THE UNIVERSITY OF CHICAGO

GENERALIZED ADAPTIVE SHRINKAGE METHODS AND APPLICATIONS IN GENOMICS STUDIES

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF STATISTICS

BY
MENGYIN LU

CHICAGO, ILLINOIS
OCTOBER 2018

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser, Matthew Stephens, for his guidance over my entire graduate school study. I am deeply grateful for all the insights and feedbacks he has shared over the years, and am truly honored to have worked with such an outstanding researcher.

I would also like to thank the other members of my advising committee, Dan Nicolae and Mary Sara McPeek, for taking time out of their busy schedules to provide insightful advice to the project.

I am also grateful to my peers, both from the Department of Statistics and the Stephens Lab, for their encouragement, support and constructive debates. In particular, I would like to thank Peter Carbonetto, Kushal Dey, David Gerard, Joyce Hsiao, Abhishek Sarkar, Lei Sun, Gao Wang, Wei Wang, Zhengrong Xing and Siming Zhao for their unconditional help and invaluable suggestions.

Finally, I would like to thank my parents for their love and support throughout my PhD.

ABSTRACT

Shrinkage procedures have played an important role in helping improve estimation accuracy for a variety of applications. In genomics studies, the gene-specific sample statistics are usually noisy, especially when sample size is limited. Hence some shrinkage methods (e.g. *limma*) have been proposed to increase statistical power in identifying differentially expressed genes. Motivated by the success of shrinkage methods, Stephens (2016) proposed a novel approach, Adaptive Shrinkage (*ash*) for large-scale hypothesis testing including false discovery rate and effect size estimation, based on the fundamental "unimodal assumption" (UA) that the distribution of the actual unobserved effects has a single mode.

Even though *ash* primarily dealt with normal or student-t distributed observations, the idea of UA can be widely applied to other types of data. In this dissertation, we propose a general flexible Bayesian shrinkage framework based on UA, which is easily applicable to a wide range of settings. This framework allows us to deal with data involving other noise distributions (gamma, F, Poisson, binomial, etc.). We illustrate its flexibility in a variety of genomics applications including: differential gene expression analysis on RNA-seq data; comparison between bulk RNA-seq and single cell RNA-seq data; gene expression distribution deconvolution for single cell RNA-seq data, etc.

Chapter 1

INTRODUCTION

Shrinkage procedures have played an important role in helping improve estimation accuracy for a variety of applications. They are often used to shrink target parameters towards some chosen prior, and typically reduce the variance of the resulting estimators by combining information from prior knowledge as well as other observations. Stein (1956) and James and Stein (1961) proposed the famous James-Stein estimator as a shrinkage estimator to tackle the classical normal means problem: if we have a single n-dimensional multivariate normal observation $\mathbf{y} \sim N(\theta, \sigma^2 I)$, the maximum likelihood estimator (MLE) for θ is the observation itself $\hat{\theta}^{\text{MLE}} = \mathbf{y}$. Even though the MLE is unbiased, it could be sensitive to random noise inherent in the observed values. The James-Stein estimator is defined as $\hat{\theta}^{\text{JS}} = (1-(n-2)\sigma^2/||\mathbf{y}||^2)\mathbf{y}$, which biases the θ estimate toward 0 compared to $\hat{\theta}^{\text{MLE}}$. Despite the higher bias present in the James-Stein estimator ($|\text{E}(\hat{\theta}^{\text{JS}}) - \theta| > 0$), it has a lower variance than the MLE, $\text{Var}(\hat{\theta}^{\text{JS}}) < \text{Var}(\hat{\theta}^{\text{MLE}})$, in line with the standard bias-variance tradeoff property of parameter estimation. Hence the James-Stein estimator would be more robust to outliers with large random noises, and reduce the mean squared estimation error in general. The success of James-Stein estimator further motivated various shrinkage methods for parameter estimation, which used biased estimators to reduce variance and thus alleviate overfitting issues. Robbins (1985) later developed the empirical Bayes method, which assumes that the parameter follows some underlying prior $\theta \sim g(\cdot)$, then estimating g by maximizing the marginal likelihood $L(g) := p(\mathbf{y}|g) = \int p(\mathbf{y}|\theta)g(\theta)d\theta$. The posterior mean (or mode) of the parameter θ

is hence a shrinkage estimator, and the amount of shrinkage is adaptive to the data due to the estimation of the prior from the same data.

However, shrinkage estimation is not as widely used in practice as might be expected given the benefits mentioned above. For example, it is quite common in genomic studies for gene specific MLEs to be directly used for effect estimation or hypothesis testing. However, the gene-specific sample statistics are usually noisy, especially when sample size is limited. To deal with this, some shrinkage methods have indeed been proposed Efron et al. (2001); Baldi and Long (2001); Lönnstedt and Speed (2002); Smyth (2004); Anders and Huber (2010); Robinson et al. (2010) to increase power in identifying differentially expressed (DE) genes in two or more conditions. Among these methods, *limma* (Smyth, 2004) has become one of the most popular approaches for different expression analysis. The widespread use of *limma* has then proven the usefulness of shrinkage methods in genomic contexts. The *limma* method developed a Bayesian hierarchical model for gene-specific variances and estimated the prior using Empirical Bayes methods, which allows *limma* to be robust to different types of datasets.

Motivated by the success of shrinkage methods, Stephens (2016) proposed a novel approach, *Adaptive Shrinkage* (*ash*) for large-scale hypothesis testing, including false discovery rate (FDR) and effect size estimation. The fundamental assumption of *ash* is the "unimodal assumption" (UA), whereby the distribution of the actual (unobserved) effects has a single mode. This turns out to be a reasonable assumption in many contexts: in many applications most effect sizes are often concentrated around some centroid, and smaller/larger effects become less probable. Suppose we observe

a list of normal observations $y_j \sim N(\beta_j, s_j^2)$ (or student-t observations) around the true unknown effects β_j, *ash* assumes that $\beta_j \sim g(\cdot)$ where $g(\cdot)$ is a unimodal prior estimated by the empirical Bayes approach. The posterior mean of β_j is thus used as a shrinkage estimator for the true effect β_j. The *ash* shrinkage is quite adaptive and can be applied to generic datasets, since UA is more flexible than any specific parametric distribution assumption. At the same time, the unimodal constraint also allows *ash* to avoid issues with over-fitting. The posterior distribution of β_j can also be used in situations other than parameter estimation, for example hypothesis testing, false sign rate controlling etc. In summary, *ash* demonstrates a between flexibility, generality, and simplicity on the one hand, and statistical efficiency and principle on the other.

Even though *ash* primarily dealt with normal or student-t distributed observations, the idea of UA can be widely applied to other types of data. In this thesis, we propose a general flexible Bayesian shrinkage framework based on UA, which is easily applicable to a wide range of settings. This framework allows us to deal with data involving other noise distributions (gamma, F, Poisson, binomial, etc.). We illustrate its flexibility in a variety of genomics applications other than differential expression analysis.

This thesis consists of the following chapters:

- Chapter 2: Variance adaptive shrinkage, which aims at shrinking variance estimates of expression data. While *limma* suggests a single inverse-gamma prior for gene-specific variances, we use a more flexible unimodal inverse-gamma mixture prior to approach the problem.

- Chapter 3: Using adaptive shrinkage methods to detect differentially expressed genes from RNA-Seq data, where we have to tackle issues caused by count data, small sample size and unwanted variation (Rocke et al., 2015).

- Chapter 4: Adaptive shrinkage for observations from general distributions. Some special cases (F, binomial, Poisson distributed observations) and applications are further discussed.

- Chapter 5: Using Poisson adaptive shrinkage methods to deconvolve gene expression distribution from single cell RNA-Seq data, and comparison with other expression distribution deconvolution methods.

Chapter 2

VARIANCE ADAPTIVE SHRINKAGE

2.1 Introduction

For differential expression analysis, a typical pipeline for identifying differentially expressed genes computes a p-value for each gene using a t-test (two condition experiments) or F-test (multiple condition experiments), both of which require an estimate of the variance in expression of each gene among samples. In the classical t-test or F-test, sample variances are used as plug-in estimates of gene-specific variances. However, when the sample size is small, sample variances can be inaccurate, resulting in loss of power (Murie et al., 2009). Hence, many methods have been proposed to improve variance estimation. For example, several papers (Tusher et al., 2001; Efron et al., 2001; Broberg et al., 2003) suggested adding an offset standard deviation to stabilize small variance estimates. A more sophisticated approach (Baldi and Long, 2001; Lönnstedt and Speed, 2002) used parametric hierarchical models to combine information across genes, using an inverse gamma prior for the variances, and a Gamma likelihood to model the observed sample variances. This idea was further developed by Smyth (2004) into an Empirical Bayes approach that estimates the parameters of the prior distribution from the data. This improves performance by making the method more adaptive to the data. Smyth (2004) also introduces the "moderated t-test", which modifies the classical t-test by replacing the gene-specific sample variances with estimates based on their posterior distribution. This pipeline, implemented in the software `limma`, is widely used in genomics thanks to

its adaptivity, computational efficiency and ease of use.

While assuming an inverse-gamma distribution for the variances yields simple procedures, the actual distribution of variances may be more complex. Motivated by this, Phipson et al. (2016) (*limma* with robust option, denoted by *limmaR*) modified the procedures from Smyth (2004), in a somewhat *ad hoc* way, to allow that some small proportion of outlier genes may have higher variability than expected under the inverse-gamma assumption. They showed that, in the presence of such outliers, this procedure could improve on the standard limma pipeline.

Here we consider a more general and adaptive approach to this problem. Our method is based on the assumption that the distribution of the variances (or, alternatively, the precisions) is unimodal. This unimodal assumption involves compromises between flexibility and robustness. On the one hand, it provides more flexibility and generality than specific parametric models. One the other hand, the unimodal constraint would lessen the typical over-fitting issues of non-parametric methods. We use a mixture of (possibly a large number of) inverse-gamma distributions to flexibly model this unimodal distribution, and provide simple computational procedures to fit this empirical Bayes model by maximum likelihood of the mixture proportions. Our procedure provides a posterior distribution on each variance or precision, as well as point estimates (posterior mean). The methods are an analogue of the "adaptive shrinkage" methods for mean parameters introduced in Stephens (2016), and are implemented in the R package `vashr` (for "variance adaptive shrinkage in R"). We compare our method with both *limma* and *limmaR* in various simulation studies, and also assess its utility on real gene expression data.

Our R package `vashr` is available from http://github.com/mengyin/vashr. The R code for simulations and analysis results are available from http://github.com/mengyin/vash.

2.2 Methods

2.2.1 Models

Suppose that we observe variance estimates $\hat{s}_1^2, ..., \hat{s}_G^2$ that are estimates of underlying "true" variances $s_1^2, ..., s_G^2$. Motivated by standard normal theory, we assume that

$$\hat{s}_g^2 | s_g^2 \sim s_g^2 \chi_{d_g}^2 / d_g, \quad i.e. \quad \hat{s}_g^2 | s_g^2 \sim \text{Gamma}(d_g/2, d_g/(2s_g^2)). \tag{2.1}$$

where the degrees of freedom d_g depends on the sample size and we assume it to be known.

Empirical Bayes (EB) approaches to estimating s_g^2 (eg (Smyth, 2004)) are commonly used to improve accuracy, particularly when the degrees of freedom d_g for each observation are modest. The EB approach typically assumes that the variances s_g^2 are independent and identically distributed from some underlying parametric distribution g:

$$s_g^2 \sim g(\cdot; \theta) \tag{2.2}$$

where the parameters θ are to be estimated from the data. Equivalently, that the precisions (inverse variances), s_g^{-2}, are i.i.d. from some $h(\cdot; \theta)$. A standard approach (Smyth) assumes that g is an inverse-gamma distribution (i.e. h is a gamma dis-

tribution) which simplifies inference because of conjugacy. Here we introduce more flexible assumptions for g or h: specifically that either g or h is *unimodal*. By using a mixture of inverse gamma distributions for g (i.e. a mixture of gamma distributions for h), we can flexibly capture a wide variety of unimodal distributions for g or h, while preserving many of the computational benefits of conjugacy.

2.2.2 A unimodal distribution for the variances

Let $\text{InvGamma}(\cdot; a, b)$ denote the density of an inverse-gamma distribution with shape a and rate b. This distribution is unimodal with mode at $c = b/(a+1)$. To obtain a more flexible family of unimodal distributions with mode at c we consider a mixture of inverse-gamma distributions, each with mode at c:

$$g(\cdot; \pi, \mathbf{a}, c) = \sum_{k=1}^{K} \pi_k \text{InvGamma}(\cdot; a_k, b_k), \tag{2.3}$$

where

$$b_k := (a_k + 1)c. \tag{2.4}$$

Each component in (2.3) has mode at c, and the variance about this mode is controlled by a_k, with large a_k corresponding to small variance. By setting $\mathbf{a}$ to a fixed grid of values that range from "small" to "large", we obtain a flexible family of distributions, with hyperparameters π, that are unimodal about c. See below for details on choice of grid for $\mathbf{a}$. This approach is analogous to Stephens (2016), which uses mixtures of normal or uniform distributions, with a fixed grid of variances, to model unimodal distributions for mean parameters. In practice modest values of K (e.g.

10-16) are sufficiently large to give reasonable performance. Even though the overall estimation accuracy of prior distribution is critical, the values of hyper-parameters are not of our primary interest.

2.2.3 Estimating hyper-parameters

For $K = 1$ we estimate the hyperparameters (a, c) by maximizing the likelihood

$$L(a, c; \hat{s}_1^2, ..., \hat{s}_G^2) := P(\hat{s}_1, ..., \hat{s}_G | a, c) \tag{2.5}$$

$$= \prod_{g=1}^{G} p(\hat{s}_g; a, c) \tag{2.6}$$

$$\tag{2.7}$$

where

$$p(\hat{s}_g; a, c) = \int P(\hat{s}_g^2 | s_g^2) g(s_g^2 | a, c) ds_g^2 \tag{2.8}$$

$$= (d_g/2)^{d_g/2} \frac{\hat{s}_g^{d_g - 1/2} \Gamma(a + d_g/2) b^a}{\Gamma(d_g/2)\Gamma(a)(d_g \hat{s}_g^2/2 + b)^{a + d_g/2}}, \tag{2.9}$$

$$[b = (a + 1)/c]. \tag{2.10}$$

We use the R command `optim` to numerically maximize this likelihood. The approach is similar to Smyth (2004), except that we use maximum likelihood instead of moment matching.

For $K > 1$, as noted above, we use K "large" (e.g. 10-16), fix the values of a_k to a grid of values from "small" to "large", and estimate the hyper-parameters c, π by

maximizing the likelihood

$$L(\pi, c; \mathbf{a}, \hat{s}_1^2, ..., \hat{s}_G^2) = P(\hat{s}_1, ..., \hat{s}_G | \pi, \mathbf{a}, c) \tag{2.11}$$

$$= \prod_{g=1}^{G} \sum_k \pi_k p(\hat{s}_g; a_k, c) \tag{2.12}$$

where $p(\hat{s}_g; a_k, c)$ is given by (2.9). We center the grid of a_k values on the point estimate $\hat{a}$ obtained for $K = 1$, to ensure that the grid values span a range consistent with the data (typically a_k lies between 0 and 100). Moreover, if the data are consistent with $K = 1$ then the estimated π will be concentrated on the component with $a_k = \hat{a}$, and thus lead to similar results to *limma*.

To maximize the likelihood we use an iterative procedure that alternates between updating c and π, with each step increasing the likelihood. Given c, we update π using a simple EM step (Dempster et al., 1977). Given π we update c by optimizing (2.12) numerically using `optim`. We use SQUAREM (Varadhan, 2010) to accelerate convergence of the overall procedure. See Appendix A.1 for details.

2.2.4 Posterior calculations

Using (2.3) as a prior distribution for s_g^2, and combining with the likelihood (2.1) the posterior distribution of s_g^2 is also a mixture of inverse-gamma distributions:

$$P(s_g^2 | \hat{s}_g^2) = \sum_k \tilde{\pi}_{gk} \text{InvGamma}(s_g^2; \tilde{a}_{gk}, \tilde{b}_{gk}), \tag{2.13}$$

where

$$\tilde{a}_{gk} := a_k + d_g/2, \tag{2.14}$$

$$\tilde{b}_{gk} := b_k + d_g \hat{s}_g^2/2, \tag{2.15}$$

$$\tilde{\pi}_{gk} := \frac{\pi_k \hat{s}_g^{d_g-2} \frac{\Gamma(a_k+d_g/2)}{\Gamma(a_k)} \frac{b_k^{a_k}}{(b_k+d_g\hat{s}_g^2/2)^{a_k+d_g/2}}}{\sum_{k'} \pi_{k'} \hat{s}_g^{d_g-2} \frac{\Gamma(a_{k'}+d_g/2)}{\Gamma(a_{k'})} \frac{b_{k'}^{a_{k'}}}{(b_{k'}+d_g\hat{s}_g^2/2)^{a_{k'}+d_g/2}}}. \tag{2.16}$$

Following Smyth (2004) we use the posterior mean of s_g^{-2} as a point estimate for the precision s_g^{-2}:

$$\tilde{s}_g^{-2} = \mathbb{E}(s_g^{-2}|\hat{s}_g^2) = \sum_k \tilde{\pi}_{gk} \frac{\tilde{a}_{gk}}{\tilde{b}_{gk}}. \tag{2.17}$$

Note that each $\tilde{a}_{gk}/\tilde{b}_{gk}$ can be interpreted as a shrinkage-based estimate of s_g^{-2}, since it lies between the observation $\hat{s}_g^{-2}$ and the prior mean of the kth mixture component a_k/b_k.

