Shared Differential Clustering across Single-cell RNA Sequencing Datasets with the Hierarchical Dirichlet Process

Shared Differential Clustering across Single-cell RNA Sequencing Datasets with the Hierarchical Dirichlet Process

Jinlu Liu[a], Sara Wade[a], Natalia Bochkina[a]

[a] *University of Edinburgh, James Clerk Maxwell Building, Edinburgh, EH9 3FD, Scotland, United Kingdom*

Abstract

Single-cell RNA sequencing (scRNA-seq) is a powerful technology that allows researchers to understand gene expression patterns at the single-cell level and uncover the heterogeneous nature of cells. Clustering is an important tool in scRNA-seq analysis to discover groups of cells with similar gene expression patterns and identify potential cell types. Integration of multiple scRNA-seq datasets is a pressing challenge, and in this direction, a novel model is developed to extend clustering methods to appropriately combine inference across multiple datasets. The model simultaneously addresses normalization to deal with the inherent noise and uncertainty in scRNA-seq, infers cell types, and integrates multiple datasets for shared clustering in principled manner through a hierarchical Bayesian framework. A Gibbs sampler is developed that copes with the high-dimensionality of scRNA-seq through consensus clustering. The methodological developments are driven by experimental data from embryonic cells, with the aim of understanding the role of PAX6

URL: Jinlu.Liu@ed.ac.uk (Jinlu Liu), Sara.Wade@ed.ac.uk (Sara Wade), N.Bochkina@ed.ac.uk (Natalia Bochkina)

in prenatal development, and more specifically how cell-subtypes and their proportions change when knocking out this factor.

Keywords: Clustering, Hierarchical Dirichlet process, Log-fold change, Markov Chain Monte Carlo, Single-cell RNA sequencing

---

1. Introduction

Technological developments in the detection of genetic sequences, such as single-cell RNA-sequencing (scRNA-seq), have enabled scientists to measure gene expression on a single-cell level. As opposed to bulk sequencing experiments, which measure average expression levels across the bulk cell population, scRNA-seq enables investigation into the heterogeneity of cells in the population. Clustering is an important tool in scRNA-seq analysis, which is used to disentangle this heterogeneity and discover groups of cells with similar gene expression profiles, yielding potential cell types. Experiments routinely collect multiple scRNA-seq datasets across samples (e.g. individuals, experimental conditions, disease subtypes, time points, etc.) and methods to effectively integrate these datasets are required (as highlighted in the grand challenges of single-cell data science (Lähnemann et al., 2020)). In this work, we aim to extend clustering methods to appropriately integrate multiple datasets in order to identify potential cell types and understand how their proportions differ across datasets in a principled manner, as well as identify potential unique and/or rare patterns that may be present in only a subset of the datasets.

Our work is motivated by experimental data collected to shed light on the development and fates of embryonic cells and the importance of the

transcription factor PAX6 in this process (Mi et al., 2013). More generally, a single cell develops into an estimated 30 trillion cells in humans, and there is great interest in using single-cell technology to understand this process. The transcription factor PAX6 plays an important role in early development and is believed to control the expression of receptors that allow cells to respond correctly to signals from other cells during development (Cabellaro et al., 2014). To empirically study the effect of PAX6, the experimental data was collected at day E13.5 from mouse embryos under control and mutant conditions in which PAX6 is deleted. By employing integrated clustering methods for combined inference across the control and mutant datasets, we can discover potential cell types, which may be shared or unique to a condition, and utilize statistical tools to examine how their proportions change when knocking out this factor, providing insight into the role of PAX6 in cellular development.

However, challenges arise in scRNA-seq data analysis due to the increased uncertainty and noise in measurements when moving from bulk to single-cell RNA-seq (Lähnemann et al., 2020). Specifically, only a small fraction of the total RNA present can be recorded, and thus, the data are very sparse, with zero values representing either true zero counts or missing values, also called dropouts. In addition, the fraction of transcripts recovered, also called the capture efficiency, varies across cells, causing high variability in expression levels and dropout rates. Moreover, batch effects may be present across experimental conditions, e.g. the control and mutant group, leading to further variability in capture efficiencies.

There has been a large amount of research in the field of normalization

of observed gene counts Vallejos et al. (2017b) to account for dropouts, imputation, over-dispersion and batch effects. Often methods originally developed for bulk RNA-seq are employed; however as single cells are highly heterogeneous, the assumptions of these methods are typically not met, potentially leading to adverse consequences in downstream analysis Vallejos et al. (2017b). More recently, methods tailored to scRNA-seq have been introduced Finak et al. (e.g. 2015); Kharchenko et al. (e.g. 2014); L. Lun et al. (e.g. 2016), in which normalization is carried out in a preprocessing step by dividing the raw counts by estimated cell-specific scaling factors (i.e. capture efficiencies). Subsequently, the data are simply log-transformed, after adding an offset to avoid the log of zero, in order to apply standard statistical tools. Alternatively, capture efficiencies can be jointly estimated with other parameters of interest, such as gene expression, through approaches that account for the count nature of the data directly, for example, through a negative-binomial model which also allows for over-dispersion (Vallejos et al., 2015; Risso et al., 2018; Tang et al., 2020).

In order to cluster cells and identify potential cell types, most approaches first apply such normalization strategies to the data in a preprocessing step, followed by clustering in a downstream analysis. In addition, many methods also employ some form of dimension reduction, typically via principal component analysis (PCA) (Senabouth et al., 2017; Satija et al., 2015; Lin et al., 2017) or t-distributed stochastic neighbour embedding (t-SNE) (Qiu et al., 2017). Subsequently, a variety of clustering methods have been employed, such as hierarchical clustering (Senabouth et al., 2017; Lin et al., 2017; Gassen et al., 2015; Kiselev et al., 2017), k-means (Zurauskiene and Yau, 2016; Yang

et al., 2017), density-based methods (Ester et al., 1996; Jiang et al., 2016), or model-based clustering (de Souto et al., 2008; Ji and Ji, 2016; Chandra et al., 2023). For recent reviews of clustering methods for scRNA-seq data, we refer the reader to Kiselev et al. (2019) and Petegrosso et al. (2020).

However, separating the workflow into the steps of normalization, dimension reduction and clustering can adversely affect the analysis, resulting in improper clustering and characterization of cell subtypes (Prabhakaran et al., 2016; Vallejos et al., 2017a). More recent proposals integrate normalization, parameter estimation, and clustering in a combined model-based framework; not only does this allow for simultaneous recovery of clusters, inference of cell subtypes and normalization of the data based on cells with similar expression patterns, but it also provides measures of uncertainty through the model-based approach. Proposals include 1) Prabhakaran et al. (2016) who employ a Dirichlet process (DP) mixture of log-normal distributions and demonstrate the superiority of their approach compared with global normalization followed by clustering; 2) Sun et al. (2018) and Duan et al. (2019) who consider a DP mixture of multinomial distributions; and 3) Wu and Luo (2019) who combine the nested-hierarchical DP (Rodriguez et al., 2008) with a zero-inflated Poisson-log-normal distribution to cluster both subjects and cells in a nested fashion.

In this direction, we construct a novel Bayesian model which combines normalization, parameter estimation, and clustering, and integrates multiple datasets through a hierarchical framework for shared clustering across datasets. In particular, we build on the *bayNorm* model (Tang et al., 2020), which directly accounts for the count nature and overdispersion of the data through

a negative-binomial model and addresses normalization and imputation by estimating the capture efficiencies through an empirical Bayes approach. Specifically, we extend bayNorm to incorporate clustering through mixture models and integrate multiple datasets for shared clustering with a hierarchical framework based on the hierarchical Dirichlet process (HDP, Teh et al., 2006). This allows us to identify potential cell types and infer varying cell-type proportions across datasets in a principled fashion. Moreover, cells are clustered based on both mean expression and dispersion, allowing us to directly account for the mean-variance relationship, which provides robust estimates, particularly for sparse data and/or small clusters (Eling et al., 2018).

An important aspect of clustering scRNA-seq data is the discovery and identification of genes that distinguish one cluster from the others, often referred to as *marker genes* (Petegrosso et al., 2020). Most methods identify marker genes after clustering, by performing some statistical tests, e.g. Satija et al. (2015) identify marker genes by applying the Wilcoxon rank-sum test to the expression values and Guo et al. (2015) use a mixture of the rank-sum test and Welch's t-test depending on the sample size. Motivated by Vallejos et al. (2016), we go beyond simple comparison of mean expression levels; specifically, marker genes that characterize differences between cell subpopulations are detected based on the log-fold change (LFC) of expression values and dispersions across subpopulations. Other methods identify marker genes simultaneously within the clustering process (e.g. Zeisell et al., 2015; Olsson et al., 2016; Zhang et al., 2018).

In summary, we develop a novel Bayesian model (Section 2) that simulta-

neously normalizes the data, infers cell subtypes with unique mean expression and dispersion patterns, and integrates multiple datasets for shared clustering. A posterior inference scheme that copes with the high-dimensionality of scRNA-seq through consensus clustering (Coleman et al., 2022), as well as marker gene detection and posterior predictive checks are developed and described in Section 3. The effectiveness and robustness of the proposed model and probabilistic tools for detection of marker genes are demonstrated on simulated datasets in Section 4, and results on the motivating experimental data to examine how cell-type proportions change with PAX6 is deleted are presented in Section 5.

2. The Model

To introduce notation, we observe multiple scRNA-seq datasets over the same genes, and the raw RNA counts for each dataset d are collected in the matrix $\mathbf{Y}_d$ for $d = 1, \dots D$. Each $\mathbf{Y}_d$ has elements $y_{c,g,d}$, with rows representing cells $c = 1, \dots, C_d$ and columns representing genes $g = 1, \dots, G$. The number of genes G is common across the datasets, while the number of cells C_d is dataset-specific.

2.1. The bayNorm Model

In this work, we build on *bayNorm* (Tang et al., 2020), an integrated modelling approach to address simultaneously normalisation (correcting for variability in capture efficiencies), imputation (accounting for dropouts), and batch effects. Specifically, *bayNorm* assumes a binomial likelihood for the observed raw counts, given the unobserved true latent counts, denoted by

$y^0_{c,g,d}$, and the cell-specific capture efficiencies, denoted by $\beta_{c,d}$:

$$y_{c,g,d} \mid y^0_{c,g,d}, \beta_{c,d} \overset{ind}{\sim} \text{Bin}(y^0_{c,g,d}, \beta_{c,d}).$$

The binomial distribution is a simple model for the transcript capture in scRNA-seq data, assuming the the observed count is obtained through independent Bernoulli experiments determining whether each of the true transcripts is captured, with a constant cell-specific probability (or capture efficiency). The latent counts are modelled with a negative-binomial distribution:

$$y^0_{c,g,d} \mid \mu_g, \phi_g \overset{ind}{\sim} \text{NB}(\mu_{g,d}, \phi_{g,d}), \tag{1}$$

with gene-specific mean expression level $\mu_{g,d}$ and dispersion parameter $\phi_{g,d}$. The negative-binomial, which can be represented as a Poisson-gamma mixture, is required when modelling RNA counts to capture the burstiness and excess variability observed, compared with a Poisson model. The latent counts can be marginalized to obtain the model:

$$y_{c,g,d} \mid \mu_{g,d}, \phi_{g,d}, \beta_{c,d} \overset{ind}{\sim} \text{NB}(\mu_{g,d}\beta_{c,d}, \phi_{g,d}). \tag{2}$$

We remark that potential identifiability issues are apparent in the marginalized model in eq. (2); specifically, if all capture efficiencies are multiplied by a common factor and all mean expressions are divided by that same factor, the model is unchanged. To address this, an informative approach is used to estimate the capture efficiencies, which assumes they are proportional to cell-specific scaling factors times the estimated global mean capture efficiency across all cells for each experiment (See Supplementary Note 1, Tang et al., 2020). We also note that batch effects are mitigated by allowing both the capture efficiencies and mean expression and dispersion to be batch-specific.

An important aspect of *bayNorm* is that it allows imputation of the latent counts by computing the posterior of $y^0_{c,g,d}$, which is shown to be a shifted negative-binomial distribution:

$$y^0_{c,g,d} = y_{c,g,d} + \zeta_{c,g,d},$$

where $\zeta_{c,g,d}$ represents the *lost* count, which has a negative-binomial distribution with mean $\mu_{g,d}(1-\beta_{c,d})(y_{c,g,d}+\phi_{g,d})/(\mu_{g,d}\beta_{c,d}+\phi_{g,d})$ and size $y_{c,g,d}+\phi_{g,d}$. This accounts for dropouts through imputation of $y^0_{c,g,d}$, in contrast to other normalization schemes (e.g. Finak et al., 2015; Kharchenko et al., 2014; L. Lun et al., 2016), which simply rescale the raw data by the estimated cell-specific scaling factors (i.e. zero counts remain zero after rescaling).

2.2. The NormHDP model

We develop a novel model (NormHDP) that extends *bayNorm* to incorporate shared clustering across multiple datasets through a hierarchical Bayesian framework. Specifically, we allow for *cell-specific* mean expression $\mu_{c,g,d}$ and dispersion $\phi_{c,g,d}$ in eq. (1) and assume that they are generated from unknown, discrete, data-specific distributions P_d for $d = 1, \dots D$, which are modelled with a hierarchical Dirichlet process (HDP, Teh et al., 2006):

$$y_{c,g,d} \mid \mu_{c,g,d}, \phi_{c,g,d}, \beta_{c,d} \overset{ind}{\sim} \mathrm{NB}(\mu_{c,g,d}\beta_{c,d}, \phi_{c,g,d}),$$
$$(\boldsymbol{\mu}_{c,d}, \boldsymbol{\phi}_{c,d})|P_d \overset{ind}{\sim} P_d, \quad P_d|P \overset{iid}{\sim} \mathrm{DP}(\alpha P) \quad \text{and} \quad P \sim \mathrm{DP}(\alpha_0 P_0),$$

where $\boldsymbol{\mu}_{c,d} = (\mu_{c,1,d}, \dots, \mu_{c,G,d})^T \in \mathbb{R}^G_+$ and $\boldsymbol{\phi}_{c,d} = (\phi_{c,1,d}, \dots, \phi_{c,G,d})^T \in \mathbb{R}^G_+$ collect the mean expression and dispersion parameters for the c-th cell in dataset d across all genes; $\alpha > 0$ and $\alpha_0 > 0$ are the concentration parameters and P_0 is the base measure of the HDP; and DP denotes the Dirichlet process (Ferguson, 1973) discussed below.

2.2.1. The Hierarchical Dirichlet Process

The HDP (Teh et al., 2006) defines a distribution over a set of exchangeable random probability measures and is a widely adopted nonparametric prior due to its large support, interpretable parameters, and tractability. Realisations of the HDP are discrete with probability one, and an explicit construction is provided by the stick-breaking representation:

$$P_d = \sum_{j=1}^{\infty} p_{j,d}\delta_{\boldsymbol{\theta}_j^*},$$

with $\boldsymbol{\theta}_j^* = (\boldsymbol{\mu}_j^*, \boldsymbol{\phi}_j^*)$ and $\boldsymbol{\theta}_j^* \overset{iid}{\sim} P_0$,

$$p_{j,d} = v_{j,d}\prod_{j'<j}(1-v_{j',d}), \qquad v_{j,d}|(p_1,\ldots,p_j) \sim \text{Beta}\left(\alpha p_j, \alpha\left(1-\sum_{j'=1}^{j} p_j\right)\right),$$

$$p_j = v_j\prod_{j'<j}(1-v_{j'}) \quad \text{and} \quad v_j \sim \text{Beta}(1,\alpha_0).$$

Notice that the probability measures P_d share a common set of atoms $\boldsymbol{\theta}_j^*$, representing different cell subtypes with unique expression levels $\boldsymbol{\mu}_j^*$ and dispersion $\boldsymbol{\phi}_j^*$, but have cell-subtype proportions $p_{j,d}$ that differ across datasets. The discrete nature will induce ties in the cell-specific values of $\boldsymbol{\theta}_{c,d} = (\boldsymbol{\mu}_{c,d}, \boldsymbol{\phi}_{c,d})$ with positive probability, and thus a random clustering of the cells is obtained, where two cells belong to the same cluster if they share the same expression level $\boldsymbol{\mu}_j^*$ and dispersion $\boldsymbol{\phi}_j^*$. Moreover, clusters can be shared across multiple datasets. In fact, the law of this random clustering can be analytically obtained and described by the hierarchical Chinese restaurant franchise (Teh et al., 2006). The HDP is a nonparametric prior that avoids pre-specifying a finite number of cell subtypes and instead assumes the number of cell subtypes in any finite sample is data-driven and grows with the number of cells.

The HDP can also be constructed as the limit of a finite-dimensional HDP defined as

$$P_d^J = \sum_{j=1}^{J} p_{j,d}^J \delta_{\boldsymbol{\theta}_j^*}, \tag{3}$$

with $\boldsymbol{\theta}_j^* \overset{iid}{\sim} P_0$,

$$(p_{1,d}^J, \ldots, p_{J,d}^J) | (p_1^J, \ldots, p_J^J) \sim \text{Dir}(\alpha p_1^J, \ldots, \alpha p_J^J) \tag{4}$$

and

$$(p_1^J, \ldots, p_J^J) \sim \text{Dir}\left(\frac{\alpha_0}{J}, \ldots, \frac{\alpha_0}{J}\right). \tag{5}$$

Here, J represents the truncation level of the finite-dimensional approximation. As shown in Teh et al. (2006) and from the results of Kingman (1975), it follows that $P_d^J \Rightarrow P_d$.

Figure 1 illustrates a simulation from the model in a simple setting with only one gene and a truncation at $J = 5$ cell subtypes. On the left, the mean and dispersion for each cell subtype is depicted, with size reflecting the cell-subtype proportion p_j of the base measure. The concentration parameter of the base measure is $\alpha_0 = 5$; as $\alpha_0 \to \infty$, the Dirichlet prior degenerates to a point mass at $(1/J, \ldots, 1/J)$, while as $\alpha_0 \to 0$, the Dirichlet prior places all mass on the vertices of the simplex. Thus, large values of α_0 favor equal cell-subtype proportions in the base measure and small values of α_0 result in a large proportion for a single cell subtype. On the right of Figure 1, the cell-subtype proportions across two datasets are depicted for varying values of the HDP concentration parameter α, which controls the similarity of the data-specific cell-subtype proportions $p_{j,d}$ to the overall cell-subtype

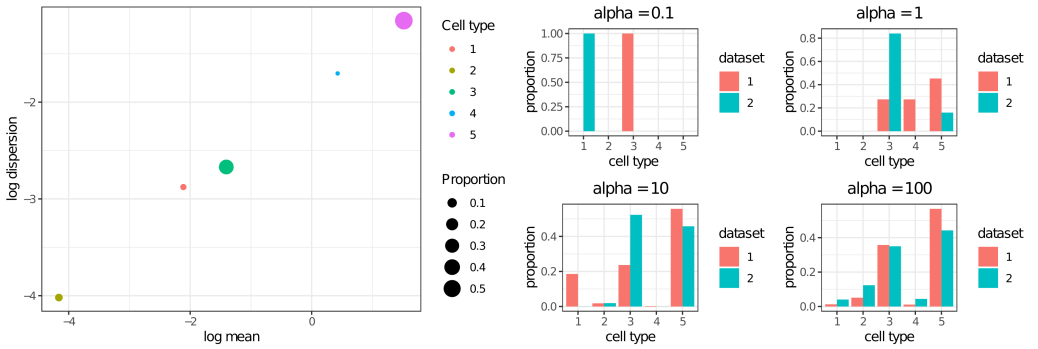


Figure 1: Simulation from the finite-dimensional HDP model, with $G = 1$ genes and $J = 5$ cell subtypes. Left: illustration of the mean and dispersion for each cell subtype, with size reflecting the cell-subtype proportion in the base measure. Right: cell-subtype proportions across two datasets for varying values of α. For a large value of $\alpha = 100$, the cell-subtype proportions for both datasets are similar to those of the base measure. As α decreases, the cell-subtype proportions become more distinct across datasets and degenerate to a single cell subtype per dataset as $\alpha \to 0$.

proportions p_j of the base measure. For a large value of $\alpha = 100$, the cell-subtype proportions for both datasets are similar to those of the base measure. As α decreases, the cell-subtype proportions become more distinct across datasets, and each dataset will contain only a single cell subtype in the limit as $\alpha \to 0$. Based on initial discussions with neuroscientists, such clustering structure reflects the anticipated clustering in the motivating data; specifically, we anticipate similar cell-subtypes, with some difference across control and mutant conditions, and possibly the presence of small unique cell-subtypes.

2.2.2. Prior Specification

The NormHDP model is completed with prior specification for the other parameters, namely, the base measure P_0 of the HDP, the capture efficiencies $\beta_{c,d}$ and additional hyperpriors (e.g. for α, α_0).

The base measure P_0. The prior for the atoms $\boldsymbol{\theta}_j^*$, which characterize the different cell subtypes, is determined by the base measure P_0 of the HDP. One main difference with other proposals in literature (e.g. Wu and Luo (2019)) is that our model allows both the mean expression and dispersion to be cell subtype-specific, i.e. the atoms are $\boldsymbol{\theta}_j^* = (\boldsymbol{\mu}_j^*, \boldsymbol{\phi}_j^*)$. The motivation for this is two-fold; it allows 1) more general patterns to characterize differences between cell subpopulations and 2) inclusion of prior dependence between expression levels and dispersions to account for the mean-variance relationship. Recent studies have demonstrated the utility of exploring more general patterns, beyond focusing solely on differential expression (Korthauer et al., 2016). For example, Vallejos et al. (2015) and Vallejos et al. (2016) develop tools to assess differential variability, which has led to novel biological insights (Martinez-Jimenez et al., 2017). Moreover, a strong relationship is typically observed between mean expression and variability (Brennecke et al., 2013), suggesting that marker genes which are differentially expressed across subpopulations tend also to be differentially dispersed. Also, including prior dependence between expression levels and dispersions has shown to be important for sparse data and/or small sample sizes (Eling et al., 2018); for our motivating dataset, we show that gene counts are truly sparse (Appendix C). Based on preliminary analysis of our motivating dataset (Appendix C), a parametric linear dependence appears sensible (in contrast to the nonlinear dependence

in Eling et al. (2018)). Thus, the base measure is assumed to have the form:

$$P_0(d\boldsymbol{\mu}^*) = \prod_{g=1}^{G} \text{log-N}(\mu_g^* | m_u, a_u^2)$$

and

$$P_0(d\boldsymbol{\phi}^* | \boldsymbol{\mu}^*) = \prod_{g=1}^{G} \text{log-N}(\phi_g^* | b_0 + b_1 \log(\mu_g^*), a_\phi^2). \tag{6}$$

To enhance flexibility and assess robustness, we also consider a simple extension of eq. (6) based on a quadratic prior relationship between the mean-dispersion parameters.

Hyperparameters of P_0. For the hyperparameters of the mean-dispersion model $\mathbf{b} = (b_0, b_1)$ and a_ϕ^2, we set $\mathbf{b}|a_\phi^2 \sim \text{N}(\mathbf{m}_b, a_\phi^2 \mathbf{V}_b)$ and $a_\phi^2 \sim \text{IG}(\nu_1, \nu_2)$, with default values of $\mathbf{m}_b = \mathbf{0}$, $\mathbf{V}_b = \mathbf{I}$, $\nu_1 = 2$, and $\nu_2 = 1$ (Eling et al., 2018). We also consider an empirical prior by setting the values of $\mathbf{m}_b$, v_1 and v_2 based on the estimated linear relationship using the *bayNorm* estimates of the mean and dispersion parameters. The parameters m_u and a_u^2 and can also be set to default values of $m_u = 0$ and $a_u^2 = 0.5$ (Eling et al., 2018) or set empirically based on the mean and range of the mean expression estimates from *bayNorm*.

Capture efficiencies. Vallejos et al. (2017b) acknowledge the inherent randomness in the capture efficiencies; if cells were processed twice, the related scaling factors would vary. Thus, instead of using fixed estimates as in *bayNorm*, we model the capture efficiencies as $\beta_{c,d} \overset{iid}{\sim} \text{Beta}(a_d^\beta, b_d^\beta)$ to account for their randomness. However, as highlighted in Section 2.1, identifiability issues exist. Interestingly, parallels can be made with other domains, namely, the bias

issues in scRNA-seq due to dropouts are akin to under-reporting of events for economic, health, and social indicators (Lopes de Oliveira et al., 2022). In this field, compound models are considered, which involve modelling the latent true event count (similar to the latent count $y^0_{c,g,d}$) and associated reporting probabilities (related to the capture efficiencies). To mitigate identifiability issues, introducing prior information on the reporting probabilities is necessary. One approach is to make use of a validation dataset on the reporting process (Whittemore and Gong, 1991; Stamey et al., 2006; Dvorzak and Wagner, 2016), however this is rarely available. Alternatively, informative priors based expert knowledge have been successful (Moreno and Giron, 1998; Schmertmann and Gonzaga, 2018), as well as a more recent hierarchical approach with an informative prior on only the mean reporting probability (Stoner et al., 2019). Following this framework, we employ informative priors by setting a^β_d and b^β_d empirically based on the mean and variance of the $\widehat{\beta}_{c,d}$ obtained from *bayNorm*. As suggested by one reviewer, the capture efficiencies also play an interesting role in inducing dependence across the genes, and a further discussion on this is provided in Appendix A.1.

Concentration parameters. The HDP concentration parameters influence the overall number of cell subtypes and the amount of information borrowed across datasets. Thus, we infer and account for their uncertainty through the hyperpriors: $\alpha \sim \text{Gam}(1,1)$ and $\alpha_0 \sim \text{Gam}(1,1)$.

3. Posterior Inference

Due to the infinite number of parameters, inference schemes for HDP models often rely on an approximation based on a finite truncation to $J < \infty$

components. Truncations based on the stick-breaking construction (Ishwaran and James, 2001) are widely used but can suffer from an inflated proportion p_J for the last atom if the truncation level is not sufficiently large. Alternatively, the finite-dimensional construction in eq. (3) can be used, which provides a good approximation of the HDP for sufficiently large J (see Ishwaran and Zarepour (2002) for thorough study of the finite-dimensional Dirichlet approximation to the DP); moreover, it has the nice feature of exchangeability of the proportions p_j. We further note that finite-dimensional Dirichlet approximations belong to the class of over-fitted or sparse mixtures, which have shown to be consistent for the true number of clusters, when this number is finite but unknown (Rousseau and Mengersen, 2011). In contrast, DP mixtures can lead to posterior inconsistency for the number of clusters, if the true number of clusters (in an infinite sample) is finite. However, this consistency requires correct specification of local likelihood (Miller and Dunson, 2015).

Thus, focusing on the finite-dimensional approximation of the HDP in eqs. (3)-(5) and introducing the latent allocation variables $z_{c,d}$, where $z_{c,d} = j$ if $\boldsymbol{\theta}_{c,d} = \boldsymbol{\theta}^*_j$, we define the augmented model:

$$
\begin{aligned}
y_{c,g,d} \mid z_{c,d} = j, \mu^*_{j,g}, \phi^*_{j,g}, \beta_{c,d} &\overset{ind}{\sim} \text{NB}(\mu^*_{j,g}\beta_{c,d}, \phi^*_{j,g}), \\
z_{c,d}|(p^J_{1,d}, \ldots, p^J_{J,d}) &\overset{ind}{\sim} \text{Cat}(p^J_{1,d}, \ldots, p^J_{J,d}), \\
(p^J_{1,d}, \ldots, p^J_{J,d})|(p^J_1, \ldots, p^J_J) &\sim \text{Dir}(\alpha p^J_1, \ldots, \alpha p^J_J), \\
(p^J_1, \ldots, p^J_J) &\sim \text{Dir}\left(\frac{\alpha_0}{J}, \ldots, \frac{\alpha_0}{J}\right), \\
\mu^*_{j,g} \overset{iid}{\sim} \text{log-N}(m_u, a^2_u), \quad \phi^*_{j,g}|\mu^*_{j,g} &\overset{ind}{\sim} \text{log-N}(b_0 + b_1 \log(\mu^*_{j,g}), a^2_\phi), \\
\beta_{c,d} &\overset{iid}{\sim} \text{Beta}(a^\beta_d, b^\beta_d)
\end{aligned}
$$

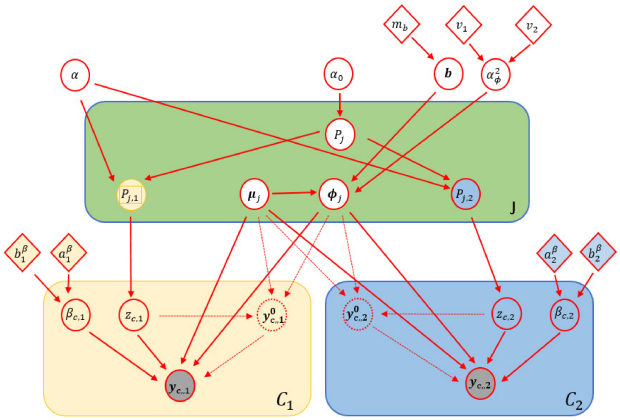


Figure 2: Graphical model of NormHDP with two datasets. Observed gene counts are shown in grey circles and latent genes counts are shown in circles with dashed outlines. Shapes with white fills are for global parameters, and shapes with yellow and blue fills are specific to dataset 1 and 2, respectively. Circles denote parameters of interest, and diamonds denote fixed hyperparameters.

and hyperpriors: $\alpha \sim \text{Gam}(1,1)$; $\alpha_0 \sim \text{Gam}(1,1)$; and $(\mathbf{b}, a_\phi^2) \sim \text{NIG}(\mathbf{m}_b, \mathbf{V}_b, \nu_1, \nu_2)$. A graphical representation of the NormHDP model is shown in Figure 2.

