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

The ZINB team

2016-09-24

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 standard bulk RNA sequencing, which is only capable of measuring gene expression levels averaged over millions of cells. Such averaged gene expression profiles may describe and characterize the global state of a 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

For any $\mu \geq 0$ and $\theta > 0$, let $f_{NB}(\cdot\,;\mu,\theta)$ denote the probability mass function (p.m.f.) of the negative binomial (NB) distribution with mean μ and inverse dispersion parameter θ, namely:

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

Note that another parametrization of the NB p.m.f. is in terms of the dispersion parameter $\phi = \theta^{-1}$ (although θ is also sometimes called dispersion parameter in the literature). In both cases, the mean of the NB distribution is μ and its variance is:

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

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

For any $\pi \in [0,1]$, let $f_{ZINB}(\cdot\,;\mu,\theta,\pi)$ be the p.m.f. of the zero-inflated negative binomial (ZINB) distribution given by:

$$\forall y \in \mathbb{N}, \quad f_{ZINB}(y;\mu,\theta,\pi) = \pi\delta_0(y) + (1-\pi) f_{NB}(y;\mu,\theta) ,$$

where $\delta_0(\cdot)$ is the Dirac function. Here, π can be interpreted as the probability that a 0 is observed instead of the actual count, resulting in an inflation of zeros compared to the NB distribution, hence the name ZINB.

Given n samples (typically, n single cells) and J features (typically, J genes) that can be counted in each sample, let Y_{ij} be the count of feature j (for $j = 1, \ldots, J$) in sample i ($i = 1, \ldots, n$). To account for various technical and biological effects frequent in particular in single-cell sequencing technologies, we model Y_{ij} as a random variable following a ZINB distribution with parameters μ_{ij}, θ_{ij}, and π_{ij}, and consider the following model for the parameters:

$$\begin{aligned}
\ln(\mu_{ij}) &= \left(X\beta_\mu + (V\gamma_\mu)^\top + W\alpha_\mu + O_\mu\right)_{ij} , && (1)\\
\text{logit}(\pi_{ij}) &= \left(X\beta_\pi + (V\gamma_\pi)^\top + W\alpha_\pi + O_\pi\right)_{ij} , && (2)\\
\ln(\theta_{ij}) &= \zeta_j , && (3)
\end{aligned}$$

where

$$\text{logit}(\pi) = \ln\left(\frac{\pi}{1-\pi}\right)$$

and:

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

This model deserves a few comments.

- The model extends the RUV framework to the ZINB distribution (thus far, RUV had only been implement for linear and log-linear regression). It differs in interpretation from RUV in the $W\alpha$ factor, which is not necessarily considered unwanted; this term generally corresponds to unknown low-dimensional variation, that could be due to unwanted techical effects (as in RUV), such as batch effects, or to biological effects which could be of interest, such as cell cycle or cell differentiation, as typically assumed in other factor models such as principal component analysis (PCA) or independent component analysis (ICA).
- By default, X and V contain a constant column of ones, to account, respectively, for gene-specific (e.g., mean expression level) and cell-specific (e.g., size factors) variation. 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 a vector of gene-specific intercepts and $\gamma^1 \in \mathbb{R}^{1 \times n}$ a vector of cell-specific intercepts. The representation $\mathbf{1}_n \beta^1 + (\mathbf{1}_J \gamma^1)^\top$ is then not unique, but could be made unique by adding a constant and constraining β^1 and γ^1 to each have elements summing to zero.
- 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 notation 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 $V = \mathbf{1}_J$, the model is 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 θ_{ij} could be investigated, such as a model similar to μ_{ij} or a functional of the form $\theta_{ij} = f(\mu_{ij})$, but we restrict ourselves to a simpler model that has been shown to be largely sufficient in bulk RNA-Seq analysis.

Parameter estimation

The input to the model are the matrices X, V, O_μ, and O_π and the integer K; the parameters to be inferred are $\beta = (\beta_\mu, \beta_\pi)$, $\gamma = (\gamma_\mu, \gamma_\pi)$, W, $\alpha = (\alpha_\mu, \alpha_\pi)$, and ζ. Given an $n \times J$ matrix of counts Y, the log-likelihood function is

$$\ell(\beta, \gamma, W, \alpha, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{J} \ln f_{ZINB}(Y_{ij}; \mu_{ij}, \theta_{ij}, \pi_{ij}) \,,$$

where μ_{ij}, θ_{ij}, and π_{ij} depend on $(\beta, \gamma, W, \alpha, \zeta)$ through (1)-(3).

