Zero-inflated models for single-cell RNA-seq data

The ZINB team

2016-06-21

Introduction

Single cell RNA-sequencing (scRNA-seq) is a powerful and relatively young technique to characterize molecular states of individual cells through their transcriptional profiles. It represents a major advance with respect to the standard RNA-sequencing which is only capable of detecting gene expressions averaged over millions of cells. Such averaged gene expression profiles may describe and characterize the global state of the tissue but cannot afford to study its heterogeneity and completely mask signals coming from individual cells. Accessing cell-to-cell variability is crucial for understanding many important biological processes such as tissue development and cancer. To be continued...

Model

Let Y_{ij} be the observed read count for gene $j = 1, \ldots, J$ in cell $i = 1, \ldots, n$, and $Z_{ij} \in \{0, 1\}$ an unobserved detection indicator, such that $Z_{ij} = 1$ when gene j is not detected in sample i (i.e., technical zero inflation) and 0 otherwise. We consider a general zero-inflated negative binomial (ZINB) model to account for zero inflation and over-dispersion of the observed counts:

$$\forall y \in \mathbb{N}, \quad \Pr(Y_{ij} = y) = \pi_{ij} f_0(y) + (1 - \pi_{ij}) f(y; \mu_{ij}, \phi_{ij}) \,,$$

where π_{ij} denotes the zero-inflation (ZI) probability, $f_0(\cdot)$ the probability mass function (p.m.f.) of the count distribution when the gene is not detected (typically, the Dirac function at 0), $f(\cdot\,; \mu, \phi)$ the negative binomial (NB) p.m.f. with mean μ and dispersion parameter ϕ, i.e.,

$$f(y; \mu, \phi) = \frac{\Gamma(y + \phi^{-1})}{\Gamma(y+1)\Gamma(\phi^{-1})} \left(\frac{1}{1+\mu\phi}\right)^{\phi^{-1}} \left(\frac{\mu}{\mu + \phi^{-1}}\right)^{y} .$$

Note that another parametrization of the NB p.m.f. is in terms of the inverse dispersion parameter $\theta = \phi^{-1}$ (sometimes also called dispersion parameter in the literature), i.e.,

$$f(y; \mu, \theta) = \frac{\Gamma(y + \theta)}{\Gamma(y+1)\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^{\theta} \left(\frac{\mu}{\mu + \theta}\right)^{y} .$$

In both cases, the mean of the NB distribution is μ, and its variance is:

$$\sigma^2 = \mu + \phi\mu^2 = \mu + \frac{\mu^2}{\theta} \,.$$

In particular the NB distribution boils down to a Poisson distribution when $\phi = 0 \Leftrightarrow \theta = +\infty$.

The mean, ZI probability and dispersion parameter for the count of the gene j in cell i are modeled as follows:

$$\log(\mu_{ij}) = \left(X\beta_\mu + (V\gamma_\mu)^\top + W\alpha_\mu + O_\mu\right)_{ij} \,, \tag{1}$$

$$\mathrm{logit}(\pi_{ij}) = \left(X\beta_\pi + (V\gamma_\pi)^\top + W\alpha_\pi + O_\pi\right)_{ij} \,, \tag{2}$$

$$\phi_{ij} = \phi_j \,, \tag{3}$$

where

- X is a known $n \times M$ design matrix corresponding to M covariates for each cell, and $\beta = (\beta_\mu, \beta_\pi)$ its associated M × J matrices of parameters. X can typically include covariates that induce a variation of interest, such as cell types, or covariates that induce unwanted variations, such as batch or QC measures. It can also include a constant column $\mathbf{1}_n$ to account for gene-specific offsets.
- V is a known $J \times L$ matrix corresponding to known gene-level covariates, such as gene length or GC-content, and $\gamma = (\gamma_\mu, \gamma_\pi)$ its associated $L \times n$ matrices of parameters. It can also include a constant column $\mathbf{1}_J$ to account for cell-specific offsets, such as size factors representing differences in library sizes.
- W is an unobserved $n \times K$ matrix corresponding to unknown cell covariates, which could be of "unwanted variation" as in RUV or of interest (such as biological variations), and $\alpha = (\alpha_\mu, \alpha_\pi)$ its associated $K \times J$ matrices of parameters.
- O_μ and O_π are known $n \times J$ matrices of offsets.
- With a slight overload of notation, $\phi \in \mathbb{R}^J$ is a vector of gene-specific dispersion parameters.