A Markov chain Monte Carlo (MCMC) algorithm is developed for full posterior inference. The algorithm is a Gibbs sampler which produces asymptotically exact samples from the posterior by iteratively sampling the parameters in blocks. For the allocation variables $(z_{c,d})$, mean-dispersion hyperparameters $(\mathbf{b}, \alpha_\phi^2)$ and dataset-specific component probabilities $(\mathbf{p}_d = (p_{1,d}^J, \ldots, p_{J,d}^J))$, the full conditional distributions correspond to standard distributions and can be sampled from directly. For the remaining variables, samples are obtained via adaptive Metropolis-Hastings (Griffin and Stephens, 2013). Full implementation details are provided in Appendix A. We note that each iteration of the Gibbs sampling algorithm has a computational complexity of

$\mathcal{O}((C_1 + \cdots + C_D)JG)$.

However, while in theory, running a single long MCMC chain provides convergence to the posterior of interest, in practice, this produces unsatisfactory results, due to the computational issues that arise in clustering high-dimensional data (Celeux et al., 2019), such as sRNA-seq. Namely, in high-dimensions, MCMC chains are highly sensitive to initialization and tend to get trapped very quickly in local modes, with little to no movement in the cluster allocations after the burn-in period. To overcome such issues and reduce computational cost, Coleman et al. (2022) develop a general scheme to explore the posterior based on an ensemble of Bayesian clustering results. The basic idea is to run a large number of chains (referred to as the width) and a small number of iterations (called the depth). Following Coleman et al. (2022), the width and depth are selected by monitoring the mean absolute difference in the posterior similarity matrix (see Section 3.2), comparing successive iterations for fixed width to determine when the chains have reached the local modes and successive widths for fixed depth to understand if the uncertainty in the clustering has stabilized. Figure 3 provides an example for the data in Section 5, highlighting the utility of this approach to improve exploration of the posterior. After combining the MCMC draws across the multiple chains, in the following, we use T to denote the total number of MCMC draws and use the superscript notation $z_{c,d}^{(t)}$ to denote the t-th sample. The MCMC is re-run conditional on the clustering estimate from consensus clustering to make further inference on the patterns within each cluster; it is important to note that these inferences are conditioned on the fixed clustering and thus do not account for its uncertainty.

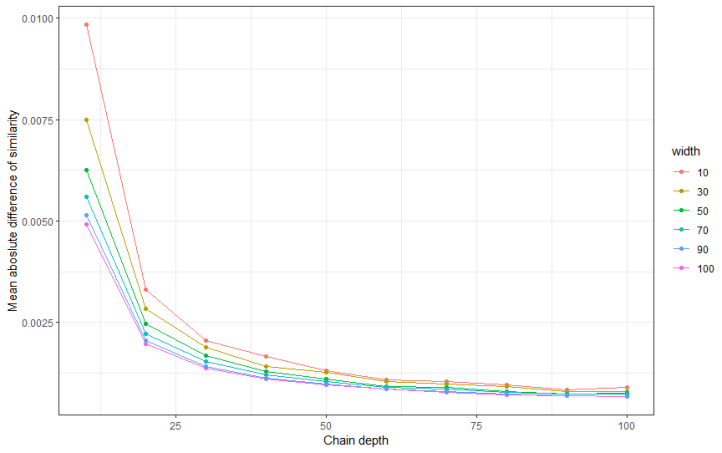


Figure 3: The mean absolute difference between the sequential consensus matrices for the experimental scRNA-seq data, for different chain widths of $10, 30, 50, 70, 80, 100$ and chain depths of $10, 20, \ldots, 100$. The mean absolute difference levels off before 100 iterations.

3.1. Latent Counts

To estimate the normalized count of gene g in cell c, we report the posterior mean of the latent count $y^0_{c,g,d}$, along with posterior summaries to characterize uncertainty. Following Tang et al. (2020), it can be shown the posterior of the latent count given the allocation variables, capture efficiencies and unique parameters has a shifted negative-binomial distribution:

$$y^0_{c,g,d} = y_{c,g,d} + \zeta_{c,g,d},$$

where $\zeta_{c,g,d}$ represents the *lost* count, which has a negative-binomial distribution with mean $\mu^*_{z_{c,d},g}(1-\beta_{c,d})(y_{c,g,d}+\phi^*_{z_{c,d},g})/(\mu^*_{z_{c,d},g}\beta_{c,d}+\phi^*_{z_{c,d},g})$ and size $y_{c,g,d}+\phi^*_{z_{c,d},g}$. Thus, the posterior mean of the latent count given the allocation variables, capture efficiencies and unique parameters can be written as:

$$\mathbb{E}[y^0_{c,g,d}|y_{c,g,d}, z_{c,d}=j, \beta_{c,d}, \boldsymbol{\mu}^*_j, \boldsymbol{\phi}^*_j] = y_{c,g,d}\frac{\mu^*_{j,g}+\phi^*_{j,g}}{\mu^*_{j,g}\beta_{c,d}+\phi^*_{j,g}} + \mu^*_{j,g}\frac{\phi^*_{j,g}(1-\beta_{c,d})}{\mu^*_{j,g}\beta_{c,d}+\phi^*_{j,g}},$$

and the posterior mean of the latent counts can be approximated by the MCMC average:

$$\mathbb{E}[y^0_{c,g,d}|\mathbf{Y}] \approx \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[y^0_{c,g,d}|y_{c,g,d}, z^{(t)}_{c,d} = j, \beta^{(t)}_{c,d}, \boldsymbol{\mu}^{*(t)}_j, \boldsymbol{\phi}^{*(t)}_j],$$

where $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_D)$. We can also examine the full posterior of the latent counts and compute credible intervals by imputing multiple values of $y^0_{c,g,d}$ from the shifted negative-binomial distribution at each MCMC draw.

3.2. Clustering

To summarize the posterior of the allocation variables and uncertainty in the clustering structure, we construct the posterior similarity matrix (PSM) to measure the similarity between individual cells, both within and across datasets. In particular, each element of the posterior similarity matrix, $\text{PSM}_{c,c'}$, represents the posterior probability that cells c and c' are clustered together, which is approximated by

$$\text{PSM}_{c,c'} \approx \frac{1}{T}\sum_{t=1}^{K}\text{I}(z^{(t)}_c = z^{(t)}_{c'}).$$

Cells are ordered in blocks corresponding to the different datasets; thus, the diagonal blocks represent the posterior similarity matrix within each dataset and the off-diagonal blocks represent the posterior similarity matrix across datasets. Within each block, cells are sorted based on hierarchical clustering to improve visualization. Based on the posterior similarity matrix, we obtain a point estimate of the clustering structure by minimizing the posterior expected variation of information (Wade and Ghahramani, 2018). To subsequently, analyze the patterns and uncertainty within each cluster for

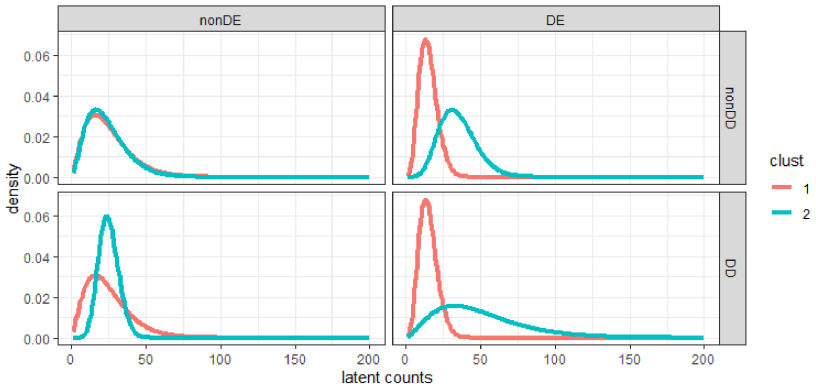
this optimal clustering, an additional MCMC draws values of all parameters with the allocation variables fixed at the optimal clustering (see Appendix A).

3.3. Detecting Marker Genes

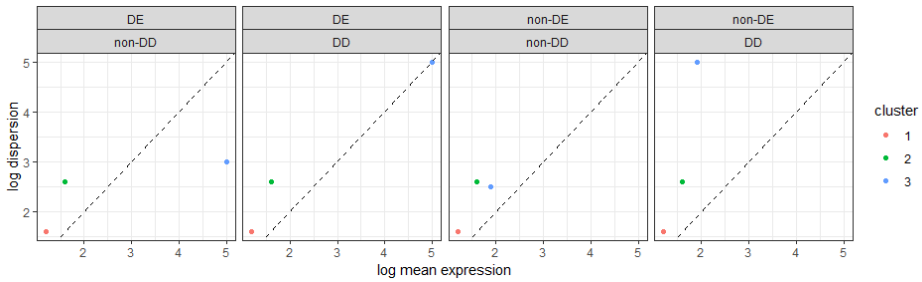
Motivated by Vallejos et al. (2016), probabilistic tools are developed based on the LFC to detect marker genes that distinguish between different cell types. Vallejos et al. (2016) focus on comparing two cell populations; thus, a simple extension is proposed which takes all clusters into consideration. Going beyond comparison of mean expression levels, we aim to detect marker genes both in terms of differential mean expression (DE) and differential dispersion (DD). For example, DD allows identification of genes whose expression may be less stable in one cell subtype.

We define two types of marker genes; *global* marker genes differ between at least two clusters, whereas *local* marker genes for a given cluster differ compared with all other clusters.

Given the clustering allocation $\mathbf{Z} = (z_{c,d})_{c=1,d=1}^{C_d,D}$ of all cells across all datasets, we first focus on comparing the mean expression and dispersion across two clusters j and j'. We highlight that different dispersion parameters quantify changes in heterogeneity across cell subpopulations, while also accounting for the well-known mean-variance relationship in count data (Vallejos et al., 2016). Figure 4 depicts four scenarios for a single gene: on the top right, cells are differently expressed across the cell types with similar heterogeneity; on the bottom left, overall expression levels are similar but less stable with varying heterogeneity; and on the bottom right, cell types differ in both overall expression and heterogeneity. The mean-variance relationship



(a) Density of hypothetical latent counts



(b) Different types of global marker genes

Figure 4: Top: Density of hypothetical latent counts across the four combinations of non-DE/DE and non-DD/DD genes. For illustration, we have assumed a total of 2 clusters. Bottom: a demonstration of four different types of global marker genes with 3 clusters.

induces apriori correlation in the chance of DE and DD, i.e. genes that are DE tend also to be DD, but Figure 4 demonstrates how the other cases also occur.

To measure these changes, the posterior probability associated with the LFC decision rule is computed for each pair of clusters j and j' and each gene g. Specifically, let $P_g(j, j')$ and $L_g(j, j')$ be the posterior tail probabilities that the absolute LFC of the mean expression and dispersion between clusters j

and j' is greater than the threshold τ_0 and ω_0, respectively;

$$P_g(j, j') = \Pr\left(\left|\log\left(\frac{\mu^*_{j,g}}{\mu^*_{j',g}}\right)\right| > \tau_0 \mid \mathbf{Z}, \mathbf{Y}\right)$$

and

$$L_g(j, j') = \Pr\left(\left|\log\left(\frac{\phi^*_{j,g}}{\phi^*_{j',g}}\right)\right| > \omega_0 \mid \mathbf{Z}, \mathbf{Y}\right).$$

In light of the correlation between the unique parameters, we also developed tools to compare the LFC in the residual overdisperion (Eling et al., 2018) (method outlined in Appendix A.6).

3.3.1. Global Marker Genes

Global marker genes are identified by considering the maximum of the posterior tail probabilities across all pairs of clusters:

$$P^*_g = \max_{(j,j')} P_g(j, j') \quad \text{and} \quad L^*_g = \max_{(j,j')} L_g(j, j').$$

Genes with high values of P^*_g or L^*_g have a high posterior probability that the LFC in the mean expression or dispersion is greater than a threshold across at least two clusters. Formally genes are classified as DE if the maximum probability, P^*_g, is greater than the threshold value α_M, and genes are classified as DD if the maximum probability, L^*_g, is greater than the threshold value α_D. By default, these threshold values (α_M, α_D) are set to control the expected false discovery rate (EFDR) to 5 percent (Vallejos et al., 2016). In our context, these are given by:

$$\text{EFDR}_{\alpha_M}(\tau_0) = \frac{\sum_{g=1}^{G}\left(1 - P^*_g(\tau_0)\right)\text{I}(P^*_g(\tau_0) > \alpha_M)}{\sum_{g=1}^{G}(1 - P^*_g(\tau_0))}$$

and

$$\text{EFDR}_{\alpha_D}(\omega_0) = \frac{\sum_{g=1}^{G} \left(1 - L_g^*(\omega_0)\right) \text{I}(L_g^*(\omega_0) > \alpha_D)}{\sum_{g=1}^{G}(1 - L_g^*(\omega_0))}.$$

Global marker genes can be computed by conditioning on the clustering estimate to detect important genes that distinguish between the identified cell subpopulations. Alternatively, uncertainty in the clustering structure can also be incorporated by integrating the maximum probability P_g^* or L_g^* with respect to the posterior of $\mathbf{Z}$, which can be approximated by averaging P_g^* or L_g^* across the MCMC samples.

3.3.2. Local Marker Genes

While global marker genes distinguish between at least two cell subtypes, one might also be interested in identifying *local* marker genes, or cluster-specific marker genes, with unique expression or dispersion for a specified cell-subtype j in comparison with all others. In this case, the minimum of the posterior tail probabilities is computed:

$$P_{g,j}^* = \min_{j' \neq j} P_g(j, j') \quad \text{and} \quad L_{g,j}^* = \min_{j' \neq j} L_g(j, j').$$

For cluster j, genes with high values of $P_{g,j}^*$ or $L_{g,j}^*$ have a high posterior probability that the LFC in the mean expression or dispersion is greater than a threshold between cluster j and any other cluster. If the minimum posterior tail probability is greater than a threshold (calibrated through EFDR), the gene is detected as locally DE or DD for the specified cluster. Figure 5 illustrates the hypothetical latent count density for two genes and four cell subtypes. Both genes are global markers, but the gene on the left is a local marker for cell-subtypes 3 and 4 only, while the gene on the right is not a local marker for any cell subtype.

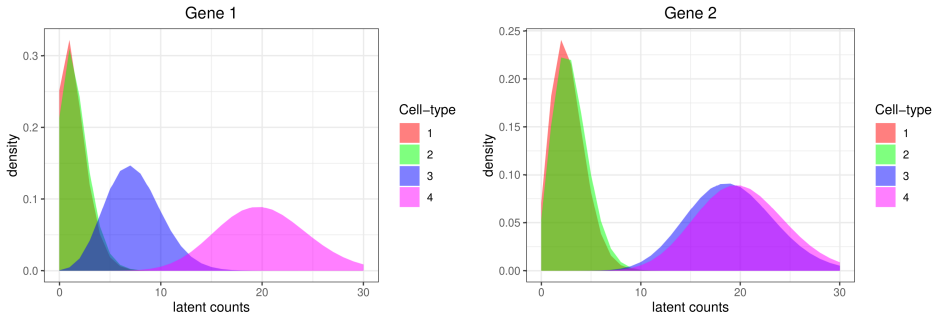


Figure 5: Density of hypothetical latent counts across four cell subtypes for two genes. Both genes are global markers, while the gene on the left is a local marker for cell-subtypes 3 and 4 and the gene on the right is *not* a local marker for any cell subtype.

3.4. Detecting Differences in Datasets

From NormHDP, it is possible to investigate if differences exist between the dataset-specific component probabilities. Specifically, given the clustering estimate, for each cluster j and for each pair of datasets d and d', we compare their component probabilities, $p_{j,d}$ and $p_{j,d'}$, by computing the posterior probability

$$\pi(p_{j,d} > p_{j,d'} \mid \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}(p_{j,d}^{(t)} > p_{j,d'}^{(t)}).$$

In the scRNA case, if the posterior probability is less than 0.05, we classify the cluster as being *over-represented* in the mutant group. If the posterior probability is greater than 0.95, we classify the cluster as being *under-represented* in the mutant group. The remaining clusters are the stable clusters.

3.5. Posterior Predictive Checks

Posterior predictive checks are used to assess the fit of the model together with the inferred parameters to the observed data. In particular, we generate replicated datasets from the posterior predictive distribution (approximated

based on the MCMC draws) and compare key statistics between the observed and replicated data, such as the mean and standard deviation of the log counts (shifted by 1) and the dropout probabilities. Following Lewin et al. (2007) in the context of gene expression data, we employ mixed posterior predictive checks, where the posterior is used to simulate a subset of parameters and the prior is used to simulate the remaining variables. In particular, we generate replicated datasets after first simulating dispersions ($\boldsymbol{\phi}$) from their log-normal prior given the posterior samples of $\mathbf{b}$, α_ϕ^2 and $\boldsymbol{\mu}$. Pseudo-code for generating replicated datasets is given in Appendix A.5.

4. Simulation Study

We consider three simulated scenarios to examine different aspects of our model. In Simulation 1, data is simulated based on the proposed model, and we investigate the ability of the model to recover the true parameters and clustering. In Simulation 2, we study robustness of the model under misspecification of the true mean-variance relationship. In Simulation 3, we assume that only a fraction of genes distinguish between clusters to demonstrate the effectiveness of the proposed probabilistic tools for detecting global marker genes. For each of the simulated scenarios, we run 100 parallel chains, each with 100 iterations to obtain the clustering estimate; in all runs, the algorithm is able to find the correct clustering in less than four iterations. The subsequent MCMC with fixed clustering is run for $T = 8,000$ iterations, following a burn-in of $5,000$ and using thinning of 5; traceplots (shown in Appendix B.4) suggest convergence.

4.1. Simulation 1 and 2

In Simulation 1, we assume the true relationship between the mean-dispersion parameters is linear on the log-scale, while in Simulation 2, we assume it is non-linear and non-quadratic, but monotonically increasing on the log-scale (Figure 6). For both, we simulate data with $C_1 = 50$ cells in dataset 1 and $C_2 = 100$ cells in dataset 2, with $G = 50$ genes. We assume there are 3 clusters, with true cell proportions $(p_{1,1}, p_{2,1}, p_{3,1}) = (0.6, 0.4, 0)$ for dataset 1 and $(p_{1,2}, p_{2,2}, p_{3,2}) = (0.4, 0, 0.6)$ for dataset 2; simulation details are provided in Appendix B.1 and Appendix B.2.

Note that to avoid simulating datasets with empty cells and genes that are not expressed, we generate the true capture efficiencies with a mean of 0.70 which is much higher than the default value of 0.06 for droplet based protocol (Klein et al., 2015). The capture efficiencies in *bayNorm* are estimated as proportional to cell-specific scaling factors with the global mean set to this default value (Supplementary Note 1 of Tang et al., 2020, and also detailed in Appendix A.4); thus, the empirical prior based on the *bayNorm* capture efficiencies estimates has prior mean equal to the default value. Due to the identifiability issues discussed in Section 2, recovery of the true capture efficiencies is difficult if the true mean is far from the prior. Hence, for this task, we employ more informative priors by setting the global mean capture efficiency in *bayNorm* to 0.70. In Appendix B.6, we demonstrate good recovery of the true mean capture efficiencies under our Bayesian approach even when the prior mean is slightly misspecified, which is instead problematic for *bayNorm*.

4.1.1. Results

For both simulations, we investigate prior sensitivity by comparing general priors based on standard hyperparameter values with empirical priors based on hyperparameters specified using initial *bayNorm* estimates (as described in Section 2.2.2). In addition, to enhance flexibility, we consider both linear and quadratic relationships in the prior model for the mean-variance relationship. Results presented focus on the quadratic model with empirical priors (the remaining results are shown in Appendix B.4). But, we highlight that recovery of the true clustering is robust to the choice of priors and model specification (Table 1); for Simulation 2, however, the quadratic model is better in recovering the true mean-dispersion relationship in comparison to the linear model. In Table 1, we also compare the clustering solutions based on our proposed model (and under the different settings) to three competing methods for clustering scRNA-seq data: 1) Seurat (Satija et al., 2015), 2) CIDR (Lin et al., 2017), and 3)TSCAN (Ji and Ji, 2016). The VI and adjusted Rand index (ARI) are used to compare the clustering solutions with

Table 1: Comparison of the different clustering solutions based on the VI and ARI to measure distance between the true and estimated clustering for Simulations 1 and 2.

	General		Empirical		Seurat	CIDR	TSCAN
	Linear	Quadratic	Linear	Quadratic			
Simulation 1 - VI	0.00	0.00	0.00	0.00	0.31	1.30	0.58
Simulation 2 - VI	0.00	0.00	0.00	0.00	0.31	1.38	0.75
Simulation 1 - ARI	1.00	1.00	1.00	1.00	0.87	0.28	0.72
Simulation 2 - ARI	1.00	1.00	1.00	1.00	0.87	0.26	0.63

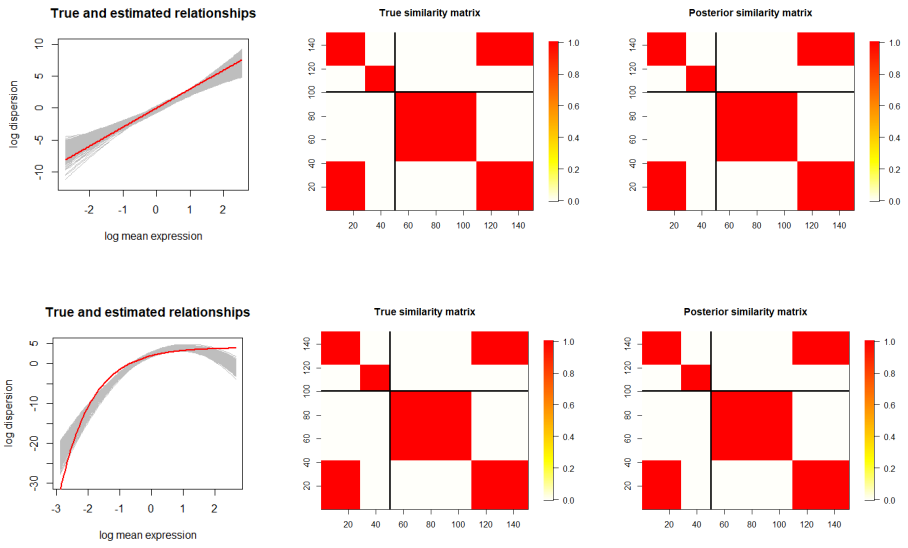


Figure 6: Summary of Simulation 1 (top row) and 2 (bottom row). Left: the true (red) and posterior (grey) relationship between mean-dispersion parameters on the log-scale. Middle: true similarity matrix. Right: posterior similarity matrix.

the truth, where a small value of VI and large value of ARI indicate good performance. In both simulations, NormHDP performs the best, followed by Seurat, TSCAN and CIDR.

Posterior predictive checks are presented in Figure 7 and 8 for Simulation 1 and 2, respectively. For the simulated and replicated datasets, we use kernel density estimation (KDE) to estimate densities of key statistics, namely the 1) mean of log shifted counts, 2) the standard deviation of log shifted counts and 3) the dropout probabilities of each gene. The KDE of the simulated dataset is similar to the KDEs of the replicated datasets, highlighting the sensible fit of the proposed model.

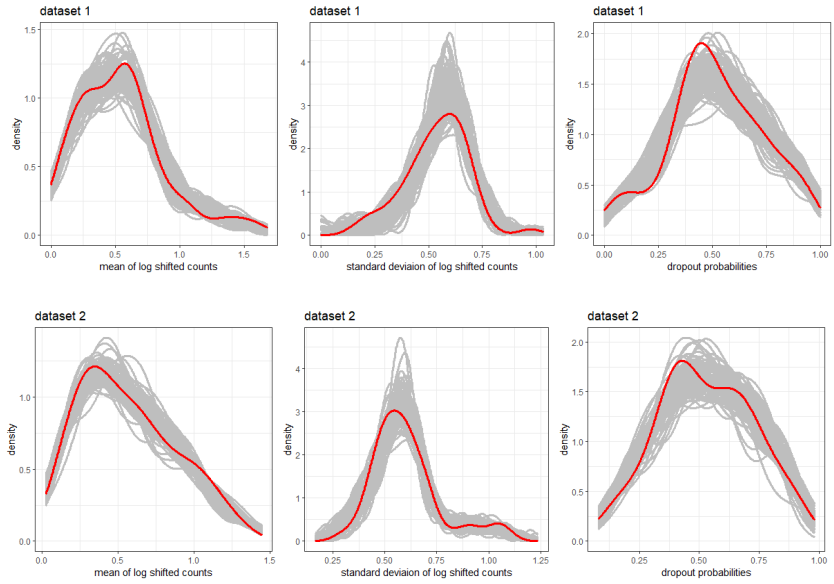


Figure 7: Posterior predictive checks for Simulation 1. Grey and red lines are the KDEs of the replicated and true simulated datasets, respectively. Left to right: the density of mean of log shifted counts, standard deviation of log shifted counts and dropout probabilities.

4.2. Simulation 3

For Simulation 3, we generate 20 replicated sets of data, with $C_1 = 300$ and $C_2 = 400$ cells, for dataset 1 and 2, respectively, with $G = 150$ genes. We assume a total of 3 cell subtypes, with proportions $(p_{1,1}, p_{2,1}, p_{3,1}) = (0.8, 0.2, 0)$ and $(p_{1,2}, p_{2,2}, p_{3,2}) = (0.8, 0, 0.2)$ for dataset 1 and 2, respectively. In addition, we assume that the first 70 percent of the genes are both DE and DD, and remaining genes are both non-DE and non-DD. Unique parameters and allocation variables used to simulate each set of data are identical across replicates. We also assume that the mean expressions $(\mu_{j,g})$ for DE genes follows a log-normal distribution with log-mean m_j for each cluster j and $m_{1:J} = (-3, 5, 8)$. The true relationship between mean and dispersions is

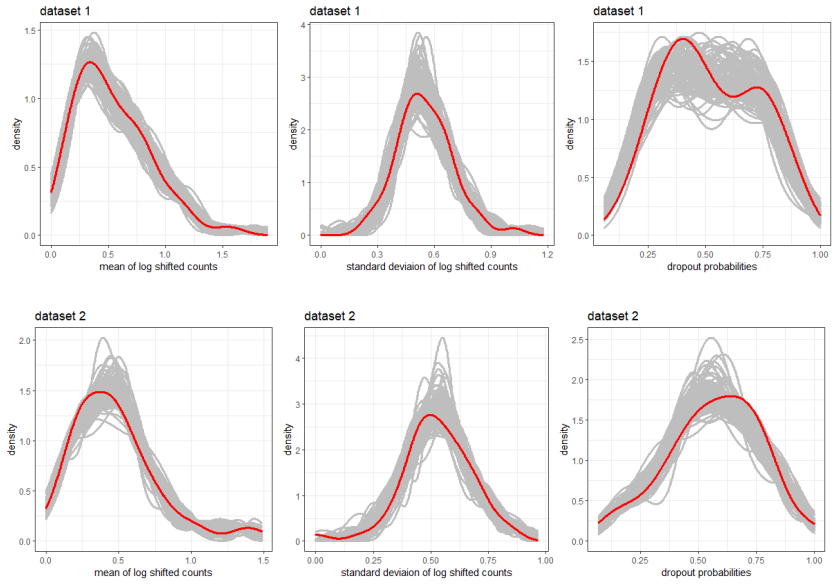


Figure 8: Posterior predictive checks for Simulation 2. Grey and red lines are the KDEs of the replicated and true simulated datasets, respectively. Left to right: the density of mean of log shifted counts, standard deviation of log shifted counts and dropout probabilities.

assumed to be linear on the log-scale. Simulation details are provided in Appendix B.3. In this case, we focus on the linear model with empirical priors. Figure 9 shows that NormHDP recovers well the true mean and dispersion parameters (result based on single set of data). In addition, heat-maps for the true and estimated latent counts show similar patterns (Appendix B.5).