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

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

where $\mathrm{Pen}(\cdot)$ is a regularization term to reduce overfitting and improve the numerical stability of the optimization problem in the setting of many parameters. For a set of nonnegative regularization parameters $(\epsilon_\beta, \epsilon_\gamma, \epsilon_W, \epsilon_\alpha, \epsilon_\zeta)$, we set

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

where β^0 and γ^0 denote the matrices β and γ without the rows corresponding to the intercepts if an unpenalized intercept is included in the model, $\|\cdot\|$ is the Frobenius matrix norm, and $\mathrm{Var}(\zeta) = 1/(J-1)\sum_{i=1}^{J}\left(\zeta_i - (\sum_{j=1}^{J}\zeta_j)/J\right)^2$ is the variance of the elements of ζ (using the unbiased sample variance statistic). The penalty tends to shrink the estimated parameters to 0, except for the cell- and gene-specific intercepts which are not penalized and 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 the structure 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:*

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

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 that 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 $S = s^{-1/2}\tilde{S} = s^{-1/2}R_L(st)^{1/4}R_\Sigma^{1/2} = (t/s)^{1/4}R_L R_\Sigma^{1/2}$. A similar argument 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

$$\epsilon_\beta = \frac{\epsilon}{J} , \quad \epsilon_\gamma = \frac{\epsilon}{n} , \quad \epsilon_W = \frac{\epsilon}{n} , \quad \epsilon_\alpha = \frac{\epsilon}{J} , \quad \epsilon_\zeta = \epsilon \,.$$

In particular, from Lemma 1, we easily deduce the following characterization of the penalty on W and α, which shows that the entries in the matrices W and α have similar standard deviation after optimization:

Corollary 1. *For any $n \times J$ matrix R and positive scalars ϵ, the following holds.*

$$\min_{W,\alpha\,:\,R=W\alpha} \frac{\epsilon}{2}\left(\frac{1}{n}\|W\|^2 + \frac{1}{J}\|\alpha\|^2\right) = \frac{\epsilon}{\sqrt{nJ}}\|R\|_* \,.$$

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

$$W = \left(\frac{n}{J}\right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}}\,, \quad T = \left(\frac{J}{n}\right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R\,.$$

In particular, for any $i = 1, \ldots, \min(n, J)$,

$$\frac{1}{n}\sum_{j=1}^{n} W_{j,i}^2 = \frac{1}{J}\sum_{j=1}^{J} \alpha_{i,j}^2 = \frac{[R_\Sigma]_{i,i}}{\sqrt{nJ}}.$$

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 $L_{ij} = \ln(Y_{ij}) - (O_\mu)_{ij}$ 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 β_μ and γ_μ by solving the convex ridge regression problem:

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

This is a standard ridge regression problem, but with a potentially huge design matrix, with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of β_μ and γ_μ. Specifically, we initialize parameter values as:

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

and repeat the following two steps a few times (or until convergence):

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

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

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

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

5. Estimate W and α_μ by solving

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

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

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

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

$$\hat{R} \in \arg \min_{R\,:\,\mathrm{rank}(R)\leq K} \|D - R\|^2_{\mathcal{P}} + \sqrt{\epsilon_W \epsilon_\alpha}\|R\|_* \tag{4}$$

and set

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

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 (4), which we solve with the `softImpute::softImpute()` function [@Mazumder2010Spectral]. If K is smaller, then (4) becomes a non-convex optimization problem whose global optimum may be challenging to find. In that case we also use the rank-constrained version of `softImpute::softImpute()` to obtain a good local optimum.

6. Estimate β_π, γ_π, and α_π by solving the regularized logistic regression problem:

$$\min_{(\beta_\pi,\gamma_\pi,\alpha_\pi)} \sum_{(i,j)} \Big[-\hat{Z}_{ij}(X\beta_\pi + (V\gamma_\pi)^\top + \hat{W}\alpha_\pi)_{ij} + \ln\left(1 + e^{(X\beta_\pi + (V\gamma_\pi)^\top + \hat{W}\alpha_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\beta}{2}\|\beta_\pi\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\pi\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\pi\|^2\,. \tag{5}$$

This is a standard ridge logistic regression problem, but with a potentially huge design matrix, with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of β_π, γ_π, and α_π. Specifically, we initialize parameter values as:

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

and repeat the following two steps a few times (or until convergence):

(a) Optimization in γ_π:

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

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

(b) Optimization in β_π and α_π:

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

7. Initialize $\hat{\zeta} = 0$.

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:

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

   To solve this problem, we start by estimating a common dispersion parameter for all the genes, by maximizing the objective function under the constraint that $\mathrm{Var}(\zeta) = 0$; in practice, we use a derivative-free one-dimensional optimization vector over the range $[-50, 50]$. We then optimize the objective function by a quasi-Newton optimization scheme starting from the constant solution found by the first step. To derive the gradient of the objective function used by the optimization procedure, note that the derivative of the NB log-density is:

$$\frac{\partial}{\partial\theta} \ln f_{NB}(y; \mu, \theta) = \Psi(y+\theta) - \Psi(\theta) + \ln\theta + 1 - \ln(\mu+\theta) - \frac{y+\theta}{\mu+\theta} ,$$

   where $\Psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function. We therefore get the derivative of the ZINB density as follows, for any $\pi \in [0, 1]$:

   - If $y > 0$, $f_{ZINB}(y; \mu, \theta, \pi) = (1-\pi) f_{NB}(y; \mu, \theta)$ therefore

$$\frac{\partial}{\partial\theta} \ln f_{ZINB}(y; \mu, \theta) = \Psi(y+\theta) - \Psi(\theta) + \ln\theta + 1 - \ln(\mu+\theta) - \frac{y+\theta}{\mu+\theta} .$$

   - For $y = 0$, $\frac{\partial}{\partial\theta} \ln f_{NB}(0; \mu, \theta) = \ln\theta + 1 - \ln(\mu+\theta) - \frac{\theta}{\mu+\theta}$, therefore

$$\frac{\partial}{\partial\theta} \ln f_{ZINB}(y; \mu, \theta) = \frac{\ln\theta + 1 - \ln(\mu+\theta) - \frac{\theta}{\mu+\theta}}{1 + \frac{\pi(\mu+\theta)^\theta}{(1-\pi)\theta^\theta}} .$$

   The derivative of the objective function w.r.t. ζ_j, for $j = 1, \ldots, J$, is then easily obtained by

$$\sum_{i=1}^{n} \theta_j \frac{\partial}{\partial\theta} \ln f_{ZINB}(y_{ij}; \mu_{ij}, \theta_j) - \frac{\epsilon_\zeta}{J-1} \left(\zeta_j - \frac{1}{J} \sum_{k=1}^{J} \zeta_k \right) .$$

   (Note that the $J-1$ term in the denominator comes from the use of the unbiased sample variance statistic in the penalty for ζ.)

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

$$\left(\hat{\gamma}, \hat{W}\right) \leftarrow \arg\max_{(\gamma, W)} \left\{ \ell(\hat{\beta}, \gamma, W, \hat{\alpha}, \hat{\zeta}) - \frac{\epsilon_\gamma}{2} \|\gamma^0\|^2 - \frac{\epsilon_W}{2} \|W\|^2 \right\} . \tag{8}$$

   Note that this optimization can be performed independently and in parallel for each cell $i = 1, \ldots, n$. For this purpose, we consider a subroutine `solveZinbRegression`$(y, A_\mu, B_\mu, C_\mu, A_\pi, B_\pi, C_\pi, C_\theta)$ to find a set of vectors (a_μ, a_π, b) that locally maximize the log-likelihood of a ZINB model for a vector of counts y parametrized as follows:

$$\begin{aligned} \ln(\mu) &= A_\mu a_\mu + B_\mu b + C_\mu , \\ \mathrm{logit}(\pi) &= A_\pi a_\pi + B_\pi b + C_\pi , \\ \ln(\theta) &= C_\theta . \end{aligned}$$

   We give more details on how to solve `solveZinbRegression` in the next section. To solve (8) for cell i

we call `solveZinbRegression` with the following parameters:

$$\begin{cases} a_\mu & = \gamma_\mu[.,i] \\ a_\pi & = \gamma_\pi[.,i] \\ b & = W[i,.]^\top \\ y & = Y[i,]^\top \\ A_\mu & = V_\mu \\ B_\mu & = \alpha_\mu^\top \\ C_\mu & = (X_\mu[i,.]\beta_\mu + O_\mu[i,.])^\top \\ A_\pi & = V_\pi \\ B_\pi & = \alpha_\pi^\top \\ C_\pi & = (X_\pi[i,.]\beta_\pi + O_\pi[i,.])^\top \\ C_\theta & = \zeta \end{cases} .$$

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

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

   Note that this optimization can be performed independently and in parallel for each gene $j = 1, \ldots, J$, by calling `solveZinbRegression` with the following paramters:

$$\begin{cases} a_\mu & = (\beta_\mu[.,j]; \alpha_\mu[.,j]) \\ a_\pi & = (\beta_\pi[.,j]; \alpha_\pi[.,j]) \\ b & = \emptyset \\ y & = Y[.,j] \\ A_\mu & = [X_\mu, W] \\ B_\mu & = \emptyset \\ C_\mu & = (V_\mu[j,.]\gamma_\mu)^\top + O_\mu[.,j] \\ A_\pi & = [X_\pi, W] \\ B_\pi & = \emptyset \\ C_\pi & = (V_\pi[j,.]\gamma_\pi)^\top + O_\pi[.,j] \\ C_\theta & = \zeta_j \mathbf{1}_n \end{cases} .$$

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 an 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.

