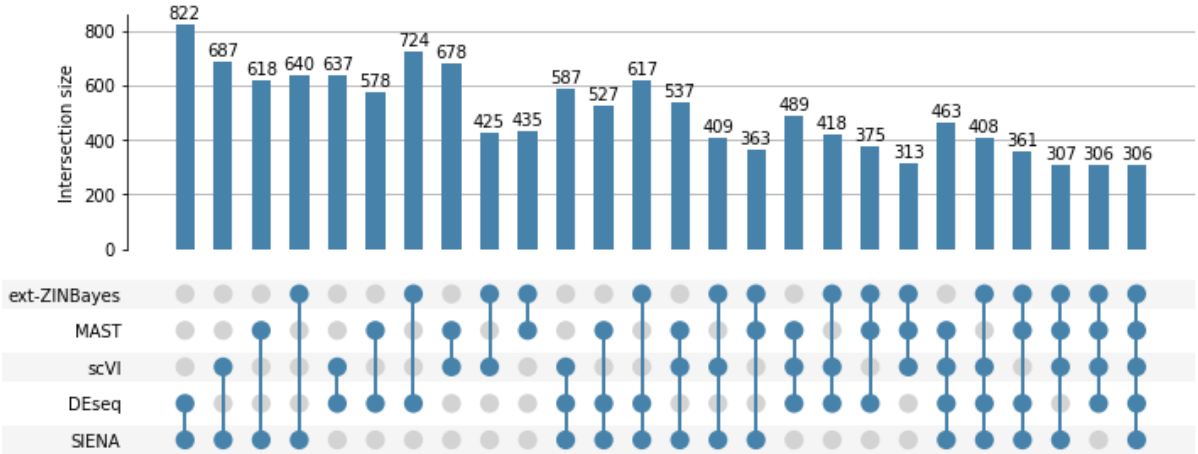


Figure 7: Intersections of the top 1000 DEG regarding the B vs Dendritic analysis. Matrix dots specify the methods combinations and the bars encode the number of DEG in common of the corresponding combination.

[11] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

[12] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Bayesian inference for a generative model of transcriptome profiles from single-cell RNA sequencing. mar 2018.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] V. Svensson. Droplet scrna-seq is not zero-inflated. *bioRxiv*, 2019.