When estimating variances we use the inverse of the estimated precision (2.17). While it may seem more natural to use the posterior mean of s_g^2 as a point estimate for s_g^2, we found that this can be very sensitive to small changes in the estimated hyper-parameters $\mathbf{a}$, and so can perform poorly. And while it may also be more natural to estimate variances on a log scale, for example using the posterior mean for $\log(s_g)$, the absence of closed-form expressions makes this less convenient.

2.2.5 *Unimodal prior assumption on variance or precision*

The above formulation is based on assuming a unimodal prior distribution for the variance s_g^2, and specifically by using a mixture of inverse-gamma distributions all with the same mode. An alternative is to assume a unimodal prior distribution for the precision $1/s_g^2$, by using a mixture of gamma distributions, all with the same mode. This is equivalent to using a mixture of inverse-gamma distributions for the variance s_g^2 as in (2.3) above, but with

$$b_k := (a_k - 1)/c \tag{2.18}$$

in place of (2.4), because the mode of a Gamma(a, b) distribution is at $c = (a-1)/b$. We present results for both approaches. In practice one can assess which of the two models provides a better fit to the data by comparing their (maximized) likelihoods (2.12). Note that in many (but not all) cases the fitted prior distributions under either or both approaches will end up being unimodal for both the variance *and* the precision. However, even in these cases, the optimal likelihood under each approach will typically differ because the family of unimodal distributions being optimized over is different.

2.2.6 *Moderated t-tests*

In differential expression analysis, testing if $\beta_g = 0$ is of primary interest. Smyth (2004) suggested using the "moderated t-test", which moderated the sample variance and degree of freedom by the shrunk variance estimates and its posterior degree of

freedom. We can also extend this moderated t-test to our mixture prior setting.

We define our t-score as follows:

$$\tilde{T}_g := \sum_k \tilde{\pi}_{gk} \frac{\hat{\beta}_g}{\tilde{s}_{gk}}. \tag{2.19}$$

Here we show that $\tilde{T}_g$ follows a mixture t-distribution:

$$\begin{aligned} P(\hat{\beta}_g|\hat{s}_g^2) &= \int P(\hat{\beta}_g|\beta_g, s_g^2) P(s_g^2|\hat{s}_g^2) ds_g & (2.20) \\ &= \int N(\hat{\beta}_g; \beta_g, s_g^2) \cdot \sum_k \tilde{\pi}_{gk} \text{InvGamma}(s_g; \tilde{a}_{gk}, \tilde{b}_{gk}) ds_g & (2.21) \\ &\sim \sum_k \tilde{\pi}_{gk} (\beta_g + \tilde{s}_{gk} t_{2\tilde{a}_k}) & (2.22) \end{aligned}$$

where $t_{2\tilde{a}_k}$ is a random variable following the t distribution with degree of freedom $2\tilde{a}_k$. Hence, our t-score follows the following mixture t-distribution under the null hypothesis $\beta_g = 0$:

$$\tilde{T}_g|\beta_g = 0 \sim \sum_k \tilde{\pi}_{gk} t_{2\tilde{a}_k}. \tag{2.23}$$

The p-value of testing $\beta_g = 0$ is thus given by

$$p_g = \sum_k \tilde{\pi}_{gk} P(|t_{2\tilde{a}_k}| > |\hat{\beta}_g / \tilde{s}_{gk}|). \tag{2.24}$$

The p-values measure the significance of gene-wise effects. To select top-ranked differential expressed candidate genes and control the false discovery rate, p-values

can be further adjusted for multiple testing using procedures like the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995), q-values (Storey, 2002), etc.

2.3 Results

2.3.1 Simulation studies

To compare and contrast our method with *limma* and *limmaR* we simulate data from the model (2.1)-(2.3), with $G = 10,000$, and degrees of freedom $df = 3, 10, 50$ (corresponding to sample sizes 4, 11 and 51 respectively) under various scenarios for the actual distribution of variances (scenarios A-D) or precisions (scenarios E-H), as summarized in Tables 2.1 and 2.2. The simulation scenarios are designed to span the range from a single inverse-gamma prior as assumed by *limma*, to more complex distributions under which we might expect our method to outperform *limma*. Specifically we consider:

- Single IG (Gamma): single component inverse-gamma prior on variance (or gamma prior on precision), which satisfies the assumptions of *limma*.
- Single IG (Gamma) with outliers: two component inverse-gamma prior on variance (or gamma prior on precision), where one component models the majority of genes and the other component, being more spread out, attempts to capture possible outliers. The method *limmaR* is specifically designed to deal with the case where large variance outliers exist.
- IG (Gamma) mixture: a more flexible inverse-gamma mixture prior on variance

(or mixture gamma prior on precision) with multiple components.

- Long tail log-normal mixture: log-normal mixture prior on variance or precision, which yields a longer tail than either the inverse-gamma or the gamma distribution.

Table 2.1: Parameters for the simulation scenarios with unimodal prior on variance

Scenario	Description	Prior of s_g^2
A	Single IG	InvGamma(10,11)
B	Single IG with outliers	0.1InvGamma(3,4)+0.9InvGamma(10,11)
C	IG mixture	0.1InvGamma(3,4) + 0.4InvGamma(5,6) + 0.5InvGamma(20,21)
D	Long tail log-normal mixture	0.7logN(0.0625,0.0625) + 0.3logN(0.64,0.64)

Table 2.2: Parameters for the simulation scenarios with unimodal prior on precision

Scenario	Description	Prior of $1/s_g^2$
E	Single gamma	Gamma(10,9)
F	Single gamma with outliers	0.1Gamma(2,1)+0.9Gamma(10,9)
G	Gamma mixture	0.1Gamma(2,1) + 0.4Gamma(5,4) + 0.5Gamma(30,29)
H	Long tail log-normal mixture	0.7logN(0.0625,0.0625) + 0.3logN(0.64,0.64)

For each simulation scenario we simulate 50 datasets and apply *limma*, *limmaR*, and our proposed method (*vash*) to estimate s_g^2 (or $1/s_g^2$). We compare the relative root mean squared errors (RRMSEs) of the shrinkage estimators, which we define by

$$\text{RRMSE}_{prec} := \frac{\sqrt{\mathbb{E}(1/s_g^2 - 1/\tilde{s}_g^2)^2}}{\sqrt{\mathbb{E}(1/s_g^2 - 1/\hat{s}_g^2)^2}},$$

$$\text{RRMSE}_{var} := \frac{\sqrt{\mathbb{E}(s_g^2 - \tilde{s}_g^2)^2}}{\sqrt{\mathbb{E}(s_g^2 - \hat{s}_g^2)^2}}.$$

The RRMSE measures the improvement of a shrinkage estimator over simply using the sample variance $\hat{s}_g^2$ or precision $1/\hat{s}_g^2$, with RRMSE=1 indicating no benefit of shrinkage. We also show the absolute root mean squared errors in the supplementary materials (Table S1, S2).

Figure 2.1 and 2.2 show the RRMSEs of *limma*, limmaR and *vash* for all scenarios. We summarize the main patterns as follows:

1. Across all scenarios, the mean RRMSE of *vash* is consistently no worse than either *limma* or *limmaR*, and is sometimes appreciably better. In contrast, limmaR sometimes performs better than *limma* and sometimes worse. In this sense *vash* is the most robust of the three methods.

2. In simulations under the simplest scenario (A and E) where the assumptions of *limma* are met, all three methods perform similarly. In particular, the additional flexibility of *vash* does not come at a cost of a drop of performance in the simpler scenarios.

3. When sample sizes are small (df=3) all methods perform similarly under all scenarios. This highlights the fact that the benefits of more flexible methods like *vash* are small if samples sizes are too small to exploit the additional flexibility. Put another way, for small sample sizes simple assumptions suffice.

4. When sample sizes are large (df=50) vash can outperform the other methods, particularly under the more complex scenarios (C,D; G,H), which most

strongly violate the assumptions of *limma*. Indeed, in these cases both *limma* and *limmaR* can have RRMSE>1, indicating that they perform worse than the unshrunken sample estimators. That is, when sample sizes available to estimate each variance are relatively large shrinkage estimates based on oversimplified assumptions can make estimation accuracy worse rather than better. (In contrast, for small sample sizes, the benefits of shrinkage greatly outweigh any cost of oversimplified assumptions.)

We also note that in scenario B where variances are sampled from a two component inverse-gamma mixture prior (one "majority" component and one "outlier" component), both *vash* and *limmaR* perform similarly on average (and slightly outperform *limma*), but results of *vash* are slightly more variable than *limmaR*. Possibly this reflects the fact that *limmaR* was specifically designed to deal with such cases.

Another metric for methods comparison is the accuracy in estimating prior distribution. We use D_{cdf}, the average distance between the cumulative distribution function (cdf) of true variance prior and that of estimated prior, to evaluate the prior estimation accuracy:

$$D_{cdf} := \frac{1}{M} \sum_{m=1}^{M} |cdf_{true}(x_m) - cdf_{fitted}(x_m)|, \tag{2.25}$$

where we take x_m ranging from 0 to 10 with increment size 0.01. Fig. S1 shows that the estimated mixture prior improves D_{cdf} by 0.02 for scenario D and 0.01 for scenario C & G comparing to the estimated single inverse-gamma prior, and this improvement keeps consistent regardless of sample size.

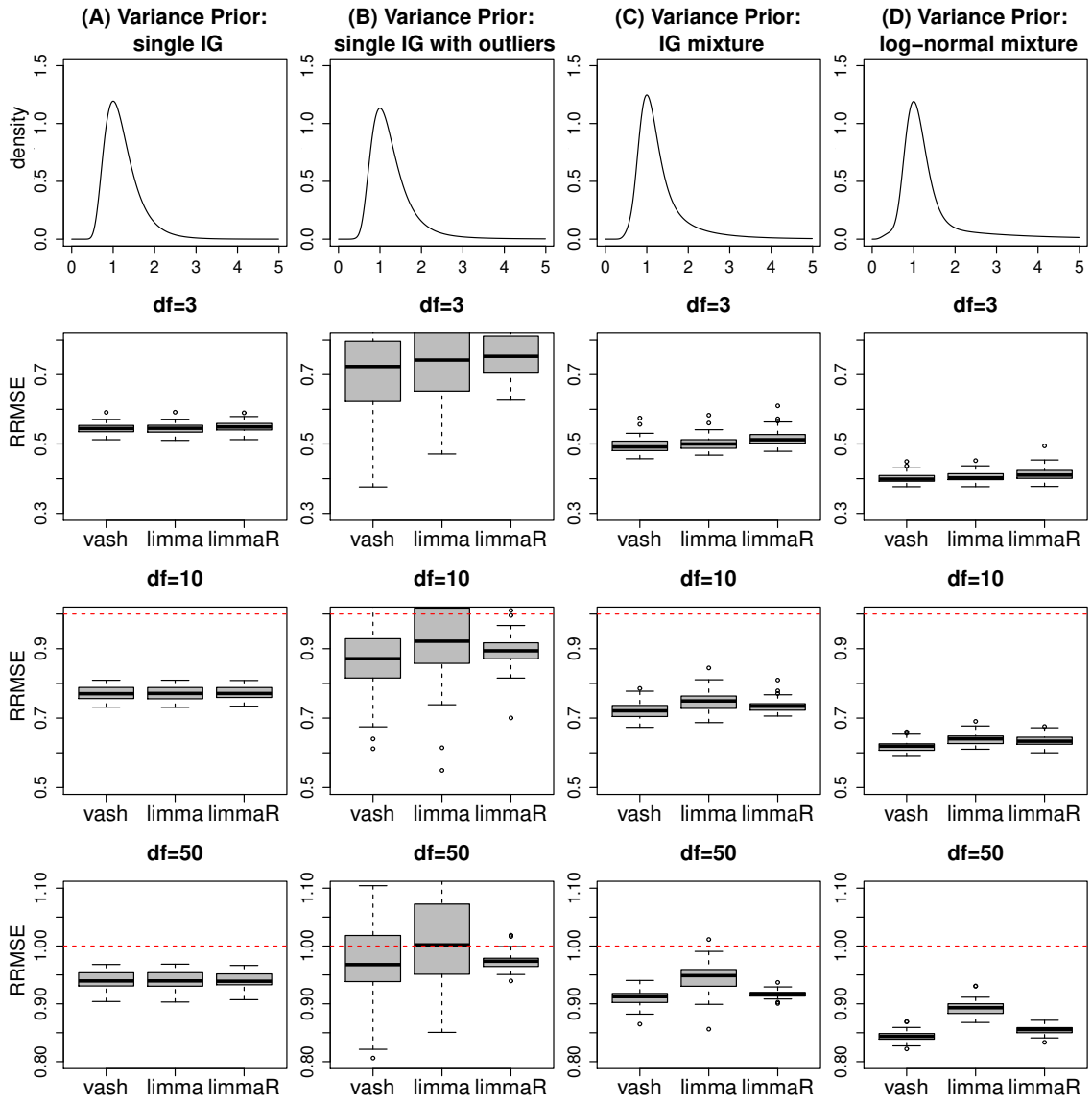

Figure 2.1: RRMSE$_{var}$ of three gene-specific variances estimators, *limma*, robust *limma* (*limmaR*) and our proposed estimator (*vash*) in the 4 simulation scenarios A-D with unimodal variance prior.

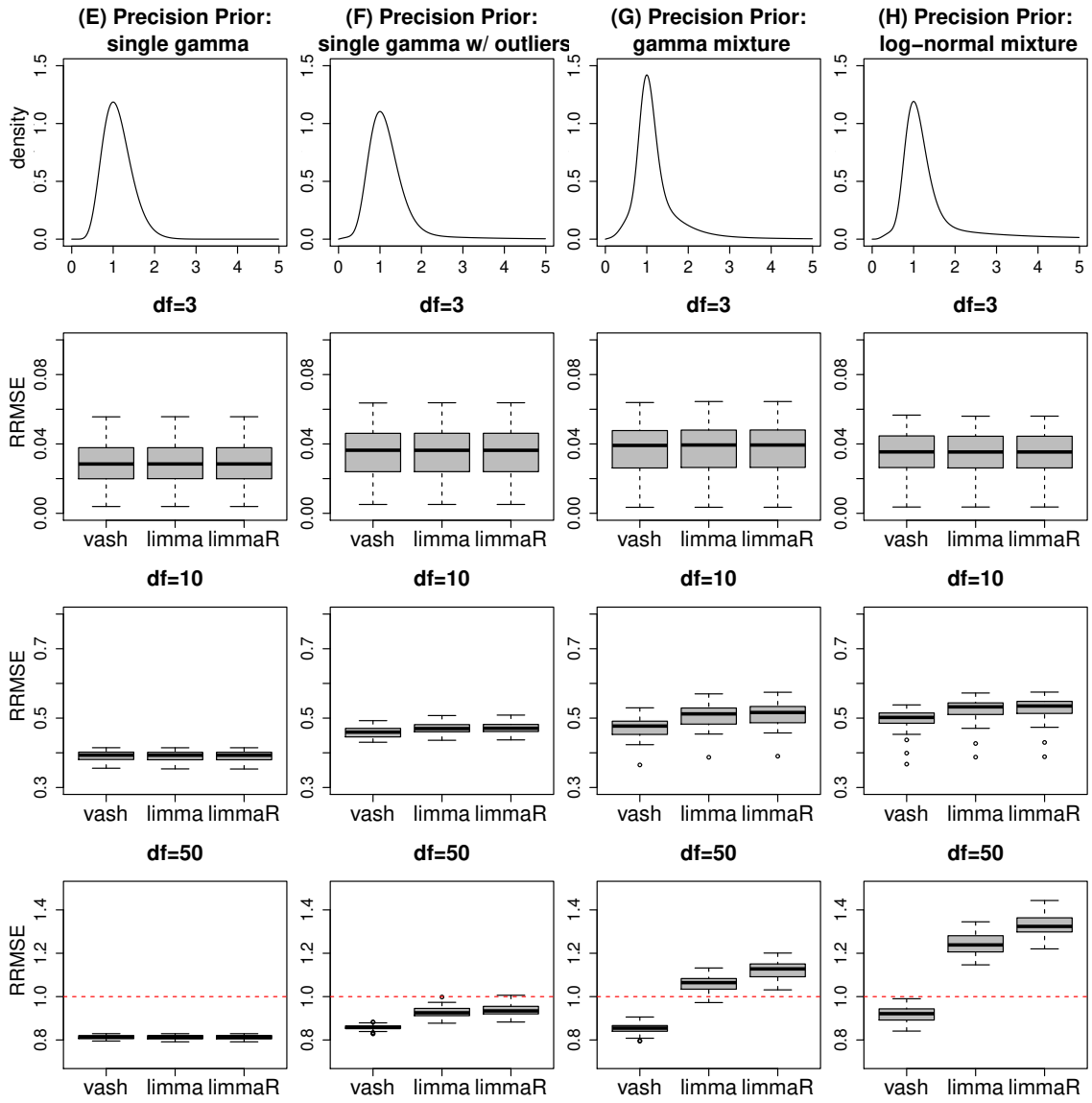

Figure 2.2: RRMSE$_{prec}$ of three gene-specific variances estimators, *limma*, robust *limma* (*limmaR*) and our proposed estimator (*vash*) in the 4 simulation scenarios E-H with unimodal precision prior.