Solving `solveZinbRegression`

Given a N-dimensional matrix of counts $y \in \mathbb{N}^N$, and matrix $A_\mu \in \mathbb{R}^{N\times p}$, $B_\mu \in \mathbb{R}^{N\times r}$, $C_\mu \in \mathbb{R}^N$, $A_\pi \in \mathbb{R}^{N\times q}$, $B_\pi \in \mathbb{R}^{N\times r}$, $C_\pi \in \mathbb{R}^N$ and $C_\theta \in \mathbb{R}^N$, for some integers p, q, r, the function `solveZinbRegression`$(y, A_\mu, B_\mu, C_\mu, A_\pi, B_\pi, C_\pi, C_\theta)$ attempts to find parameters $(a_\mu, a_\pi, b) \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^r$ that maximize the ZINB log-likelihood of y with parameters:

$$
\begin{aligned}
\ln(\mu) &= A_\mu a_\mu + B_\mu b + C_\mu \,, \\
\operatorname{logit}(\pi) &= A_\pi a_\pi + B_\pi b + C_\pi \,, \\
\ln(\theta) &= C_\theta \,.
\end{aligned}
$$

Starting from an initial guess (as explained in the different steps above), we perform a local minimization of this function $F(a_\mu, a_\pi, b)$ using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method. Let us now give some implementation details, on how the gradient of F is computed.

Given a single count y (i.e., $N = 1$), let us first explicit the derivatives of the log-likelihood of y with respect to the (μ, π) parameters of the ZINB distribution. Let us first observe that

$$\frac{\partial}{\partial \mu} \ln f_{NB}(y; \mu, \theta) = \frac{y}{\mu} - \frac{y+\theta}{\mu+\theta} \,,$$

and that by definition of the ZINB distribution the following holds:

$$
\begin{aligned}
\frac{\partial}{\partial \mu} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{(1-\pi) f_{NB}(y; \mu, \theta) \frac{\partial}{\partial \mu} \ln f_{NB}(y; \mu, \theta)}{f_{ZINB}(y; \mu, \theta, \pi)} \,, \\
\frac{\partial}{\partial \pi} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{\delta_0(y) - f_{NB}(y; \mu, \theta)}{f_{ZINB}(y; \mu, \theta, \pi)} \,.
\end{aligned}
$$

Let us explicit these expressions, depending on whether or not y is null:

- If $y > 0$, then $\delta_0(y) = 0$ and $f_{ZINB}(y; \mu, \theta, \pi) = (1-\pi) f_{NB}(y; \mu, \theta)$, so we obtain:

$$
\begin{aligned}
\frac{\partial}{\partial \mu} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{y}{\mu} - \frac{y+\theta}{\mu+\theta} \,, \\
\frac{\partial}{\partial \pi} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{-1}{1-\pi} \,.
\end{aligned}
$$

- If $y = 0$, then $\delta_0(y) = 0$, and we get

$$
\begin{aligned}
\frac{\partial}{\partial \mu} \ln f_{ZINB}(0; \mu, \theta, \pi) &= \frac{-(1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta+1}}{\pi + (1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta}} \,, \\
\frac{\partial}{\partial \pi} \ln f_{ZINB}(0; \mu, \theta, \pi) &= \frac{1 - \left(\frac{\theta}{\mu+\theta}\right)^{\theta}}{\pi + (1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta}} \,.
\end{aligned}
$$

When $N \geq 1$, using standard calculus for the differentiation of compositions and the facts that:

$$
\begin{aligned}
\left(\ln^{-1}\right)' (\ln \mu) &= \mu \,, \\
\left(\operatorname{logit}^{-1}\right)' (\operatorname{logit} \pi) &= \pi(1-\pi) \,,
\end{aligned}
$$

we finally get that

$$\begin{aligned} \nabla_{a_\mu} F &= A_\mu^\top G \,, \\ \nabla_{a_\pi} F &= A_\pi^\top H \,, \\ \nabla_b F &= B_\mu^\top G + B_\pi^\top H \,. \end{aligned}$$

where G and H are the N-dimensional vectors given by

$$\begin{aligned} \forall i \in [1, N], \quad G_i &= \mu_i \frac{\partial}{\partial \mu} \ln f_{ZINB}(y_i; \mu_i, \theta_i, \pi_i) \,, \\ H_i &= \pi_i (1 - \pi_i) \frac{\partial}{\partial \pi} \ln f_{ZINB}(y_i; \mu_i, \theta_i, \pi_i) \,. \end{aligned}$$

References

Mazumder, Hastie, R., and R. Tibshirani. 2010. "Spectral Regularization Algorithms for Learning Large Incomplete Matrices." *J. Mach. Learn. Res.* 11: 2287–2322.

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.