Using the global marker gene detection method proposed in Section 3.3, the range of the false discovery rate (FDR), across the 20 replicated sets of data, corresponding to the mean expressions is $(0, 0.019)$ and dispersions is $(0, 0.045)$, hence NormHDP is sufficient in detecting the true global marker genes. Relationships between mean absolute LFCs and tail probabilities for a single set of data are shown in Figure 10, in which case the FDR is 0 and

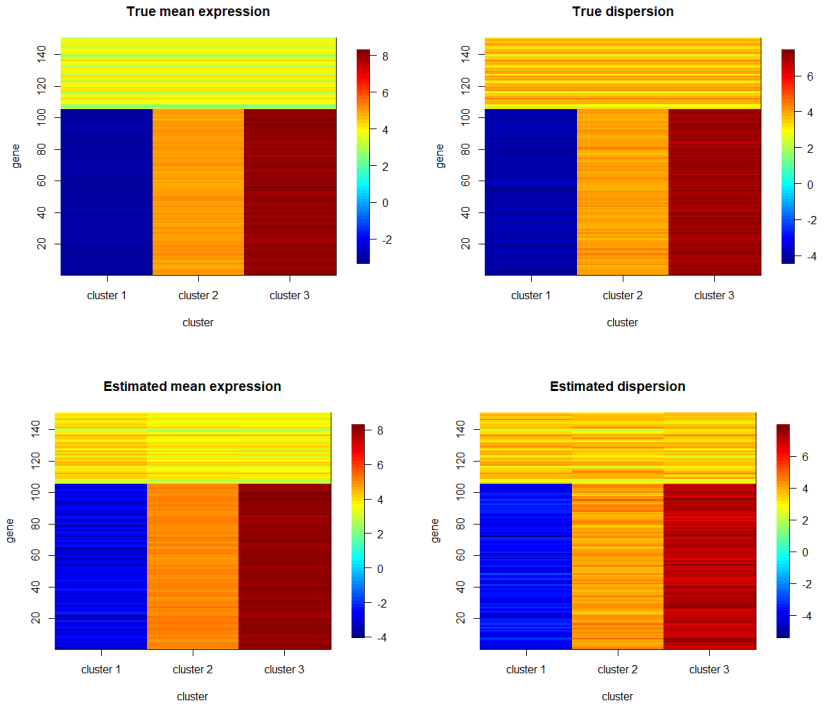


Figure 9: Simulation 3: heat-maps for true and estimated mean expressions and dispersions.

0.028 for the mean expression and dispersion, respectively. The thresholds τ_0 and ω_0 for computing the LFC are set to 1 for both DD and DE.

5. Experimental Data on Embryonic Cell Development

The experimental scRNA-seq datasets (Manuel et al., 2022) analyzed in this paper were collected and prepared by Dr. Tan Kai Boon and the research group lead by Prof. D. Price and Prof. J. Mason at the Centre for Discovery Brain Sciences, University of Edinburgh. The study aims to shed light on the importance of the transcription factor PAX6 in the development and fates of embryonic cells. Tamoxifen administration was carried out at day

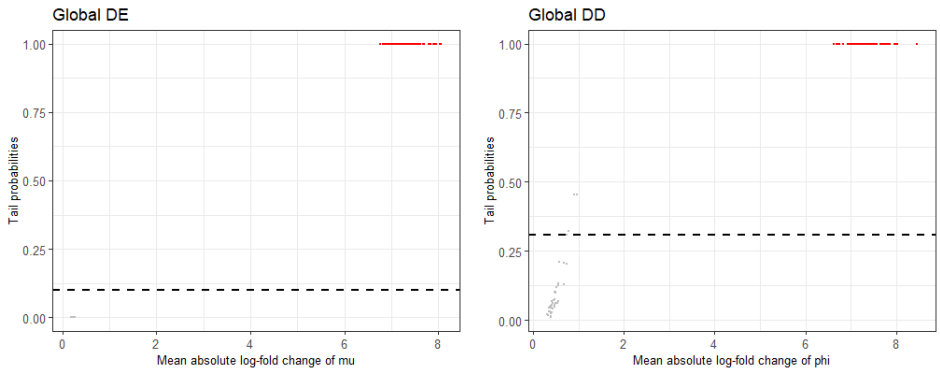


Figure 10: Simulation 3: relationship between the mean absolute LFC and tail probability for Simulation 3. The horizontal dashed lines represent the threshold for classifying DE and DD based on the tail probabilities. True DE or DD genes are represented in red and true non-DE or non-DD genes coloured grey.

E9.5, with mouse embryos sacrificed and dissected at day E13.5. Thus, the resulting scRNA-seq data, collected at day E13.5, is obtained under control (HET) and mutant (HOM) conditions in which PAX6 has been deleted. Standard pre-processing for scRNA-seq is carried out, following procedures in (Hoffman, 2023), to remove non-informative genes and cells from the raw datasets to improve model performance and avoid misinterpretation; pre-processing details are included in Appendix C.1, which involves selecting cells and genes based on quality control metrics (Stuart et al., 2019). After pre-processing, the HET and HOM datasets contain $C_1 = 3,096$ and $C_2 = 5,282$ cells, respectively, both with $G = 2,529$ genes. To investigate the role of PAX6 and differences when PAX6 is not present, the proposed NormHDP model is employed for integrative clustering across the control and mutant datasets. This allows for identification of cell subtypes, that can be shared across datasets, and detection of differences in cell-subtype proportions when

PAX6 is knocked out, providing an understanding of how PAX6 influences the presence/absence of cell subtypes.

For inference, we run consensus clustering with 100 parallel chains and 100 iterations in each chain, which as shown in Figure 3, is sufficient to explore the posterior clustering structure. After obtaining a point estimate of the clustering by considering MCMC draws from all chains, we run an additional MCMC chain to infer the remain parameters with $T = 8,000$ iterations, burn-in of 5000, and thinning of 5. For robustness to non-linearity, we focus on the quadratic model for the mean-variance relationship with empirical priors and fix the truncation level to $J = 30$. Traceplots demonstrating mixing and convergence are shown in Appendix C.2. In all chains, less than 30 components are occupied, thus $J = 30$ provides a sufficient level of truncation.

A heatmap of the posterior similarity matrix in Figure 11 provides a visualization of the clustering structure and its uncertainty, both within and across the control and mutant datasets. The estimated clustering which minimizes the variation of information contains 22 clusters, 19 of which are shared between HET and HOM; while this clustering estimate is clearly observed in Figure 11, there is also some apparent uncertainty on whether to further split some clusters. In the following, we examine the patterns within each cluster based on the subsequent MCMC run with fixed clustering; however, we note that this does not account for uncertainty in the clustering.

To further study the clustering estimate and its differences between the control and mutant groups, we determine which cell subtypes have a high posterior probability of differing in the proportions (Section 3.4). Specifically,

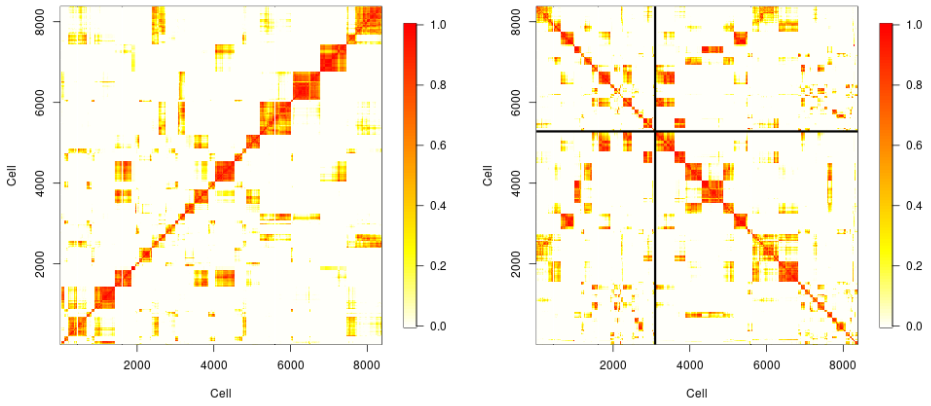


Figure 11: Left: posterior similarity matrix without distinguishing between cells from different datasets. Right: posterior similarity matrix, grouped by HET and HOM.

we define as cell subtype as stable if no difference is detected, *over-represented* in the mutant group if the proportion is larger in the mutant group with high posterior probability, and *under-represented* in the mutant group if the proportion is smaller in the mutant group with high posterior probability. Following this approach, cell subtypes $1, 3, 5, 7, 8, 10, 11, 15, 18$ are *over-represented* in the mutant group, cell subtypes $13, 14, 16, 20, 22$ are stable and cell subtypes $2, 4, 6, 9, 12, 17, 19, 21$ are *under-represented* in the mutant group. Figure 12 provides a visualization of the mean absolute difference in the cell-subtype proportions against the posterior probability of having a larger proportion when PAX6 is present. We note that although several cell subtypes are either over or under represented when PAX6 is deleted, the mean absolute difference in the proportions tends to be small. This suggests that PAX6 may play a smaller role at this early stage in the development (day E13.5).

Additional figures examining the posterior relationship between the mean

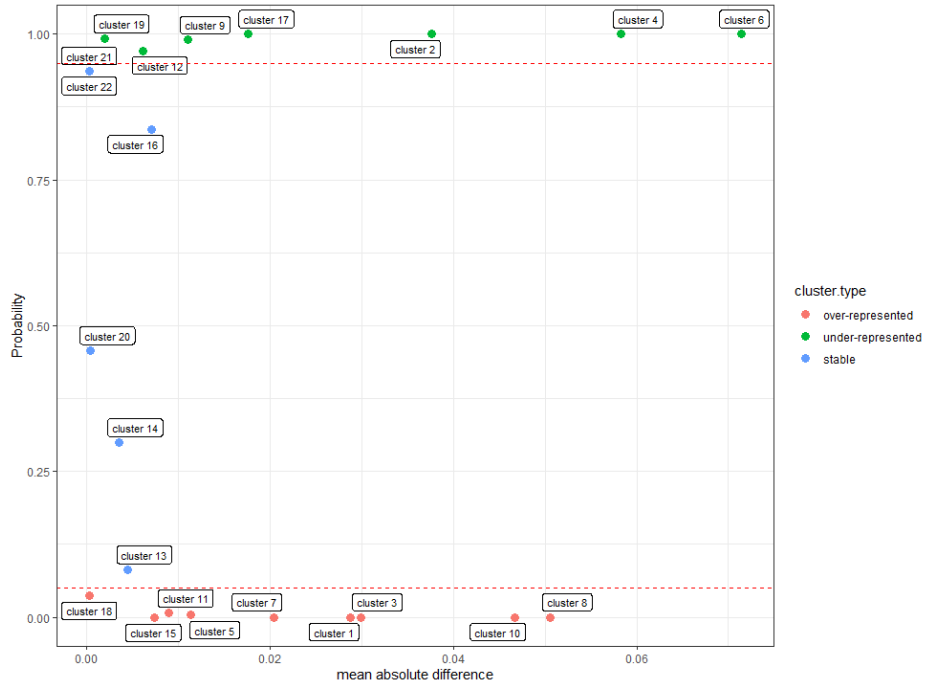


Figure 12: Detecting differences in cell-subtype proportions between the control and mutant groups. The x-axis shows the posterior mean absolute difference in the cell-subtype proportions. The y-axis shows the posterior probability of having a larger proportion when PAX6 is present. Horizontal red dashed lines are the thresholds for cell-subtype classification as stable, under or over represented when PAX6 is deleted.

expressions and dispersions for each cell subtype are provided in Appendix C.2. The following subsections provide a further analysis of the patterns that characterize each cell subtype.

5.1. Posterior Estimated Latent Counts

We compute the posterior estimated latent counts for all cells and compare between cell subtypes. Figure 13 provides a heat-map of the estimated latent counts; cells are ordered by the cell subtypes, with solid vertical lines

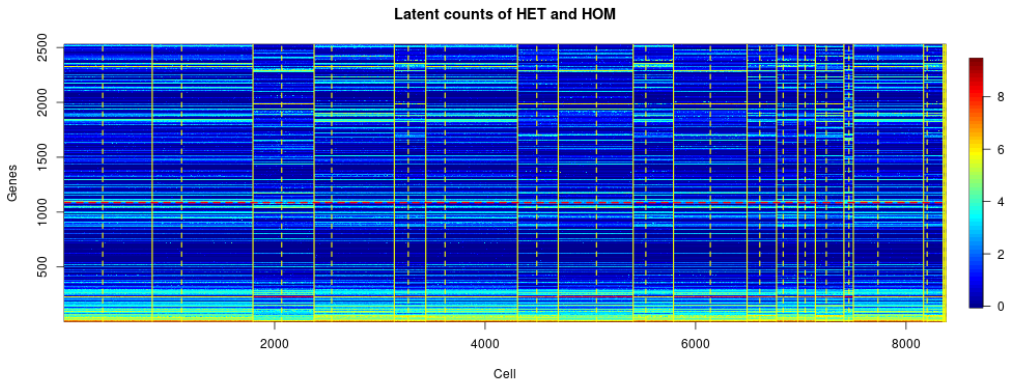


Figure 13: Heat-map of posterior estimated latent gene-counts for HET and HOM. Genes are reordered by global DE tail probabilities and genes above the red horizontal line are global DE. Cells for reordered by the cell subtype, with cells from different subtypes separated by solid lines and cells from different datasets separated by dashed lines.

separating cells from different subtypes and dashed vertical lines separating HET and HOM within cell subtype. Genes are reordered by global DE tail probabilities, with global DE genes above the horizontal line. Corresponding figures for the observed counts are shown in Appendix C.3. For each gene, posterior estimated latent counts and observed counts for cells within each cell subtype are similar, and clear differences are observed across cells from different subtypes.

In addition, we use t-SNE (a commonly used dimensional reduction method for visualising gene expressions) to visualize similarities between cells within each subtype and differences across sybtypes (Figure 14). Applying t-SNE to the posterior estimated latent counts for genes which are global DE and DD shows a clear separation between cell subtypes (Appendix C.3).

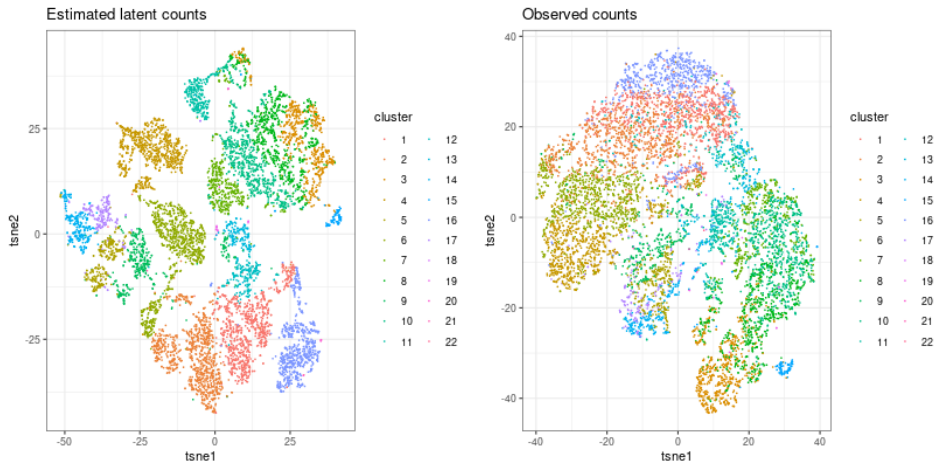


Figure 14: t-SNE plot for the posterior estimated latent counts (left) and observed counts (right). Cells from different subtypes are shown in different colours.

5.2. Global Marker Genes

The detected global marker genes for differential expression and dispersion (with $\tau_0 = 2.5$ and $\omega_0 = 2.5$) are summarized in Figures 15 and 16 and Appendix C.3. In this case, 57% of genes are global markers for DE and 24% of genes are global markers for DD. To visualize the detected global marker genes across cell subtyes, we include heat-maps of the posterior mean of the subtype-specific parameters, with rows representing genes (reordered by gene-wise tail probabilities) and columns representing cell subtyes. We observe that the rare cell subtypes, namely the over-represented cluster 18, the under-represented cluster 21, and the stable clusters 20 and 22, have slightly higher expression levels for genes that are lowly expressed in other cell subtypes and, in general, higher dispersion parameters across most of the the global DD genes and some of the global non-DD genes.

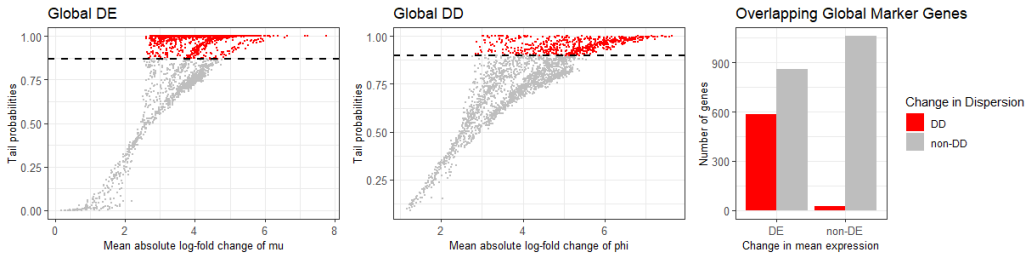


Figure 15: Relationship between mean absolute LFCs and tail probabilities, and a summary of the number of genes that are global markers.

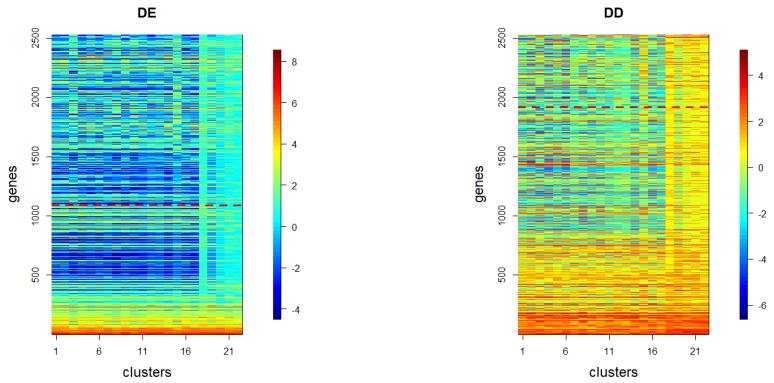


Figure 16: Heat-maps of estimated unique parameters (mean expression on the left and dispersion on the right) on the log scale, with columns representing cell subtypes and rows representing genes. Genes are reordered by tail probabilities; tail probabilities decrease as we move down the rows, with the horizontal dashed lines separating global marker genes from non-marker genes.

5.3. Local Marker Genes

For each cell subtype, we detect local marker genes to identify distinct gene expression patterns in the current cell subtype in comparison with all other subtypes. Threshold values for the LFC are set to 0.8. Figure 17 plots

the absolute LFC against tail probabilities together with summary plots. We observe that the over-represented cell subtypes $3, 8, 15$ and the rare cell subtypes $18, 20, 21, 22$ have high numbers of local marker genes in terms of both DE and DD. For DE, 1752 genes are classified as local DE genes for more than one cell subtype and gene *Neurod4* is classified as a local DE gene for the most cell subtypes (16 cell subtypes). For DD, 1234 genes are classified as local DD genes for more than one cell subtype and genes *Kcnma1* and *Scgn* are classified as a local DD gene for the most cell subtypes (18 cell subtypes). Heat-maps of the estimated mean expressions and dispersions for local marker genes are shown in Appendix C.4. We observe that the local marker DE genes for the over-represented cell subtype 15 tend to be more highly expressed in this subtype. For the rare cell subtypes $18 - 22$, the local DE genes tends to be lowly expressed in all other cell subtypes, with slightly higher expression in cell subtype 15, and most of the local DD genes have higher dispersions (less over-dispersion). In addition, for the remaining cell subtypes $1 - 14$, 16 and 17, most local DD genes have smaller dispersions, thus, these genes are over-dispersed in the corresponding cell subtypes.

Further findings and discussions on specific *important* genes provided by the research group of Prof. Price can be found in Appendix D.

5.4. Posterior Predictive Checks

To assess the fit of the proposed NormHDP model to the experimental data, we carry out mixed posterior predictive checks, as described in Section 3.5. For a single replicated dataset, we compare key statistics, namely, the mean and standard deviation of the log shifted counts and the proportion of dropouts for each gene. The statistics of the replicated dataset match well the

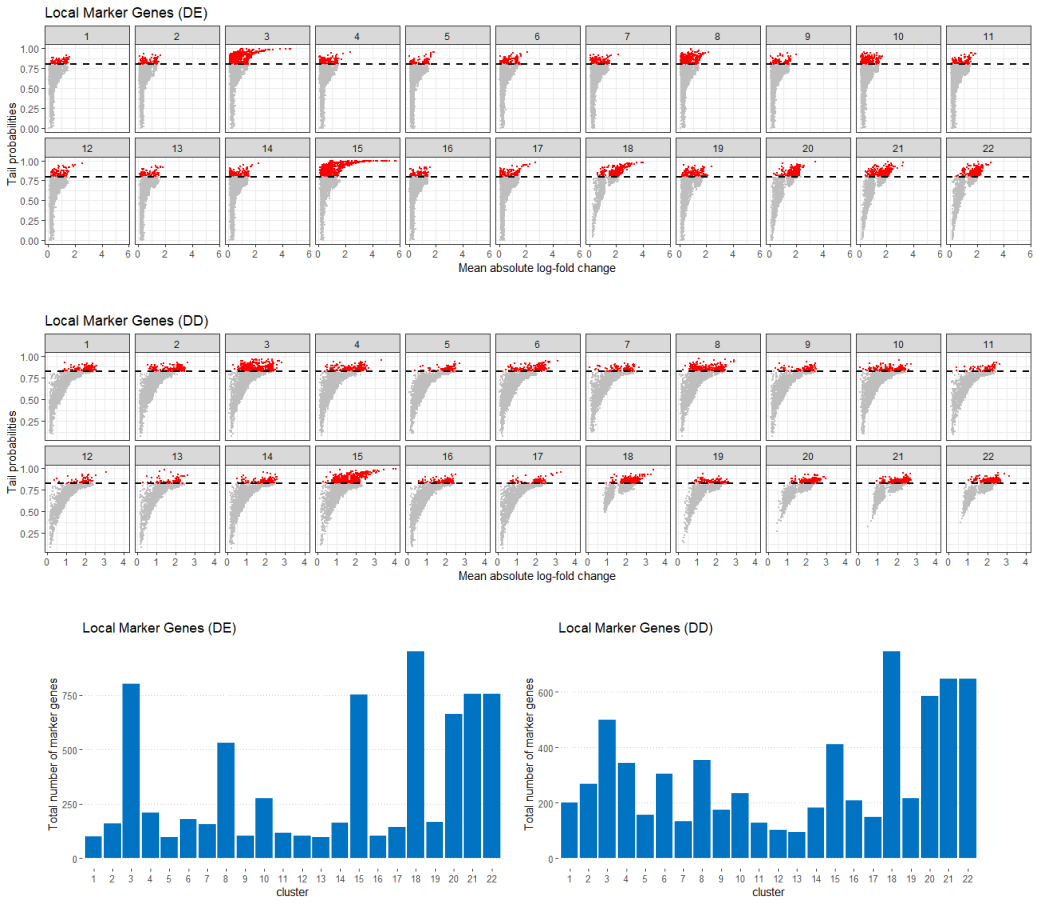


Figure 17: Detection of local marker genes. Top rows: plots of tail probabilities against mean absolute LFCs for each cell subtype, with mean expression in the top row and dispersion in the middle row. Bottom row: a summary of the number of local DE (left) and DD genes (right).

observed data, highlighting the sensible fit of the model to the data (Figure 18). For multiple replicates, we compare the KDEs of these statistics between the observed dataset and replicated datasets; the KDEs are similar, further supporting the model fit (Figure 19).

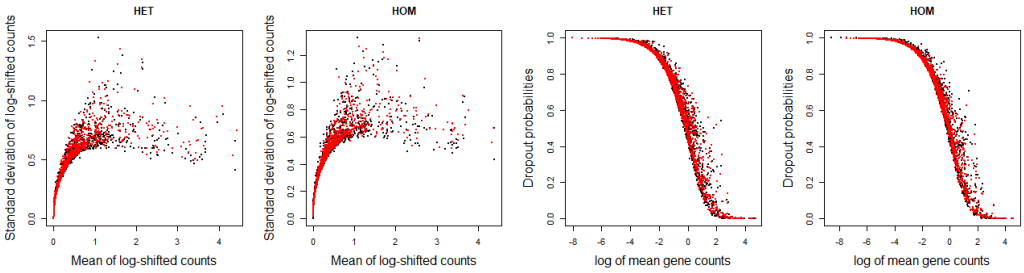


Figure 18: Posterior predictive checks for the experimental data. Comparing the relationships between key statistics (namely, the mean, standard deviation, and dropout probability) for the observed dataset (in red) and the replicated dataset (in black).

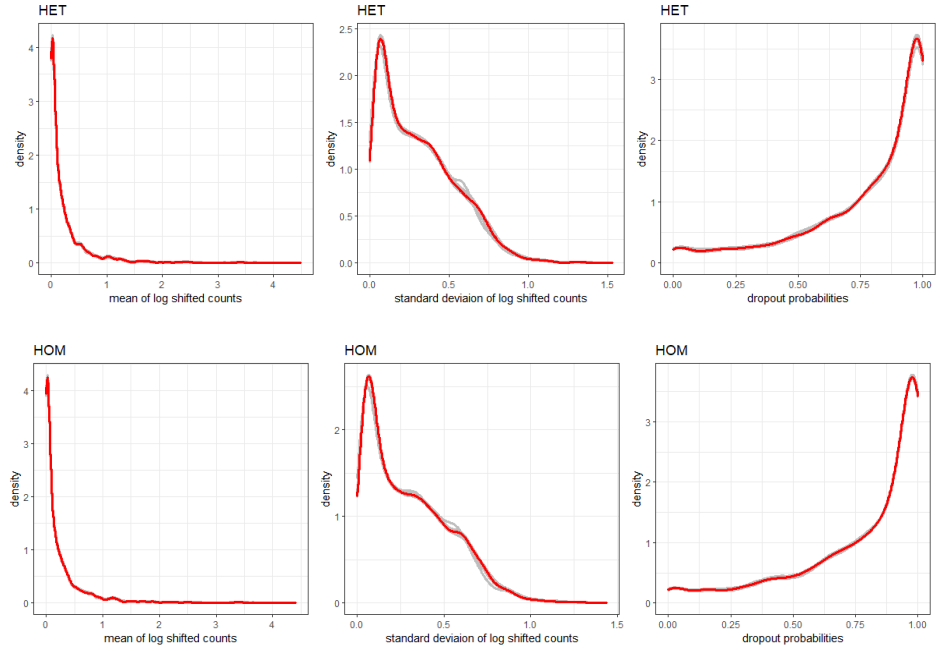


Figure 19: Comparison of the KDE of statistics from the observed and replicated datasets, with observed dataset in red and replicated datasets in grey.

6. Discussion

In this article, we constructed a novel hierarchical Bayesian model that integrates multiple RNA-seq datasets for shared clustering and identification of cell types. Moreover, it addresses the challenges of normalization and imputation in scRNA-seq by building on the *bayNorm* model. In this high-dimensional setting, the proposed model allows borrowing of information and detection of differences in cell-type proportions across datasets as well as measures of uncertainty in the estimated clustering and detection of marker genes to characterize the patterns within each cell type. In simulated datasets, our proposed NormHDP model is robust and able to recover the true parameters and clustering, as well detect correctly the marker genes.

Our work was motivated by experimental scRNA-seq data collected to shed light on the role of PAX6 in prenatal development. By applying NormHDP, we can identify cell types that form in this early stage of development and gain an understanding of the differences in cell type presence/absence when PAX6 is deleted. In the experimental data, the model estimates a total of 22 cell types, with four rare cell types. Among the identified cell types, nine are over-represented and seven are under-represented when PAX6 is deleted. However, although differences between the control and mutant groups are detected with a high posterior probability, the posterior mean absolute value of the difference tends to be small, suggesting that PAX6 plays a smaller role at this early stage in the development (E13.5). Following this, our colleagues at the Centre for Discovery Brain Sciences have collected additional data at day E14.5. In ongoing work, initial results suggest a stronger role of PAX6 at this slightly later stage of development, and a comprehensive analysis

will be conducted in future work to integrate all datasets. More generally, experiments now routinely collect multiple scRNA-seq datasets, and the proposed model is relevant in such applications for shared clustering and borrowing of information and understanding differences across datasets.

For posterior inference, we have developed a Gibbs sampling algorithm, which produces asymptotically exact posterior samples. However, in high-dimensional settings, such as scRNA-seq, where the number of cells and genes are typically in the thousands, such algorithms suffer from convergence issues. To combat this and improve computational speed, we have employed a parallel consensus clustering approach. This relies on combining multiple short chains to explore the clustering, followed by a long chain with fixed clustering to infer all other parameters. While the first phase provides uncertainty in clustering, the second phase only considers uncertainty in parameters conditional on clustering estimate. This motivated to balance speed, mixing, and uncertainty quantification, and in ongoing work, we are also developing a variational Bayes approximation (Hughes et al., 2015) for faster, approximate inference that provides uncertainty estimates unconditional on clustering structure (although this comes with other assumptions on the form of the approximate posterior). Other potential model extensions of interest include cluster-specific mean-dispersion relationships for increased flexibility and incorporation of covariate information (such as cell-specific latent time) (Bergen et al., 2020), as well as priors for clustering beyond the HDP (Argiento et al., 2020).

Supporting Information

Appendices referenced in Sections 2, 3, 4, 5, and 6 are available online. All code for model implementation and analysis is publicly available through the Github repository (`https://github.com/jinluliu550/normHDP`), along with the simulated data (Section 4).