This model deserves a few comments

- The model extends the RUV framework to ZINB variables. It differs in interpretation from RUV in the $W\alpha$ factor, which we do not consider as necessarily unwanted; it generally corresponds to an unknown low-dimensional variations in the data, that could be due to unwanted factors such as technical artifacts (as in RUV), or to biological variations which could be of interest such as cell cycle or cell differentiation, as typically assumed in other factor models such as PCA or ICA.
- By default X and V contain a constant column, to account for cell-specific (e.g., size factors) and gene-specific (e.g., mean expression level) variations. In that case, X and V are of the form $X = [\mathbf{1}_n, X^0]$ and $V = [\mathbf{1}_J, V^0]$, and we can similarly decompose the corresponding parameters as $\beta = [\beta^1; \beta^0]$ and $\gamma = [\gamma^1; \gamma^0]$, where $\beta^1 \in \mathbb{R}^{1 \times J}$ is the vector of gene-specific offsets, and $\gamma^1 \in \mathbb{R}^{1 \times n}$ is the vector of cell-specific offsets. The representation $\mathbf{1}_n \beta^1 + (\mathbf{1}_J \gamma^1)^\top$ is then not unique, but we could make it unique by adding a constant and constraining β^1 and γ^1 to have zero mean.
- The X and V matrices could differ in the modelling of μ and π if we assume that some known factors do not affect both μ and π; to keep notations simple and consistent we use the same matrices, but will implicitly assume that some parameters may be constrained to be 0 if needed.
- When $X = \mathbf{1}_n$ and $V = \mathbf{1}_J$, then the model if a factor model akin to PCA where W is a factor matrix and (α_μ, α_π) are loading matrices.
- By allowing the parameters to differ between the models of μ and π, we can model and test for differences in NB mean or in ZI probability.
- We limit ourselves to a gene-dependent dispersion parameter. More complicated models for ϕ_{ij} could be investigated, such as a model similar to μ_{ij}, or a functional of the form $\phi_{ij} = f(\mu_{ij})$, but we restrict ourselves to a simpler model that has been shown to be largely sufficient in bulk RNA-seq models.

Parameter estimation

The input to the model are the matrices X, V, O_μ, O_π and the integer K; the parameters to be inferred β, γ, W, α and ϕ. The log-likelihood of the model is

$$\begin{aligned} \ell(\beta, \gamma, W, \alpha, \phi) &= \sum_{i=1}^{n} \sum_{j=1}^{J} \ln \Pr(Y_{i,j} = y_{ij} \mid \beta, \gamma, W, \alpha, \phi) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{J} \ln \left(\pi_{ij} f_0(y_{ij}) + (1 - \pi_{ij}) f(y_{ij}; \mu_{ij}, \phi_{ij}) \right) \end{aligned} \tag{4}$$

To infer the parameters we follow a penalized maximum likelihood approach, by trying to solve

$$\max_{\beta,\gamma,W,\alpha,\phi} \{\ell(\beta,\gamma,W,\alpha,\phi) - \mathrm{Pen}(\beta,\gamma,W,\alpha,\phi)\}\,,$$

where $\mathrm{Pen}(\cdot)$ aims at reducing overfitting and improving numerical stability of the optimization problem, when many parameters must be estimated. By default, for a set of regularization parameters $(\epsilon_\beta, \epsilon_\gamma, \epsilon_W, \epsilon_\alpha, \epsilon_\phi)$, we take

$$\mathrm{Pen}(\beta,\gamma,W,\alpha,\phi) = \frac{\epsilon_\beta}{2}\|\beta^0\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma^0\|^2 + \frac{\epsilon_W}{2}\|W\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha\|^2 + \frac{\epsilon_\phi}{2}\mathrm{var}(\phi)^2\,,$$

where β^0 and γ^0 stand for the matrices β and γ without the columns corresponding to the offsets if an unpenalized offset is included in the model, and $\|\cdot\|$ is the Frobenius matrix norm. The penalty tends to shrink the estimated parameters to 0, except for the cell- and gene-specific offsets which are not penalized, and the for the dispersion parameters which are not shrunk towards 0 but instead towards a constant value across genes. Note also that the likelihood only depends on W and α through their product $R = W\alpha$, and that the penalty ensures that at the optimum W and α have specific structures as described in the following result which generalizes standard results such as (Srebro, Rennie, and Jaakkola 2005, Lemma 1; Mazumder and Tibshirani 2010, Lemma 6)