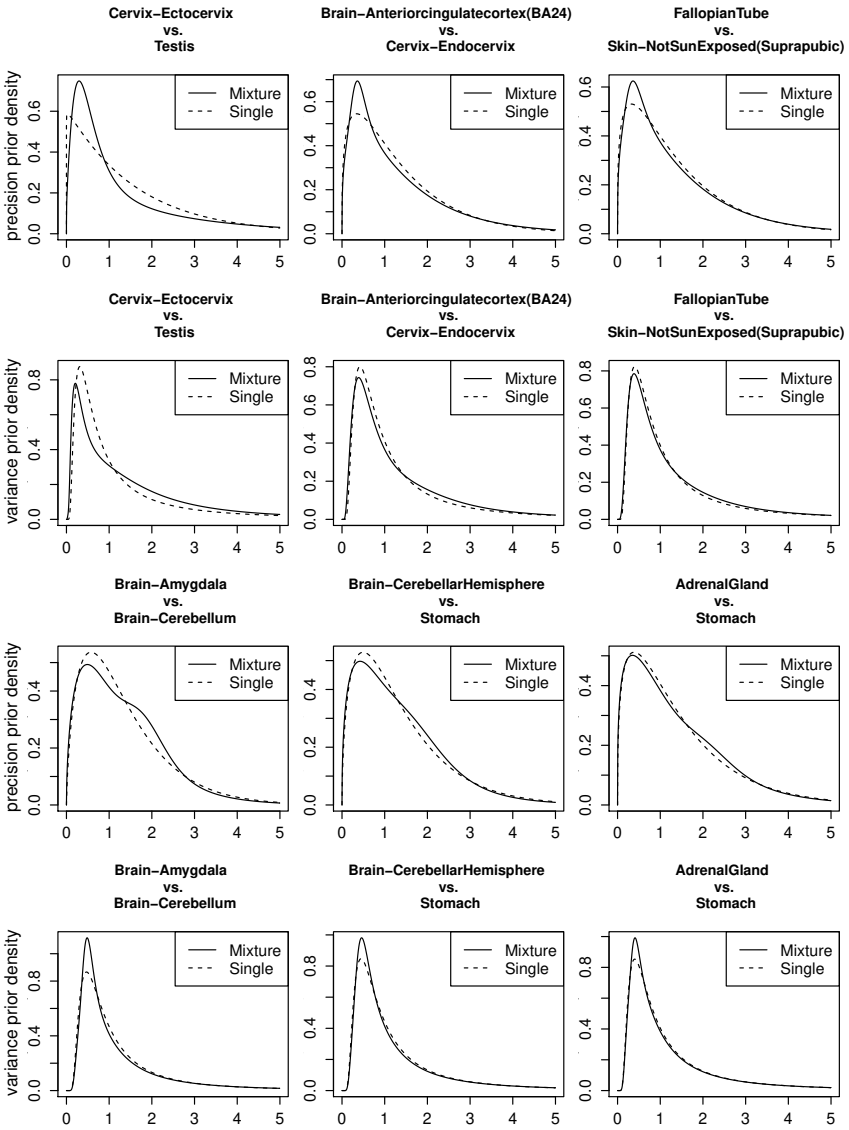

Figure 2.3: The variance priors (the 2nd and 4th row) and precision priors (the 1st and 3rd row) fitted by mixture prior model (black line) or single component prior model (red line) for 6 tissue pair comparisons. The differences in the log-likelihood between the mixture prior model and the single component prior model for tissue pair comparisons "Cervix-Ectocervix vs Testis", "Brain-Amygdala vs Brain-Cerebellum", "Brain-Anteriorcingulatecortex (BA24) vs Cervix-Endocervix", "Brain-CerebellarHemisphere vs Stomach", "Fallopian Tube vs Skin-Not Sun Exposed (Suprapubic)", "Adrenal Gland vs Stomach" are given by 705, 166, 78, 78, 44, 44 respectively (from top-left to bottom-right).

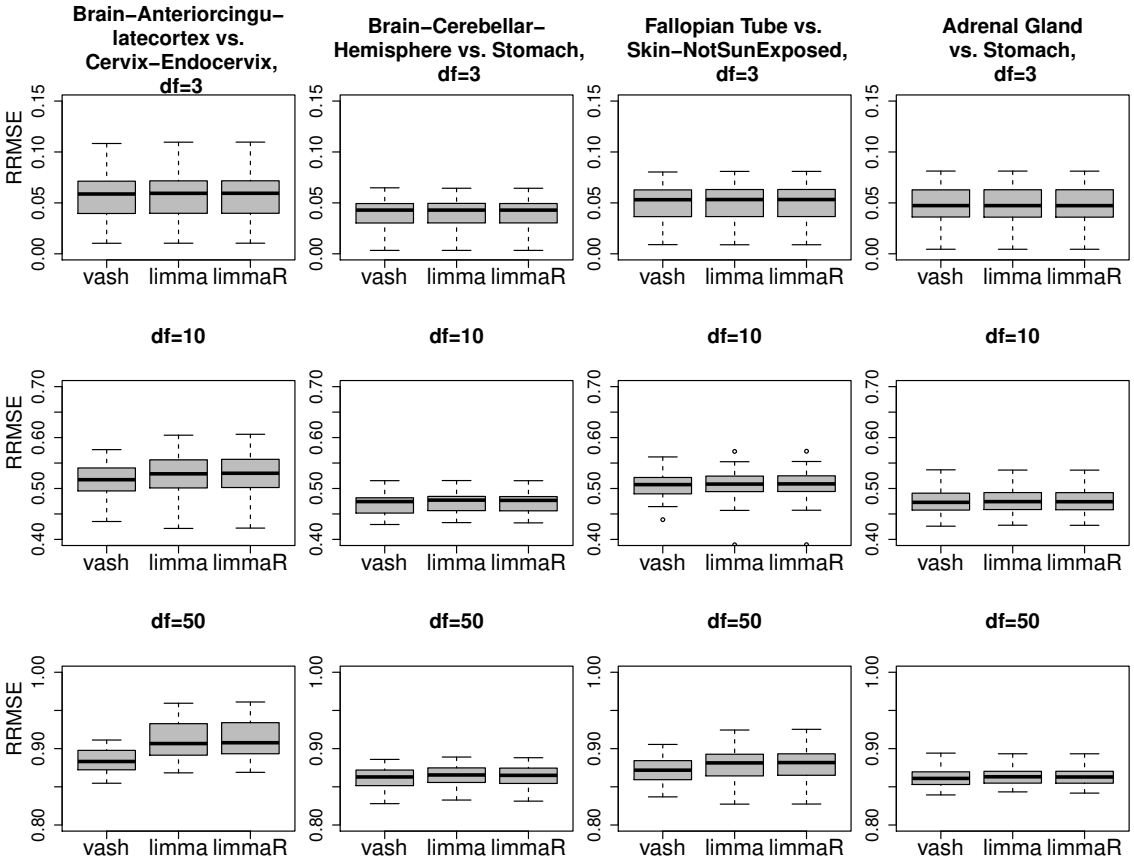

Figure 2.4: RRMSE_{prec} of three gene-specific variances estimators, *limma*, robust *limma* (*limmaR*) and our proposed estimator (*vash*) in simulation scenarios, which simulate the last four GTEx tissue pair comparisons ("Brain-Anteriorcingulatecortex (BA24) vs Cervix-Endocervix", "Brain-CerebellarHemisphere vs Stomach", "Fallopian Tube vs Skin-Not Sun Exposed (Suprapubic)" and "Adrenal Gland vs Stomach") in Figure 2.3.

2.3.2 Assessment of variances in gene expression data

The results above demonstrate that the more flexible mixture prior implemented in *vash*, can in principle provide more accurate variance and precision estimates than the simple inverse-gamma prior implemented in *limma*. However, in practice these gains will only be realized if the actual distribution of variances differs from the single inverse-gamma model. Here we examine this issue using RNA sequencing data from the Genotype-Tissue Expression (GTEx) project (Lonsdale et al., 2013). The GTEx Project is an extensive resource which studies the relationship among genetic variation, gene expression, and other molecular phenotypes in multiple human tissues. Here we consider RNA-Seq data (GTEx V6 dbGaP accession phs000424.v6.p1, release date: Oct 19, 2015, http://www.gtexportal.org/home/) on 53 human tissues from a total of 8555 samples (ranging from 6 to 430 samples per tissues).

Since in practice variance estimation is usually performed as part of a differential expression analysis (Smyth, 2004), we mimicked this set-up here: specifically we considered performing a differential expression analysis between every pair of tissues. We selected the top 20,000 highly expressed genes, transformed their read counts into log-cpm by "voom" transformation (Law et al., 2014), and used the `lmFit` function in *limma*package to estimate the effects and variances. Since there are 53 tissues this resulted in 1378 datasets of variance estimates.

First, for each data set, we quantified the improved fit of the mixture prior vs a single component prior by comparing the maximum log-likelihood under each prior. (For the mixture prior we fitted both the unimodal-variance and unimodal-precision priors, and took the one that provided the larger likelihood.) In principle the mixture

prior log-likelihood should always be larger because it includes the single component as a special case; we observed rare and minor deviations from this in practice due to numerical issues. Across all 1378 datasets the average gain in log-likelihood of the mixture prior vs the single component prior was 34.1. The 25% quantile, median, 75% quantile, 90% quantile and maximum of the difference are given by 2.9, 15.8, 42.9, 77.4 and 705.2 respectively. A log-likelihood difference of 15.8 is already quite large: for comparison the maximum difference in log-likelihood for simulations under a single component model, Scenario A, df=50, was 1.9. We therefore conclude that the mixture component prior fits the data appreciably better for many datasets.

To visualize the deviations from a single component prior present in these data, we examine the fitted priors in datasets where the log-likelihood differences are about 42.9 (75% quantile), 77.4 (90% quantile) and higher. Figure 2.3 compares the fitted single component prior and mixture prior on several typical scenarios. Generally, the mixture priors use extra components to better fit the middle portion of distribution. The single component priors can match the tails pretty well, but often fails to accurately capture the peak.

Overall, our impression from Figure 2.3 is that differences between the fitted priors seem relatively minor, and might be expected to lead to relatively small differences in accuracy of shrinkage estimates, despite the large likelihood differences. To check this impression we simulated data where the variances are generated from the fitted mixture priors for four of these datasets (the four datasets on the right hand side of Figure 2.3). Figure 2.4 compares the RRMSEs of *vash*, *limma* and *limmaR* in these four scenarios. In general the results confirm our impression: the

three methods perform very similarly in most scenarios, although *vash* shows some gain in accuracy in two scenarios with df=50.

2.4 Discussion

We have presented a flexible empirical Bayes approach ("variance adaptive shrinkage", or "vash") to shrinkage estimation of variances. The method makes use of a mixture model to allow for a flexible family of unimodal prior distributions for either the variances or precisions, and uses an accelerated EM-based algorithm to efficiently estimate the underlying prior by maximum likelihood. Although slower than *limma*, *vash* is computationally tractable for large datasets: for example, for data with 10,000 genes, *vash* typically takes about 30 seconds (*limma* takes just a few seconds).

Our results demonstrate that *vash* provides a robust and effective approach to variance shrinkage, at least in settings where the distribution of the variances (or precisions) is unimodal. When the true variances come from a single inverse-gamma prior, *vash* is no less accurate than the simpler method. When the variances come from a more complex distribution *vash* can be more accurate than simpler methods if the sample sizes to estimate each variance are sufficiently large.

In the gene expression datasets we examined here, the gains in accuracy of *vash* vs *limma* are small, and likely not practically important. While this could be viewed as disappointing, it nonetheless seems useful to show this, since it suggests that in many gene expression contexts the simpler approaches will suffice. At the same time, it remains possible that our method could provide practically useful gains in accuracy

for other data-sets, and as we have shown, it comes at little cost. In addition, our work provides an example of a general approach to empirical Bayes shrinkage – use of mixture components with a common mode to model unimodal prior distributions – that could be useful more generally.

Our method is implemented in an R package `vashr` available from http://github.com/mengyin/vashr.

Chapter 3

DETECTING DIFFERENTIALLY EXPRESSED GENES FROM RNA-SEQ DATA USING ADAPTIVE SHRINKAGE METHODS

3.1 Introduction

High-throughput sequencing of RNA (RNA-Seq) has proven to be invaluable in understanding gene regulation and its downstream effects. An important task in analyzing RNA-Seq data is to identify genes that are differentially expressed across groups of samples. Many methods have been proposed for differential expression (DE) analysis of RNA-Seq data. The idea of improving statistical power by pooling information across genes has been widely used in these methods. For example, *DESeq* (Anders and Huber, 2010), *edgeR* (Robinson et al., 2010) and *DSS* (Wu et al., 2013) use shrinkage estimators to improve gene-wise dispersion estimation accuracy, *limma* (Smyth, 2004) and *vash* (Lu and Stephens, 2016) focus on the accuracy of the variance component by shrinking the error variances, and *DESeq2* (Love et al., 2014) provides shrunk effect estimates by putting a shrinkage prior on effects to improve estimation accuracy of the effect sizes.

In a typically analysis of RNA-Seq data, gene-wise p-values are typically computed from these shrinkage quantities to test if the effects (expression differences between conditions) are significantly different from zero. Multiple testing adjustment procedures (e.g. Benjamini-Hochberg adjustment (Benjamini and Hochberg, 1995) or q-values (Storey, 2002)) are then derived from p-values to control the false

discovery rate (FDR). However, this procedure has a few drawbacks. First, they rely on the assumption that all p-values near 1 imply no differential expression. If this assumption does not hold, then we may lose much statistical power by being over-conservative. Second, they can provide poor (over-conservative) FDR estimates when the data contain both high precision measurements and low precision measurements.

Recently, Stephens (2016) proposed a novel FDR estimation approach, *Adaptive Shrinkage* (*ash*), to tackle this problem. *ash* has several advantages over the classical p-value based methods, and specifically targets the two aforementioned issues. However, while *ash* is a generic, adaptive, flexible and powerful statistical tool, there are several practical issues that need to be resolved to apply ash to RNA-Seq data:

1. How to take the count data generated by a standard RNA-Seq sequencing protocol and turn it into suitable input to *ash*, which assumes a normal means (or t means) model.

2. The inaccuracy in standard errors limits the performance of *ash* in small sample size cases. It is hence important if we can incorporate variance shrinkage (e.g. as in *limma* or *vash* in Chapter 2) into the analysis.

3. *ash* assumes independence among the tests (genes), but RNA-Seq data often have unwanted variation due to correlation structures and unmeasured confounding factors.

In this work, we propose a pipeline "*VL+eBayes+ash*" for differential expression analysis on RNA-Seq data. Our pipeline combines count data transformation *voom* (Law et al., 2014), variance modeling *vash* (Lu and Stephens, 2016) and adap-

tive shrinkage procedures *ash* (Stephens, 2016). This pipeline successfully resolves the first two issues mentioned above while retaining the advantages of all three techniques. For the third issue, we try methods based on the "empirical null" idea (Efron, 2004) to remove the unwanted variation. Some factor analysis based methods (Leek and Storey, 2007; Leek et al., 2012; Gagnon-Bartsch and Speed, 2012; Risso et al., 2014; **?**) have also been proposed to deal with the confounders present in RNA-Seq data. We show on simulated RNA-Seq data that our proposed pipeline is adaptive, statistically powerful and gives well-calibrated FDR. For small sample size cases in particular, our proposed pipeline yields noticeably better performance compared to the existing widely-used methods *DESeq2*, *edgeR* and *voom+limma*.

The R code, simulation results and analysis results are available from `http://github.com/mengyin/EBNM`.

3.2 Methods

3.2.1 Obtain shrunk effect estimates and control FDR with ash

Here we consider a typical differential gene expression analysis. Let $\beta = (\beta_1, \ldots, \beta_J)$ denote "effects" of interest for J genes.

ash assumes that the effect β_j has a unimodal prior g with the mode at 0:

$$g(\cdot) = \pi_0 \delta_0(\cdot) + (1 - \pi_0) g_1(\cdot), \tag{3.1}$$

where π_0 is the proportion of effects that are null, $\delta_0(\cdot)$ denotes a point mass at 0, and $g_1(\cdot)$ denotes the distribution of the non-zero β_j. Here we assume that g_1 is

a unimodal distribution. Note that existing methods tend to over-estimate π_0 and hence lose statistical power. However, *ash* is able to give more accurate estimates of π_0, resulting in more accurate effect size and FDR estimates.

Instead of simply modeling the p-values, *ash* models both the effect estimates $\hat{\beta}_j$ and their standard errors $\hat{s}_j$ and use the posterior distribution of β to estimate β and FDR. This allows *ash* models to account for variation in measurement precision across tests, and circumvents the main issue that plagues traditional p-value based methods: poor-precision measurements that lead to inflated FDR estimates.

Having specified the prior, suppose that the likelihood $p(\hat{\beta}|\beta, \hat{s})$ is approximated by

$$p(\hat{\beta}|\beta, \hat{s}) = \prod_j N(\hat{\beta}_j; \beta_j, \hat{s}_j^2), \tag{3.2}$$

or

$$p(\hat{\beta}|\beta, \hat{s}) = \prod_j T_\nu(\hat{\beta}_j; \beta_j, \hat{s}_j), \tag{3.3}$$

where $T_\nu(\beta_j, \hat{s}_j)$ denotes the distribution of $\beta_j + \hat{s}_j T_\nu$, in which T_ν has a standard t distribution on ν degrees of freedom.