In Appendix A, we provide further model insights on the capture efficiencies, describe the MCMC algorithm for posterior inference, explain the derivation of *bayNorm* estimates of the capture efficiencies, and provide pseudo-code for replicating data from the mixed posterior predictive. In Appendix B, we include the details of simulated examples, including data generation, results and additional experiment to investigate misspecification of the mean capture efficiency. In Appendix C, we provide further details on the PAX6 data analysis, including the the filtering process; MCMC summaries; *global* marker genes; *local* marker genes; t-SNE plots and posterior predictive checks. In Appendix D, we show the results corresponding to the set of *important* genes.

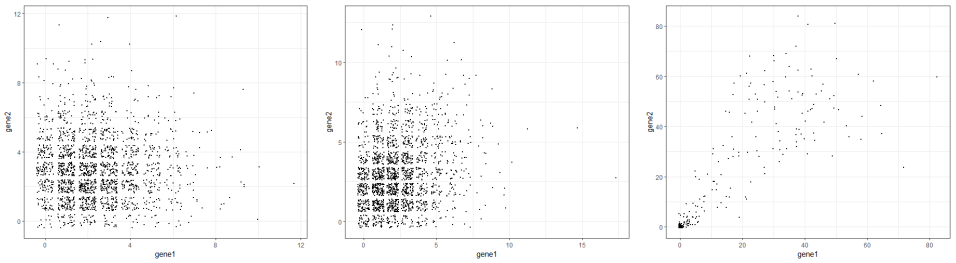
Appendix A. Posterior Inference and Model Insights

Appendix A.1. Gene Dependence through Capture Efficiencies

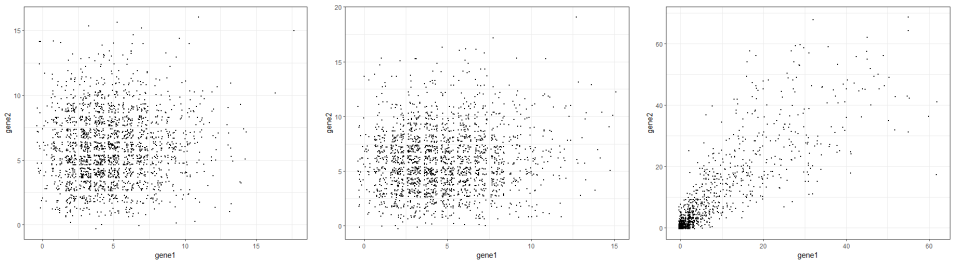
The cell-specific capture efficiencies play an important role in normalization and imputation to obtain the latent counts $y^0_{c,g,d}$. Following other fields, where under-reporting of counts also occurs, we alleviate identifiability issues by employing an informative, empirical prior based on the *bayNorm* estimates and global mean capture efficiency. As suggested by one reviewer, the capture efficiencies also play an interesting role in inducing dependence across the genes. Specifically, for each cell, dependence across genes is obtained after integrating over the capture efficiency $\beta_{c,d}$:

$$p(y_{c,\cdot,d} \mid z_{c,d}, \boldsymbol{\mu}_j, \boldsymbol{\phi}_j) = \int \prod_{g=1}^{G} p(y_{c,g,d} \mid z_{c,d}, \mu_{j,g}, \phi_{j,g}, \beta_{c,d}) \pi(\beta_{c,d}) d\beta_{c,d},$$

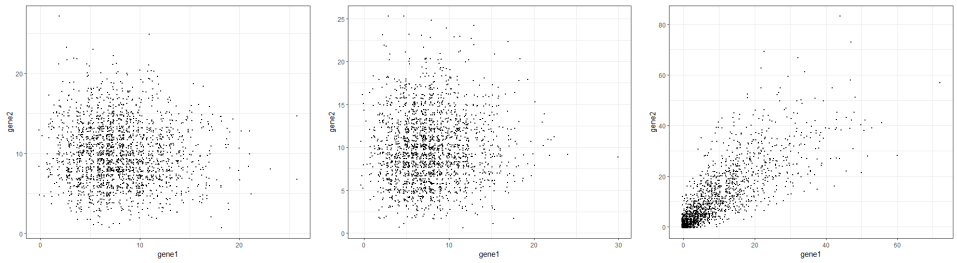
where $\beta_{c,d} \sim \text{Beta}(a_d^\beta, b_d^\beta)$. While the integral above is not available in closed form, we can simulate to understand better the induced dependence. For two genes with mean expression and dispersion set to $(40, 10)$ and $(50, 20)$ respectively, we simulate $y_{c,g}$ for $C = 2,000$ cells and under different scenarios for the hyperparameters a_d^β and b_d^β. The results are shown in Figure A.20, with rows corresponding to increasing prior mean of the capture efficiencies, equal to $0.06, 0.12, 0.2$, and columns corresponding to increasing prior variance, equal to $0, 0.0004, 0.05$. For larger variance, higher dependence is evident, but we note that for the middle column (with variance similar to the empirical variance from the bayNorm estimates), no clear dependence structure is observed.



(a) Independent: $\mathbb{E}(\beta_c) = 0.06$, $\mathbb{V}(\beta_c) = 0.06$ (b) $\mathbb{E}(\beta_c) = 0.06$, $\mathbb{V}(\beta_c) = 0.0004$ (c) $\mathbb{E}(\beta_c) = 0.06$, $\mathbb{V}(\beta_c) = 0.05$



(d) Independent: $\mathbb{E}(\beta_c) = 0.12$, $\mathbb{V}(\beta_c) = 0.12$ (e) $\mathbb{E}(\beta_c) = 0.12$, $\mathbb{V}(\beta_c) = 0.0004$ (f) $\mathbb{E}(\beta_c) = 0.12$, $\mathbb{V}(\beta_c) = 0.05$



(g) Independent: $\mathbb{E}(\beta_c) = 0.2$, $\mathbb{V}(\beta_c) = 0.2$ (h) $\mathbb{E}(\beta_c) = 0.2$, $\mathbb{V}(\beta_c) = 0.0004$ (i) $\mathbb{E}(\beta_c) = 0.2$, $\mathbb{V}(\beta_c) = 0.05$

Figure A.20: Simulated counts for $C = 2,000$ cells and two genes to examine dependence induced through the capture efficiencies. Rows correspond to increasing prior mean of the capture efficiencies, and column correspond to increasing prior variance.

Appendix A.2. MCMC

In this section, we describe the MCMC algorithm for posterior inference with the NormHDP model. In the following, we assume the relationship between the mean and dispersion parameters is linear on the log scale. Let $\mathbf{Z} = (z_{c,d})_{c=1,d=1}^{C_d,D}$; $\mathbf{Y} = (y_{c,d,g})_{c=1,d=1,g=1}^{C_d,D,G}$; $\mathbf{p}_d = (p_{1,d}^J, \ldots, p_{J,d}^J)$; $\mathbf{p} = (p_1^J, \ldots, p_J^J)$; $\boldsymbol{\beta} = (\beta_{c,d})_{c=1,d=1}^{C_d,D}$. The posterior of interest is:

$$\begin{aligned}
\pi(\mathbf{Z}, \mathbf{p}_d, \mathbf{p}, \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*, \boldsymbol{\beta}, \alpha, \alpha_0, \mathbf{b}, a_\phi^2 | \mathbf{Y}) \propto & \prod_{j=1}^{J} \prod_{(c,d):z_{c,d}=j} \prod_{g=1}^{G} \mathrm{NB}(y_{c,d,g} \mid \mu_{j,g}^* \beta_{c,d}, \phi_{j,g}^*) \\
& * \prod_{j=1}^{J} \prod_{d=1}^{D} p_{j,d}^{N_{j,d}} * \prod_{d=1}^{D} \mathrm{Dir}(\mathbf{p}_d \mid \alpha \mathbf{p}) * \mathrm{Dir}\left(\mathbf{p} \mid \frac{\alpha_0}{J}, \ldots, \frac{\alpha_0}{J}\right) \\
& * \prod_{j=1}^{J} \prod_{g=1}^{G} \mathrm{LN}(\mu_{j,g}^* \mid m_u, a_u^2) \mathrm{LN}(\phi_{j,g}^* \mid b_0 + b_1 \log(\mu_{j,g}^*), a_\phi^2) \\
& * \prod_{d=1}^{D} \prod_{c=1}^{C_d} \mathrm{Beta}(\beta_{c,d} \mid a_d^\beta, b_d^\beta) * \mathrm{Gam}(\alpha \mid 1, 1) * \mathrm{Gam}(\alpha_0 \mid 1, 1) \\
& * \mathrm{N}(\mathbf{b} \mid \mathbf{m}_b, a_\phi^2 \mathbf{V}_b) * \mathrm{IG}(a_\phi^2 \mid \nu_1, \nu_2),
\end{aligned}$$

where $N_{j,d} = \sum_{c=1}^{C_d} \mathbf{1}(z_{c,d} = j)$ is the number of cells in component j in dataset d. A Markov chain Monte Carlo (MCMC) algorithm is developed for full posterior inference. The algorithm is a Gibbs sampler which produces asymptotically exact samples from the posterior by iteratively sampling the parameters in blocks corresponding to the:

- allocation variables $\mathbf{Z} | \mathbf{p}_d, \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*, \mathbf{Y}, \boldsymbol{\beta}$,
- dataset-specific component probabilities $\mathbf{p}_d | \mathbf{Z}, \mathbf{p}, \alpha$,
- component probabilities $\mathbf{p} | \mathbf{p}_{1:D}, \alpha_0$,

- unique parameters $\boldsymbol{\mu}_j^*, \boldsymbol{\phi}_j^* \mid \mathbf{Z}, \mathbf{b}, a_\phi^2, \mathbf{Y}, \boldsymbol{\beta}$,
- concentration parameters $\alpha|\mathbf{p}_{1:D}, \mathbf{p}$ and $\alpha_0|\mathbf{p}$,
- mean-dispersion hyperparameters $\mathbf{b}, a_\phi^2|\boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*$,
- capture efficiencies $\boldsymbol{\beta}|\mathbf{Y}, \mathbf{Z}, \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*$.

The complexity of the algorithm is dominated by the update of the allocation variables, which is of order $\mathcal{O}(\text{sum}(C)JG)$. In the following subsections, we describe the update for each block of parameters.

Appendix A.2.1. Mean-dispersion Hyperparameters

The full conditional distribution for the mean-dispersion hyperparameters in the linear case is:

$$\pi(\mathbf{b}, \alpha_\phi^2|\boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*) \propto \text{N}(\mathbf{b}|\mathbf{m}_b, \alpha_\phi^2\mathbf{V}_b) * \text{IG}(\alpha_\phi^2|v_1, v_2) * \prod_{j=1}^{J}\prod_{g=1}^{G} \text{logN}(\phi_{j,g}^*|b_0 + b_1\log(\mu_{j,g}^*), \alpha_\phi^2)$$

$$\propto \left(\frac{1}{\alpha_\phi^2}\right)^{v_1+2+\frac{JG}{2}} \exp\left(-\frac{1}{\alpha_\phi^2}\left[\frac{1}{2}\sum_{j=1}^{J}\sum_{g=1}^{G}(\ln(\phi_{j,g}^*) - b_0 - b_1\log(\mu_{j,g}^*))^2 + v_2 + \frac{1}{2}[(b_0 - m_{b0})^2 + (b_1 - m_{b1})^2]\right]\right). \tag{A.1}$$

Following eq. (A.1), the full conditional for $\mathbf{b}$ conditioning on α_ϕ^2 is:

$$\pi(\mathbf{b}|\boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*, \alpha_\phi^2) \propto \exp\left(-\frac{1}{2\alpha_\phi^2}\left[\sum_{j=1}^{J}\sum_{g=1}^{G}(\ln(\phi_{j,g}^*) - b_0 - b_1\log(\mu_{j,g}^*))^2 + [(b_0 - m_{b0})^2 + (b_1 - m_{b1})^2]\right]\right).$$

Hence we have the full conditional:

$$\mathbf{b}|\boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*, \alpha_\phi^2 \sim \text{N}(\tilde{\mathbf{m}}_b, \alpha_\phi^2\tilde{\mathbf{V}}_b), \tag{A.2}$$

where

$$\tilde{\mathbf{m}}_b = \left(\sum_{j=1}^{J} \tilde{\boldsymbol{\mu}}_j^T \tilde{\boldsymbol{\mu}}_j + I \right)^{-1} \left(\sum_{j=1}^{J} \tilde{\boldsymbol{\mu}}_j^T \ln(\boldsymbol{\phi}_j^*) + \mathbf{m}_b \right),$$

$$\tilde{\mathbf{V}}_b = \left(\sum_{j=1}^{J} \tilde{\boldsymbol{\mu}}_j^T \tilde{\boldsymbol{\mu}}_j + I \right)^{-1},$$

and

$$\ln(\boldsymbol{\phi}_j^*) = \begin{bmatrix} \ln(\phi_{j,1}^*) \\ \vdots \\ \ln(\phi_{j,G}^*) \end{bmatrix}, \quad \tilde{\boldsymbol{\mu}}_j = \begin{bmatrix} 1 & \log(\mu_{j,1}^*) \\ \vdots & \vdots \\ 1 & \log(\mu_{j,G}^*) \end{bmatrix}.$$

And following eq. (A.1), the full conditional for α_ϕ^2 is:

$$\begin{aligned} \pi(\alpha_\phi^2 | \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*) &= \int \pi(\mathbf{b}, \alpha_\phi^2 | \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^*)\, d\mathbf{b} \\ &\propto \int \left(\frac{1}{\alpha_\phi^2} \right)^{v_1+1} \exp\left(-\frac{v_2}{\alpha_\phi^2} \right) \left(\frac{1}{\alpha_\phi^2} \right)^{JG/2} \left(\frac{1}{\alpha_\phi^2} \right) \\ &\quad * \exp\left(-\frac{1}{2\alpha_\phi^2} \left[(\mathbf{b} - \tilde{\mathbf{m}}_b)^T \tilde{\mathbf{V}}_b^{-1} (\mathbf{b} - \tilde{\mathbf{m}}_b) - \tilde{\mathbf{m}}_b^T \tilde{\mathbf{V}}_b^{-1} \tilde{\mathbf{m}}_b + \sum_{j=1}^{J} \ln(\boldsymbol{\phi}_j^*)^T \ln(\boldsymbol{\phi}_j^*) + \sum \mathbf{m}_b^2 \right] \right) d\mathbf{b}. \end{aligned}$$

Thus, we have:

$$\alpha_\phi^2 | \boldsymbol{\mu}_{1:J}^*, \boldsymbol{\phi}_{1:J}^* \sim \text{IG}\left(\tilde{v_1}, \tilde{v_2} \right), \tag{A.3}$$

where

$$\begin{aligned} \tilde{v_1} &= v_1 + JG/2, \\ \tilde{v_2} &= v_2 + \frac{1}{2} \left(\sum_{j=1}^{J} \ln(\boldsymbol{\phi}_j^*)^T \ln(\boldsymbol{\phi}_j^*) - \tilde{\mathbf{m}}_b^T \tilde{\mathbf{V}}_b^{-1} \tilde{\mathbf{m}}_b + \sum \mathbf{m}_b^2 \right). \end{aligned}$$

For Gibbs sampling, at each iteration, we first simulate α_ϕ^2 from the Inverse-Gamma distribution in eq. (A.3) and then conditioned on this value, simulate $\mathbf{b}$ from the Normal distribution in eq. (A.2). Details are given in Algorithm 1.

Algorithm 1 Simulation of Mean-dispersion Hyperparameters

Require: $\boldsymbol{\mu}^*_{1:J}, \boldsymbol{\phi}^*_{1:J}$

Ensure: $\alpha^2_\phi, \mathbf{b}$

$A \leftarrow$ Identity matrix of dimension 2;

$B \leftarrow$ Matrix of $\mathbf{m}_b$;

$C \leftarrow 0$;

for $j = 1, \ldots, J$ **do**

$A = A + \tilde{\boldsymbol{\mu}}_j^T \tilde{\boldsymbol{\mu}}_j; \quad B = B + \tilde{\boldsymbol{\mu}}_j^T \ln(\boldsymbol{\phi}^*_j); \quad C = C + \ln(\boldsymbol{\phi}^*_j)^T \ln(\boldsymbol{\phi}^*_j);$

end for

$\tilde{\mathbf{V}}_b \leftarrow A^{-1}; \quad \tilde{\mathbf{m}}_b \leftarrow \tilde{\mathbf{V}}_b A; \quad \tilde{v}_1 \leftarrow v_1 + JG/2; \quad \tilde{v}_2 \leftarrow v_2 + 1/2 * (C - \tilde{\mathbf{m}}_b^T \tilde{\mathbf{V}}_b^{-1} \tilde{\mathbf{m}}_b + \sum \mathbf{m}_b^2)$;

Simulate α^2_ϕ and $\mathbf{b}$ from the NIG in eq. A.3 and A.2 using the above parameters.

Appendix A.2.2. Allocation Variables

The full conditional for the allocation variables is:

$$\pi(\mathbf{Z}|\mathbf{P}_{1:D}, \boldsymbol{\mu}^*_{1:J}, \boldsymbol{\phi}^*_{1:J}, \mathbf{y}, \boldsymbol{\beta}) \propto \prod_{j=1}^{J} \prod_{(c,d):z_{c,d}=j} \prod_{g=1}^{G} \mathrm{NB}(y_{c,g,d}|\mu^*_{j,g}\beta_{c,d}, \phi^*_{j,g}) \prod_{j=1}^{J} \prod_{d=1}^{D} p_{j,d}^{N_{j,d}},$$

where $N_{j,d}$ is the total number of cells in component j in dataset d. Thus, since the allocations are conditionally independent, we have the full conditional of z for cell c in dataset d as:

$$z_{c,d} \sim \mathrm{Cat}\left(\pi_{c,d,1}, \ldots, \pi_{c,d,J}\right), \tag{A.4}$$

where $\pi_{c,d,j} = \pi(z_{c,d} = j|\mathbf{P}_{1:D}, \boldsymbol{\mu}^*_{1:J}, \boldsymbol{\phi}^*_{1:J}, \mathbf{Y}, \boldsymbol{\beta})$. To avoid numerical errors, we employ the log-sum trick to compute the probabilities:

$$\pi_{c,d,j} = \frac{\exp(\log(\tilde{\pi}_{c,d,j}) + \log(K)}{\sum_{j=1}^{J} \exp(\log(\tilde{\pi}_{c,d,j}) + \log(K))}.$$

where

$$\tilde{\pi}_{c,d,j} = \prod_{g=1}^{G} \mathrm{NB}(y_{c,g,d}|\mu^*_{j,g}\beta_{c,d}, \phi^*_{j,g})p_{j,d},$$

$$\log(K) = -\max_j \log(\tilde{\pi}_{c,d,j}).$$

Details are given in Algorithm 2.

Algorithm 2 Simulation of Allocation Variables

Require: $\mathbf{P}_{1:D}, \boldsymbol{\mu}^*_{1,J}, \boldsymbol{\phi}^*_{1,J}, \mathbf{y}, \boldsymbol{\beta}$

Ensure: $\mathbf{Z}$

$Z \leftarrow$ a list of length D;

for $d = 1, \ldots, D$ and $c = 1, \ldots, C_d$ **do**

Compute $\tilde{\pi}_{c,d,j}$ for each j; and K,

Use $\tilde{\pi}_{c,d,j}$ and K to compute $\pi_{c,d,j}$ and simulate $z_{c,d}$ from the Categorical distribution in eq. (A.4).

end for

Appendix A.2.3. Dataset-specific Component Probabilities

The full conditional distribution for the dataset-specific component probabilities is:

$$\pi(\mathbf{p}_d|\mathbf{Z}, \mathbf{p}, \alpha) \propto \prod_{j=1}^{J} p_{j,d}^{N_{j,d}} \mathrm{Dir}(\mathbf{p}_d|\alpha\mathbf{p}) \propto \prod_{j=1}^{J} p_{j,d}^{N_{j,d}+\alpha p_j - 1}.$$

Hence, $\mathbf{p}_d|\mathbf{Z}, \mathbf{p}, \alpha$ follows a Dirichlet distribution with parameters equals to $N_{j,d} + \alpha p_j$, for $j = 1, \ldots, J$. Details are given in Algorithm 3.

Algorithm 3 Simulation of Dataset-specific Component Probabilities

Require: $\mathbf{Z}, \mathbf{P}, \alpha$

Ensure: $\mathbf{p}_{1:D}$

$\mathbf{p}_{1:D} \leftarrow$ Matrix with J rows and D columns;

for $d = 1, \ldots, D$ **do**

$\tilde{\boldsymbol{\alpha}} \leftarrow \text{table}(\mathbf{Z}_d) + \alpha\mathbf{p}$

Simulate $\mathbf{p}_d$ from the Dirichlet distribution using the updated parameters $\tilde{\boldsymbol{\alpha}}$.

end for

Appendix A.2.4. Component Probabilities

The full conditional distribution for the component probabilities is:

$$\begin{aligned}\pi(\mathbf{p}|\mathbf{p}_{1:D}, \alpha_0, \alpha) &\propto \text{Dir}(\mathbf{p}|\frac{\alpha_0}{J}, \ldots, \frac{\alpha_0}{J}) * \prod_{d=1}^{D} \text{Dir}(\mathbf{p}_d|\alpha\mathbf{p}) \\ &\propto \left[\prod_{j=1}^{J} p_j^{\frac{\alpha_0}{J}-1}\right] \prod_{d=1}^{D} \frac{1}{B(\alpha\mathbf{p})} \prod_{j=1}^{J} p_{j,d}^{\alpha p_j} * \mathbf{1}\left(p_j > 0, \forall j, \quad \sum_{j=1}^{J} p_j = 1\right).\end{aligned} \tag{A.5}$$

As the full conditional distribution has no closed-form, we will use adaptive Metropolis-Hastings (Griffin and Stephens, 2013) to obtain posterior samples of $\mathbf{p}$. The log of the full conditional in eq. (A.5) can be written as:

$$\log \pi(\mathbf{p}|\ldots) = \sum_{j=1}^{J} \left[\left(\frac{\alpha_0}{J} - 1\right) \log(p_j)\right] + \sum_{d=1}^{D}\sum_{j=1}^{J} \left[\alpha p_j \log(p_{j,d}) - \log\Gamma(\alpha p_j)\right] + \text{const.}$$

Adaptive Metropolis-Hastings for $\mathbf{p}$. In the following, we describe the steps of the adaptive Metropolis-Hastings algorithm.

1. Note that since $p_j > 0$ for all $j = 1, \ldots, J$ and $\sum_{j=1}^{J} p_j = 1$, we apply

the following transformation:

$$\mathbf{p} \in \triangle^{J-1} \mapsto \mathbf{x} \in \mathbb{R}^{J-1}$$

where

$$x_j = t(p_j) := \log\left(\frac{p_j}{p_J}\right), \quad j = 1, \ldots, J-1.$$

Note that the reverse of the transformation is:

$$p_j = \frac{e^{x_j}}{1 + \sum_{j=1}^{J-1} e^{x_j}}, \quad j = 1, \ldots, J-1.$$

2. For $t \leq 100$, $\mathbf{x}_{new}$ is simulated from $\mathrm{N}(\mathbf{x}_{old}, \mathbf{I}_d)$. If $t > 100$, $\mathbf{x}_{new}$ is simulated from $\mathrm{N}(\mathbf{x}_{old}, 2.4^2/d * (\mathbf{\Sigma}_{t-1} + \epsilon \mathbf{I}_d))$, where $\mathbf{\Sigma}_{t-1}$ is the current estimate of covariance structure of $\mathbf{x}$ based on the first $t-1$ samples; d is the length of the parameters of interest, i.e. $d = J-1$; and an epsilon is small constant, i.e. $\epsilon = 0.01$.

3. To avoid re-computing $\mathbf{\Sigma}_t$ at each iteration, we compute $\mathbf{\Sigma}_t$ based on two statistics: $\tilde{\mathbf{S}}_t$ and $\mathbf{m}_t$, which can be sequentially updated. These statistics are defined as:

$$\tilde{\mathbf{S}}_t = \begin{bmatrix} \sum_{i=1}^t x_{1,i}^2 & \sum_{i=1}^t x_{1,i}x_{2,i} & \cdots & \sum_{i=1}^t x_{1,i}x_{J-1,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^t x_{J-1,i}x_{1,i} & \sum_{i=1}^t x_{J-1,i}x_{2,i} & \cdots & \sum_{i=1}^t x_{J-1,i}^2 \end{bmatrix},$$

$$\mathbf{m}_t = [\sum_{i=1}^t x_{1,i}, \ldots, \sum_{i=1}^t x_{J-1,i}]^T.$$

We can express $\mathbf{\Sigma}_t$ as:

$$\mathbf{\Sigma}_t = \frac{1}{t-1}\tilde{\mathbf{S}}_t - \frac{t}{t-1}\mathbf{m}_t\mathbf{m}_t^T,$$

where we sequentially update the required statistics:

$$\tilde{\mathbf{S}}_t = \tilde{\mathbf{S}}_{t-1} + \mathbf{x}_t \mathbf{x}_t^T;$$
$$\mathbf{m}_t = \left(1 - \frac{1}{t}\right) \mathbf{m}_t + \frac{1}{t} \mathbf{x}_t.$$

4. To evaluate the proposal distribution, we are required to compute the Jacobian of the transformation. More specifically, the proposal is:

$$q_n(\mathbf{p}_{new}|\mathbf{p}_{old}) = q_n\left(\mathbf{t}(\mathbf{p}_{new})|\mathbf{t}(\mathbf{p}_{old})\right) |J_{\mathbf{t}(\mathbf{p}_{new})}|,$$

where

$$J_{\mathbf{t}(\mathbf{p})} = \begin{pmatrix} \frac{dt_1}{dp_1} & \frac{dt_2}{dp_1} & \dots & \frac{dt_{J-1}}{dp_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dt_1}{dp_{J-1}} & \frac{dt_2}{dp_{J-1}} & \dots & \frac{dt_{J-1}}{dp_{J-1}} \end{pmatrix}$$
$$= \begin{pmatrix} \frac{1}{p_J} & \dots & \frac{1}{p_J} \\ \vdots & \ddots & \vdots \\ \frac{1}{p_J} & \dots & \frac{1}{p_J} \end{pmatrix} + \begin{pmatrix} \frac{1}{p_1} & 0 & \dots & 0 \\ 0 & \frac{1}{p_2} & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \frac{1}{p_{J-1}} \end{pmatrix} = B + A.$$

Since $\det(A+B) = \det(A) + \det(B) + Tr(A^{-1}B)\det(A)$, and in our case, we have $\det(B) = 0$ and $\det(A) = \prod_{j=1}^{J-1} \frac{1}{p_j}$, hence $\det(A+B) = \prod_{j=1}^{J-1} \frac{1}{p_j} + [1-p_J] \prod_{j=1}^{J} \frac{1}{p_j} = \prod_{j=1}^{J} \frac{1}{p_j}$. And taking the log of the determinant of the Jacobian, we have

$$\log\det J_{\mathbf{t}(\mathbf{P})} = \log\left[\prod_{j=1}^{J} \frac{1}{p_j}\right] = -\sum_{j=1}^{J} \log(p_j).$$

5. Next, we compute the acceptance probability:

$$\alpha(\mathbf{p}_{new}, \mathbf{p}_{old}) = \min\{1, \exp(\tilde{\alpha}(\mathbf{p}_{new}, \mathbf{p}_{old}))\},$$

where

$$\tilde{\alpha}(\mathbf{p}_{new}, \mathbf{p}_{old}) = \log\left[\frac{\pi(\mathbf{p}_{new})}{\pi(\mathbf{p}_{old})}\frac{q_n(\mathbf{p}_{old}|\mathbf{p}_{new})}{q_n(\mathbf{p}_{new}|\mathbf{p}_{old})}\right] = \log\left[\frac{\pi(\mathbf{p}_{new})}{\pi(\mathbf{p}_{old})}\frac{|J_{\mathbf{t}(\mathbf{p}_{old})}|}{|J_{\mathbf{t}(\mathbf{p}_{new})}|}\right]$$
$$= \log\left[\frac{\pi(\mathbf{p}_{new})}{\pi(\mathbf{p}_{old})}\right] - \sum_{j=1}^{J}\log(p_{old,j}) + \sum_{j=1}^{J}\log(p_{new,j}),$$

and $\pi(\cdot)$ is the density of the target distribution.