Lemma 1. *For any matrix R and positive scalars s and t, the following holds:*

$$\min_{S,T\,:\,R=ST} \frac{1}{2}\left(s\|S\|^2 + t\|T\|^2\right) = \sqrt{st}\|R\|_*\,,$$

and if $R = R_L R_\Sigma R_R$ is a SVD decomposition of R then a solution to this optimization problem is:

$$S = \left(\frac{t}{s}\right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}}\,, \quad T = \left(\frac{s}{t}\right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R\,.$$

Proof. Let $\tilde{S} = \sqrt{s}S$, $\tilde{T} = \sqrt{t}T$ and $\tilde{R} = \sqrt{st}R$. Then $\|\tilde{S}\|^2 = s\|S\|^2$, $\|\tilde{T}\|^2 = t\|T\|^2$ and $\tilde{S}\tilde{T} = \sqrt{st}ST$ so the optimization problem is equivalent to:

$$\min_{\tilde{S},\tilde{T}\,:\,\tilde{S}\tilde{T}=\tilde{R}} \frac{1}{2}\left(\|\tilde{S}\|^2 + \|\tilde{T}\|^2\right)\,,$$

which by [@Mazumder2010Spectral, Lemma 6] has optimum value $\|\tilde{R}\|_* = \sqrt{st}\|R\|_*$ reached at $\tilde{S} = \tilde{R}_L\tilde{R}_\Sigma^{\frac{1}{2}}$ and $\tilde{T} = \tilde{R}_\Sigma^{\frac{1}{2}}\tilde{R_R}$, where $\tilde{R}_R\tilde{R}_\Sigma\tilde{R}_R$ is a SVD decomposition of $\tilde{R}$. Observing that $\tilde{R}_L = R_L$, $\tilde{R}_R = R_R$ and $\tilde{R}_\Sigma = \sqrt{st}\tilde{R}_\Sigma$ gives that a solution of the optimization problem is $S = s^{-1/2}\tilde{S} = s^{-1/2}R_L(st)^{1/4}R_\Sigma^{1/2} = (t/s)^{1/4}R_L R_\Sigma^{1/2}$, and a similar computation for T concludes the proof. □

This lemma implies in particular that at any local maximum of the penalized log-likelihood, W and $\alpha^\top$ have orthogonal columns, which is useful for visualization or interpretation of latent factors.

To balance the penalties applied to the different matrices in spite of their different sizes, a natural choice is to fix $\epsilon > 0$ and set

$$\epsilon_\beta = \frac{\epsilon}{M_0 J}\,, \quad \epsilon_\gamma = \frac{\epsilon}{nL_0}\,, \quad \epsilon_W = \frac{\epsilon}{nK}\,, \quad \epsilon_\alpha = \frac{\epsilon}{KJ}\,,$$

where M_0 and L_0 are respectively the number of rows in β^0 and in γ^0. As for ϵ_ϕ, it should probably be considered a specific regularization parameter to control the shrinkage of the dispersion estimates across genes.

The penalized likelihood is however not concave, making its maximization computationally challenging. We instead find a local maximum starting from a smart initialization and iterating a numerical optimization scheme until local convergence, as described below.

Initialization

To initialize the set of parameters we approximate the count distribution by a log-normal distribution and explicitly separate zero and non-zero values, as follows:

1. Set $\mathcal{P} = \{(i,j) : Y_{ij} > 0\}$
2. Set $L_{ij} = \log(Y_{ij}) - (O_\mu)_{ij}$ for all $(i,j) \in \mathcal{P}$.
3. Set $\hat{Z}_{ij} = 1$ if $(i,j) \in \mathcal{P}$, $\hat{Z}_{ij} = 0$ otherwise.
4. Estimate $\hat{\gamma}_\mu$ and $\hat{\beta}_\mu$ by solving the convex ridge regression problem:

$$\min_{\beta_\mu, \gamma_\mu} \sum_{(i,j)\in\mathcal{P}} (L_{ij} - (X\beta_\mu)_{ij} - (V\gamma_\mu)_{ji})^2 + \frac{\epsilon_\beta}{2}\|\beta_\mu^0\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\mu^0\|^2 \,.$$