We take an Empirical Bayes (EB) approach to estimate the hyperparameters of the prior g. In practice g_1 is assumed to be a mixture of uniform distributions or normal distributions, which are flexible enough to approximate generic unimodal distributions. We then make inferences on β based on the posterior distribution

$$p(\beta|\hat{\beta}, \hat{s}) \propto p(\beta|\hat{s})p(\hat{\beta}|\beta, \hat{s}), \tag{3.4}$$

where the prior is $p(\beta|\hat{s}) = \prod_j g(\beta_j)$.

3.2.2 *Extend* ash *to deal with small sample size cases*

One important consideration when applying *ash* on RNA-Seq data is that of small sample sizes. In such cases the raw estimated standard errors $\hat{s}_j$ can be highly variable, and the normal likelihood approximation (3.2) is often be questionable. Even the t-likelihood (3.3) assumption in Stephens (2016) is not entirely satisfactory, since it ignores the randomness of $\hat{s}$. From standard regression theory, we have

$$\frac{\hat{\beta}_j - \beta_j}{\hat{s}_j} \sim T_\nu. \tag{3.5}$$

However, (3.5) does not imply (3.3) since $\hat{s}$ is random, and cannot result in the conditional t-likelihood in (3.3).

Fortunately, Bayesian modeling allows us to develop a proper likelihood $p(\hat{\beta}|\beta, \hat{s})$ by incorporating the variation in $\hat{s}$. Suppose $\hat{s}$ is a noisy estimate of the true standard deviation s of β (i.e. $\hat{\beta}|\beta, s \sim N(\hat{\beta}; \beta, s^2)$), Then the likelihood $p(\hat{\beta}|\beta, \hat{s})$ is given by

$$p(\hat{\beta}|\beta, \hat{s}) = \int p(\hat{\beta}|\beta, s)p(s|\hat{s})ds \tag{3.6}$$

$$= \int N(\hat{\beta}; \beta, s^2)p(s|\hat{s})ds, \tag{3.7}$$

assuming that the distribution of s does not depend on β i.e.

$$p(s|\hat{s}) = p(s|\hat{s}, \beta). \tag{3.8}$$

In genomics, it is a common practice to obtain $p(s|\hat{s})$ in (3.7) by applying EB methods (*limma* (Smyth, 2004), *vash* (Lu and Stephens, 2016)) to "moderate" (i.e. shrink) variance estimates. Assuming that all gene-specific variances come from a common prior, shrinkage variance estimates yield higher estimation accuracies by borrowing information across genes. *limma* further computes p-values from the "moderated" test statistics based on these shrinkage variance estimates, but does not model the effect estimates. Hence, combining the functionality of *ash* (modeling effects) with that of *limma* (modeling variance) is an natural way to retain the advantages of both methods.

Suppose we obtain the conjugate inverse-gamma posterior $p(s|\hat{s})$ from the variance modeling methods, resulting in

$$s_j^{-2}|\hat{s} \sim \tilde{s}_j^{-2}\frac{\chi^2_{\tilde{\nu}}}{\tilde{\nu}}, \tag{3.9}$$

where $\tilde{s}_j$ is the shrinkage estimate for s_j, $\tilde{\nu}$ denotes the "moderated" degree of freedom, and $\chi^2_{\tilde{\nu}}$ is a χ^2 random variable with $\tilde{\nu}$ degree of freedom. Integrating (3.9) into (3.7), the likelihood is thus given by

$$p(\hat{\beta}|\beta, \hat{s}) = \prod_j T_{\tilde{\nu}}(\hat{\beta}_j; \beta_j, \tilde{s}_j), \tag{3.10}$$

We propose using the above t-likelihood to account for variance shrinkage, instead of using the naive t-likelihood (3.3) suggested in Stephens (2016).

3.2.3 *Apply* ash *on RNA-Seq data*

Another important issue to consider when applying *ash* to RNA-Seq is that of the presence of counts in the data. RNA-Seq reads are often modeled by (possibly over-dispersed) Poisson (Marioni et al., 2008) or Negative Binomial distribution (Robinson et al., 2010; Anders and Huber, 2010), where the variance of distribution depends on the mean parameter. However, this mean-variance or mean-dispersion relationship results in correlations between effect sizes and their variances, and hence violates the assumption (3.8) in previous section.

Law et al. (2014) proposed *voom+limma*, a normal linear RNA-Seq modeling framework which suggests modeling the de-trended variances. *voom*transforms count data into Gaussian representations (log counts per million) with weights, and then use weighted least squares regression to estimate effects:

$$\mathbf{y}_j = X\beta_j + \mathbf{e}_j, \quad \mathbf{e}_j \sim N(0, W_j^{-1}\sigma_j^2), \tag{3.11}$$

where $\mathbf{y}_j$ is the vector of log-cpm (log counts per million) for gene j, X denotes the design matrix, W_j denotes the diagonal weight matrix, and σ_j^2 denotes the de-trended error variance. Here the mean-variance relationship is fully adjusted by the weight W_j, and thus the de-trended variance σ_j^2 no longer depends on count level. The very same *limma* pipeline can be then applied on σ_j^2 since they are exchangeable across genes.

As discussed in the previous section, we can further combine *limma* and *ash* by moderating the t-likelihood with the shrunk variance and moderated degree of

freedom. Note that in the WLS framework (3.11) the standard error of β_j is given by $\hat{s}_j^2 = (X^T W_j X)^{-1}\sigma_j^2$, so the moderated t-likelihood (3.10) should take

$$\tilde{s}_j^2 = (X^T W_j X)^{-1}\tilde{\sigma}_j^2, \tag{3.12}$$

where $\tilde{\sigma}_j^2$ is the shrunk estimate of σ_j^2.

3.2.4 Proposed pipeline: VL+eBayes+ash

In summary, we propose the following pipeline "*VL+eBayes+ash*" for differential expression analysis on RNA-Seq data:

1. Use *voom* transformation to obtain effect estimates $\hat{\beta}_j$, weights W_j, and detrended variance estimates $\hat{\sigma}_j^2$.

2. Use *limma* or *vash*to model σ_j^2, and compute the posterior estimates $\tilde{\sigma}_j^2$ & degrees of freedom $\tilde{\nu}_j$.

3. Apply *ash* on $(\hat{\beta}_j, \tilde{s}_j)$'s with degrees of freedom $\tilde{\nu}_j$, where $\tilde{s}_j^2 = (X^T W_j X)^{-1}\tilde{\sigma}_j^2$. Use the posterior distribution of β to assess its significance and estimate FDR.

3.2.5 Dealing with unwanted variation in data

Another practical concern when analyzing RNA-Seq data is the presence of unwanted variation. Rocke et al. (2015) has shown that the popular RNA-Seq analysis softwares *DESeq2*, *edgeR* and *voom+limma* all produce large numbers of false positives in cases where the null hypotheses are true by construction. Even on wholly null RNA-Seq

datasets, where samples from same condition are randomly divided into two groups (so no genes are differentially expressed), the p-values produced by these existing methods do not follow the expected uniform distribution, which implies potential dependence among the gene-wise tests.

This unwanted variation might be caused by various factors, including unmeasured confounders (e.g. batch effects, sample correlation), gene-wise correlations, etc. This issue is particularly important in practice because they can lead to strong correlations among large numbers of tests and result in anti-conservativeness and a failure to control FDR.

Some studies (Leek and Storey, 2007; Leek et al., 2012; Gagnon-Bartsch and Speed, 2012; Risso et al., 2014; **?**) propose factor models to estimate confounding factors and mitigate this issue to a certain extent, especially when control genes (known null genes where no effects exist) are provided. In such cases, we can add the estimated confounding factors V into the WLS regression framework (3.11):

$$\mathbf{y}_j = X\beta_j + V\gamma_j + \mathbf{e}_j, \quad \mathbf{e}_j \sim N(0, W_j^{-1}\sigma_j^2). \tag{3.13}$$

However, using factor models to estimate confounders is typically infeasible in when sample sizes are small, since it is hard to separate confounding effects from true effects in this case. Hence, we consider another approach to address the unwanted variation issue for small sample size applications. Note that the presence of unwanted variation makes the test statistics deviate from their expected distribution under the independence assumption (no correlations or confounders). Efron (2004, 2007, 2010) proposed the "empirical null" methods to deal with the general

cases where z scores do not follow the theoretical null distribution $N(0,1)$. They use an estimated empirical null distribution $f_0 = N(0, \lambda_1^2)$ to replace the theoretical null distribution$N(0,1)$, and then adjust the test results accordingly. Schwartzman (2010) pointed out that positive correlation among the tests often result in deflated z-score variance ($\lambda_1 < 1$), and unobserved confounders can lead to inflated z-score variance ($\lambda_1 > 1$).

We adopt and extend this "empirical null" idea to address the unwanted variation issue for RNA-Seq data. Assume that under the null

$$\frac{\hat{\beta}_j - \beta_j}{\lambda_1 \tilde{s}_j + \lambda_2} \sim T_{\tilde{\nu}}. \tag{3.14}$$

Compared to (3.10), we have two additional parameters λ_1, λ_2 to adjust the distribution of actual t-scores. While Efron (2004) only uses a scale parameter λ_1 to control the empirical variance of z-scores, our assumption (3.14) has more flexibility when modeling the empirical null distribution.

If there exist control genes, λ_1, λ_2 are directly estimated by matching the moments. Since

$$\mathrm{E}(\beta_j^2) = \frac{\nu}{\nu - 2}(\lambda_1 \tilde{s}_j + \lambda_2)^2, \tag{3.15}$$

we solve λ_1, λ_2 from the following equations using the least squares method:

$$\hat{\beta}_j \approx \sqrt{\frac{\nu}{\nu - 2}}(\lambda_1 \tilde{s}_j + \lambda_2). \tag{3.16}$$

If no control genes are provided, we first estimate λ_1 as in Efron (2004), and then

estimate λ_2 by matching the second moments of effect sizes. We then replace the standard error $\tilde{s}_j$ in (3.10) by $\lambda_1 \tilde{s}_j + \lambda_2$ in our proposed pipeline *VL+eBayes+ash*.

Note that in real data the "variance deflation" ($\lambda_1 < 1$ and/or $\lambda_2 < 0$) pattern arises on occasion. Nevertheless, deflating the standard errors in differential expression analyses can be dangerous and lead to high false discovery rates. In this case, it is advisable to set constraints $\lambda_1 \geq 1$ and $\lambda_2 \geq 0$ to avoid such anti-conservativeness.

3.3 Simulation studies

3.3.1 Simulation scheme

Rocke et al. (2015) claimed that even if existing methods perform well on simulations tailored specifically to RNA-Seq data (e.g. Negative Binomial distribution with specific parameter structures), they often failed to control FDR in real data. Hence, a more generic and realistic simulation scheme should be designed to investigate the performance of RNA-Seq analysis methods.

Here we propose the following simulation scheme based on real RNA-Seq data, making distributional assumptions only sparingly:

1. Sample $2N$ RNA-Seq samples from the same group (such that there are no systematic differences among the samples) in a real RNA-Seq dataset, and divide them into two groups (A,B) with equal sizes N. This is an entirely null dataset by construction where all effects are truly zero. Let C_{ji} denotes the read count for gene j and sample i.

2. Suppose J denotes the number of genes. We randomly select $J(1-\pi_0)$ genes as

"alternative genes", and then generate their effects (log2 fold-changes between two groups) β_j's from an unimodal distribution g_1.

3. For these alternative genes, if $\beta_j > 0$ (such that group B should be more highly expressed), we use Poisson thinning to achieve the desired fold-change 2^{β_j} i.e. thin the read counts in group A as follows,

$$C^*_{ji} \sim Binomial(C_{ji}, 2^{-\beta_j}), \quad \forall i \in A. \tag{3.17}$$

Similarly if $\beta_j < 0$, thin the read counts in group B:

$$C^*_{ji} \sim Binomial(C_{ji}, 2^{-\beta_j}), \quad \forall i \in B. \tag{3.18}$$

Replacing C_{ji} by C^*_{ji} will result in a new RNA-Seq dataset, where the true effects roughly follow the unimodal assumption (3.1).

This simulation scheme allows us to generate RNA-Seq datasets with our desired effect distributions, while still preserving most of the structure (correlation, magnitude, etc) of the actual RNA-Seq data.

3.3.2 Simulate RNA-Seq data with independent genes

As discussed in Section 3.2.5, the subsampled null dataset from real RNA-Seq data likely preserves the correlation structure among genes due to unmeasured confounding factors. We can manually remove this dependence structure by moderating the first simulation step in Section 3.3.1: For each gene, we randomly subsample $2N$

samples from the raw RNA-Seq dataset, and record the corresponding read counts for this gene. Since different genes use read counts from different subsets of samples, all genes are independent from one other by design.

Following this moderated simulation scheme, we run simulations based on RNA-Seq data from the Genotype-Tissue Expression (GTEx) project (Lonsdale et al., 2013). The GTEx Project studies the relationship among genetic variation, gene expression, and other molecular phenotypes in multiple human tissues, and is an extensive source of data. The GTEx RNA-Seq data (GTEx V6 dbGaP accession phs000424.v6.p1, release date: Oct 19, 2015, http://www.gtexportal.org/home/) consists of 8555 samples on 53 human tissues (ranging from 6 to 430 samples per tissues). We use the liver tissue samples (119 samples in total) to construct the initial null RNA-Seq datasets, and then use the 10,000 top expressed genes for use.

Since no genes are supposed to be differentially expressed in null datasets, the null datasets can be used to examine the conservativeness of the DE methods. Rocke et al. (2015) investigated the performances of *DESeq2*, *edgeR* and *VL+eBayes* on original RNA-Seq data where all samples belong to the same group, and concluded that no methods gave well-calibrated p-values. However, it was unclear whether this anti-conservativeness was due to the unwanted variation in real data or due to the limitations of methods. In this work we also compare these three methods on simulated null RNA-Seq datasets which have removed the dependence structure.

Figure 3.1 shows the proportion of genes with p-values under a threshold (0.001, 0.01, 0.1) and proportion of discoveries if declaring genes with q-values under this threshold as positives. Ideally the p-values are uniformly distributed under the null

assumption, so the proportion of genes with p-values under the threshold should be around this threshold. Even though the dependence among genes has been removed, *DESeq2* and *edgeR* still fail to provide nicely calibrated p-values. The method *VL+eBayes* is the only method that consistently gets satisfactory estimated null proportion (close to 1), well calibrated p-values and hardly any discoveries on null data. The results suggest that *DESeq2* and *edgeR* are at a disadvantage compared to *VL+eBayes* even on the "perfect null RNA-Seq" data.

Since these null simulations confirm that *VL+eBayes* generally has advantages over *DESeq2* and *edgeR*, from now on we mainly focus on *VL+eBayes* and *ash* based methods.

We perform simulations under scenarios with various alternative distribution g_1 (scenario settings shown in Table 3.1). For each scenario, we simulate 50 datasets with $N = 2, 4, 10$ respectively, in which π_0's are uniformly drawn from [0,1]. For different N, we have the scaling factors S_N ($S_2 = 0.125, S_4 = 0.5, S_{10} = 1.5$) to adjust for the sample sizes, such that the alternative distribution of t-scores have similar magnitudes as shown in Figure 3.2. We perform differential expression analysis on the simulated datasets. The dependence structure among genes has already been removed, and it allows us to study different methods under the "ideal" cases.