Appendix A.2.5. Concentration Parameters

The full conditional distribution for α is:

$$\pi(\alpha|\mathbf{p}_{1:D}, \mathbf{p}) \propto \text{Gam}(\alpha|1,1) * \prod_{d=1}^{D}\text{Dir}(\mathbf{p}_d|\alpha\mathbf{p}) \propto \exp(-\alpha)\prod_{d=1}^{D}\frac{1}{B(\alpha\mathbf{p})}\prod_{j=1}^{J}p_{j,d}^{\alpha p_j}$$
$$\propto \exp(-\alpha)\prod_{d=1}^{D}\frac{\Gamma(\alpha)}{\prod_{j=1}^{J}\Gamma(\alpha p_j)}\prod_{j=1}^{J}p_{j,d}^{\alpha p_j} * \mathbf{1}(\alpha > 0). \tag{A.6}$$

The full conditional distribution for α_0 is:

$$\pi(\alpha_0|\mathbf{p}) \propto \text{Gam}(\alpha_0|1,1) * \text{Dir}\left(\mathbf{p}|\frac{\alpha_0}{J}, \ldots, \frac{\alpha_0}{J}\right) \propto \exp(-\alpha_0)\frac{1}{B(\frac{\alpha_0}{J})}\prod_{j=1}^{J}p_j^{\frac{\alpha_0}{J}}$$
$$\propto \exp(-\alpha_0)\frac{\Gamma(\alpha_0)}{[\Gamma(\frac{\alpha_0}{J})]^J}\prod_{j=1}^{J}p_j^{\frac{\alpha_0}{J}} * \mathbf{1}(\alpha_0 > 0). \tag{A.7}$$

We obtain no closed-form distributions for both concentration parameters α and α_0, hence we apply adaptive Metropolis-Hastings to obtain posterior samples. The log of the full condition in eq. (A.6) is:

$$\log\pi(\alpha|\mathbf{p}_{1:D}, \mathbf{p}) = -\alpha + D\log(\Gamma(\alpha)) + \sum_{d=1}^{D}\sum_{j=1}^{J}[\alpha p_j\log(p_{j,d}) - \log(\Gamma(\alpha p_j))] + \text{const.},$$

and in eq. (A.7) is:

$$\log\pi(\alpha_0|\mathbf{p}) = -\alpha_0 + \log(\Gamma(\alpha_0)) - J\log(\Gamma(\frac{\alpha_0}{J})) + \sum_{j=1}^{J}\frac{\alpha_0}{J}\log(p_j) + \text{const.}$$

Adaptive Metropolis-Hastings for α.

1. Since α is always greater than zero, we apply the following transformation to map α to the real axis:

$$\alpha \in \mathbb{R}_+ \mapsto x \in \mathbb{R},$$

   where $x = t(\alpha) = \log(\alpha)$.

2. We apply an adaptive random walk in the transformed space, where the variance of proposal is adapted in similar fashion to Appendix Appendix A.2.4, with the sample variance computed by sequentially updating the required statistics.

3. To compute the Jacobian of the transformation, we differentiate $t(\alpha)$ with respect to α:

$$\frac{dt(\alpha)}{d\alpha} = \frac{d\log(\alpha)}{d\alpha} = \frac{1}{\alpha}.$$

4. Hence the acceptance probability simplifies to:

$$\alpha(\alpha_{new}, \alpha_{old}) = \min\{1, \tilde{\alpha}(\alpha_{new}, \alpha_{old})\},$$

   where

$$\tilde{\alpha}(\alpha_{new}, \alpha_{old}) = \log\left[\frac{\pi(\alpha_{new})}{\pi(\alpha_{old})}\frac{q_n(\alpha_{old}|\alpha_{new})}{q_n(\alpha_{new}|\alpha_{old})}\right] = \log\left[\frac{\pi(\alpha_{new})}{\pi(\alpha_{old})}\frac{\alpha_{new}}{\alpha_{old}}\right].$$

Appendix A.2.6. Unique Parameters

The full conditional distribution for $\boldsymbol{\mu}^*$ and $\boldsymbol{\phi}^*$ is:

$$\pi(\boldsymbol{\mu}_j^*, \boldsymbol{\phi}_j^*|\mathbf{Z}, \mathbf{b}, \alpha_\phi^2, \mathbf{Y}, \boldsymbol{\beta}) \propto \prod_{g=1}^{G} \text{logN}(\mu_{j,g}^*|m_u, \alpha_\mu^2)\ \text{logN}(\phi_{j,g}^*|b_0 + b_1\log(\mu_{j,g}^*), \alpha_\phi^2)$$

$$* \prod_{(c,d):z_{c,d}=j} \prod_{g=1}^{G} \text{NB}(y_{c,g,d}|\mu_{j,g}^*\beta_{c,d}, \phi_{j,g}^*).$$

Since the genes are conditionally independent, we can write the full condition for $\mu^*_{j,g}$ and $\phi^*_{j,g}$ as:

$$\pi(\mu^*_{j,g}, \phi^*_{j,g} | \mathbf{Z}, \mathbf{b}, \alpha^2_\phi, \mathbf{Y}, \beta) \propto \prod_{(c,d):z_{c,d}=j} \binom{y_{c,g,d} + \phi^*_{j,g} - 1}{\phi^*_{j,g} - 1} \left(\frac{\phi^*_{j,g}}{\mu^*_{j,g}\beta_{c,d} + \phi^*_{j,g}}\right)^{\phi^*_{j,g}} \left(\frac{\mu^*_{j,g}}{\mu^*_{j,g}\beta_{c,d} + \phi^*_{j,g}}\right)^{y_{c,g,d}} \left(\frac{1}{\mu^*_{j,g}\phi^*_{j,g}}\right) \exp\left(-\frac{1}{2\alpha^2_\mu}(\ln \mu^*_{j,g} - m_u)^2 - \frac{1}{2\alpha^2_\phi}(\ln \phi^*_{j,g} - (b_0 + b_1 \log \mu^*_{j,g}))^2\right). \tag{A.8}$$

The full conditional distribution has no closed-form, hence we apply adaptive Metropolis-Hastings to obtain posterior samples. We first compute the log-likelihood of eq. (A.8):

$$\begin{aligned}\log \pi\left(\mu^*_{j,g}, \phi^*_{j,g} | \dots\right) = &- \log(\mu^*_{j,g}\phi^*_{j,g}) - \frac{1}{2\alpha^2_\mu}(\ln \mu^*_{j,g} - m_u)^2 \\ &- \frac{1}{2\alpha^2_\phi}(\ln \phi^*_{j,g} - (b_0 + b_1 \log \mu^*_{j,g}))^2 \\ &+ \sum_{(c,d):z_{c,d}=j} \left[\log \binom{y_{c,g,d} + \phi^*_{j,g} - 1}{\phi^*_{j,g} - 1} + \phi^*_{j,g} \log\left(\frac{\phi^*_{j,g}}{\mu^*_{j,g}\beta_{c,d} + \phi^*_{j,g}}\right) \right. \\ &\left. + y_{c,g,d} \log\left(\frac{\mu^*_{j,g}}{\mu^*_{j,g}\beta_{c,d} + \phi^*_{j,g}}\right)\right] + \text{const.}\end{aligned}$$

Adaptive Metropolis Hastings for μ and ϕ.

1. Since (μ^*, ϕ^*) are always positive, we transform to the real axis:

$$(\mu^*, \phi^*) \in \mathbb{R}^2_+ \mapsto \mathbf{x} \in \mathbb{R}^2,$$

   where $t_1(\mu^*, \phi^*) = x_1 = \log(\mu^*)$ and $t_2(\mu^*, \phi^*) = x_2 = \log(\phi^*)$.

2. We apply an adaptive random walk in the transformed space, where the covariance matrix of proposal is adapted in similar fashion to Appendix

Appendix A.2.4, with the sample covariance matrix computed by sequentially updating the required statistics.

3. The Jacobian matrix of the transformation is:

$$J_{\mathbf{t}(\mu^*,\phi^*)} = \begin{pmatrix} \frac{dt_1}{d\mu^*} & \frac{dt_1}{d\phi^*} \\ \frac{dt_2}{d\mu^*} & \frac{dt_2}{d\phi^*} \end{pmatrix} = \begin{pmatrix} \frac{1}{\mu^*} & 0 \\ 0 & \frac{1}{\phi^*} \end{pmatrix};$$

and the log determinant of the Jacobian is:

$$\log\det J_{\mathbf{t}(\mu^*,\phi^*)} = \log\left[\frac{1}{\mu^*\phi^*}\right] = -\log(\mu^*) - \log(\phi^*).$$

4. The corresponding acceptance probability is given by:

$$\alpha((\mu^*,\phi^*)_{new},(\mu^*,\phi^*)_{old}) = \min\{1, \exp[\tilde{\alpha}((\mu^*,\phi^*)_{new},(\mu^*,\phi^*)_{old})]\},$$

where

$$\tilde{\alpha}((\mu^*,\phi^*)_{new},(\mu^*,\phi^*)_{old}) = \log\left[\frac{\pi((\mu^*,\phi^*)_{new})}{\pi((\mu^*,\phi^*)_{old})}\right] - \log(\mu^*_{old}) - \log(\phi^*_{old}) + \log(\mu^*_{new}) - \log(\phi^*_{new}).$$

The above process is repeated for all unique parameters corresponding to occupied components. For non-occupied components, we can sample directly from the prior.

Note the random walk used here is specific to each cluster and gene (j, g). For empty components, i.e. in the case when $\sum_{d=1}^{D} N_{j,d} = 0$, we sample the mean expression and dispersion from the prior. The adaptive covariance matrix is updated in either case.

Appendix A.2.7. Capture Efficiencies

The full conditional distribution for $\boldsymbol{\beta}$ is:

$$\pi(\boldsymbol{\beta}|\mathbf{Y},\mathbf{Z},\mu^*_{1:J},\phi^*_{1:J}) \propto \prod_{j=1}^{J}\prod_{(c,d):z_{c,d}=j}\prod_{g=1}^{G}\mathrm{NB}(y_{c,g,d}|\mu^*_{j,g}\beta_{c,d},\phi^*_{j,g}) \\ * \prod_{d=1}^{D}\prod_{c=1}^{C_d}\mathrm{Beta}(\beta_{c,d}|a^\beta_d,b^\beta_d).$$

Since all β are conditionally independent, they can be sampled in parallel, and we can write the full conditional distribution for cell c and gene g as:

$$\pi(\beta_{c,d}|\mathbf{Y},\mathbf{Z},\mu^*_{1:J},\phi^*_{1:J}) \propto \mathrm{Beta}(\beta_{c,d}|a^\beta_d,b^\beta_d)\prod_{g=1}^{G}\mathrm{NB}(y_{c,g,d}|\mu^*_{j,g}\beta_{c,d},\phi^*_{j,g})$$

$$\propto \left[\prod_{g=1}^{G}\left(\frac{1}{\phi^*_{j,g}+\mu^*_{j,g}\beta_{c,d}}\right)^{\phi^*_{j,g}+y_{c,g,d}}(\beta_{c,d})^{y_{c,g,d}}\right] \\ * (\beta_{c,d})^{a^\beta_d-1}(1-\beta_{c,d})^{b^\beta_d-1}\,\mathbf{1}(\beta_{c,d}\in[0,1]).$$

As no closed-form is obtained, we apply adaptive Metropolis-Hastings to obtain posterior samples. First, the full-conditional on the log-scale is:

$$\log\pi(\beta_{c,d}|\dots) = (a^\beta_d-1)\log(\beta_{c,d}) + (b^\beta_d-1)\log(1-\beta_{c,d}) \\ -\sum_{g=1}^{G}(\phi^*_{j,g}+y_{c,g,d})\log(\phi^*_{j,g}+\mu^*_{j,g}\beta_{c,d}) - y_{c,g,d}\log(\beta_{c,d}) + \text{const.}$$

Adaptive Metropolis-Hastings for β.

1. We apply a logit transformation to transform β into x, where x belongs to the real axis:

$$\beta\in[0,1]\mapsto x\in\mathbb{R},$$

where $t(\beta)=\log(\frac{\beta}{1-\beta})=x$.

2. We apply an adaptive random walk in the transformed space, where the variance of proposal is adapted in similar fashion to Appendix Appendix A.2.4, with the sample variance computed by sequentially updating the required statistics.
3. To Jacobian of the transformation is:

$$\frac{dt(\beta)}{d\beta} = \frac{d}{d\beta}(\log(\beta) - \log(1-\beta)) = \frac{1}{\beta(1-\beta)},$$

and the log of the determinant of the Jacobian is:

$$\log\left(\frac{1}{\beta(1-\beta)}\right) = -\log(\beta) - \log(1-\beta).$$

4. The acceptance probability is given by:

$$\alpha(\beta_{new}, \beta_{old}) = \min\{1, \exp[\tilde{\alpha}(\beta_{new}, \beta_{old})]\},$$

where

$$\tilde{\alpha}(\beta_{new}, \beta_{old}) = \log\left[\frac{\pi(\beta_{new})}{\pi(\beta_{old})}\right] + \log(\beta_{new}) + \log(1-\beta_{new}) - \log(\beta_{old}) - \log(1-\beta_{old}).$$

Appendix A.3. MCMC with Fixed Clustering

After obtaining the optimal clustering which minimizes the posterior expected VI, we subsequently rerun the MCMC algorithm described above, but omitting the step described in Appendix A.2.2 and fixing the allocation variables to the optimal estimate. This allows us to analyze the patterns and uncertainty within each cluster. We note that updating the allocation variables is the most expensive step in the MCMC algorithm; thus, this subsequent run is much faster.

Appendix A.4. bayNorm Estimates of the Capture Efficiencies

In the bayNorm approach, capture efficiencies are estimated using the following approach:

$$\hat{\beta}_{c,d}^{\text{bay}} = \frac{\sum_{g=1}^{G} Y_{c,g,d}}{\frac{1}{C_d}\sum_{c=1,g=1}^{C_d,G} Y_{c,g,d}} \times \lambda,$$

where λ is the mean of estimated capture efficiencies. Under the default setting of bayNorm, $\lambda = 0.06$. The estimates are used to construct empirical priors for the capture efficiencies in our NormHDP model.

Appendix A.5. Mixed Posterior Predictive Checks

The pseudo-code below demonstrates the steps taken to replicate data from the mixed posterior predictive.

Algorithm 4 Replicating data for mixed posterior predictive checks

Require: Number of replicated datasets M, Cell allocations $\mathbf{Z}$, Posterior draws $(\boldsymbol{\mu}_{1:J}^{*(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{b}^{(t)}, a_{\phi}^{2\,(t)})$ for $t = 1, \ldots, T$.

for $m = 1, \ldots M$ **do**

Index $t_m \leftarrow$ randomly sampled from $1 : T$.

Set parameters based on the tth posterior draw:

$$\boldsymbol{\mu}_{1:J}^{*} = \boldsymbol{\mu}_{1:J}^{*(t_m)}, \boldsymbol{\beta} = \boldsymbol{\beta}^{(t_m)}, \mathbf{b} = \mathbf{b}^{(t_m)}, \alpha_{\phi}^{2} = a_{\phi}^{2\,(t_m)}.$$

for $j = 1, \ldots, J$ and $g = 1, \ldots, G$ **do**

Simulate $\boldsymbol{\phi}_{j,g}^{*} \sim \text{logN}(b_0 + b_1 \log(\mu_{j,g}^{*}), \alpha_{\phi}^{2})$.

end for

for $c = 1, \ldots, C_d$, $g = 1, \ldots, G$ and $d = 1, \ldots, D$ **do**

Simulate replicated data: $y_{c,g,d}^{\text{rep},m} \mid z_{c,d} = j \overset{ind}{\sim} \text{NB}(\mu_{j,g}^{*}\beta_{c,d}, \phi_{j,g}^{*})$.

end for

end for

Appendix A.6. Compare LFC in the residual overdispersion

We can also classify global marker genes based on the residuals of dispersions (Eling et al., 2018), where the residual $\epsilon_{j,g}$ is defined as the difference between estimated dispersion $\phi_{j,g}$ and the fitted dispersion based on the regression trend, i.e.: in the linear case, we have

$$\epsilon_{j,g} = \phi_{j,g} - (b_0 + b_1 \log \mu_{j,g}^{*})$$

and the conditionally differentially dispersed genes are the ones with a large proportion of $|\epsilon_{j,g} - \epsilon_{j',g}|$ being greater than a threshold value between some cluster pairs (j, j').

Appendix B. Simulations

Appendix B.1. Simulation 1: Data Generation

In the first simulation, the two datasets are simulated using the following model:

$$\begin{aligned} y_{c,g,d}|z_{c,d} = j, \mu^*_{j,g}, \beta_{c,d}, \phi^*_{j,g} &\sim \text{NB}(\mu^*_{j,g}\beta_{c,d}, \phi^*_{j,g}), \\ z_{c,d}|(p^J_{1,d}, \cdots, p^J_{J,d}) &\sim \text{Cat}(p^J_{1,d}, \cdots, p^J_{J,d}), \\ \mu^*_{j,g} &\sim \text{logN}(0, \alpha^2_\mu), \\ \phi^*_{j,g}|\mu^*_{j,g} &\sim \text{logN}(b_0 + b_1 \log(\mu^*_{j,g}), \alpha^2_\phi), \\ \beta_{c,d} &\sim \text{Beta}(a^\beta_d, b^\beta_d), \end{aligned}$$

where we set $b_1 = 0; b_2 = 3; \alpha^2_\mu = 1; \alpha^2_\phi = 1$ and $a^\beta_d = 1; b^\beta_d = 0.5$ for both datasets. The first dataset contains $C_1 = 50$ cells and the second contains $C_2 = 100$ cells, with $G = 50$ genes. We assume there are $J = 3$ clusters, with true cell proportions $(p_{1,1}, p_{2,1}, p_{3,1}) = (0.6, 0.4, 0)$ for dataset 1 and $(p_{1,2}, p_{2,2}, p_{3,2}) = (0.4, 0, 0.6)$ for dataset 2.

Appendix B.2. Simulation 2: Data Generation

In the second simulation, we assume a non-linear relationship between the mean expressions and dispersions, to assess robustness to misspecification. The two datasets are generated as in Simulation 1 (Appendix B.1), however, we assume the following relationship:

$$\phi^*_{j,g}|\mu^*_{j,g} \sim \text{logN}(4 - 2/\mu_{j,g}, \alpha^2_\phi),$$

and values of α^2_μ; α^2_ϕ; a^β_d and b^β_d are set to be the same as in Simulation 1.

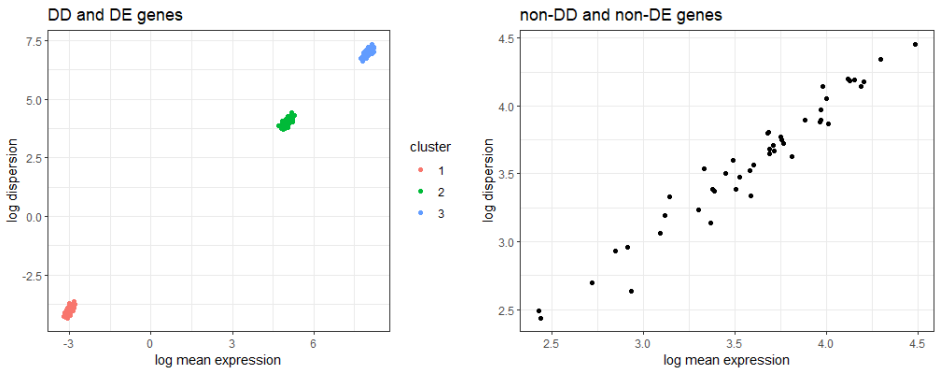


Figure B.21: True mean-dispersion relationships for DE and DD genes (left) and non-DE (also non-DD) genes (right). On the left, different colors are used to represent mean expressions and dispersions from different clusters.

Appendix B.3. Simulation 3: Data Generation

In Simulation 3, we test the ability of our algorithm to detect global differentially expressed and dispersed genes. In this case, we have $G = 150$ genes; $C_1 = 300$ cells in dataset 1 and $C_2 = 400$ cells in dataset 2. We assume that there are $J = 3$ clusters, with dataset-specific allocation proportions $(0.8, 0.2, 0)$ and $(0.8, 0, 0.2)$ for dataset 1 and 2, respectively. For simplicity, we assume the first 70 percent of the genes are DD and DE, specifically, these are the genes with indices 1 to 105.

Based on the results from the real data, non-DE genes tend to be highly expressed. Thus, we set the true mean for non-DE genes $\mu^*_{j,g}$ to μ^*_g, with $\mu^*_g \sim \text{logN}(3.5, 0.5)$. Instead, for DE genes, we set the true mean $\mu^*_{j,g} \sim \text{logN}(m_j, 0.1)$, where $m_{1:J} = (-3, 5, 8)$. We assume all DE genes are also DD and vice versa. For non-DD (and non-DE) genes, we set the true dispersion $\phi^*_{j,g} = \phi^*_g$ with $\phi^*_g \sim \text{logN}(b_0 + b_1 \log(\mu^*_g), 0.1)$, where $b_0 = -1$ and $b_1 = 1$. For DD (and DE) genes, we set the true dispersion $\phi^*_{j,g} \sim \text{logN}(b_0 + b_1 \log(\mu^*_{j,g}), 0.1)$.

The true relationships between mean expressions and dispersions are shown in Figure B.21.

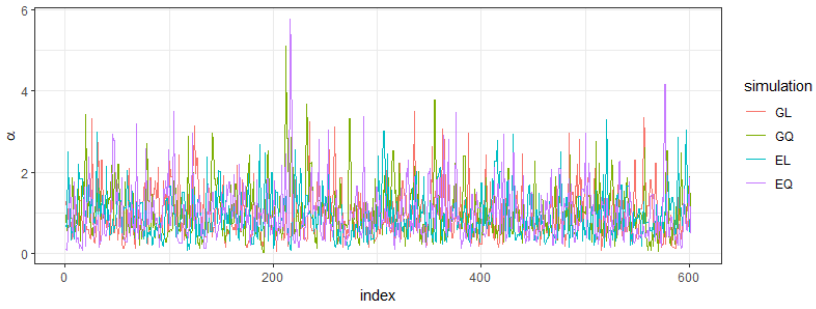
Appendix B.4. Simulations 1 and 2: Results

For both simulations, we investigate prior sensitivity by comparing general priors based on standard hyperparameter values with empirical priors based on hyperparameters specified using initial *bayNorm* estimates (as described in Section 2.2.2). In addition, to enhance flexibility, we consider both linear and quadratic relationships in the prior model for the mean-variance relationship. This results in four settings for the proposed NormHDP model: linear model with general priors (GL), quadratic model with general priors (GQ), linear model with empirical priors (EL) and quadratic model with empirical priors (EQ).

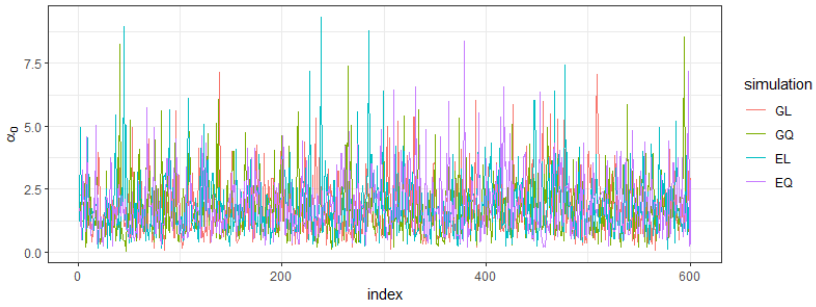
Traceplots of concentration parameters and α_ϕ^2 for Simulation 1 and Simulation 2 provided in Figures B.22 and B.23 suggest convergence:

In Figure B.24 and B.25, we compare the true similarity matrix with the posterior similarity matrix obtained from the different prior (general vs. empirical) and model (linear vs. quadratic) choices, for Simulation 1 and 2 respectively. In all cases, the results highlight that the true clustering structure is well recovered.

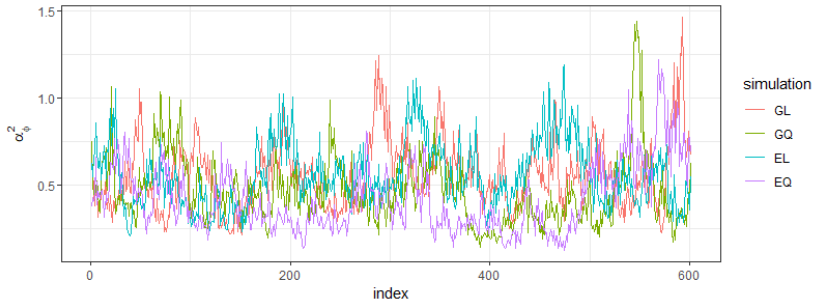
Figure B.26 and B.27 compares the true capture efficiencies with estimated capture efficiencies for Simulation 1 and Simulation 2, respectively. The figures highlight that the informative prior choice helps to mitigate indentifiability issues.



(a) Trace plot of α.

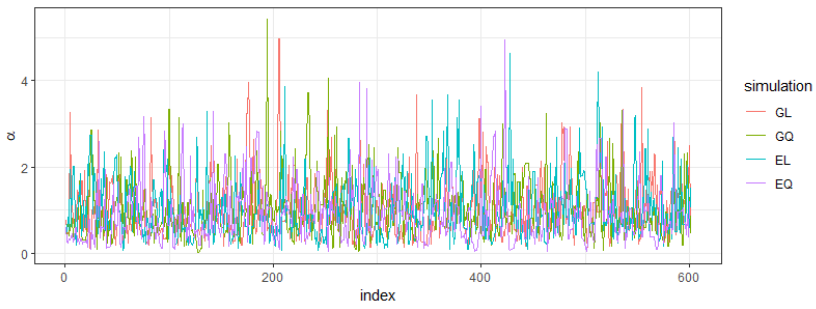


(b) Trace plot of α_0.

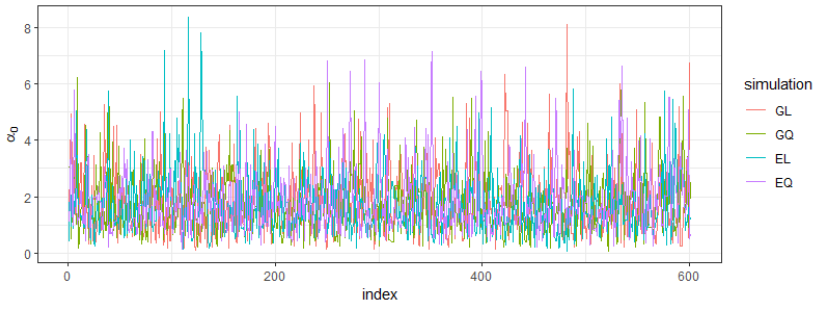


(c) Trace plot of α^2_ϕ.

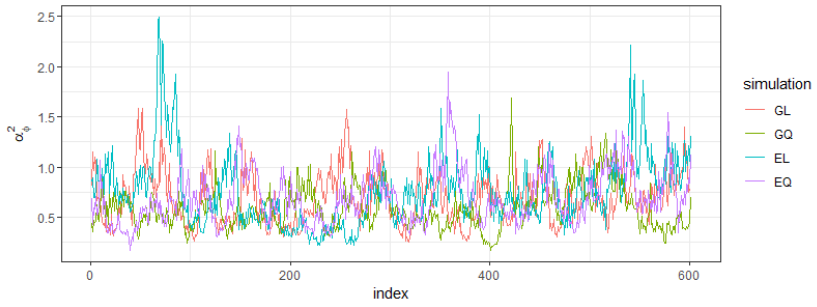
Figure B.22: Traceplots of the concentration parameters and regression parameter α^2_ϕ for Simulation 1. Colors represent the chains under the different prior and model choices.



(a) Trace plot of α.



(b) Trace plot of α_0.



(c) Trace plot of α_ϕ^2.

Figure B.23: Traceplots of the concentration parameters and regression parameter α_ϕ^2 for Simulation 2. Colors represent the chains under the different prior and model choices.

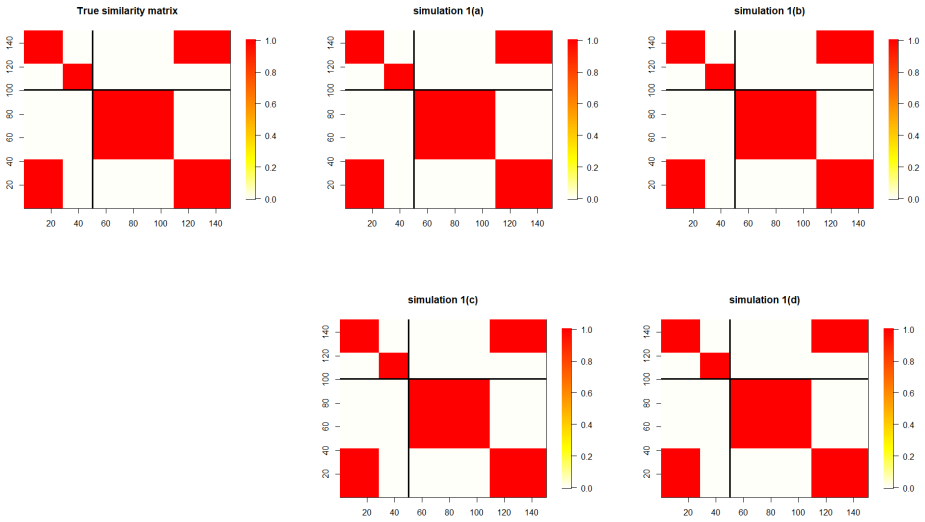


Figure B.24: Comparison of the true and posterior similarity matrix for Simulation 1, with the true similarity matrix (left) and the posterior similar matrix with (a) the linear model and general prior, (b) the linear model and empirical prior, (c) the quadratic model and general prior, and (d) the quadratic model and empirical prior.