   This is just a ridge regression problem, but with potentially huge design matrix with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of $\hat{\beta}_\mu$ and $\hat{\gamma}_\mu$ by initializing:

$$\hat{\beta}_\mu \leftarrow 0\,, \quad \hat{\gamma}_\mu \leftarrow 0\,,$$

   and repeating a few times (or until convergence):

   (a) Optimization in γ_μ, which can be performed independently an in parallel for each cells:

$$\hat{\gamma}_\mu \in \arg\min_{\gamma\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (X\hat{\beta}_\mu)_{ij} - (V\gamma_\mu)_{ji}\right)^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\mu^0\|^2 \,.$$

   (b) Optimization in β_μ, which can be performed independently an in parallel for each gene:

$$\hat{\beta}_\mu \in \arg\min_{\beta\mu} \sum_{(i,j)\in\mathcal{P}} (L_{ij} - (V\hat{\gamma}_\mu)_{ji} - (X\beta_\mu)_{ij})^2 + \frac{\epsilon_\beta}{2}\|\beta_\mu^0\|^2 \,.$$

5. Estimate $\hat{W}$ and $\hat{\alpha}$ solving

$$\left(\hat{W}, \hat{\alpha}_\mu\right) \in \arg\min_{W,\alpha_\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (V\hat{\gamma}_\mu)_{ji} - (X\hat{\beta}_\mu)_{ij} - (W\alpha_\mu)_{ij}\right)^2 + \frac{\epsilon_W}{2}\|W\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\mu\|^2 \,.$$

   Denoting by $D = L - X\hat{\beta} - (V\hat{\gamma})^\top$, this problem can be rewritten:

$$\min_{W,\alpha} \|D - W\alpha\|_{\mathcal{P}}^2 + \frac{1}{2}\left(\epsilon_W\|W\|^2 + \epsilon_\alpha\|\alpha\|^2\right) \,,$$

   where $\|A\|_{\mathcal{P}}^2 = \sum_{(i,j)\in\mathcal{P}} A_{ij}^2$. By Lemma 1, if K is large enough, this can be solved by first solving the convex optimization problem:

$$\hat{R} \in \arg\min_{R\,:\,\mathrm{rank}(R)\le K} \|D - R\|_{\mathcal{P}}^2 + \sqrt{\epsilon_W\epsilon_\alpha}\|R\|_* \,, \tag{5}$$

   and setting

$$W = \left(\frac{\epsilon_\alpha}{\epsilon_W}\right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}}\,, \quad \alpha = \left(\frac{\epsilon_W}{\epsilon_\alpha}\right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R \,.$$

   where $\hat{R} = R_L R_\Sigma R_R$ is the SVD of $\hat{R}$. This solution is exact when K is at least equal to the rank of the solution of the unconstrained problem (5), which we solve with the `softImpute::softImpute()` function [@Mazumder2010Spectral]. If K is smaller, then (5) becomes a non-convex optimization problem whose global optimum may be challenging to find. In that case we also use the rank-constrainted version of `softImpute::softImpute()` to obtain a good local optimum.

6. Estimate $\hat{\beta}_\pi$, $\hat{\gamma}_\pi$ and $\hat{\alpha}_\pi$ by solving the regularized logistic regression problem:

$$\min_{(\beta_\pi,\alpha_\pi,\gamma_\pi)} \sum_{(i,j)} \Big[- \hat{Z}_{ij}(X\beta_\pi + (V\gamma_\pi)^\top + \hat{W}\alpha_\pi)_{ij} + \log\left(1 + e^{(X\beta_\pi+(V\gamma_\pi)^\top+\hat{W}\alpha_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\beta}{2}\|\beta_\pi\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\pi\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\pi\|^2 . \quad (6)$$

This is just a ridge logistic regression problem, but with potentially huge design matrix with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of $\hat{\beta}_\pi$, $\hat{\alpha}_\pi$ and $\hat{\gamma}_\pi$ by initializing:

$$\hat{\beta}_\pi \leftarrow 0\,, \quad \hat{\gamma}_\pi \leftarrow 0\,, \quad \hat{\alpha}_\pi \leftarrow 0\,,$$

and repeating a few times (or until convergence):