We analyzed each simulated dataset with the following methods:

- *VL+eBayes*: the DE pipeline proposed by Law et al. (2014), which provides the effect size estimates $\hat{\beta}$ and standard errors $\hat{s}$ by `voom` (transoform count data to Gaussian data) and `lmFit` (WLS model) functions in `limma` package (we denote this step as *VL*). Then the EB shrinkage function `eBayes` is applied on

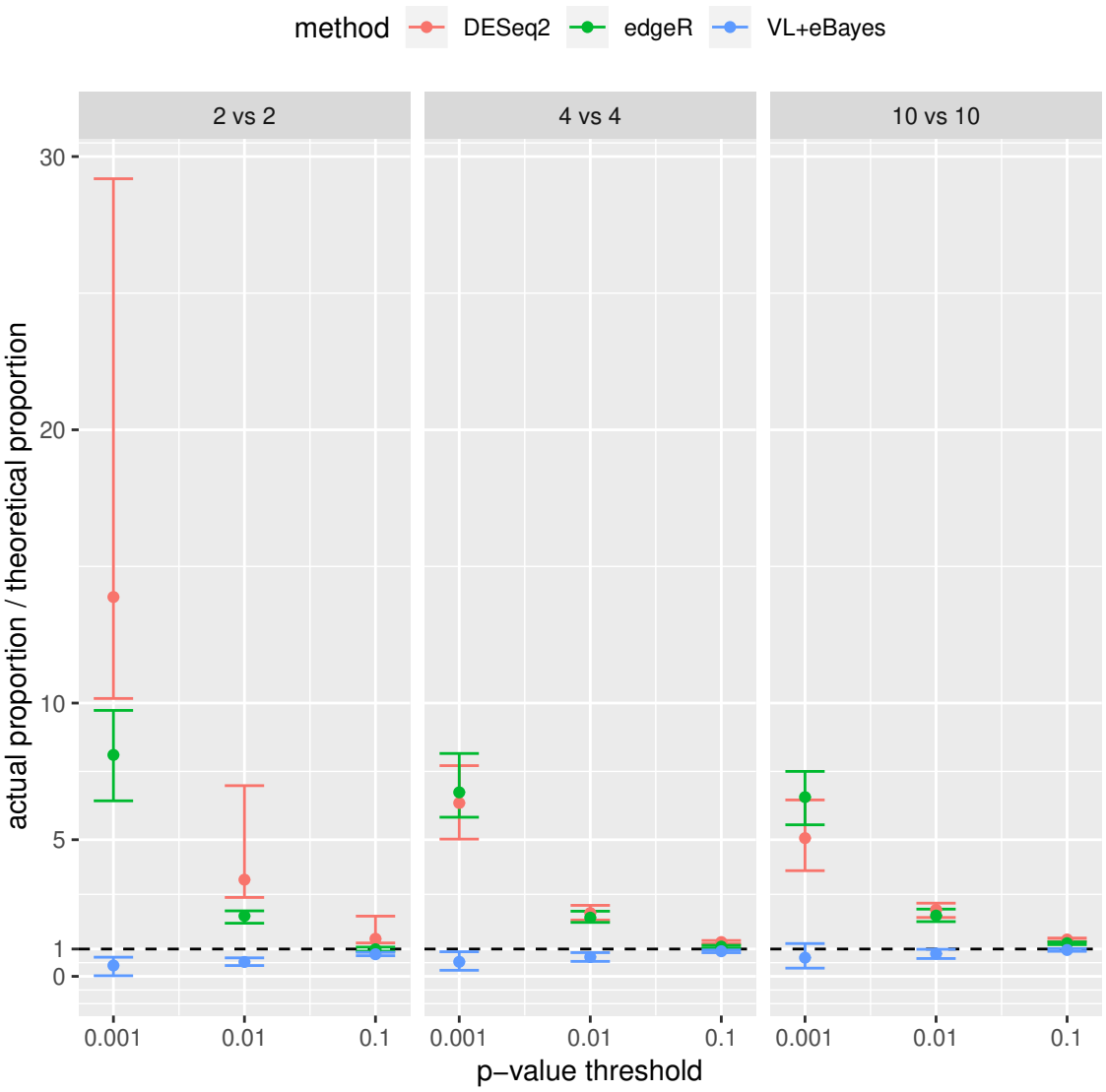

Figure 3.1: Comparison of proportion of genes with p-values under a threshold (0.001, 0.01, 0.1) on null simulations with independent genes. We show the 95% error bar of the ratio of observed proportion and theoretical proportion (which is exactly the threshold). *VL+eBayes* is the only method that can keep the proportion under its expected value (equals to the threshold).

Scenario	Alternative distribution, g_1
spiky	$\left[0.4N(0, 0.25^2) + 0.2N(0, 0.5^2) + 0.2N(0, 1^2), 0.2N(0, 2^2)\right] / S_N$
near-normal	$\left[2/3N(0, 1^2) + 1/3N(0, 2^2)\right] / S_N$
flat-top	$[(1/7)[N(-1.5, .5^2) + N(-1, .5^2) + N(-.5, .5^2) +$ $N(0, .5^2) + N(0.5, .5^2) + N(1.0, .5^2) + N(1.5, .5^2)]/S_N$
big-normal	$\left[N(0, 4^2)\right] / S_N$
bimodal	$\left[0.5N(-2, 1^2) + 0.5N(2, 1^2)\right] / S_N$

Table 3.1: Summary of simulation scenarios considered

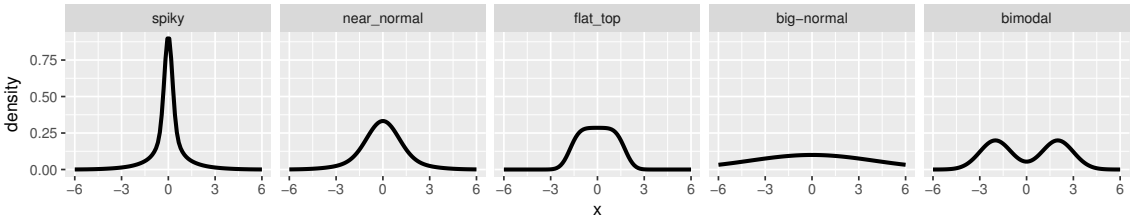

Figure 3.2: Densities of non-zero effects, g_1, used in simulations.

standard errors. The p-values are derived from the moderated t-tests and can then be adjusted for multiple testing by procedures like Benjamini-Hochberg or q-values.

- *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1*: our proposed pipeline, which first use the above *VL+eBayes* pipeline to obtain the estimated effect sizes $\hat{\beta}$ and moderated standard errors $\tilde{s}$ & degrees of freedom $\tilde{\nu}$, then feed them to the *ash* framework to further shrink $\hat{\beta}$ and compute q-value, with $\alpha = 0$ for *VL+eBayes+ash* and $\alpha = 1$ for *VL+eBayes+ash.alpha=1*.

- *VL+ash*: directly feed the *VL* estimates $\hat{\beta}$ and standard errors $\hat{s}$ into *ash* framework, without any variance shrinkage step.

- *VL+pval2se+ash*: convert the t-likelihood problem into a normal likelihood

problem. After obtaining $\hat{\beta}$ and p-value from *VL+eBayes*, compute the "adjusted standard error" s' where the z-score $|\hat{\beta}|/s'$ results in the same p-value. Then feed $\hat{\beta}$ and s' into the *ash* framework using the normal likelihood model. Since the normal likelihood is typically much easier to work with than the t likelihood (e.g. with normal prior), the method could be useful if its performance is comparable with that of *VL+eBayes+ash*, even though the procedure to obtain s' seems a bit "ad hoc".

FDR calibration An important issue in differential expression analysis is the calibration of FDR. Typically the overall false discovery rate should be controlled under a certain threshold (e.g. 0.05) when declaring significance. For the p-value based methods, it is customary to use multiple testing adjustment procedures (e.g. Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995), *qvalue* (Storey, 2002)) to estimate FDR. Here we use the R package `qvalue` to compute q-values from the p-values of *VL+eBayes*. Our method *VL+eBayes+ash* directly uses the q-values provided by `ashr` package, which is computed from the local false discovery rates. Even though both *qvalue* and *ash*'s q-values estimate the actual FDR, they have different underlying assumptions: *qvalue* assumes that alternative genes cannot have effect sizes close to 0 ("Zero Assumption", denote as ZA), while *ash* assumes the effect sizes of alternative genes come from an unimodal distribution with mode at 0. As a result, *qvalue* q-values are expected to be more conservative *ash* q-values, since *qvalue* is more likely to treat genes with small effect sizes as null genes.

Both `qvalue` and `ashr` package output the estimated proportion of null genes (π_0). If the estimated π_0 is significantly lower than the actual π_0, then the q-values

will likely lead to anti-conservative results and hence false significant genes. However, over-estimating π_0 is not desirable either, since being over-conservative can lead to low statistical power. Figure 3.3 compares the estimated null proportion π_0 with the true π_0 in our simulations.

Note that our data are simulated under the $\alpha = 0$ model (effect sizes are exchangeable), but *qvalue* assumes the $\alpha = 1$ model (z-scores are exchangeable). To better illustrate the differences between *ash* and *qvalue*, we also include method *VL+eBayes+ash.alpha=1* which assumes the z-scores are exchangeable and follow the UA.

Similar to the results shown in (Stephens, 2016), *VL+eBayes+qvalue* always provides much more conservative π_0 estimates than *VL+eBayes+ash* or *VL+eBayes+ash.alpha=1* regardless of sample size. Although *VL+eBayes+qvalue* and *VL+eBayes+ash.alpha=1* both use the $\alpha = 1$ model, their estimated null proportions can be quite different due to the discrepancy between ZA and UA. For the unimodal scenarios ("spiky", "near normal", "flat top" and "big-normal"), *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* give much more accurate π_0 estimates than *VL+eBayes+qvalue*. This is especially true for the "flat top" and "big-normal" scenarios, where *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* almost perfectly estimate π_0. For the "bimodal" scenario, *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* sometimes underestimate π_0, especially when the true π_0 is low. Therefore, as long as the unimodal assumption of *ash* model holds, *ash* yields more accurate π_0 estimates than *qvalue*, which agrees with the conclusions in (Stephens, 2016). *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* still preserve the desired FDR thresholds since its π_0 estimates are no smaller than

the true π_0's. However, applying *ash* to data which violate the unimodal assumption might result in anti-conservative results. The over-conservativeness of *qvalue* directly leads to loss of statistical power. According to Figure 3.5, *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* have the ability to discover more significant genes than *VL+eBayes+qvalue*, and still maintain desired levels of FDR when the true effect sizes follow an unimodal distribution (Figure 3.4).

Another point we would like to emphasis here is the necessity of the variance shrink step (*limma* or *vash*). We also try *voom+ash*, which simply feeds the unshrunk standard errors and degrees of freedom to *ash*. As we discussed in Section 3.2.2, this likelihood assumption is incorrect in small sample cases. Figure 3.3 and 3.4 show that *VL+ash* notably underestimates π_0 and produces extremely high false discovery rates. On the other hand, after moderating the t likelihood with EB shrunk standard errors, *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* get satisfactory π_0 estimates and false discovery rates.

Effect estimates In many applications, the estimates of effect sizes (β) are also of interest. We compare the estimated β of the methods with the true β by computing the root mean squared error (RMSE), $RMSE := \sqrt{\sum_j (\tilde{\beta}_j - \beta_j)^2}$. To illustrate the differences among the methods, we choose *VL+eBayes* as the baseline level, and compute the relative RMSE as the ratio of the method's RMSE and baseline RMSE. The relative RMSE's are shown in Figure 3.6.

We see that the RMSE of *edgeR* is always slightly higher than that of *VL+eBayes*. Note that *edgeR* and *VL+eBayes* simply obtain $\hat{\beta}$ from gene-wise regressions and do not shrink these raw estimates. *edgeR* uses count-based regression model to estimate

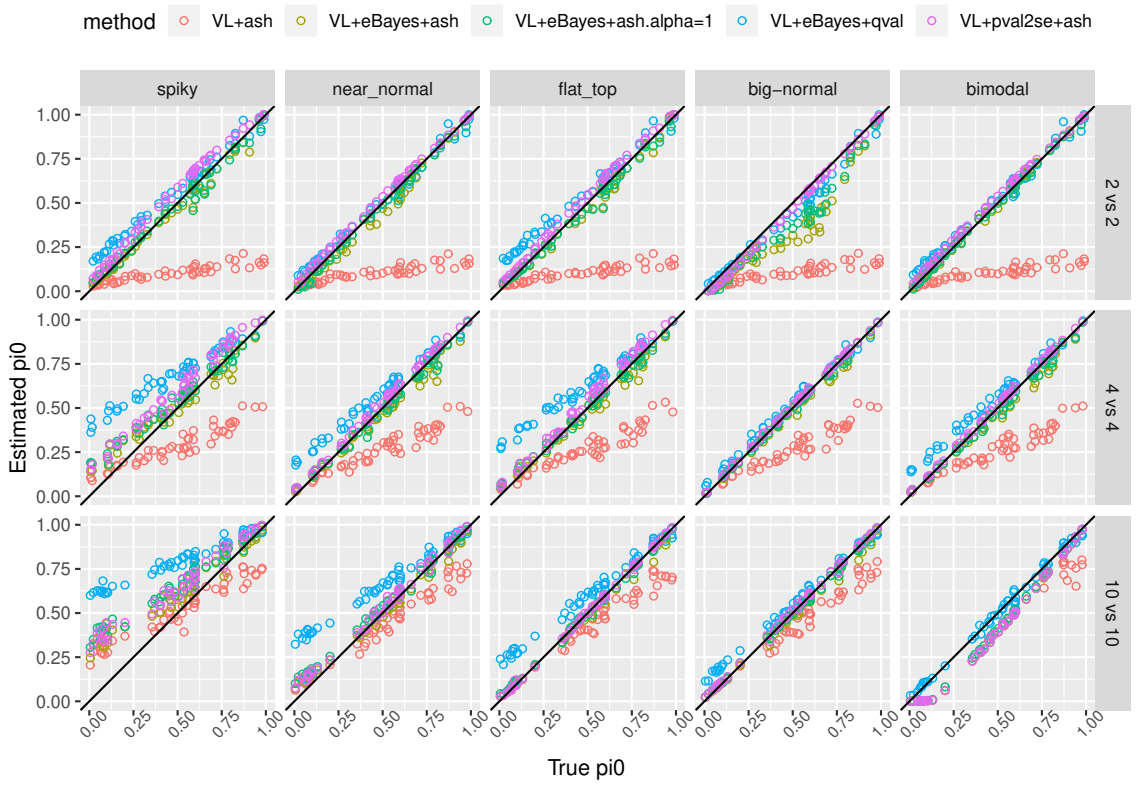

Figure 3.3: Comparison of true and estimated values of π_0 on simulations with independent genes. Generally *VL+ash* is very anti-conservative with extremely low estimates for π_0. When the UA holds the other three methods yield conservative (over-)estimates for π_0, with *VL+eBayes+ash*, *VL+eBayes+ash.alpha=1* and *VL+pval2se+ash* being less conservative, and hence more accurate. When the UA does not hold ("bimodal" scenario) the *VL+eBayes+ash* estimates are slightly anti-conservative.

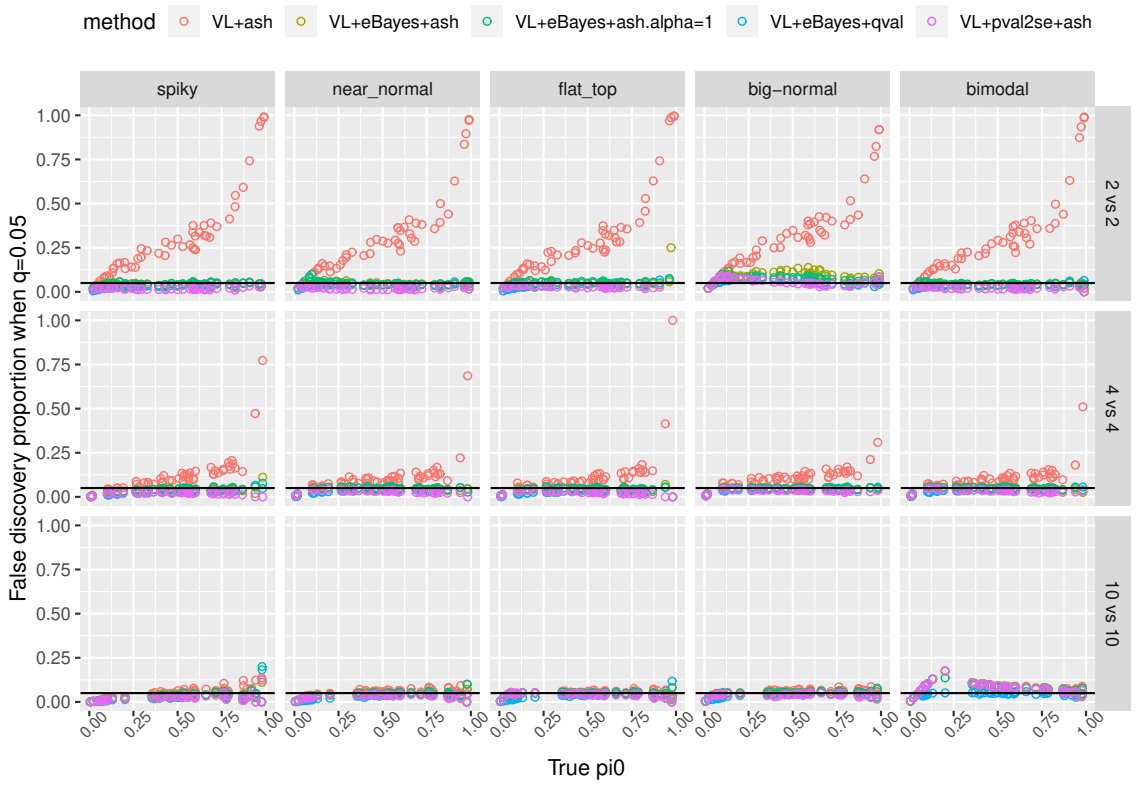

Figure 3.4: Comparison of actual false discovery proportions on simulations with independent genes if declaring genes with q-values under 0.05 as positives. *VL+eBayes+qvalue* and *VL+eBayes+ash* are generally able to control the false discovery proportion under 0.05 regardless of sample size. *VL+eBayes+ash* can be slightly anti-conservative when the UA does not hold ("bimodal" scenario) and π_0 is less than 0.5.