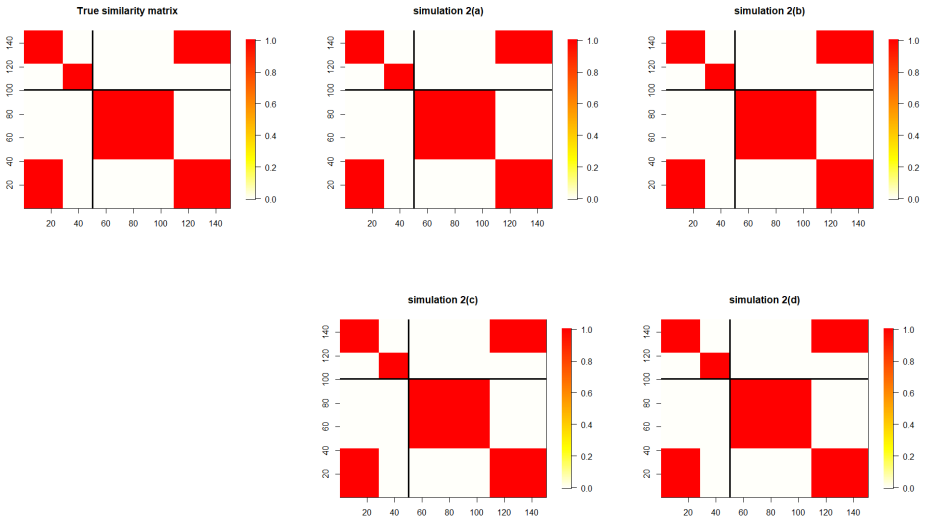


Figure B.25: Comparison of the true and posterior similarity matrix for Simulation 2, with the true similarity matrix (left) and the posterior similar matrix with (a) the linear model and general prior, (b) the linear model and empirical prior, (c) the quadratic model and general prior, and (d) the quadratic model and empirical prior.

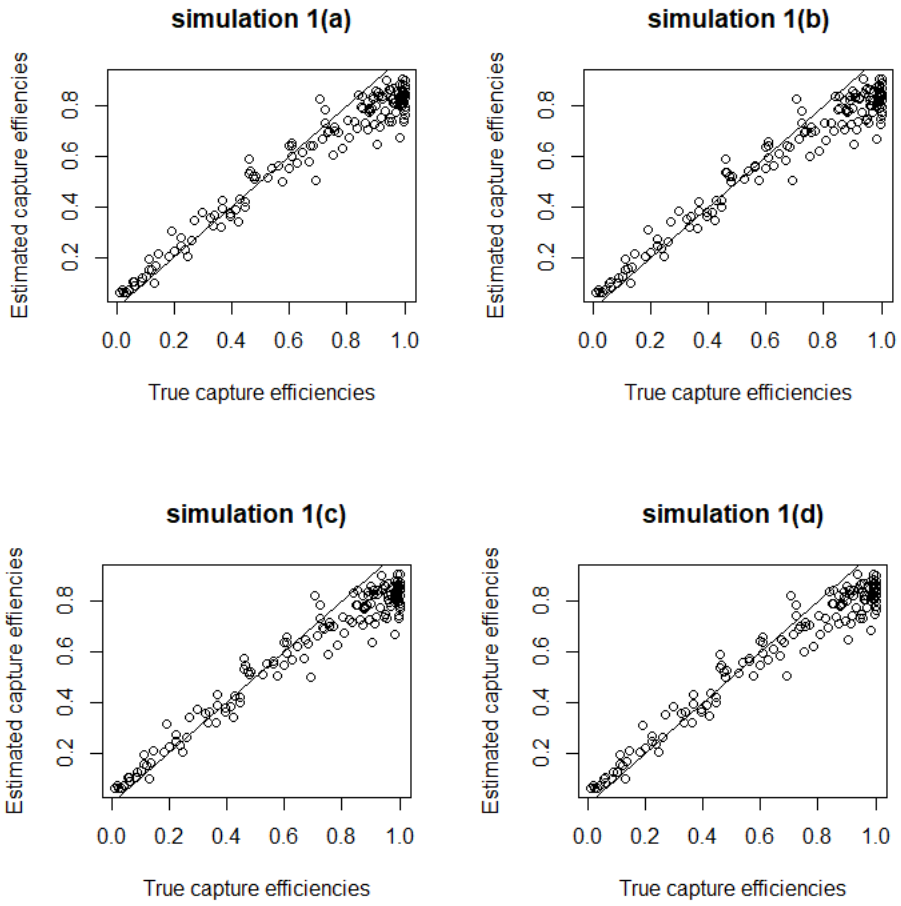


Figure B.26: Comparison between the true capture efficiencies and the posterior mean of capture efficiencies for Simulation 1 with (a) the linear model and general prior, (b) the linear model and empirical prior, (c) the quadratic model and general prior, and (d) the quadratic model and empirical prior..

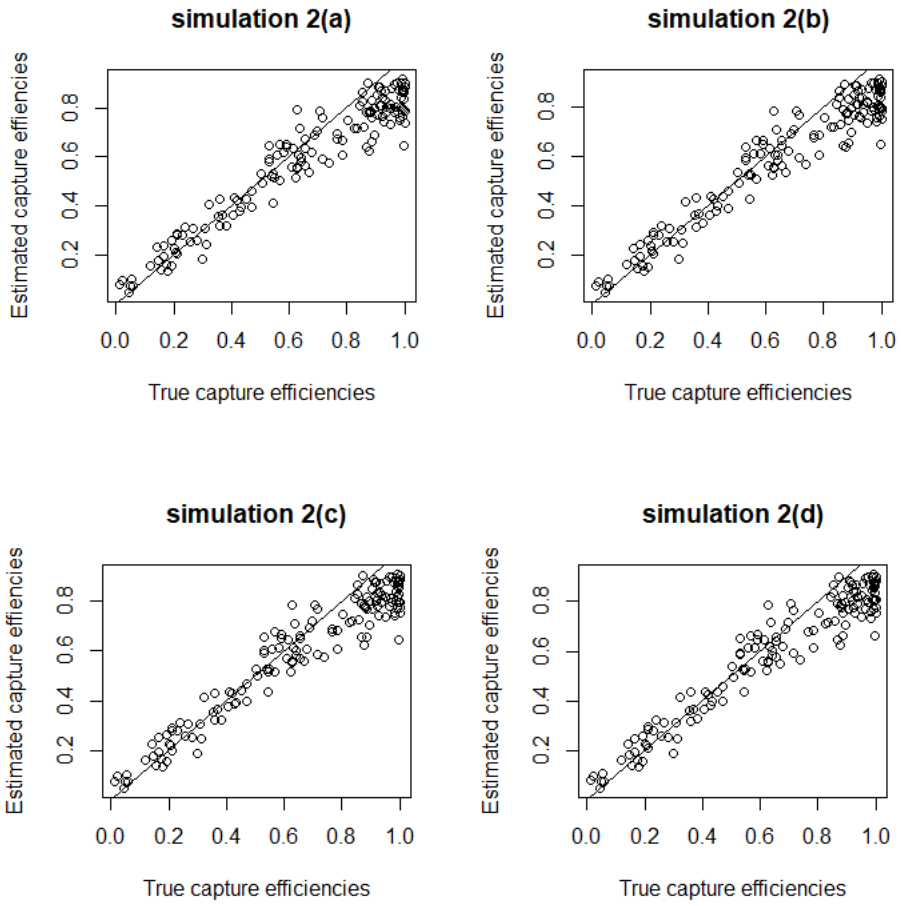


Figure B.27: Comparison between the true capture efficiencies and the posterior mean of capture efficiencies for Simulation 2 with (a) the linear model and general prior, (b) the linear model and empirical prior, (c) the quadratic model and general prior, and (d) the quadratic model and empirical prior..

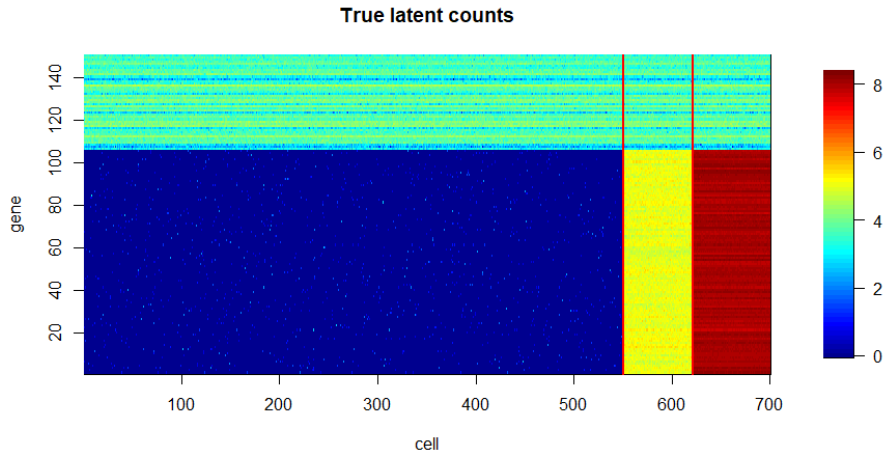


Figure B.28: Heat map of true latent gene-counts, with rows representing genes and columns representing cells. Cells are reordered by the true clustering, with cells from different clusters separated by vertical lines.

Appendix B.5. Simulation 3: Results

The true latent counts and the posterior estimated latent counts are illustrated in Figures B.28 and B.29, respectively, for Simulation 3. A comparison of the figures demonstrates that the latent counts are well recovered.

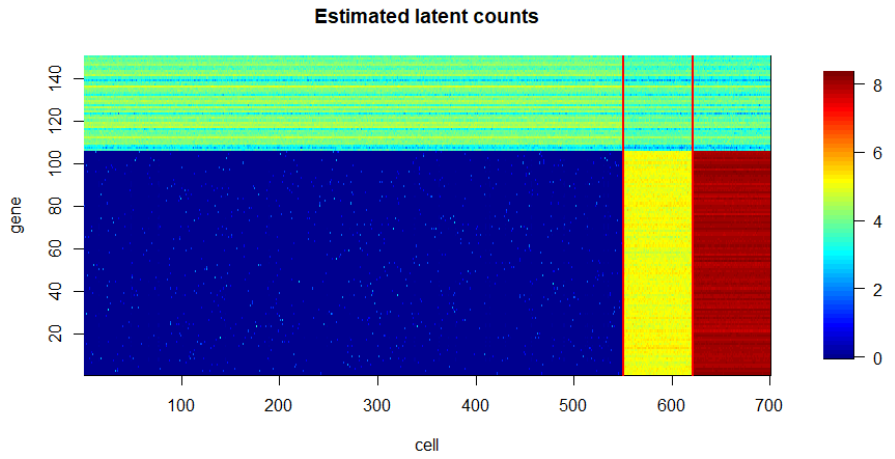


Figure B.29: Heat map of posterior estimated latent gene-counts, with rows representing genes and columns representing cells. Cells are reordered by the posterior estimated clustering, with cells from different clusters separated by vertical lines.

Appendix B.6. Additional Simulations with Mis-specified Mean Capture Efficiencies

In the previous simulations, we used an informative prior for the capture efficiencies to mitigate identifiability issues. In addition, we carried out further simulations to investigate the effect of misspecification of the mean capture efficiency in our informative prior. Specifically, we set the true mean capture efficiency for dataset 1 and 2 as 0.06 and 0.10, respectively, and simulate the true unique parameters and capture efficiencies under the following model:

$$\begin{aligned}
\beta_{c,1} &\sim \text{Unif}(0.04, 0.08), \\
\beta_{c,2} &\sim \text{Unif}(0.08, 0.12), \\
\phi_g &\sim \text{logN}(3, 0.5), \\
\mu_g &\sim \text{logN}(-1 + 0.5 \log(\phi_g), 0.1).
\end{aligned}$$

We compare the *bayNorm* estimates of the capture efficiencies and unique parameters with our empirical Bayesian approach. Results for dataset 1 and 2 are shown in Figure B.30 and B.31, respectively. In the first dataset, the mean capture efficiency is correctly specified and both approaches are able to recover the true mean expression, dispersion, and capture efficiencies (Figure B.30). However, in the second dataset, the mean capture efficiency is misspecified; while bayNorm underestimates the capture efficiencies and overestimates the mean expressions, our empirical Bayesian approach is more robust to such minor misspecifications.

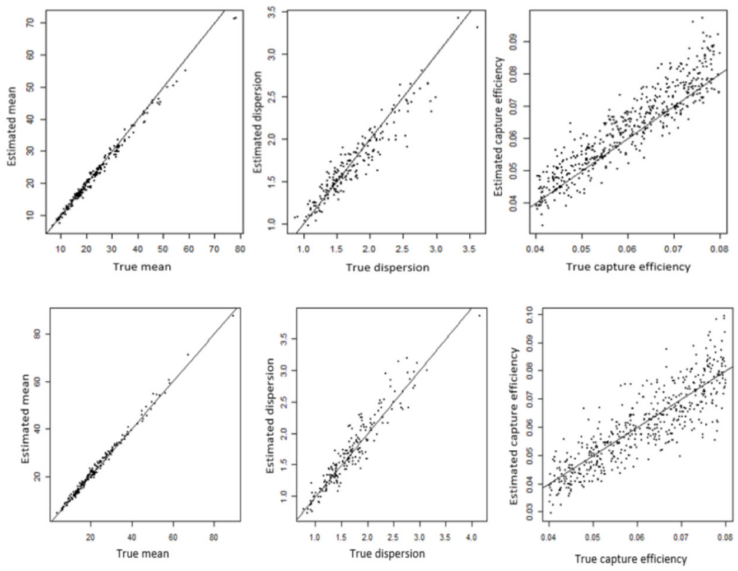


Figure B.30: Comparison between our Bayesian approach (top row) and *bayNorm* (bottom row) in recovering the mean-expression, dispersion and capture efficiency (left to right) for dataset 1.

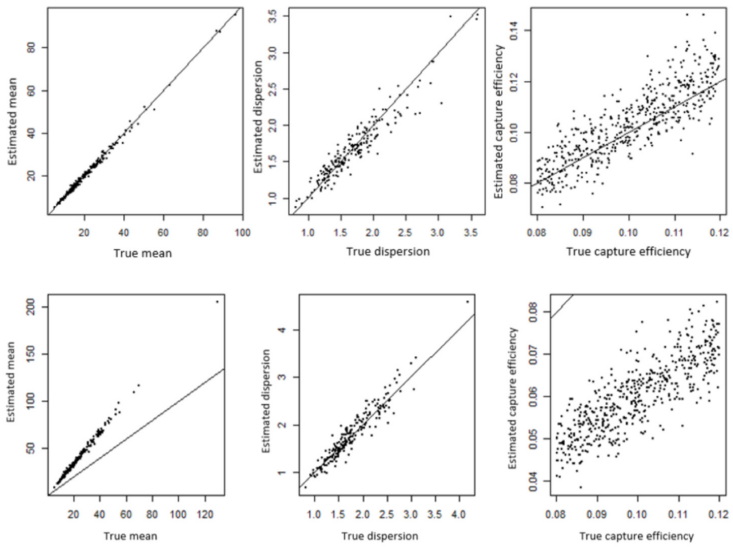


Figure B.31: Comparison between our Bayesian approach (top row) and *bayNorm* (bottom row) in recovering the mean-expression, dispersion and capture efficiency (left to right) for dataset 2.

Appendix C. PAX6 Data Analysis

Appendix C.1. Filtering Process

Similar to the approach taken in (Hoffman, 2023), we filter the raw dataset based on empirical statistics to remove extreme observations, and we select the top $2,000$ genes with the largest cell-to-cell variability:

1. Genes that are expressed by less than 5 cells are excluded (threshold chosen to remove half of the genes).
2. Cells with less than 2000 expressed genes are excluded (threshold chosen to remove cells with gene expressions less than 1.5 standard deviations below the empirical mean).
3. Cells with greater than 3.7 percent mitochondrial counts are excluded (threshold chosen to remove cells with mitochondrial counts greater than 1.5 standard deviations above than the empirical mean).
4. Cells with gene counts greater than 17500 are excluded (threshold chosen to remove cells with gene counts greater than 1.5 standard deviations above than the empirical mean).
5. 2000 genes with the greatest variability are selected.

Finally, we use the union of genes remaining after applying all above filters to each dataset to form the final processed data. Note that all the 'important' genes are also included for each dataset (Table D.6). The empirical relationship between *bayNorm* estimates of the mean expressions and dispersions and the distribution of log shifted counts based on the final processed data are shown in Figure C.32.

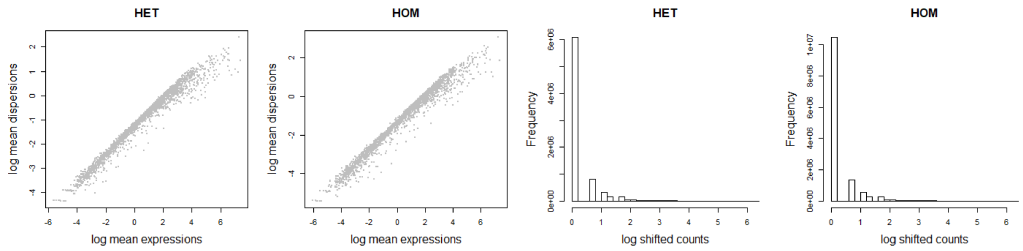


Figure C.32: Left two plots: empirical relationship between the log mean expressions and log dispersions. Right two plots: histogram of shifted latent gene-counts.

Appendix C.2. Results

For the posterior estimate of clustering, we show the number of genes in each cluster and the proportion of HET/HOM genes in each cluster in Figure C.33. The traceplots of concentration parameters and hyperparameters are shown in Figures C.34 and C.35, respectively, suggest convergence. The posterior and prior distributions of these parameters are compared in Figures C.36 and C.37, highlighting the influence of the data. The relationship between the posterior estimated mean expressions and dispersions on the log-scale for each cluster are shown in Figure C.38; the relationships are similar across clusters. In addition, we compare the posterior estimated latent counts of two selected genes, namely 'Fabp5' and 'H2afz' across different clusters in Wed Figure C.39. For gene 'Fabp5', there are no evident differences between the estimated latent counts across clusters, whereas for gene 'H2afz', differences across clusters are more apparent. Within each cluster, we observe greater variability of posterior estimated latent counts across cells for 'H2afz'. For the capture efficiencies, we compare the bayNorm estimates and NormHDP posterior mean estimates of the capture efficiencies in Figure C.40 and present

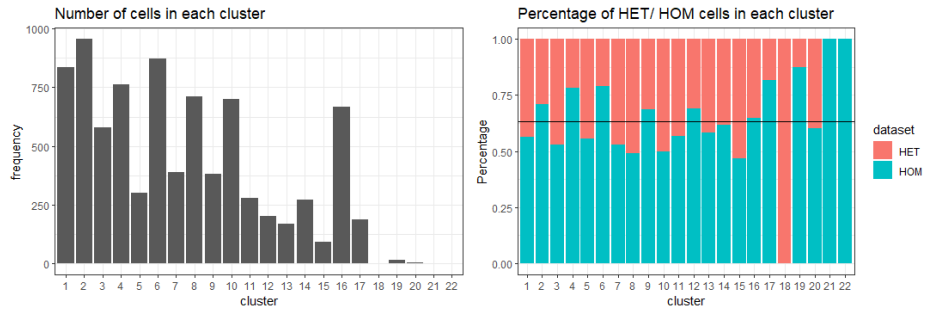
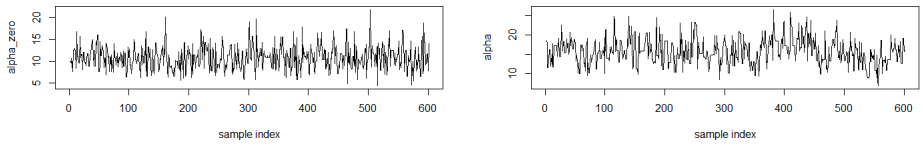


Figure C.33: Left: the number of cells within each cluster. Right: proportion of HET and HOM cells within each cluster, the horizontal line indicates the overall proportion of HOM cells across both datasets.

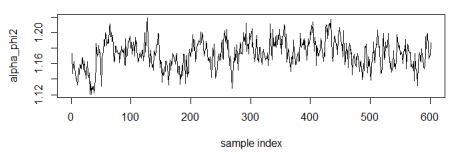


(a) Trace plot of α_0.

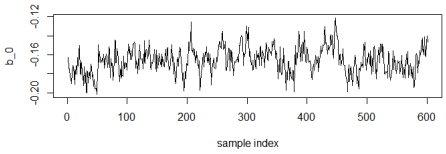
(b) Trace plot of α.

Figure C.34: Trace plot of concentration parameters.

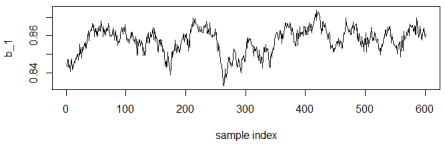
boxplots of the posterior mean capture efficiencies for cells in each cluster in Figure C.41. We observe that bayNorm tends to produce slightly larger estimates of the capture efficiencies compared with our model (Figure C.40). In addition, when comparing across clusters (Figure C.41), there is no evident difference, with the exception of the small group of cells in cluster 20, which have lower capture efficiencies. Another important aspect of our model is the quantification of uncertainty in the estimated latent counts; we show for two gene 'Fabp5' and 'Ran' the range of uncertainty in the posterior estimated latent counts in Figures C.42 and C.43.



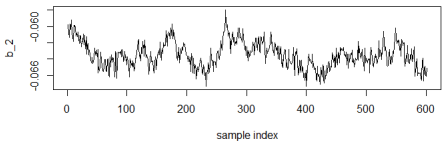
(a) Trace plot of α_ϕ^2.



(b) Trace plot of b_0.

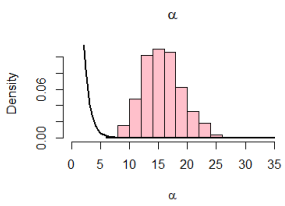


(c) Trace plot of b_1.

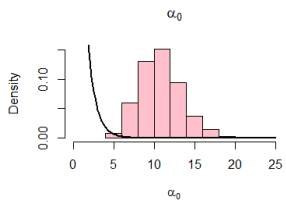


(d) Trace plot of b_2.

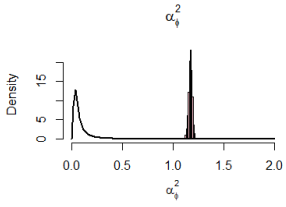
Figure C.35: Trace plot of regression parameters.



(a) Density of α

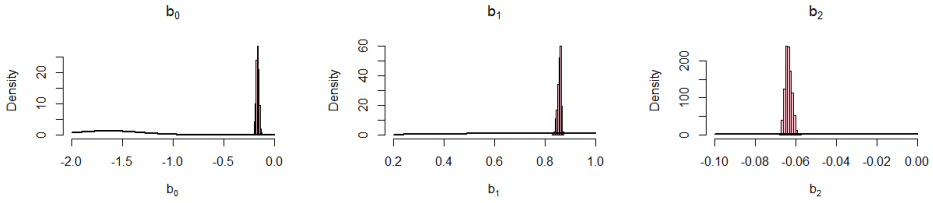


(b) Density of α_0



(c) Density of α_ϕ^2

Figure C.36: Comparison between prior and posterior of α, α_0 and α_ϕ^2 (left to right). Prior densities are shown with black lines and posterior densities are shown with histograms.



(a) Density of b_0 (b) Density of b_1 (c) Density of b_2

Figure C.37: Comparison between priors and posteriors of b_0, b_1 and b_2 (left to right). Prior densities are shown with black lines and posterior densities are shown with histograms.

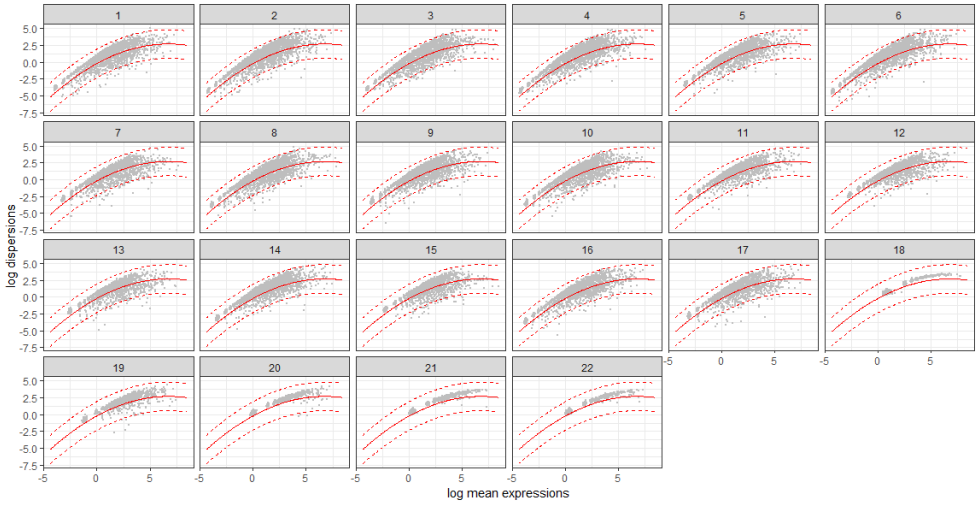


Figure C.38: Posterior estimated relationships between the mean expressions and dispersions for each cluster. The posterior means of the mean expressions and dispersions are plotted. The red dashed lines are the lower and upper bound of the 95 percent credible band obtained by considering posterior estimates of regression parameters $\mathbf{b}$ and α_ϕ^2. The red solid line is the posterior estimated relationship.

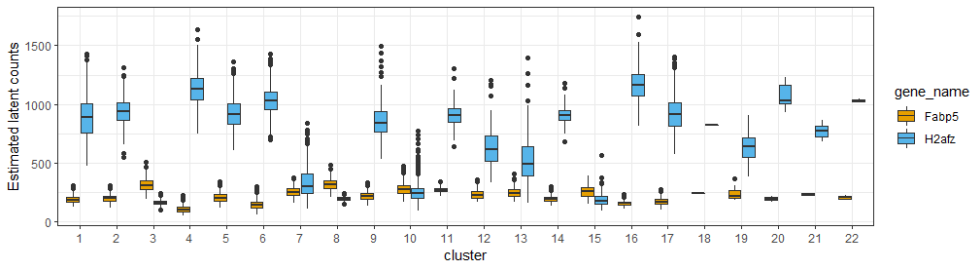


Figure C.39: Boxplots of posterior estimated latent counts across cells for genes Fabp5 and H2afz.

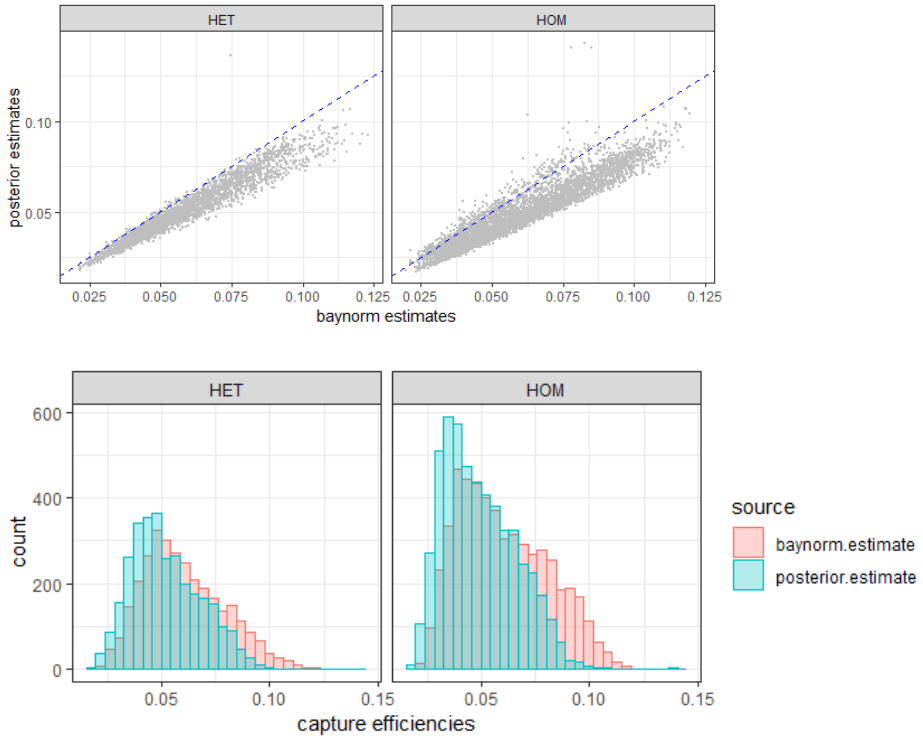


Figure C.40: Comparison of the bayNorm and posterior mean of the capture efficiencies. For the top two plots: bayNorm estimates are plotted against posterior estimates, and the dashed lines represents when the two values are equivalent. For the bottom two plots: histograms are drawn to compare distribution of the posterior mean capture efficiencies, with the bayNorm and posterior estimates in blue and red, respectively.

