Bayesian inference is implemented using an Adaptive Metropolis (AM) within Gibbs Sampling (GS) algorithm as introduced by [1]. If all model parameters and random effects are sampled throughout the algorithm, most of the full conditionals are conjugate, with AM updates only being required for $\delta_1, \ldots, \delta_{q_0}$ and $\theta$. Although this can be easily implemented, the algorithm is affected by a very poor mixing, mainly related to the updates of the $\rho_{ij}$’s (particularly critical when $n$ and/or $q_0$ are large). To overcome this problem, we implemented Bayesian inference based on the marginal model obtained after integrating out the $\rho_{ij}$’s, i.e.

$$X_{ij} | \mu_i, \delta_i, \phi_j, \nu_j, \theta \sim \begin{cases} \text{Neg-Binomial} & \left( \delta_i^{-1}, \frac{\phi_j \nu_j \mu_i}{\phi_j \nu_j \mu_i + \delta_i^{-1}} \right), \quad i = 1, \ldots, q_0, \ j = 1, \ldots, n; \\ \text{Poisson} & \nu_j \mu_i, \quad i = q_0 + 1, \ldots, q, \ j = 1, \ldots, n, \end{cases}$$  \tag{S1}

for which the associated likelihood function is given by

$$\prod_{i=1}^{q_0} \prod_{j=1}^{n} \frac{\Gamma(x_{ij} + 1/\delta_i)}{\Gamma(1/\delta_i) x_{ij}!} \left( \frac{\phi_j \nu_j \mu_i}{\phi_j \nu_j \mu_i + 1/\delta_i} \right)^{1/\delta_i} \left( \frac{\phi_j \nu_j \mu_i}{\phi_j \nu_j \mu_i + 1/\delta_i} \right)^{x_{ij}} e^{-\phi_j \nu_j \mu_i} \left( \prod_{j=q_0+1}^{n} \nu_j^{x_{ij}} e^{-\nu_j \mu_i} \right)^{1/\theta} e^{-\nu_j / (\theta s_j)}, \tag{S2}
$$

By doing this, the mixing of the sampler was substantially improved. However, the conjugacy of the full conditionals required for the GS is lost. In fact, the full conditionals required for the GS implementation are

$$\pi(\mu_i | \cdots) \propto \frac{\sum_{j=1}^{n} x_{ij}^{-1}}{\prod_{j=1}^{n} (\phi_j \nu_j \mu_i + 1/\delta_i)^{x_{ij} + 1/\delta_i}}, \quad i = 1, \ldots, q_0, \tag{S3}
$$

$$\pi(\nu_j | \cdots) \propto \prod_{i=1}^{q_0} \frac{\nu_j^{x_{ij}}}{(\phi_j \nu_j \mu_i + 1/\delta_i)^{x_{ij} + 1/\delta_i}} \left[ \prod_{j=q_0+1}^{n} \nu_j^{x_{ij}} e^{-\nu_j \mu_i} \right] = \frac{\sum_{j=1}^{n} x_{ij} + 1/\theta - 1}{\prod_{j=1}^{n} (\phi_j \nu_j \mu_i + 1/\delta_i)^{x_{ij} + 1/\delta_i}} e^{-\nu_j \left( \sum_{j=q_0+1}^{n} \nu_j \mu_i + 1/\theta s_j \right)}, \quad j = 1, \ldots, n, \tag{S4}
$$

$$\pi(\theta | \cdots) \propto \frac{\prod_{j=1}^{n} (\nu_j / s_j)^{1/\theta}}{\Gamma(n/(1/\theta))} e^{-\nu_j / (\theta s_j) - b_0 \theta}, \tag{S5}
$$

$$\pi(\delta_i | \cdots) \propto \prod_{j=1}^{n} \frac{\Gamma(x_{ij} + 1/\delta_i)}{\Gamma(1/\delta_i)} \left( \phi_j \nu_j \mu_i + 1/\delta_i \right)^{x_{ij} + 1/\delta_i} \delta_i^{\alpha_2 - 1} e^{-\delta_i b_4}, \quad i = 1, \ldots, q_0, \tag{S6}
$$

$$\pi(\kappa_j | \cdots) \propto \frac{1}{\Gamma^n(1/\delta_i)} \left[ \prod_{j=1}^{n} \frac{\Gamma(x_{ij} + 1/\delta_i)}{\phi_j \nu_j \mu_i + 1/\delta_i} \right] \delta_i^{\alpha_2 - (n/\delta_i) - 1} e^{-\delta_i b_4}, \quad j = 1, \ldots, q_0, \tag{S7}
$$

$$\pi(s_j | \cdots) \propto s_j^{\alpha_2 - (1/\theta) - 1} \exp \left\{ -\frac{\nu_j}{s_j} - s_j b_4 \right\}, \quad j = 1, \ldots, n \tag{S8}
$$

$$\phi_j$$ as in eq. (11) of the paper. $j = 2, \ldots, n$
Although the expression in (S10) is proportional to a Generalized Inverse Gaussian (GIG) density with parameters $a_s - 1/\theta$, $2b_s$, and $2\nu_j/\theta$ (the GIG distribution is parametrised as in [2], p. 478), AM updates are implemented when sampling all other model parameters. We implemented the adaptive proposals, such that the adaptation stops after a fixed number of iterations (to be specified by the user). Because of numerical issues related to the evaluation of the Gamma function in extremely large values, we truncated the value of $\theta$ to be larger than $1 \times 10^{-4}$. Nonetheless, this does not have practical consequences in posterior inference, because the range $(0, 1 \times 10^{-4})$ does not discard interesting cases.

Bayesian inference was implemented using a combination of R and C++ via the library Rcpp. Code is provided as supplementary material and available at [https://github.com/catavallejos/BASiCS](https://github.com/catavallejos/BASiCS).

References


---

1Draws from this distribution are generated using the C code provided by Ester Pantaleo and Robert B. Gramacy (https://r-forge.r-project.org/scm/viewvc.php/pkg/ghyp/src/rgig.c?view=markup&root=rmetrics&pathrev=4804).