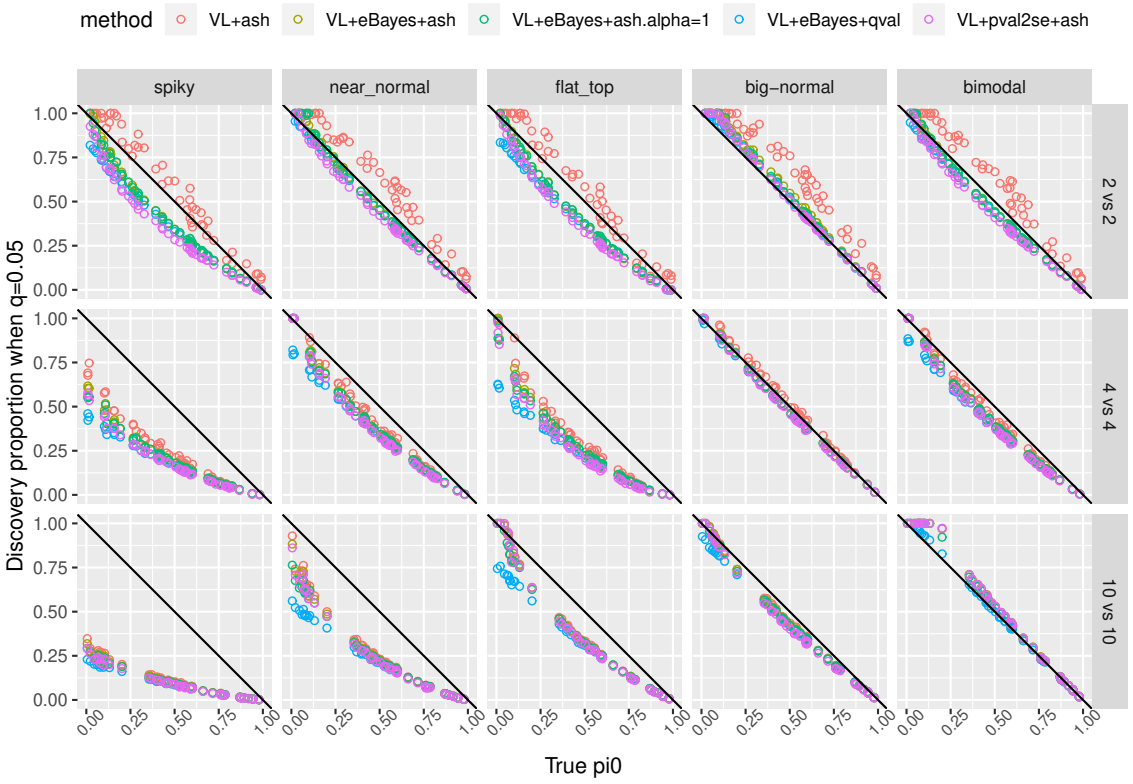

Figure 3.5: Comparison of proportion of discoveries on simulations with independent genes if declaring genes with q-values under 0.05 as positives. Typically *VL+eBayes+ash* and *VL+eBayes+ash.alpha=1* have notably more discoveries compared to *VL+eBayes+qvalue*, while still keeping the actual FDR under control as we showed in Figure 3.4.

β, while *VL+eBayes* uses a normal-based model on log-transformed data. Between the two, our simulations show that *VL+eBayes* is better at estimating the raw effect sizes.

Figure 3.6 also shows that *DESeq2* and *VL+eBayes+ash* have significantly lower RMSE's than that of *VL+eBayes*. Both *DESeq2* and *VL+eBayes+ash* further shrink $\hat{\beta}$ by pooling information across genes. According to our simulation results, the shrunk estimates are indeed closer to the truth compared to the unshrunk estimates. *DESeq2* can decrease the baseline RMSE by up to 40% in "spiky", "near normal" and "flat top" scenarios, while *VL+eBayes+ash* can decrease the baseline RMSE by up to 90% in all scenarios. In particular, when the null proportion is large, the shrunk estimates are much more accurate than the non-shrunk estimates, since the Empirical Bayes prior puts more shrinkage on the $\hat{\beta}$'s. Our method *VL+eBayes+ash* superior to the other methods in terms of estimating the effect sizes, for all scenarios.

Apart from the point estimate of β, the uncertainty of the estimates may also be of interest. *DESeq2* uses the MAP estimates to estimate β, so deriving their standard errors and confidence intervals is non-trivial and requires asymptotic approximations. On the contrary, our method *VL+eBayes+ash* uses the posterior mean to estimate β, and so the posterior distribution naturally captures its uncertainty. Table 3.2 reports the coverage rates of 95% credible intervals for the posterior mean estimate $\tilde{\beta}$. The coverage rates are generally satisfactory. The "spiky" scenarios has coverage rates slightly below 95%, since π_0 is often over-estimated because of the null-biased penalty option in `ashr` package. This pattern is consistent with the results in Stephens (2016).

Figure 3.6: Comparison of RRMSE (relative root mean squared error) of effect estimates on simulations with independent genes. We choose *VL* as the baseline level, and divide the RRMSE's of the other methods by that of *VL*. *VL+eBayes+ash* significantly reduces the MSE and gives much more accurate effect estimates in all scenarios, especially when π_0 is close 1. *VL+pval2se+ash* performs similar to *VL+eBayes+ash* when $N = 10$ for scenarios other than big-normal, but seems not that satisfying for small sample and small π_0 cases.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.80	0.93	0.96	0.91	0.93
N=4	0.93	0.96	0.96	0.96	0.95
N=10	0.97	0.97	0.96	0.96	0.95

(a) All observations. Coverage rates are generally satisfactory, except for the big-normal scenario case when N=2.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.23	0.75	0.94	0.65	0.74
N=4	0.76	0.95	0.94	0.95	0.95
N=10	0.95	0.94	0.94	0.95	0.94

(b) "Significant" negative discoveries. Coverage rates are generally satisfactory when $N = 10$ and $N = 4$ (except for big-normal scenario), but are mostly not good when $N = 2$. These results might due to inaccurate estimates of the tails of g in small sample size cases. The uniform prior sometimes substantially underestimate the length of the tail of true g.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.98	0.96	0.96	0.96	0.96
N=4	0.96	0.96	0.96	0.96	0.96
N=10	0.95	0.96	0.95	0.96	0.96

(c) "Significant" positive discoveries. Coverage rates are generally satisfactory.

Table 3.2: Table of empirical coverage for nominal 95% lower credible bounds on simulations with independent genes.

3.3.3 Simulate RNA-Seq data with unwanted variation

We also simulate RNA-Seq datasets which preserve the unwanted variation and dependence structures often present in real RNA-Seq data. We use the same simulation scheme as discussed in Section 3.3.2 (see Table 3.1), but do not permute the samples for each gene. The 10,000 top expressed genes are selected for use. For each scenario, we simulate 50 datasets with $N = 2, 4, 10$ respectively, where the π_0's are uniformly drawn from [0,1].

We perform differential expression analysis on these simulated datasets that mimic the real RNA-Seq data. We also compare the results of methods *DESeq2*, *edgeR*, *VL+eBayes* and *VL+eBayes+ash* from the following aspects.

FDR calibration Figure 3.8 shows the estimated null proportion of the methods *DESeq2*, *edgeR*, *VL+eBayes* and *VL+eBayes+ash*. The p-value based methods use the `qvalue` R package to calculate q-values, and *VL+eBayes+ash* uses the q-values provided by `ashr` package. Compared to the results for simulations with independent genes (see Section 3.3.2), all methods now give less conservative π_0 estimates, especially when sample cases are extremely small ($N = 2$). Our method *VL+eBayes+ash* has some anti-conservative π_0 estimates on occasion when the true π_0 goes up.

Figure 3.9 shows the actual FDR if we declare all genes with q-values under 0.05 as significant. All four methods fail to calibrate the FDR, and sometimes produce incredibly high FDRs (over 50%). Note that even though *VL+eBayes+ash* gives much more anti-conservative π_0 estimates than the other three methods, its FDR

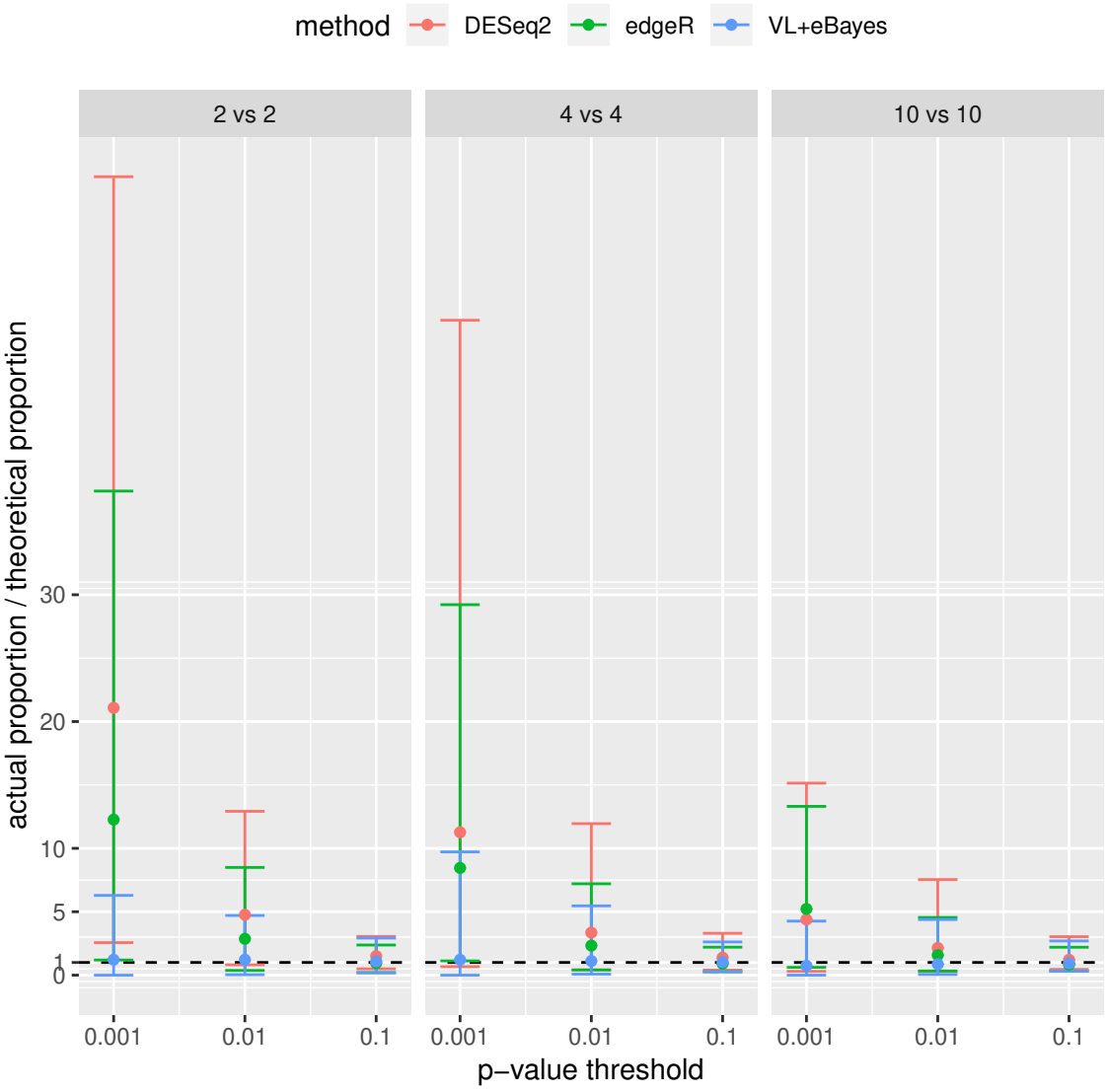

Figure 3.7: Comparison of proportion of genes with p-values under a threshold (0.01, 0.05, 0.1) on null simulations with unwanted variation. All three methods are not guaranteed to control the proportion under its expected value (the threshold), but *VL+eBayes* typically gives better calibrated p-values compared to *DESeq2* and *edgeR*.

control are actually no worse than those of *DESeq2* and *edgeR*. The small sample size leads to low degree of freedom in the t-likelihood of *ash*, and hence the posterior local false discovery rate is naturally constrained to be extremely small.

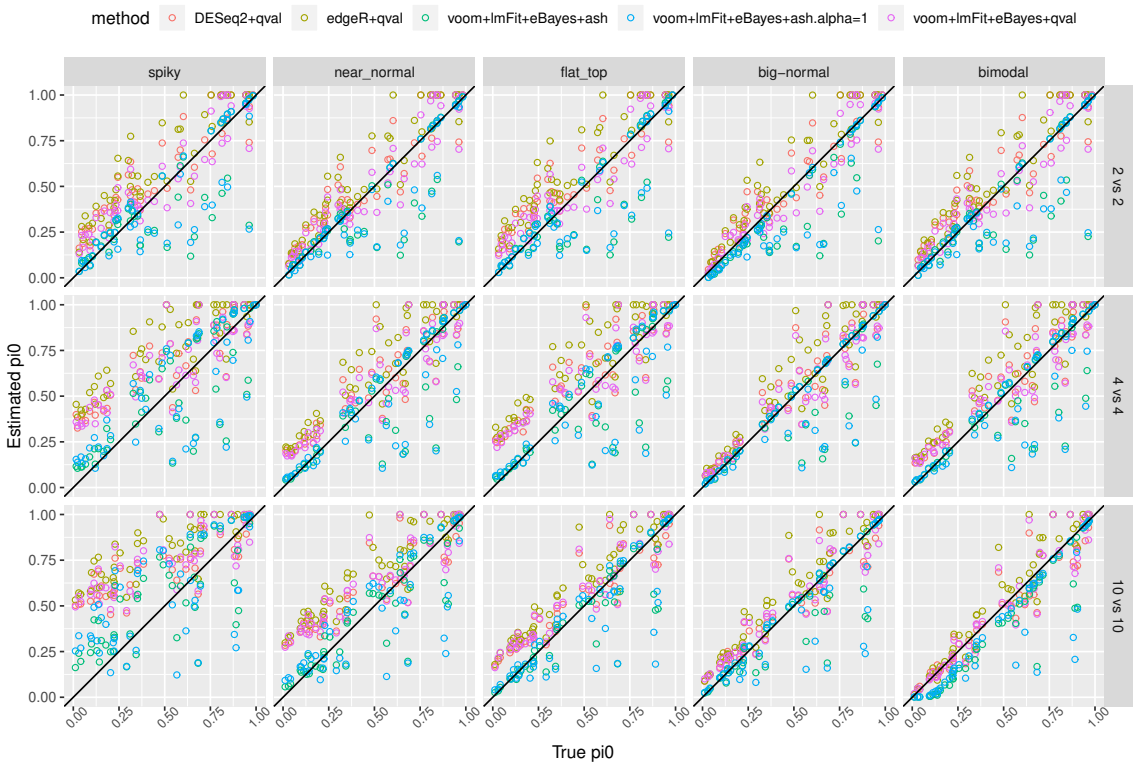

Figure 3.8: Comparison of true and estimated values of π_0 on simulations with unwanted variation.

These results reveal a worrying phenomenon in differential expression studies on

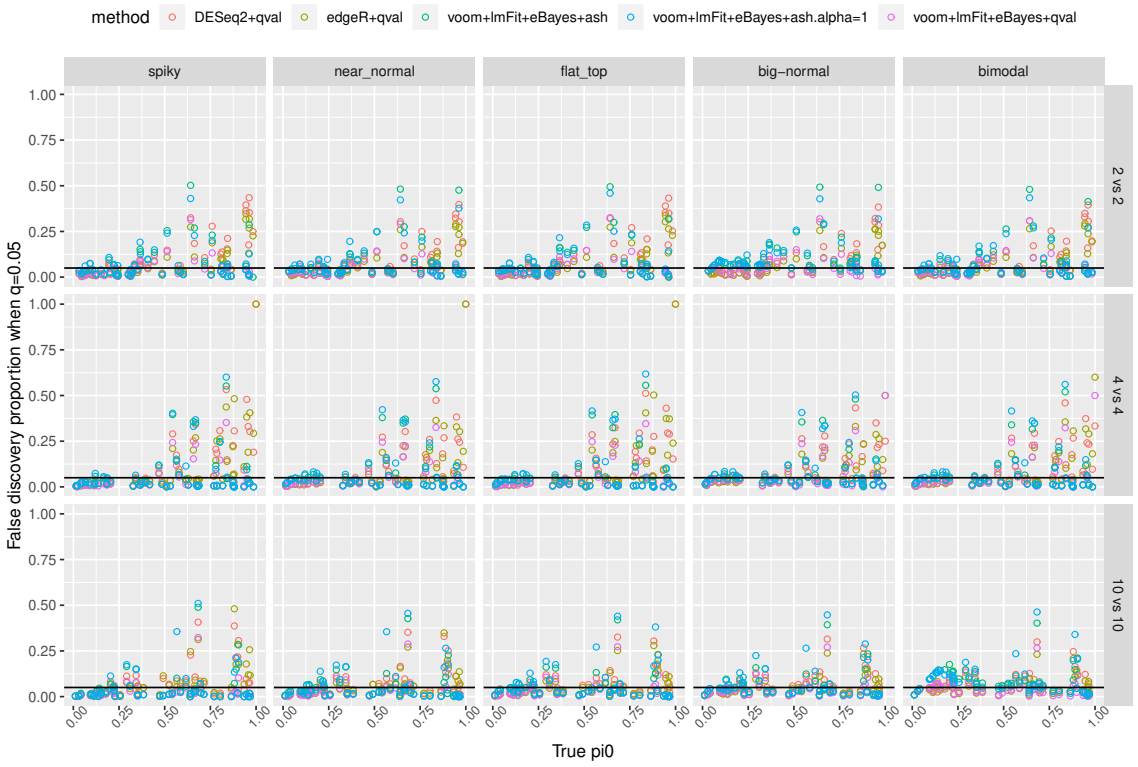

Figure 3.9: Comparison of actual false discovery proportions on simulations with unwanted variation if declaring genes with q-values under 0.05 as positives.

real RNA-Seq data: in presence of unwanted variation or correlation structures, none of the popular DE methods are conservative, and most "significant" genes are likely false discoveries. Here we attempt to use the "empirical null" like methods discussed in Section 3.2.5 to remedy the situation. First we apply the method assuming no control genes are provided (method denote as *VL+eBayes+ash+inflate*), and then apply the method which uses 100 true null genes as "control genes" (method denote as *VL+eBayes+ash+inflate.ctl*). Many RNA-Seq experiments supply about 100 spike-ins, so having 100 control genes is a reasonable assumption.