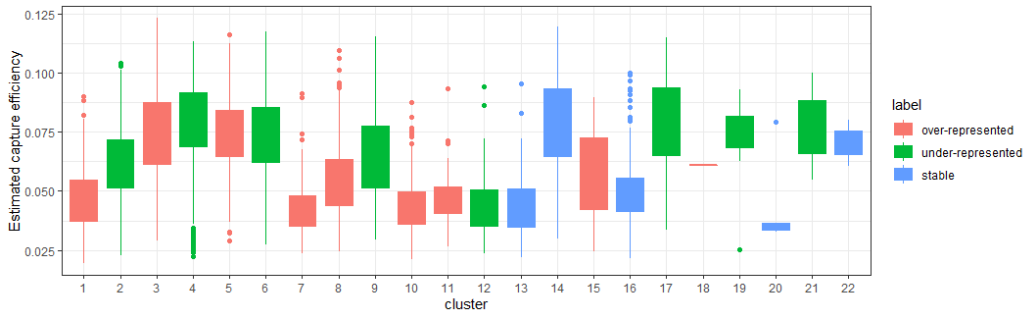


Figure C.41: box plots of the estimated capture efficiencies for cells in each cluster.

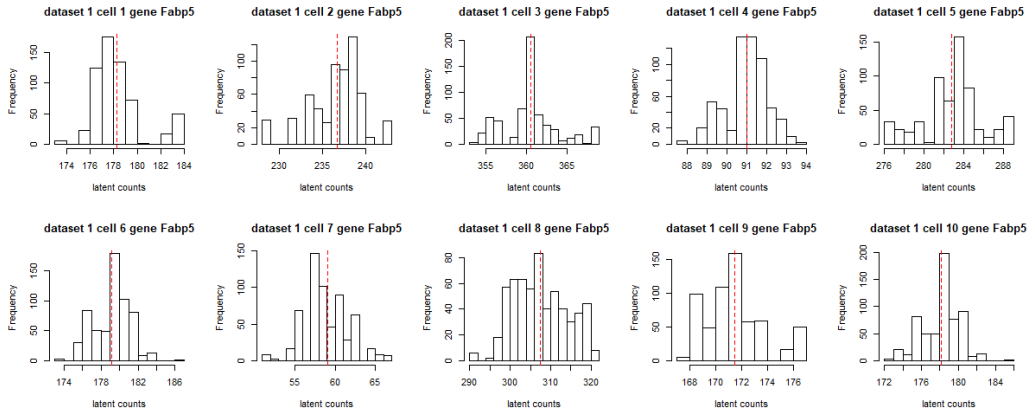


Figure C.42: Histograms of the posterior estimated latent counts for cell 1 to 10 for gene 'Fabp5' in dataset 1. For each MCMC iteration, we compute the mean estimated latent count. The red vertical line indicates the overall mean latent count, averaged across all MCMC iterations, for a given gene and cell.

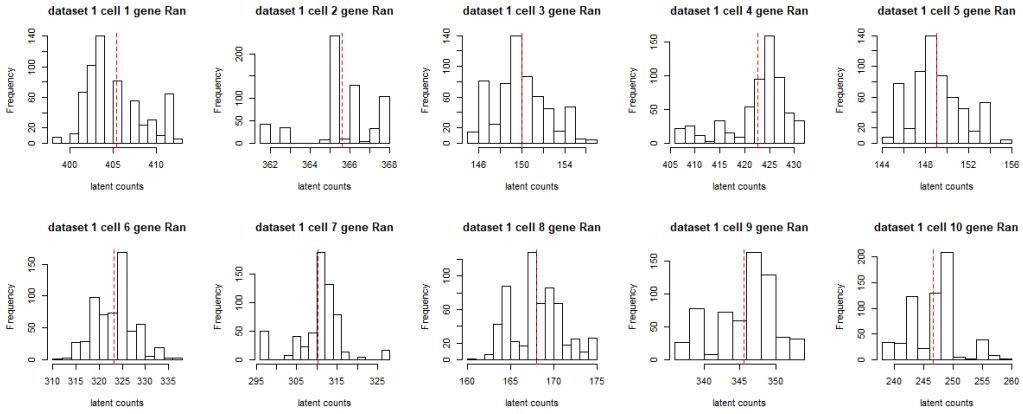


Figure C.43: Histograms of the posterior estimated latent counts for cell 1 to 10 for gene 'Ran' in dataset 1. For each MCMC iteration, we compute the mean estimated latent count. The red vertical line indicates the overall mean latent count, averaged across all MCMC iterations, for a given gene and cell.

Table C.2: Top 20 Global DE genes

Lhx1	1700063D05Rik	Nr2f2	Lhx1os	Trp73	Dlx2	Ppp2r2c	Gpr88	Gm27199	Foxo1
Gm30648	Elfn2	Insm2	Psd	Kcnj5	Ramp1	Lhx5	Mab21l1	Rspo2	AI593442

Table C.3: Top 20 Global DD genes

Ripor2	Spdl1	Trp73	Kif14	Wwtr1	Cxcl12	Ank3	Etv1	Cenpe	Nrp1
Plk4	Mfng	Zic1	Cdkn1a	Nt5dc2	Nusap1	Rtn1	Elavl3	Neurod2	Eomes

Appendix C.3. Global Marker Genes

Based on the heat-maps, we show that the posterior estimated mean expressions and dispersions tend to be lower for global DE and DD genes in comparison to the non-DE and non-DD genes for each cluster, apart from the small clusters 18 and $20 - 22$ (Figure C.44, C.45 and C.46). In addition, we compare the posterior of the mean expression and dispersion within cluster for some global marker genes that are identified as DE or DD in Figure C.47. Further, heatmaps of the observed gene-counts with genes reordered by tail probabilities are presented in Figure C.48 and C.49 for DE and DD, respectively; genes above the horizontal red line are identified as global markers. t-SNE plots with only the global marker genes are shown in Figure C.50 for both the observed and posterior estimated latent counts; separation between clusters is more evident in the t-SNE plot based on the latent counts.

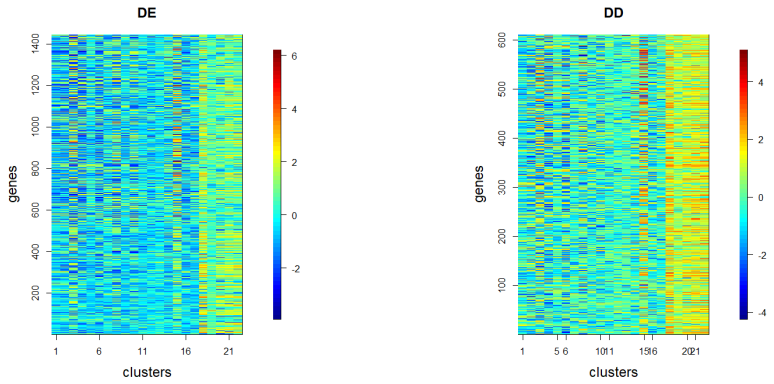


Figure C.44: Heat-maps of estimated relative unique parameters on the log-scale, with columns representing clusters and rows representing genes. Only global marker genes are included in the heat-maps. The relative value is defined as the estimated value minus the average across all clusters for that gene.

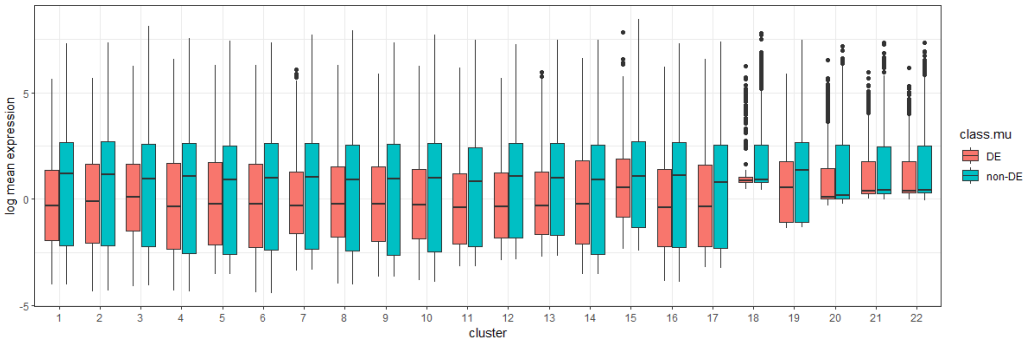


Figure C.45: Comparison of the posterior estimated log mean expressions between global DE and non-DE genes for each cluster.

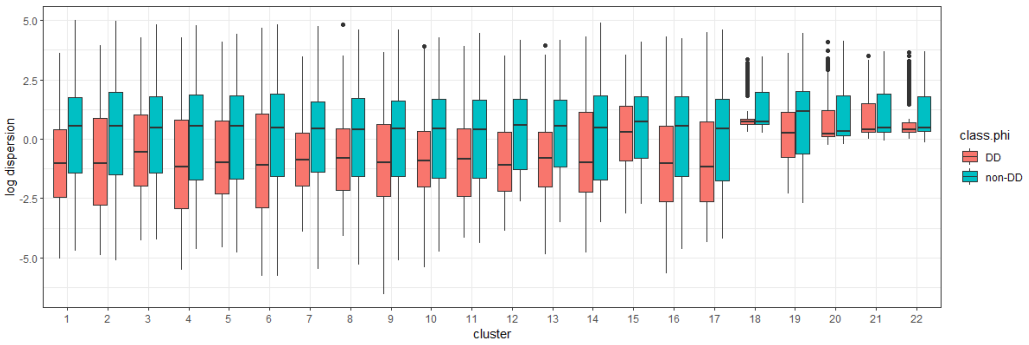


Figure C.46: Comparison of the posterior estimated log dispersions between global DD and non-DD genes for each cluster.

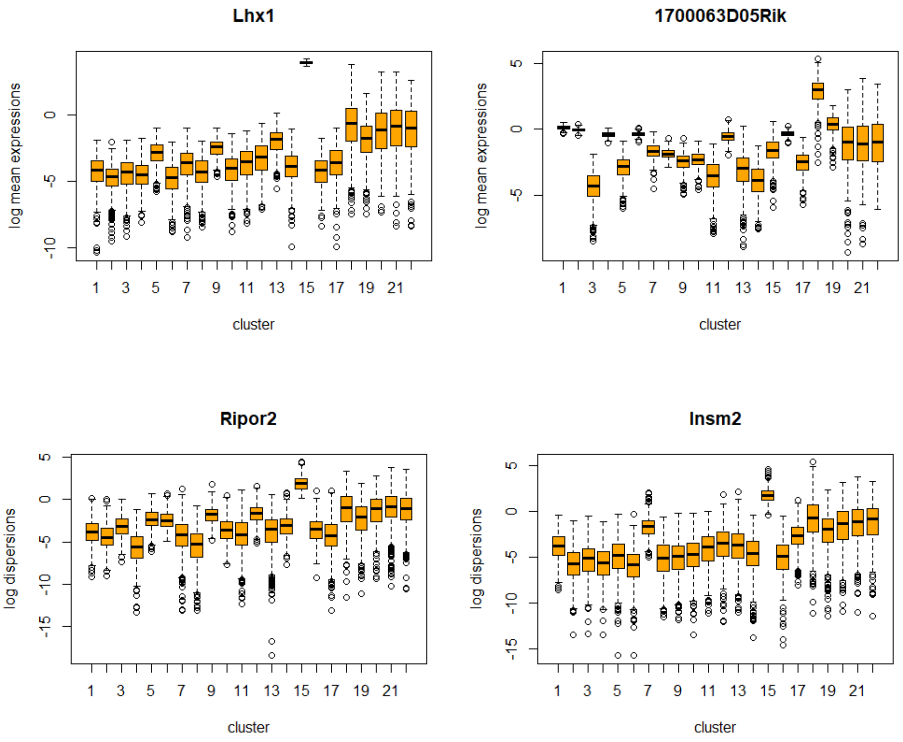


Figure C.47: Posterior of the unique parameters of global marker genes. Lhx1 and 1700063D05Rik are examples of global DE genes. Ripor2 and Insm2 are examples of global DD genes.

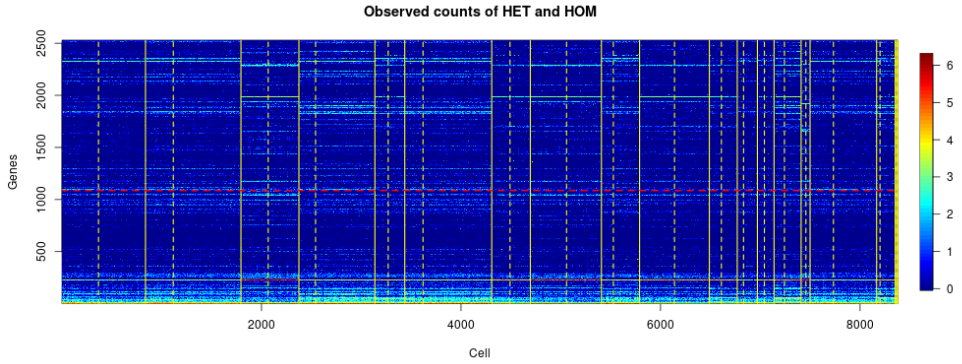


Figure C.48: Heat-map of the observed counts for HET and HOM data. Cells from different clusters are separated by yellow vertical lines. Cells from different datasets are separated by yellow dashed lines. DE and non-DE genes are separated by the horizontal line such that genes above the horizontal line are global marker genes.

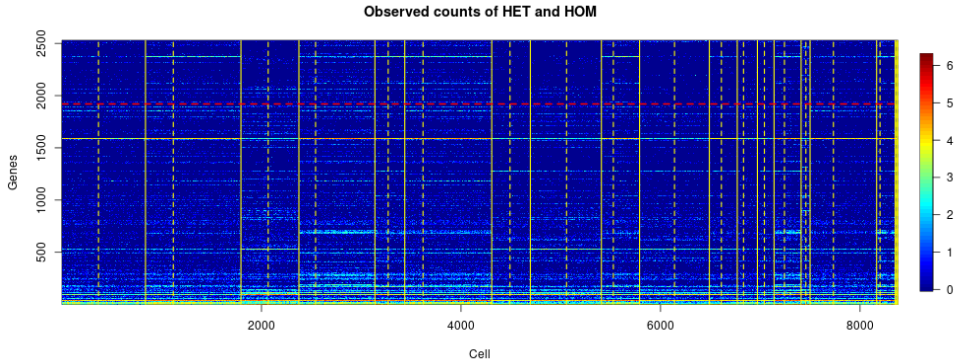


Figure C.49: Heat-map of observed counts for HET and HOM data. Cells from different clusters are separated by yellow vertical lines. Cells from different datasets are separated by yellow dashed lines. DD and non-DD genes are separated by the horizontal line such that genes above the horizontal line are global marker genes.

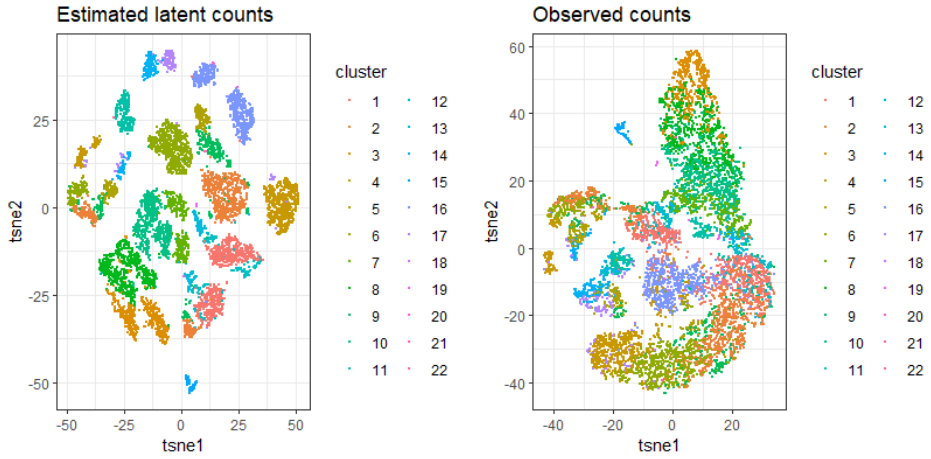


Figure C.50: t-SNE plot based on the estimated latent counts (left) and the observed counts (right) with global marker genes only. Cells are colored by cluster membership.

Table C.4: Top 5 local DE genes for each cluster.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Arhgap29	Rit2	Smc6	Gm40421	Gsx2	Ifitm3	Cnih2	Mapk3
Ifi27	Prc1	Neto2	0610010F05Rik	Hecw1	Gm36660	Rfc3	Ttk
1700093K21Rik	Tenm2	Smim18	Cnih2	Srrm4	Gm28209	Hist1h1e	Rab3a
Zfp935	1700093K21Rik	Timeless	Nlgn1	Mcm4	Msra	Nav3	Smc6
Tmem150c	Cdh4	Kcnk13	Brd8	Fzd1	Spock1	Mycbp2	Ina
Cluster 9	Cluster 10	Cluster 11	Cluster 12	Cluster 13	Cluster 14	Cluster 15	Cluster 16
Clic1	Comt	Prokr2	Nup85	Clic1	Lrrc7	Agtr1a	Txlnb
Nup62	Fbln1	Gm10457	1110017D15Rik	Serp2	Nr2f1	Kif4	Dcaf17
Schip1	Mapk3	Smc4	Xlr3a	Sobp	Loxl2	Dct	Snhg12
Mcm5	Tfap2c	1500035N22Rik	Calb1	Zic4	Mcm7	Lamp5	Dkc1
Gm28196	Dsn1	Dll1	Nefm	Reep2	Spag5	Syp	Manf
Cluster 17	Cluster 18	Cluster 19	Cluster 20	Cluster 21	Cluster 22		
Tagln2	Olfr655	Comt	Fst	Pcgf5	Pcgf5		
Dusp6	Gm16152	Pcgf5	Igfbpl1	Syn1	Dusp4		
Shb	Gm29771	Timeless	Cbfa2t3	Synpr	Nek6		
Efnb2	BC030500	Wnt7b	Mt3	Spats2l	Scg3		
Foxm1	Khdrbs2	Gpsm2	Wnt10a	Kcnh4	Shpk		

Appendix C.4. Local Marker Genes

We present the heatmaps to compare estimated unique parameters of local marker genes for each cluster are shown in Figure C.51 and C.52. Further, we present the relationship between the mean expressions and dispersions and highlight the local marker genes for each cluster in Figure C.53. No clear pattern is observed between the local features of the genes and the relationship between the unique parameters. In addition, we compare the estimated unique parameters between local marker and non-marker genes for each cluster; differences in the posterior estimated unique parameters are evident for local marker genes in 18 out of 22 clusters. (Figure C.54 and C.55).

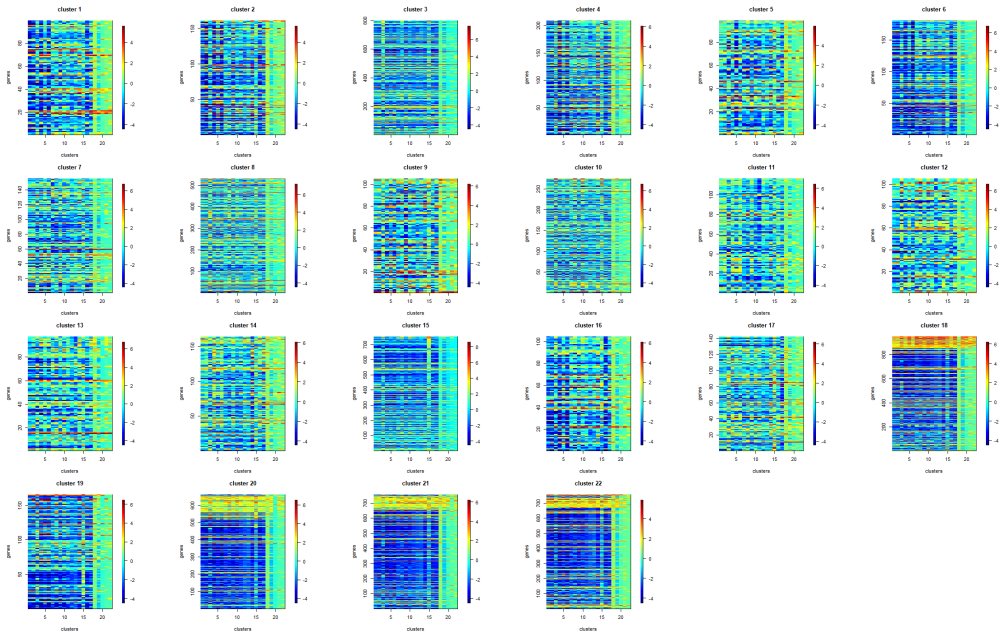


Figure C.51: Heat-maps to compare estimated unique parameters of local marker genes for each cluster. Columns in each heat map represent clusters and rows represent genes. The first two rows are the estimated mean expressions for the local DE genes and the last two rows are the estimated dispersions for the local DD genes. For all heat-maps, rows are reordered by local tail probabilities, hence genes on the top rows have a higher probability of being locally DE.

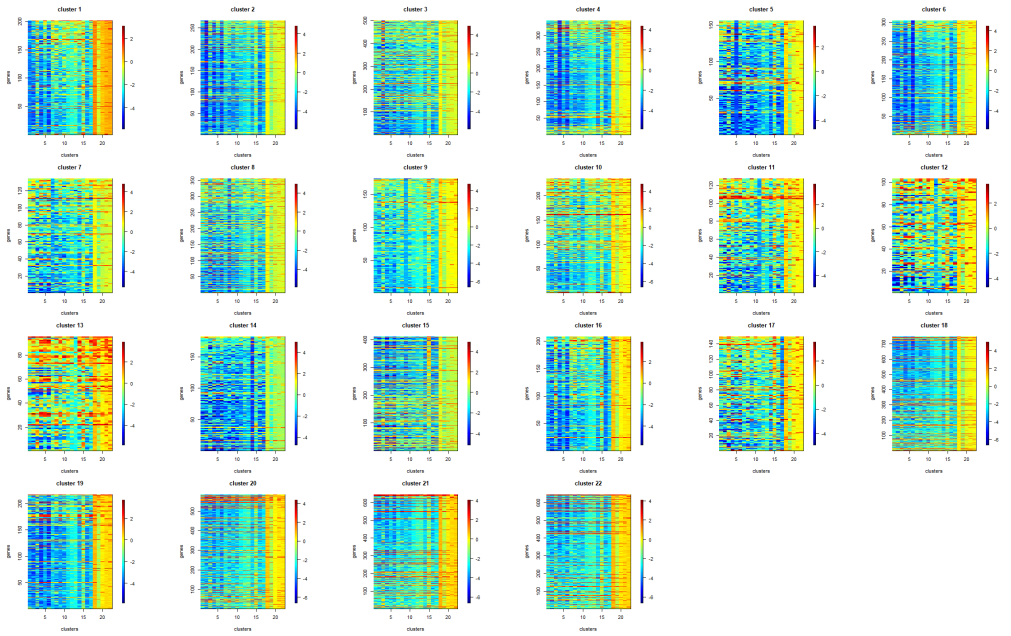


Figure C.52: Heat-maps to compare estimated unique parameters of local marker genes for each cluster. Columns in each heat map represent clusters and rows represent genes. The first two rows are the estimated mean expressions for the local DE genes and the last two rows are the estimated dispersions for the local DD genes. For all heat-maps, rows are reordered by local tail probabilities, hence genes on the top rows have a higher probability of being locally DD.

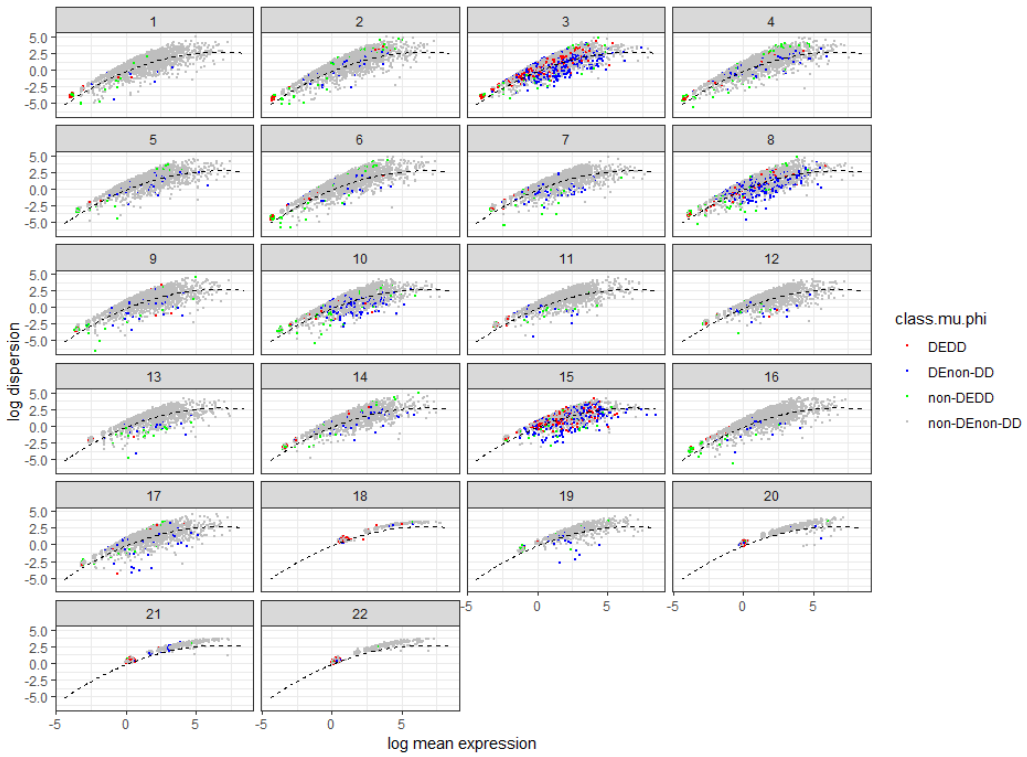


Figure C.53: Posterior estimated relationships between the mean expressions and dispersions for each cluster. The posterior means of the mean expressions and dispersions are plotted. To compare the local marker genes for each cluster, we highlight local DE and DD genes in red, local DE and non-DD genes in green, and local DD and non-DE genes in blue.

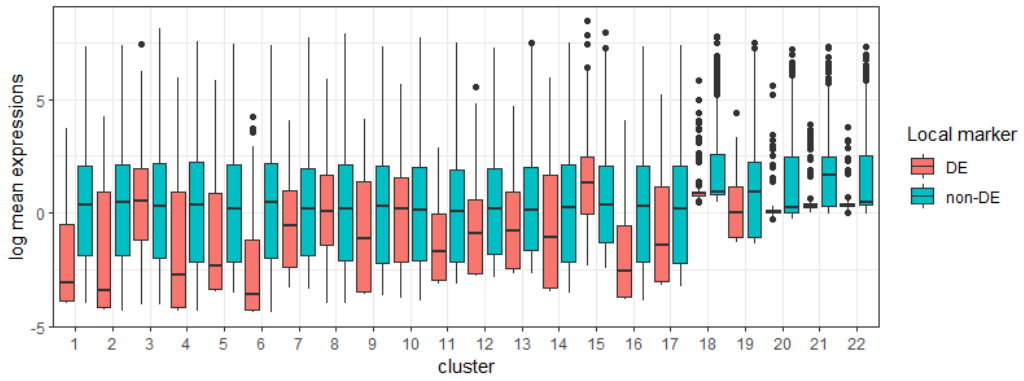


Figure C.54: Distribution of posterior estimates of mean expressions for local DE and non-DE genes for each cluster.

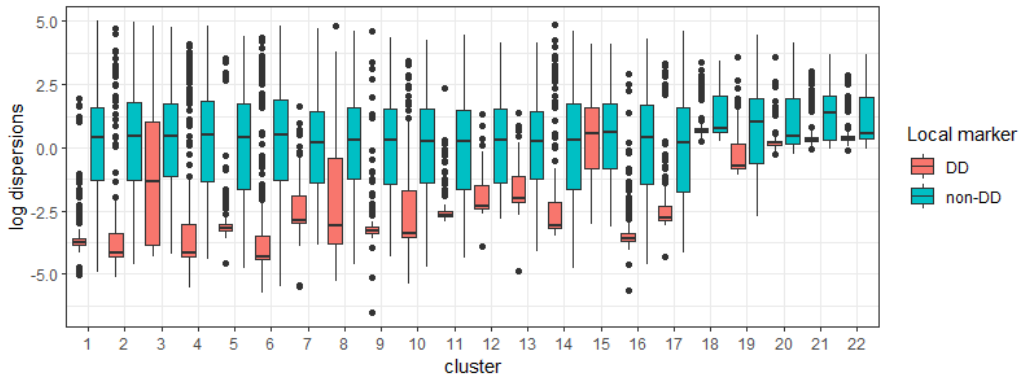


Figure C.55: Distribution of posterior estimates of dispersions for local DD and non-DD genes for each cluster.