(a) Optimization in γ_π:

$$\hat{\gamma}_\pi \in \arg\min_{\gamma_\pi} \sum_{(i,j)} \Big[- \hat{Z}_{ij}(X\hat{\beta}_\pi + (V\gamma_\pi)^\top + \hat{W}\hat{\alpha}_\pi)_{ij} + \log\left(1 + e^{(X\hat{\beta}_\pi+(V\gamma_\pi)^\top+\hat{W}\hat{\alpha}_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\gamma}{2}\|\gamma_\pi\|^2 . \quad (7)$$

Note that this problem can be solved for each cell (i) independently and in parallel. When there is no gene covariate besides the constant offset, the problem is easily solved by setting $(\hat{\gamma}_\pi)_i$ to the logit of the proportion of zeros in each cell.

(b) $\hat{\beta}_\pi$ and $\hat{\alpha}_\pi$:

$$\left(\hat{\beta}_\pi, \hat{\alpha}_\pi\right) \in \arg\min_{(\beta_\pi,\alpha_\pi)} \sum_{(i,j)} \Big[- \hat{Z}_{ij}(X\beta_\pi + (V\hat{\gamma}_\pi)^\top + \hat{W}\alpha_\pi)_{ij} + \log\left(1 + e^{(X\beta_\pi+(V\hat{\gamma}_\pi)^\top+\hat{W}\alpha_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\beta}{2}\|\beta_\pi\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\pi\|^2 . \quad (8)$$

7. Initialize $\hat{\phi} = 1$.

Optimization

After initialization, we maximize locally the penalized log-likelihood by alternating optimization over the dispersion parameters and left- and right-factors, iterating the following steps until convergence:

1. Dispersion optimization:

$$\hat{\phi} \leftarrow \arg\max_{\phi} \left\{ \ell(\hat{\beta}, \hat{\gamma}, \hat{W}, \hat{\alpha}, \phi) - \frac{\epsilon_\phi}{2}\mathrm{var}(\phi)^2 \right\} .$$

2. Left-factor (cell-specific) optimization:

$$\left(\hat{\gamma}, \hat{W}\right) \leftarrow \arg\max_{(\gamma,W)} \left\{ \ell(\hat{\beta}, \gamma, W, \hat{\alpha}, \hat{\phi}) - \frac{\epsilon_\gamma}{2}\|\gamma^0\|^2 - \frac{\epsilon_W}{2}\|W\|^2 \right\} .$$

Note that this optimization can be performed independently and in parallel for each cell i.

3. Right-factor (gene-specific) optimization:

$$\left(\hat{\beta}, \hat{\alpha}\right) \leftarrow \arg\max_{(\beta,\alpha)} \left\{ \ell(\beta, \hat{\gamma}, \hat{W}, \alpha, \hat{\phi}) - \frac{\epsilon_\beta}{2}\|\beta^0\|^2 - \frac{\epsilon_\alpha}{2}\|\alpha\|^2 \right\} ,$$

Note that this optimization can be performed independently and in parallel for each gene j.

4. Orthogonalization:

$$\left(\hat{W}, \hat{\alpha}\right) \leftarrow \arg \min_{(W,\alpha)\,:\,W\alpha=\hat{W}\hat{\alpha}} \frac{1}{2}\left(\epsilon_W \|W\|^2 + \epsilon_\alpha \|\alpha\|^2\right) .$$

   This is obtained by applying Lemma 1, starting from a SVD decomposition of the current $\hat{W}\hat{\alpha}$. Note that this step not only allows to maximize locally the penalized log-likelihood, but also ensures that the columns of W stay orthogonal to each other during optimization.

TODO: write gradients and explain how steps 1-3 are implemented using a single function

References

Mazumder, Hastie, R., and R. Tibshirani. 2010. "Spectral Regularization Algorithms for Learning Large Incomplete Matrices." *J. Mach. Learn. Res.* 11: 2287–2322. http://www.jmlr.org/papers/v11/mazumder10a.html.

Srebro, N., J. D. M. Rennie, and T. S. Jaakkola. 2005. "Maximum-Margin Matrix Factorization." In *Adv. Neural. Inform. Process Syst. 17*, edited by L. K. Saul, Y. Weiss, and L. Bottou, 1329–36. Cambridge, MA: MIT Press. http://papers.nips.cc/paper/2655-maximum-margin-matrix-factorization.