Figure 3.10 and 3.11 compare the estimated null proportion and actual FDR of the methods *VL+eBayes+ash*, *VL+eBayes+ash+inflate* and *VL+eBayes+ash+inflate.ctl*. Although *VL+eBayes+ash+inflate* do not noticeably improve FDR calibration, *VL+eBayes+ash+inflate.ctl* does indeed produce much better π_0 estimates and reasonable FDRs. Hence, even 100 control genes provide extensive information about the unwanted variation among data, and allow us to greatly improve the calibration of actual FDR in small sample size cases.

Effect estimates Figure 3.12 shows the relative RMSEs of the methods *DESeq2*, *edgeR*, *VL+eBayes* and *VL+eBayes+ash*. Similar to Section 3.3.2, we find that the non-shrunk estimators often (*VL+eBayes* and *edgeR*) have noticeably higher RMSEs than that of the shrunk methods (*VL+eBayes+ash* and *DESeq2*). However, with presence of unwanted variation,*DESeq2* presents even higher RMSEs than *VL+eBayes*, but *VL+eBayes+ash* still grants the lowest RMSEs in almost all scenarios (except for the "big-normal" scenario when $N = 2$ and true π_0 is small).

We also inspect the coverage rates of *VL+eBayes+ash*'s posterior mean estimator.

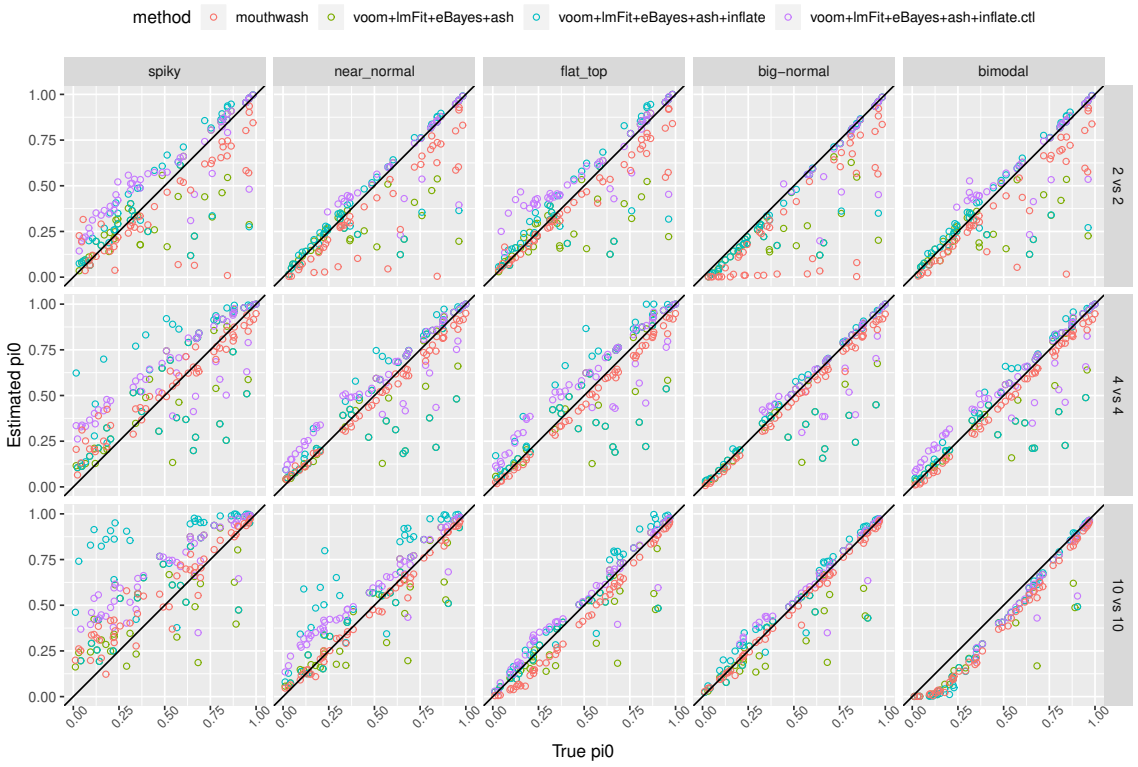

Figure 3.10: Comparison of true and estimated values of π_0 on simulations with unwanted variation. The method *VL+eBayes+ash+inflate.null* uses 100 true null genes as "control genes" to estimate λ_1, λ_2.

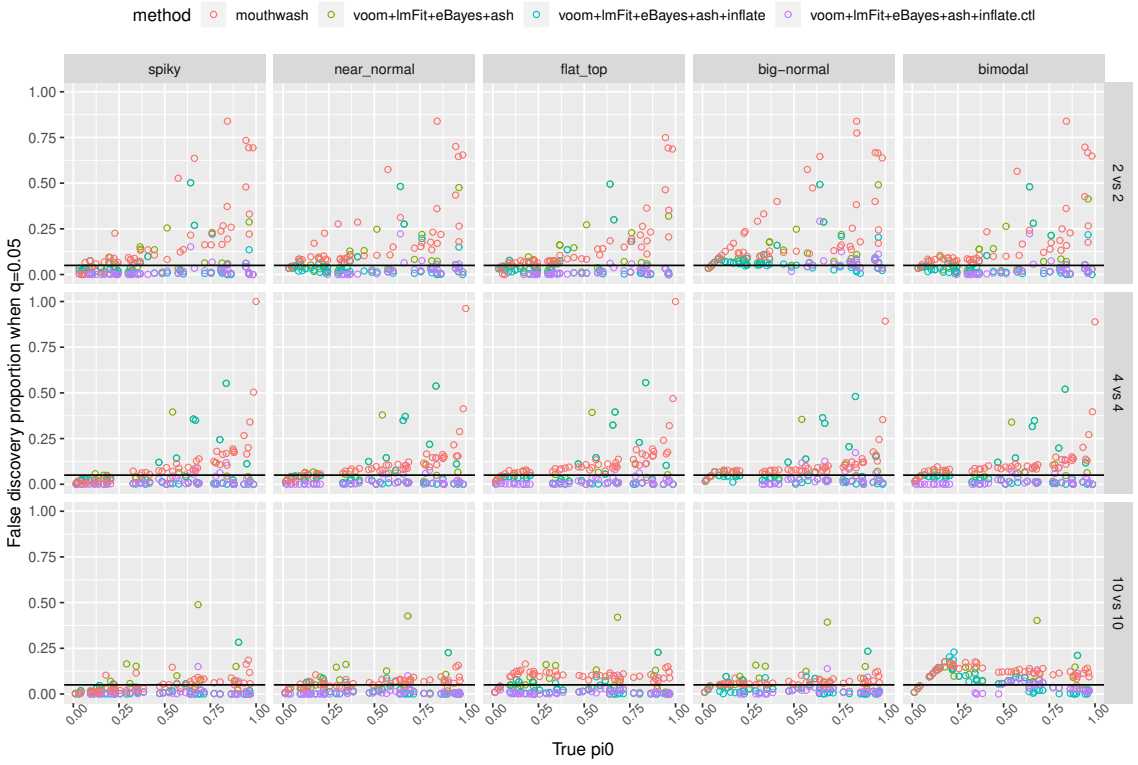

Figure 3.11: Comparison of actual false discovery proportions on simulations with unwanted variation if declaring genes with q-values under 0.05 as positives. The method *VL+eBayes+ash+inflate.null* uses 100 true null genes as "control genes" to estimate λ_1, λ_2.

Table 3.3 shows that the coverage rates are still generally satisfactory, except for the "spiky" scenario. When $N = 2$, the coverage rate of significant negative discoveries in "flat top" scenario and that of significant positive discoveries in "near normal" scenario is slightly smaller than 0.95. These results might due to inaccurate estimates of the tails of g in small sample size cases. Since we use a mixture of uniform distributions to estimate the prior distribution, the length of the tail of true g might be underestimated in some cases.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.80	0.93	0.96	0.91	0.93
N=4	0.94	0.97	0.96	0.97	0.96
N=10	0.97	0.97	0.96	0.96	0.95

(a) All observations. Coverage rates are generally satisfactory, except for the big-normal scenario case when N=2.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.24	0.76	0.94	0.66	0.76
N=4	0.76	0.95	0.94	0.95	0.95
N=10	0.95	0.94	0.94	0.95	0.95

(b) "Significant" negative discoveries. Coverage rates are generally satisfactory when $N = 10$ and $N = 4$ (except for big-normal scenario), but are mostly not good when $N = 2$. These results might due to inaccurate estimates of the tails of g in small sample size cases. The uniform prior sometimes substantially underestimate the length of the tail of true g.

	big-normal	bimodal	flat-top	near-normal	spiky
N=2	0.98	0.96	0.96	0.96	0.95
N=4	0.96	0.96	0.95	0.96	0.96
N=10	0.96	0.96	0.95	0.95	0.94

(c) "Significant" positive discoveries. Coverage rates are generally satisfactory.

Table 3.3: Table of empirical coverage for nominal 95% lower credible bounds on simulations with unwanted variation.

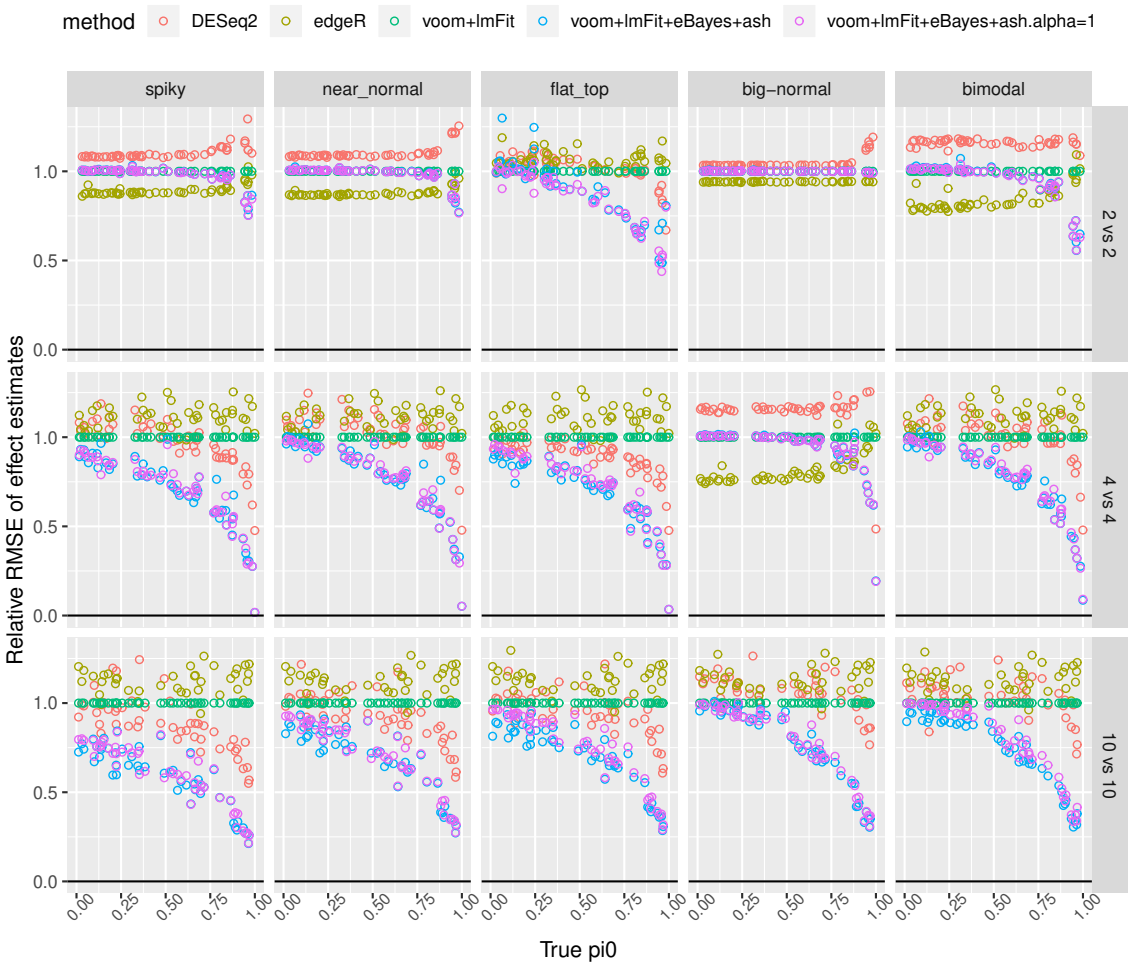

Figure 3.12: Comparison of RRMSE (relative root mean squared error) of effect estimates on simulations with unwanted variation. We choose *voom+limma* as the baseline level, and divide the RRMSE's of the other methods by that of *voom+limma*. *VL+eBayes+ash* still significantly reduces the MSE and gives much more accurate effect estimates in all scenarios, especially when π_0 is close 1.

3.4 Discussion

In this work, we propose a new pipeline *VL+eBayes+ash* for differential expression analysis on RNA-Seq data. Our method aims to combine the advantages of count transformation, variance shrinkage and effect size shrinkage. We simulate RNA-Seq data with and without dependence structures, and show that *VL+eBayes+ash* has the following improvements over the existing widely-used methods *DESeq2*, *edgeR* and *VL+eBayes*.

Higher statistical power. The count-based methods *DESeq2* and *edgeR* typically yields lower AUC than the log-transformation based method *VL+eBayes*. On the other hand, our method *VL+eBayes+ash* further improves *VL+eBayes*'s ranking of genes by incorporating *ash* into the pipeline. The *ash* step alternates the ranking of genes by incorporating standard errors into the likelihood. and thus avoids the problem with p-value based methods, whereby low precision measurements can inflate the estimated FDR.

Better FDR calibration. Even in the simulations with independent genes, *DESeq2* and *edgeR* can occasionally result in incredibly high FDR. The methods *VL+eBayes* and *VL+eBayes+ash* perform much better in controlling the actual FDR when there is no unwanted variation in data. In the cases where dependence is present RNA-Seq data, all methods can be anti-conservative and thus result in false discoveries. However, we proposed a procedure which utilizes a small amount of control genes to fix the issue of anti-conservativeness in small sample size cases.

More accurate estimates of effect sizes. Apart from the two points mentioned above for DE analysis, our method can be also used for effect size estimation. Our

shrinkage estimator improve upon the effect size estimates by pooling information across genes. *VL+eBayes+ash* achieves much higher estimation accuracy than that of *DESeq2*, which also uses a shrinkage estimator. Furthermore, *VL+eBayes+ash* naturally provides the distributions of the shrunk estimates, which allow us to easily obtain standard errors and credible intervals for the estimated effect sizes.

In summary, our proposed pipeline *VL+eBayes+ash* is statistically powerful and hence results in more true discoveries of DE genes than *DESeq2*, *edgeR* and *VL+eBayes*, while controlling the FDR at the same time. Our pipeline *VL+eBayes+ash* is also computationally efficient. A typical run of *VL+eBayes+ash* takes only seconds to complete on a dataset with 10,000 genes and less than 100 samples, which is significantly faster than the count model based methods mentioned here.

Chapter 4

GENERAL ADAPTIVE SHRINKAGE

4.1 Introduction

From the previous sections, we see that the adaptive shrinkage (*ash*) methods can be extended to deal with data from various distributions and hence used in gene expression analysis. The normal *ash* started from observations with normal likelihood. For variance shrinkage problems, we developed models for gamma (chi-squared) distributed observed variances. In RNA-Seq applications, the student t likelihood was used in *ash* to deal with the small sample size issues. The key idea of *ash* and the above extensions is the use of unimodal prior assumption (UA), which is highly adaptive to data and sensible in many contexts (not limited to genomics studies). Hence, apart from data with normal, gamma and t likelihood, it is natural for us to explore potential adaptive shrinkage approaches for generic data.

In practice, we typically use a finite mixture of uniforms to approximate any unimodal prior. Fortunately, the convolution between a unimodal distribution and general likelihood is generally straightforward using existing software. Here we exploit this to develop a general ash framework that can be applied to many commonly encountered likelihoods (binomial, Poisson, etc.).

The R code and analysis results are available from `http://github.com/mengyin/general_ash`.

4.2 Methods

4.2.1 Models

Suppose we observe Y_j $(j = 1, ..., J)$, which is a random variable with likelihood $\phi_j(\theta_j) := p(Y_j|\theta_j)$, and the parameter of our interest is θ_j. Our goal is to make inference (hypothesis testing, estimation) on θ_j. Under the *ash* (Stephens, 2016) framework, we use a Bayesian model to borrow information across the observations and use the posterior distribution to estimate or test θ_j.

We assume that after some transformation $h(\cdot)$, the true parameters θ_j come from a common unimodal prior $g(\cdot)$:

$$h(\theta_j) \sim g(\cdot), \tag{4.1}$$

where $h(\cdot)$ is the link function. We assume h is a strictly monotone increasing function.