Table C.5: Top 5 local DD genes for each cluster.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Ntm	Tenm2	Doc2b	Gm40421	Fzd1	Gm28209	Tssc4	Glra1
1700093K21Rik	1700093K21Rik	Numbl	Gm13425	Gsx2	Ifitm3	Mgat4c	Vit
Ifi27	Cnrip1	Fbln1	Nlgn1	Kcnb2	Msra	Sobp	4430402I18Rik
Arhgap29	Gm48283	A830011K09Rik	Cnih2	Mcm4	Spock1	Phf24	Calb1
Zfp935	Rgs8	Bnip3	Gabra2	Ppp2r2c	Htra1	Lsm3	Rspo3
Cluster 9	Cluster 10	Cluster 11	Cluster 12	Cluster 13	Cluster 14	Cluster 15	Cluster 16
Gm17322	Fbln1	1500035N22Rik	Nup85	Prim2	Tgfb2	Pcdhb6	Snhg12
Tenm1	Zcchc18	Prokr2	Ncapg2	Kif4	Lrrc7	Gfra4	Txlnb
Rhbdl2	Efnb1	Sfxn3	Atcay	Cck	Nr2f1	Suz12	Nlgn1
Zfpm2	Ccno	Acss1	9330159F19Rik	Sobp	Mcm7	Hspa8	H19
Clic1	Ackr3	Zbtb20	Xlr3a	Reep2	Gm13425	Chd5	Rec114
Cluster 17	Cluster 18	Cluster 19	Cluster 20	Cluster 21	Cluster 22		
Tagln2	Pcgf5	Slc25a5	Igfbpl1	Srrm4	Traf4		
Cdh10	Khdrbs2	Wnt7b	Acrbp	Syn1	Dusp4		
Efnb2	Rbms1	Etv4	Wnt10a	Tbx5	Rest		
Dusp6	Tubb3	Nrcam	Myl6b	Bmp3	Spc25		
Arhgap29	Gm16152	Mirt1	Bcl11a	Abhd11	Arhgef25		

Appendix C.5. Posterior Estimated Latent Counts

We compute the posterior estimated latent counts for all cells and compare between different clusters. Figure C.56 provides a heat-map of the estimated latent counts; cells are ordered by the clustering estimate, with solid vertical lines separating cells from different clusters and dashed vertical lines separating HET and HOM within cluster. Genes are reordered by global DD tail probabilities, with global DD genes above the horizontal line. Corresponding figures for the observed counts are shown in Section Appendix C.3.

For each gene, posterior estimated latent counts and observed counts for cells within each clusters are similar, and clear differences are observed across cells from different clusters. In addition, we use t-SNE (a commonly used dimensional reduction method for visualising gene expressions) to visualize similarities between cells within each cluster and differences across clusters. Applying t-SNE to the posterior estimated latent counts for genes which are global DE and DD shows a clear separation between clusters.

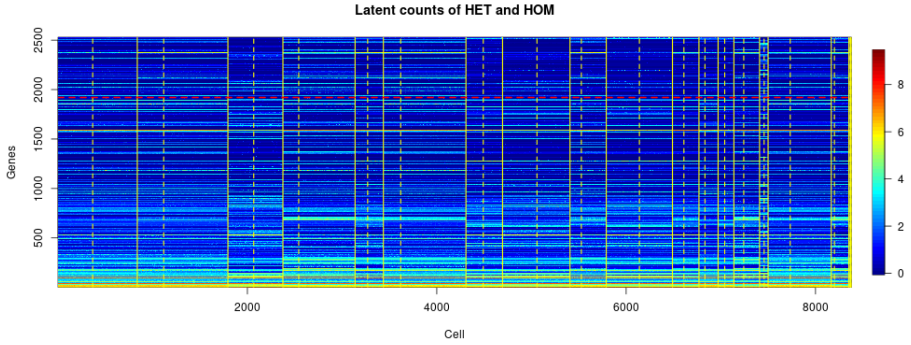


Figure C.56: Heat-map of posterior estimated latent gene-counts for HET and HOM. Genes are reordered by global DD tail probabilities, genes above the red horizontal line are global DD, and vice versa. Cells for reordered by the point estimate of posterior allocations. Cells from different clusters are separated by solid lines and cells from different datasets are separated by dashed lines.

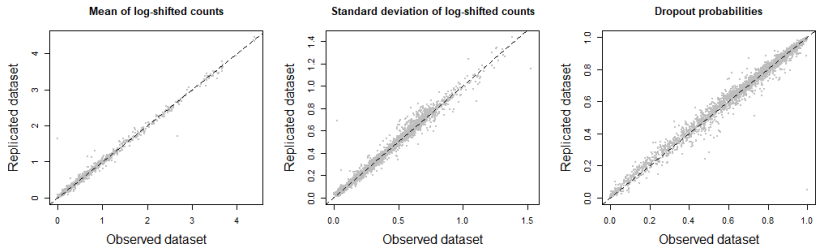


Figure C.57: Gene-wise comparison of the mean of the log shifted counts, standard deviation of the log shifted counts and dropout probabilities for posterior predictive checks with single replicate.

Appendix C.6. Posterior Predictive Checks

By comparing the observed and replicated statistics match gene-wise in Figure C.57, we show the observed and replicated statistics are similar which further supporting the model fit.

Table D.6: List of important genes.

Dlx6os1	Sp9	Nrxn3	Dlx1	Ccnd2	Arx	Dlx5	Top2a	Rrm2	Pclaf
Hmgb2	Cdca7	Gm13889	Etv1	Cenpf	Gm26917	Sp8	Gad2	Hmgn2	Cenpe
Insm1	Nusap1	Tpx2	Neurod6	Cntn2	Mef2c	Mapt	Tbr1	Nrp1	Wnt7b
Id2	Neurod1	Nrxn1	Satb2	Neurog2	Crabp1	Lhx2	Zic1	Mfap4	Nrp2
Ccnd2	Nhlh1	Plcb1	Nhlh2	Lhx9	Lmo4	Prdm13	Emx2	Cited2	Insm1
Ptn	Cux2	Wnt7b	Pou3f3	Cux1	Pou3f1	Zbtb20	Nfix	Pfn2	Ube2c
Fezf2	Sox2	Neurod2	Sox5	Slain1	Fgfr1	Pou3f2	Robo2	Dlx2	Smc2

Appendix D. *Important* Genes

We were provided with a list of 70 *important* genes, that are of particular interest for this experimental data. Information and summary statistics including the posterior estimated mean expressions and dispersions and global features of these *important* genes are shown below.

In the following, we present the list of *important* genes (Table D.6), and those which are classified as global marker genes are shown in Table D.7 and D.8. Further, we present the relationship between posterior estimated mean expressions and dispersions with different local features in Figure D.58, D.59, D.60 and D.61. Lastly, we show the heatmap of estimated unique parameters for the important genes in Figure C.53.

Table D.7: List of important genes which are classified as DE.

Pou3f3	Satb2	Nrp2	Cntn2	Lhx9	Nhlh1	Cenpf	Tbr1	Dlx1	Dlx2
Cdca7	Sp9	Neurod1	Gm13889	Nusap1	Plcb1	Insm1	Tpx2	Ube2c	Arx
Sox2	Nhlh2	Neurog2	Cenpe	Lmo4	Prdm13	Pou3f2	Smc2	Pou3f1	Dlx6os1
Dlx5	Ptn	Neurod6	Ccnd2	Sox5	Cited2	Hmgb2	Nrp1	Crabp1	Pclaf
Zic1	Mfap4	Neurod2	Top2a	Mapt	Mef2c	Rrm2	Id2	Etv1	Nrxn3
Sp8	Wnt7b	Robo2	Nrxn1	Emx2					

Table D.8: List of important genes which are classified as DD.

Lhx9	Dlx2	Cdca7	Neurod1	Nusap1	Insm1	Arx	Nhlh2	Cenpe	Neurod6
Ccnd2	Hmgb2	Nrp1	Pclaf	Zic1	Mfap4	Neurod2	Etv1	Nrxn3	Robo2
Gm26917	Nrxn1	Emx2							

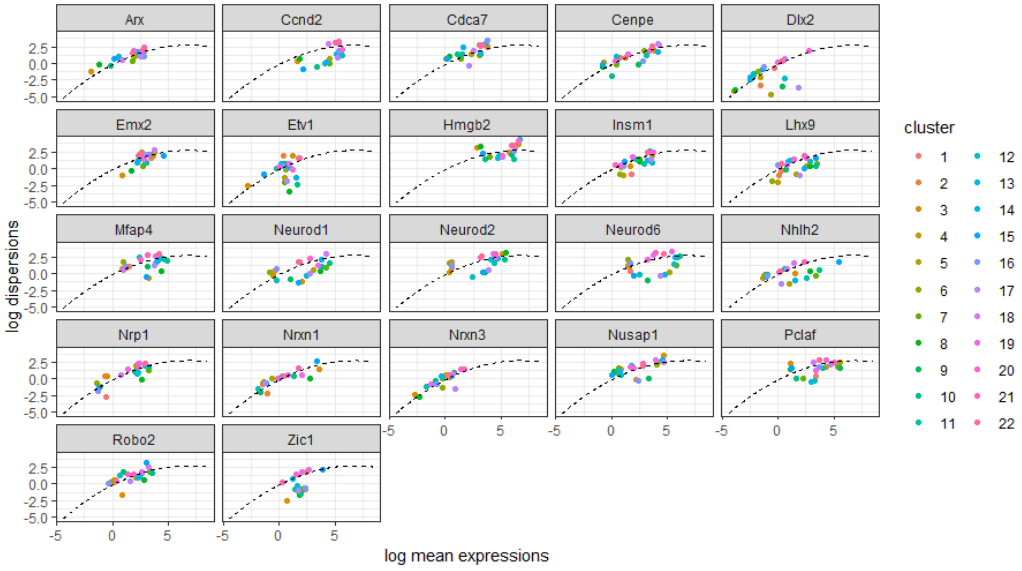


Figure D.58: Posterior estimated mean expressions and dispersions (on the log-scale) for the important genes which are classified as both DE and DD. The dashed line shows the posterior estimated relationship between the mean expressions and dispersions.

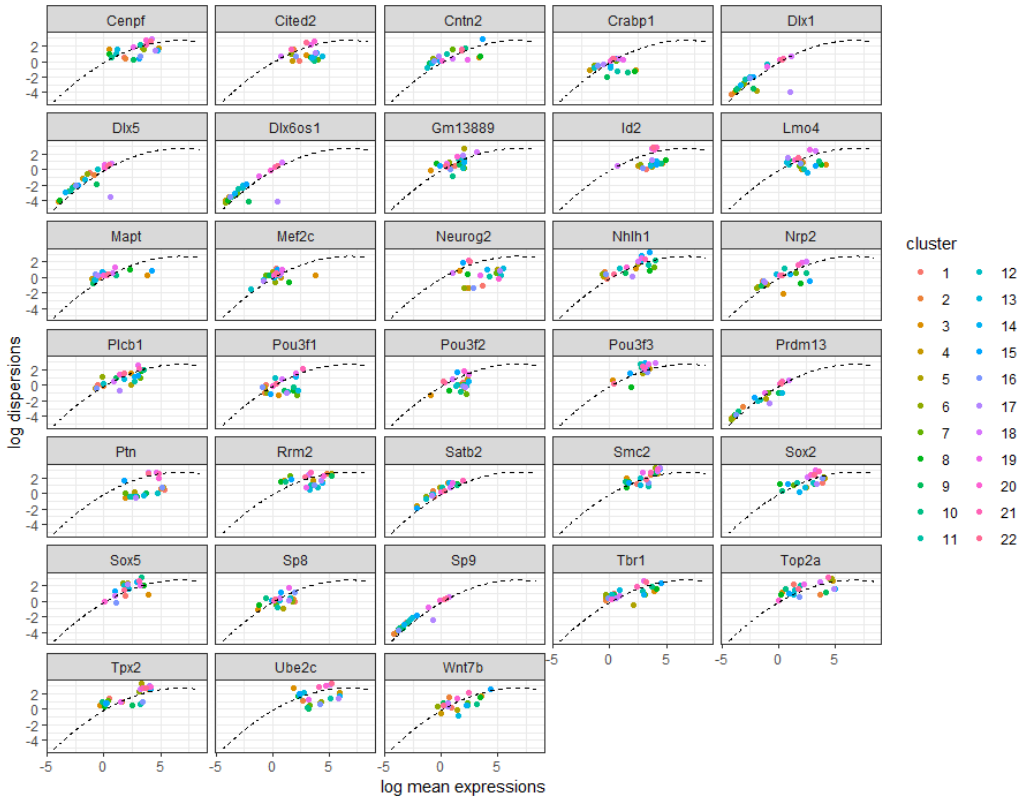


Figure D.59: Posterior estimated mean expressions and dispersions (on the log-scale) for the important genes which are classified as DE, but not DD. The dashed line shows the posterior estimated relationship between the mean expressions and dispersions.

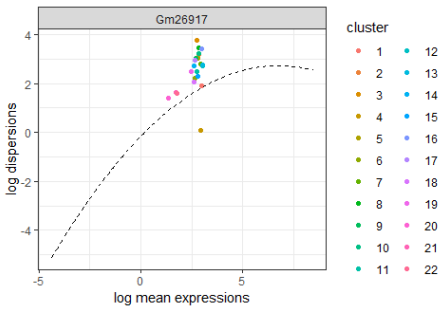


Figure D.60: Posterior estimated mean expressions and dispersions (on the log-scale) for the important genes which are classified as DD, but not DE. The dashed line shows the posterior estimated relationship between the mean expressions and dispersions.

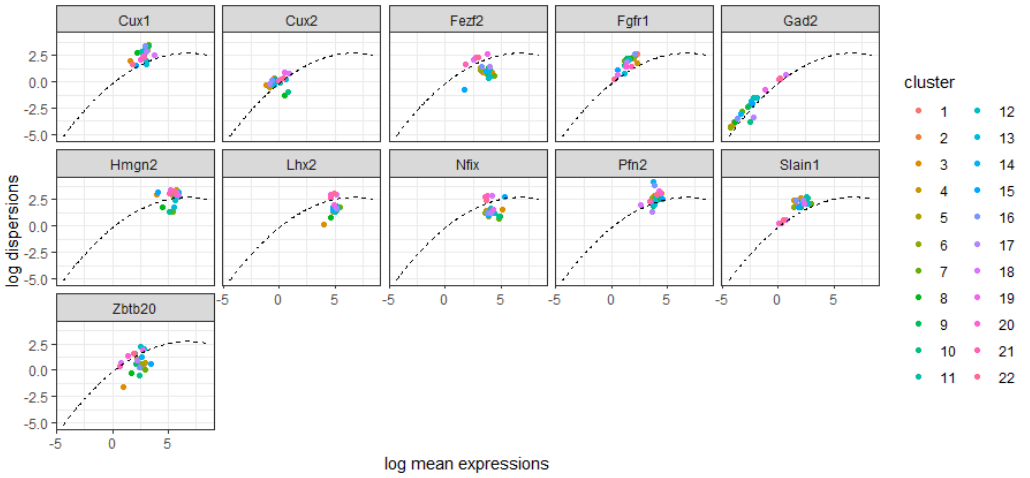


Figure D.61: Posterior estimated mean expressions and dispersions (on the log-scale) for the important genes which are classified as both not DE and DD. The dashed line shows the posterior estimated relationship between the mean expressions and dispersions.

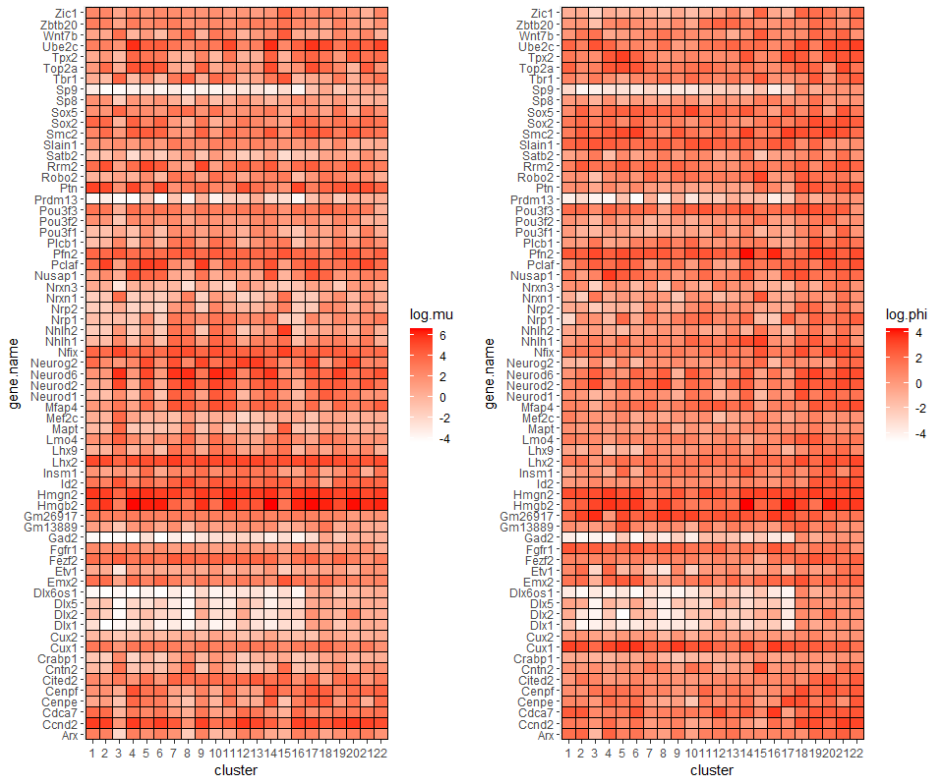


Figure D.62: Posterior means of the unique parameters of important genes for all clusters for mean expression (left) and dispersion (right).

Acknowledgements

We thank Dr. Tan, Prof. Price, Prof. Mason, Dr. Kozic and their team for providing the datasets, as well as for the insightful descriptions, motivations, suggestions and comments. We also thank Dr. Catalina Vallejos at the MRC Human Genetics Unit, University of Edinburgh for valuable suggestions and comments.

References

Argiento, R., Cremaschi, A., Vannucci, M., 2020. Hierarchical normalized completely random measures to cluster grouped data. Journal of the American Statistical Association 115 (529), 318–333.

Bergen, V., Lange, M., Peidli, S., Wolf, F. A., Theis, F. J., 2020. Generalizing RNA velocity to transient cell states through dynamical modeling. Nature Biotechnology 38 (12), 1408–1414.

Brennecke, P., Anders, S., Kim, J., Kolodziejczyk, A., Zhang, X., Proserpio, et al., V., 2013. Accounting for technical noise in single-cell RNA-seq experiments. Nature Methods 10 (11), 1093.

Cabellaro, I., Manuel, M., Molinek, M., Urzainqui, I., Mi, D., Shimogori, et al., T., 2014. Cell-Autonomous Repression of Shh by Transcription Factor Pax6 Regulates Diencephalic Patterning by Controlling the Central Diencephalic Organizer. Cell Reports 8, 1405–1418.

Celeux, G., Kamary, K., Malsiner-Walli, G., Marin, J.-M., Robert, C. P., 2019. Computational solutions for Bayesian inference in mixture models. Handbook of Mixture Analysis, 73–96.

Chandra, N. K., Canale, A., Dunson, D. B., 2023. Escaping the curse of dimensionality in Bayesian model-based clustering. J. Mach. Learn. Res. 24, 144–1.

Coleman, S., Kirk, P. D., Wallace, C., 2022. Consensus clustering for Bayesian mixture models. BMC Bioinformatics 23 (1), 1–21.

de Souto, M., Costa, I., de Araujo, D., Ludermir, T., Schliep, A., 2008. Clustering cancer gene expression data: a comparative study. BMC Bioinformatics 9 (1), 1–14.

Duan, T., Pinto, J., Xie, X., 2019. Parallel clustering of single cell transcriptomic data with split-merge sampling on Dirichlet process mixtures. Bioinformatics 35 (6), 953–961.

Dvorzak, M., Wagner, H., 2016. Sparse bayesian modelling of underreported count data. Statistical Modelling 16 (1), 24–46.

Eling, N., Richard, A., Richardson, S., Marioni, J., Vallejos, C., 2018. Correcting the mean-variance dependency for differential variability testing using single-cell RNA sequencing data. Cell Systems 7 (3), 284–294.

Ester, M., Kriegel, H., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. AAAI Press 96, 226–231.

Ferguson, T., 1973. A Bayesian analysis of some nonparametric problems. The Annals of Statistics, 209–230.

Finak, G., McDavid, A., Yajima, M., 2015. Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA-sequencing data. Genome Biology 16 (278), 1–13.

Gassen, S., Callebaut, B., Helden, M., Lambrecht, B., Demmester, P., Dhaene, et al., T., 2015. Flowsom: using self-organizing maps for visualization and interpretation of cytometry data. Cytometry A. 7 (87), 636–645.

Griffin, J., Stephens, D., 2013. Advances in Markov chain Monte Carlo. In: Bayesian Theory and Applications. Oxford University Press Oxford, pp. 104–144.

Guo, M., Wang, S., Potte, S., Whitsett, J., Xu, Y., 2015. SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. PLOS Computational Biology 11 (11).

Hoffman, P., 2023. Seurat - guided clustering tutorial.
URL `https://satijalab.org/seurat/articles/pbmc3k_tutorial.html`

Hughes, M., Kim, D., Sudderth, E., 2015. Reliable and scalable variational inference for the hierarchical Dirichlet process. In: Artificial Intelligence and Statistics. PMLR, pp. 370–378.

Ishwaran, H., James, L., 2001. Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association 96 (453), 161–173.

Ishwaran, H., Zarepour, M., 2002. Exact and approximate sum representations for the Dirichlet process. Canadian Journal of Statistics 30 (2), 269–283.

Ji, Z., Ji, H., 2016. TSCAN: pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. 13 (44).

Jiang, L., Chen, H., Pinello, L., Yuan, G., 2016. GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. Genome Biology 17 (144).

Kharchenko, P. V., Silberstein, L., Scadden, D. T., 2014. Bayesian approach to single-cell differential expression analysis. Nature Methods 11 (7), 740–742.

Kingman, J., 1975. Random discrete distributions. Journal of the Royal Statistical Society: Series B 37 (1), 1–15.

Kiselev, V., Andrews, T., Hemberg, M., 2019. Challenges in unsupervised clustering of single-cell RNA-seq data. Genetics 20, 273–282.

Kiselev, V., Kirschner, K., Schaub, M., Andrews, T., Yiu, A., Chandra, et al., T., 2017. SC3: consensus clustering of single-cell RNA-seq data. Nat Methods. 5 (14), 483–486.

Klein, A., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, et al., V., 2015. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 161 (5), 1187–1201.

Korthauer, K., Chu, L., Newton, M., Li, Y., Thomson, J., Stewart, et al., R., 2016. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. Genome Biology 17 (1), 1–15.

L. Lun, A. T., Bach, K., Marioni, J. C., 2016. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biology 17, 1–14.

Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D. J., Hicks, S. C., Robinson, M. D., Vallejos, C. A., Campbell, K. R., Beerenwinkel, N., Mahfouz, A., et al., 2020. Eleven grand challenges in single-cell data science. Genome Biology 21 (1), 1–35.

Lewin, A., Bochkina, N., Richardson, S., 2007. Fully Bayesian Mixture Model for Differential Gene Expression: Simulation and Model Checks. Statistical Applications in Genetic and Molecular Biology 6.

Lin, P., Troup, M., Ho, J., 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biology 1 (18).

Lopes de Oliveira, G., Argiento, R., Helena Loschi, R., Martins Assunção, R., Ruggeri, F., D'Elia Branco, M., 2022. Bias correction in clustered underreported data. Bayesian Analysis 17 (1), 95–126.

Manuel, M., Tan, K., Kozic, Z., Molinek, M., Marcos, T., Razak, et al., M., 2022. PAX6 limits the competence of developing cerebral cortical cells to respond to inductive intercellular signals. PLOS Biology*In press*.

Martinez-Jimenez, C., Eling, N., Chen, H., Vallejos, C., Kolodziejczyk, A., Connor, et al., F., 2017. Aging increases cell-to-cell transcriptional variability upon immune stimulation. Science 355 (6332), 1433–1436.

Mi, D., Huang, Y., Kleinjan, D., Mason, J., Price, D., 2013. Identification of Genomic Regions Regulating Pax6 Expression in Embryonic Forebrain Using YAC Reporter Transgenic Mouse Lines. PloS ONE 8 (11).

Miller, J., Dunson, D., 2015. Robust Bayesian Inference via Coarsening. Journal of American Statistical Association 114 (527), 1113–1125.

Moreno, E., Giron, J., 1998. Estimating with incomplete count data: A Bayesian approach. Journal of Statistical Planning and Inference 66 (1), 147–159.

Olsson, A., Venkatasubramanian, M., Chaudhri, V., Aronow, B., Salomonis, N., Singh, et al., H., 2016. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. Nature (537), 698–702.

Petegrosso, R., Li, Z., Kuang, R., 2020. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. Bioinformatics 21 (4), 1209–1223.

Prabhakaran, S., Azizi, E., Carr, A., Pe'er, D., 2016. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In: International Conference on Machine Learning. PMLR, pp. 1070–1079.

Qiu, X., Miao, Q., Tang, T., Wang, L., Chawla, R., Pliner, et al., H., 2017. Reversed graph embedding resolves complex single-cell trajectories. Nature Methods 4 (10), 979–982.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., Vert, J., 2018. A general and flexible method for signal extraction from single-cell RNA-seq data. Nature Communications 9 (1), 284.

Rodriguez, A., Dunson, D., Gelfand, A., 2008. The Nested Dirichlet Process. Journal of the American Statistical Association 103 (483), 1134–1154.

Rousseau, J., Mengersen, K., 2011. Asymptotic behaviour of the posterior distribution in overfitted mixture models. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 73 (5), 689–710.

Satija, R., Farrell, J., Gennert, D., Schier, A., Regev, A., 2015. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 33 (5), 495–502.

Schmertmann, C. P., Gonzaga, M. R., 2018. Bayesian estimation of age-specific mortality and life expectancy for small areas with defective vital records. Demography 55 (4), 1363–1388.

Senabouth, A., Lukowski, S., Hernandez, J., Andersona, S., Mei, X., Nguyen, et al., Q., 2017. ascend: R package for analysis of single cell RNA-seq data. bioRxiv.

Stamey, J. D., Young, D. M., Boese, D., 2006. A Bayesian hierarchical model for Poisson rate and reporting-probability inference using double sampling. Australian & New Zealand Journal of Statistics 48 (2), 201–212.

Stoner, O., Economou, T., Drummond Marques da Silva, G., 2019. A hierarchical framework for correcting under-reporting in count data. Journal of the American Statistical Association 114 (528), 1481–1492.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., III, et al., W. M., 2019. Comprehensive Integration of Single-Cell Data. Cell 177, 1888–1902.
URL https://doi.org/10.1016/j.cell.2019.05.031

Sun, Z., Wang, T., Deng, K., Wang, X., Lafyatis, R., Ding, et al., Y., 2018. DIMM-SC: a Dirichlet mixture model for clustering droplet-based single cell transcriptomic data. Bioinformatics 34 (1), 139–146.

Tang, W., Bertaux, F., Thomas, P., Stefanelli, C., Saint, M., Marguerat, et al., S., 2020. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. Bioinformatics 36 (4), 1174–1181.

Teh, Y., Jordan, M., Beal, M., Blei, D., 2006. Hierarchical Dirichlet processes. Journal of the American Statistical Association 101 (476).

Vallejos, C., Marioni, J., Richardson, S., 2015. BASiCS: Bayesian analysis of single-cell sequencing data. PLoS Computational Biology 11 (6).

Vallejos, C., Richardson, S., Marioni, J., 2016. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. Genome Biology 17 (1), 70.

Vallejos, C., Risso, D., Scialdone, A., Dudoit, S., Marioni, J., 2017a. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nature Methods 14 (6), 565.

Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S., Marioni, J. C., 2017b. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nature Methods 14 (6), 565–571.

Wade, S., Ghahramani, Z., 2018. Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). Bayesian Analysis 13 (2), 559–626.

Whittemore, A. S., Gong, G., 1991. Poisson regression with misclassified counts: application to cervical cancer mortality rates. Journal of the Royal Statistical Society: Series C (Applied Statistics) 40 (1), 81–93.

Wu, Q., Luo, X., 2019. A nonparametric Bayesian approach to simultaneous subject and cell heterogeneity discovery for single cell RNA-seq data. arXiv preprint arXiv:1912.08050.

Yang, Y., Huh, R., Culpepper, H., Lin, Y., Love, M., Li, Y., 2017. SAFE-clustering: Single-cell aggregated (from ensemble) clustering for single-cell RNA-seq data. bioRxiv.

Zeisell, A., Munoz-Manchado, A., Codeluppi, S., Lonnerberg, P., Manno, G., Jureus, et al., A., 2015. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science 347 (6226), 1138–1142.

Zhang, J., Fan, J., Fan, H., Rosenfeld, D., Tse, D., 2018. An interpretable framework for clustering single-cell RNA-Seq datasets. BMC Bioinformatics 19 (93), 1–12.

Zurauskiene, J., Yau, C., 2016. pcaReduce: hierarchical clustering of single cell transcruptional profiles. BMC Bioinformatics. 1 (17).