4.2.2 Estimate prior distribution g

As in Stephens (2016), we use a mixture of uniform distributions to approximate g:

$$g \sim \sum_{k=1}^{K} Z_k U[a_k, b_k], \tag{4.2}$$

where $(Z_1, ..., Z_K)$ are latent binary indicators for mixture components following multinomial distribution with $n = 1$ and $P(Z_k = 1) = \pi_k$, $U[a_k, b_k]$ denotes a

uniform random variable on $[a_k, b_k]$. Given a fixed grid of a_k and b_k, we use the empirical Bayes method to estimate the mixture proportion π. To do this we first compute the matrix $L = (L_{jk})$ where each entry L_{jk} is the likelihood of θ_j for the k'th prior component:

$$\begin{aligned} L_{jk} &:= p(Y_j|Z_{jk} = 1) && (4.3)\\ &= \int p(Y_j|\theta_j)p(\theta_j|Z_{jk} = 1)d\theta_j && (4.4)\\ &= \frac{1}{b_k - a_k}\int_{h^{-1}(a_k)}^{h^{-1}(b_k)} \phi_j(\theta_j)|h'(\theta_j)|d\theta_j, && (4.5) \end{aligned}$$

where h' is the derivative of h. If the mixture component is a point mass, i.e. $a_k = b_k$, the likelihood is simply given by the probability density $L_{jk} = \phi_j(h^{-1}(a_k))|h'(h^{-1}(a_k))|$.

Then the mixture proportions π are estimated by maximizing the log-likelihood:

$$l(\pi) = \sum_j \log\left(\sum_k \pi_k L_{jk}\right), \tag{4.6}$$

$$\hat{\pi} = \arg\max_{\pi} l(\pi). \tag{4.7}$$

This can be done using the same methods as in the normal case (Stephens, 2016).

The integral of $\phi_j(\theta_j)|h'(\theta_j)|$ in (4.5) does not necessarily have analytical form. However, if the link function is identity link $h(x) = x$, we have

$$L_{jk} = \frac{\Phi_j(b_k) - \Phi_j(a_k)}{b_k - a_k}, \tag{4.8}$$

where $\Phi_j(x) := \int_{-\infty}^{x} \phi_j(y)dy$.

Now we define ψ_j as the density of the distribution proportional to $\phi_j(x)|h'(x)|$:

$$\psi_j(x) := \frac{\phi_j(x)|h'(x)|}{\int_{-\infty}^{\infty} \phi_j(x)|h'(x)|dx}, \tag{4.9}$$

and $\Psi(x) := \int_{-\infty}^{x} \psi_j(y)dy$ is the corresponding cdf.

If $\phi_j(\cdot;\theta)$ belongs to the exponential family and h is its natural link, $\psi_j(x)$ would be a distribution in the conjugate distribution family of ϕ_j, and its cdf can be used to compute the integral in (4.5). Some examples are illustrated in Section 4.2.5.

In practice, $\psi_j(\cdot)$ and $\Psi_j(\cdot)$ should be provided to compute the likelihood matrix L in order to fit the prior distribution. Otherwise, we can use numerical integral to calculate (4.5), but the computational stability might not be guaranteed.

4.2.3 Posterior distribution $p(\theta_j|Y_j, \hat{\pi})$

For any distribution (density) $f(x)$, we denote $f^{\text{trunc}}(x; a, b)$ as its truncated distribution on interval $[a, b]$:

$$f^{\text{trunc}}(x; a, b) := \frac{f(x)}{\int_a^b f(y)dy}, \tag{4.10}$$

and denote $M^f(a, b)$ as the mean of $f^{\text{trunc}}(x; a, b)$:

$$M^f(a, b) := \int_{-\infty}^{\infty} x f^{\text{trunc}}(x; a, b)dx. \tag{4.11}$$

For the corner case $a = b$, $f^{\text{trunc}}(x; a, a) := \delta_a$ which is the point mass on a, and $M^f(a, a) = a$.

The posterior distribution of θ_j given observation Y_j and fitted prior mixture proportions $\hat{\pi}$ is given by:

$$p(\theta_j|Y_j,\hat{\pi}) = \frac{p(Y_j|\theta_j)p(\theta_j)}{\int p(Y_j|\theta_j)p(\theta_j)d\theta_j} \tag{4.12}$$

$$= \frac{\phi_j(\theta_j)g(h(\theta_j))|h'(\theta_j)|}{\sum_k \hat{\pi}_k L_{jk}} \tag{4.13}$$

$$= \sum_k \tilde{\pi}_{jk}\psi_j^{\text{trunc}}(\theta_j;\tilde{a}_k,\tilde{b}_k), \tag{4.14}$$

where:

$$\psi_j(x) = \frac{\phi_j(x)|h'(x)|}{\int_{-\infty}^{\infty}\phi_j(x)|h'(x)|dx}, \tag{4.15}$$

$$\tilde{\pi}_{jk} = \frac{\hat{\pi}_k L_{jk}}{\sum_{k'}\hat{\pi}_{k'}L_{jk'}}, \tag{4.16}$$

$$\tilde{a}_k = h^{-1}(a_k), \tag{4.17}$$

$$\tilde{b}_k = h^{-1}(b_k). \tag{4.18}$$

In other words, the posterior distribution of θ_j is a mixture of truncated ψ_j distribution, truncated on $(\tilde{a}_k,\tilde{b}_k)$, with mixture proportions $\tilde{\pi}_{jk}$:

$$\theta_j|Y_j,\hat{\pi} \sim \sum_{k=1}^{K}\tilde{Z}_{jk}\psi_j^{\text{trunc}}(\theta_j;\tilde{a}_k,\tilde{b}_k), \tag{4.19}$$

where $(\tilde{Z}_{j1},...,\tilde{Z}_{jK})$ are latent binary indicators for mixture components, following multinomial distribution with $n=1$ and probability $P(\tilde{Z}_{jk}=1)=\tilde{\pi}_{jk}$.

Following the posterior distribution, we can calculate other quantities to estimate

or test θ_j:

- Posterior mean:

$$\mathrm{E}(\theta_j|Y_j,\hat{\pi}) = \sum_k \tilde{\pi}_k M^{\psi_j}(\tilde{a}_k,\tilde{b}_k), \tag{4.20}$$

  which can be used as a shrinkage estimator for θ_j.

- Local false discovery rate (lfdr): if the prior includes a mixture component corresponding to the null hypothesis $\theta_j = 0$, i.e. $h^{-1}(a_k) = h^{-1}(b_k) = 0$, then lfdr for θ_j is given by

$$\mathrm{lfdr}_j = P(\theta_j = 0|Y_j,\hat{\pi}), \tag{4.21}$$

  which is the posterior mixture proportion for that null component.

- Local false sign rate (lfsr) as defined in (Stephens, 2016):

$$\mathrm{lfsr}_j = P(\theta_j = 0|Y_j,\hat{\pi}) + \min(P(\theta_j > 0|Y_j,\hat{\pi}), P(\theta_j < 0|Y_j,\hat{\pi})), \tag{4.22}$$

  where

$$P(\theta_j < 0|Y_j,\hat{\pi}) = \sum_k \frac{\tilde{\pi}_k \Psi_j(0)}{\Psi_j(\tilde{b}_k) - \Psi_j(\tilde{a}_k)}, \tag{4.23}$$

  and $P(\theta_j > 0|Y_j,\hat{\pi}) = 1 - P(\theta_j = 0|Y_j,\hat{\pi}) - P(\theta_j < 0|Y_j,\hat{\pi})$.

4.2.4 Estimate unknown mode

In previous sections we assume that for the unimodal prior g, the uniform mixture components $\{a_k, b_k\}$ are fixed. However in some cases, the mode is unknown and we

would like to estimate the prior using the empirical Bayes method:

$$\hat{g} = \arg \max_{g \text{ unimodal}} l(g), \tag{4.24}$$

hence $\hat{g}$ optimizes the log-likelihood.

In practice, we solve this optimization problem as follows: for each given mode c, we construct a grid $\{a_k, b_k\}$ which is anchored at mode c and covers a sufficient wide range, and estimate the mixture proportions $\hat{\pi}$ which achieves the maximum log-likelihood (denote by l_c). Thereby, l_c itself is a function of the mode c. We use the numerical optimization function `stats::optimize` in R to search for the optimizer $\hat{c} = \arg\max_c l_c$.

4.2.5 Special cases

Table 4.1 lists some special cases of general *ash*, where the likelihood ϕ_j is a commonly used distribution. We will discuss *fash*, Poisson *ash* and Binomial *ash* in detail in Section 4.3. Here we define the non-standard log-F distribution $\log F(\cdot; \mu, \nu_1, \nu_2)$ as follows: if for a random variable X, we have $\exp(X - \mu) \sim F(\nu_1, \nu_2)$, then we say X follows the distribution $\log F(X; \mu, \nu_1, \nu_2)$.

4.3 Applications

*4.3.1 Adaptive shrinkage of F statistics (*fash*)*

A special case for the *ash* methods with general likelihood would be the adaptive shrinkage of F statistics. F statistics are normally used for testing equality of two variances, or multiple-comparison ANOVA problems. In genomic contexts, pooling information across genes may help improve the statistical power of gene-specific F tests. Smyth (2004) suggested using the moderated error variance estimates to adjust F statistics, assuming that the gene-specific variances come from a common inverse-gamma prior. Nevertheless, we can directly work on the gene-specific F-statistics and fit their prior more adaptively by a unimodal distribution.

Suppose we have the expression matrix Y for G genes and N samples from $M(>=2)$ conditions. Consider the following two problems related to F test: variability comparison and variance decompositions.

Table 4.1: Special cases of general *ash*

	Model		Posterior		
Case	ϕ_j	$h(x)$	ψ_j	$\tilde{a}_k$	$\tilde{b}_k$
ash	$N(Y_j;\theta_j,s_j^2)$	x	$N(\theta_j;Y_j,s_j^2)$	a_k	b_k
fash	$\log F(Y_j;\theta_j,\nu_1,\nu_2)$	x	$\log F(\theta_j;Y_j,\nu_2,\nu_1)$	a_k	b_k
Poisson *ash*	$\text{Poisson}(Y_j;c_j\theta_j)$	x	$\text{Gamma}(\theta_j;Y_j+1,c_j)$	a_k	b_k
Poisson *ash*	$\text{Poisson}(Y_j;c_j\theta_j)$	$\log(x)$	$\text{Gamma}(\theta_j;Y_j,c_j)$	e^{a_k}	e^{b_k}
Binomial *ash*	$\text{Bin}(Y_j;n_j,\theta_j)$	x	$\text{Beta}(\theta_j;Y_j+1,n_j-Y_j+1)$	a_k	b_k
Binomial *ash*	$\text{Bin}(Y_j;n_j,\theta_j)$	$\text{logit}(x)$	$\text{Beta}(\theta_j;Y_j,n_j-Y_j)$	$\frac{1}{1+e^{-a_k}}$	$\frac{1}{1+e^{-b_k}}$

Variability comparison Suppose we would like to compare the expression variability within condition A and the variability within condition B. The statistical model is defined by:

$$Y_{gi} = \mu_g + \beta_{g,c(i)} + e_{gi}, \tag{4.25}$$

$$e_{gi} \sim N(0, \sigma^2_{g,c(i)}), \tag{4.26}$$

where g is the index for gene, i is the index for samples and $c(i)$ is the condition indicator, either A or B. Suppose there are N_A and N_B samples in group A and B respectively. All observations are independent with each other.

A straightforward way to estimate the true variance ratio $\frac{\sigma^2_{gA}}{\sigma^2_{gB}}$ (denoted by α_g) is using the ratio of sample variances $\frac{\hat{\sigma}^2_{gA}}{\hat{\sigma}^2_{gB}}$ (denoted by F_g). Its sampling distribution is given by

$$F_g \sim \alpha_g \times F, \tag{4.27}$$

where F is a F-distributed random variable with degrees of freedom $N_A - 1$ and $N_B - 1$. Let the null hypothesis be: the two conditions have same expression variability ($H_0 : \sigma_{gA} = \sigma_{gB}$), then under the null $\alpha_g = 1$.

Transforming (4.27), we have

$$\log(F_g) - \log(\alpha_g) | \log(\alpha_g) \sim \log F, \tag{4.28}$$

where $\log F$ is the logarithm of F-distributed random variable with d.f. $N_A - 1$ and $N_B - 1$. Note that (4.28) meets the form of general *ash* problem, where $\log(\alpha_g)$ is

our parameters of interest with log-F likelihood.

Analogous to (4.1), assuming $\log(\alpha_g)$ come from a common unimodal prior, the general *ash* framework can be further used to improve estimates of $\log(\alpha_g)$. According to Table 4.1, the posterior distribution of $\log(\alpha_g)$ is given by a mixture of truncated $\log F(\cdot; \log(F_g), N_B - 1, N_A - 1)$ distribution (with different truncation limits for different mixture components). By pooling information across genes, the posterior estimates of $\log(\alpha_g)$ are presumably more accurate than the raw noisy estimates $\log(F_g)$.

Variance decomposition Suppose we would like to compare the expression variability explained by conditions to the variability due to noise (or the total variability). The statistical model is defined by:

$$Y_{gi} = \mu_g + \beta_{g,c(i)} + e_{gi}, \tag{4.29}$$

$$e_{gi} \sim N(0, \sigma^2_{ge}), \tag{4.30}$$

$$\beta_{g,c(i)} \sim N(0, \sigma^2_{gc}), \tag{4.31}$$

where $c(i)$ is the condition level of sample i, $\beta_{g,c(i)}$ is the random condition effect. Suppose the design is balanced so we have equal number of samples for each condition (M conditions in total).

Our goal is to compare σ^2_{gc} (variance of condition effect) against σ^2_{ge} (variance among replicates). The ANOVA F-statistic F_g can be used for variance decomposi-

tion, and its sampling distribution is given by

$$F_g = \frac{\text{SST}_g/(M-1)}{\text{SSE}_g/(N-M)} \tag{4.32}$$

$$= \frac{(\sum_i (\bar{Y}_{g,c(i)} - \bar{Y}_g)^2)/(M-1)}{(\sum_i (Y_{g,c(i)} - \bar{Y}_{g,c(i)})^2)/(N-M)} \tag{4.33}$$

$$\sim (1 + M\sigma^2_{gc}/\sigma^2_{ge}) \times F \tag{4.34}$$

$$\sim \alpha_g \times F \quad (\alpha_g := 1 + M\sigma^2_{gc}/\sigma^2_{ge}) \tag{4.35}$$

where F is a F-distributed random variable with d.f $M-1$ and $N-M$, $\bar{Y}_{g,c(i)}$ is the condition $c(i)$'s expression mean (average of all $Y_{g,c(i)}$'s with condition $c(i)$), and $\bar{Y}_g$ is the overall expression mean for gene g. Let the null hypothesis be: there are no condition effects ($H_0 : \sigma_{gc} = 0$), then under the null $\alpha_g = 1$.

Similarly, we can use the general *ash* framework to fit a unimodal prior for $\log(\alpha_g)$ and use posterior means to estimate $\log(\alpha_g)$. Then the ratio of condition variance and error variance $\sigma^2_{gc}/\sigma^2_{ge}$ can be estimated by transforming the estimate of $\log(\alpha_g)$. Table 4.1 shows the analytical form of the posterior of $\log(\alpha_g)$: a mixture of truncated $\log F(\cdot; \log(F_g), M-1, N-M)$ distribution (with different truncation limits for different mixture components).

Note that this method can only apply to balanced dataset with equal number of samples for each condition, since (4.34) does not hold for unbalanced dataset.

Example: variance decomposition for stem cell expression data We have the microarray gene expression data from Burrows et al. (2016). The dataset has four individuals. Each individual has four samples types - Fibroblast, LCL, F-iPSC,

L-iPSC, where L-iPSC refers to iPSCs derived from LCLs, F-iPSC refers to iPSCs derived from Fibroblasts. The L-iPSC type has three replicates A, B and C, and the other three types only have one replicate, so there are 6 samples for each individual.

Burrows et al. (2016) were interested in the proportion of expression variance explained by cell type of origin versus that explained by individual in the iPSCs. They performed a linear mixed model with a fixed effect for cell type of origin (i.e. L-iPSC vs F-iPSC) and a random effect for individual. This model did no use the LCLs or the Fibroblasts from these individuals.

We use the naive ANOVA F-test and *fash* to analyze the proportion of variation explained by cell-type or individual. Specifically, we assume that gene expression y_{gij} comes from the following model:

$$y_{gij} = \mu_g + \beta_{gi} + \gamma_{gj} + e_{gij}, \tag{4.36}$$

where g, i, j are the indices for gene, individual and cell type respectively. β and γ are random effects for individuals and cell-types respectively. Suppose $\beta_{gi} \sim N(0, v_g^{(\text{ind})})$, $\gamma_{gj} \sim N(0, v_g^{(\text{ct})})$ and $e_{gij} \sim N(0, v_g^{(\text{err})})$, we are interested in estimating the "PVE" (proportion of variance explained) by individual or cell-type defined as follows:

$$\text{PVE}_g^{(\text{ind})} := \frac{v_g^{(\text{ind})}}{v_g^{(\text{ind})} + v_g^{(\text{ct})} + v_g^{(\text{err})}}, \tag{4.37}$$

$$\text{PVE}_g^{(\text{ct})} := \frac{v_g^{(\text{ct})}}{v_g^{(\text{ind})} + v_g^{(\text{ct})} + v_g^{(\text{err})}}. \tag{4.38}$$

Note that this dataset has unbalanced design (three L-iPSC replicates but just

one F-iPSC sample for each individual), and it is infeasible to use *fash* on unbalanced dataset for PVE analysis as we discussed before. Hence we choose an ad-hoc way: each time we simply use one of the three L-iPSC replicates to form a balanced dataset, and compare the results of three trials. Fortunately the three trials give very similar results. Figure 4.1 shows the posterior mean of gene-specific PVEs of cell-type or individual estimated by F-test and *fash* for the subsets using each of the L-iPSC replicate.

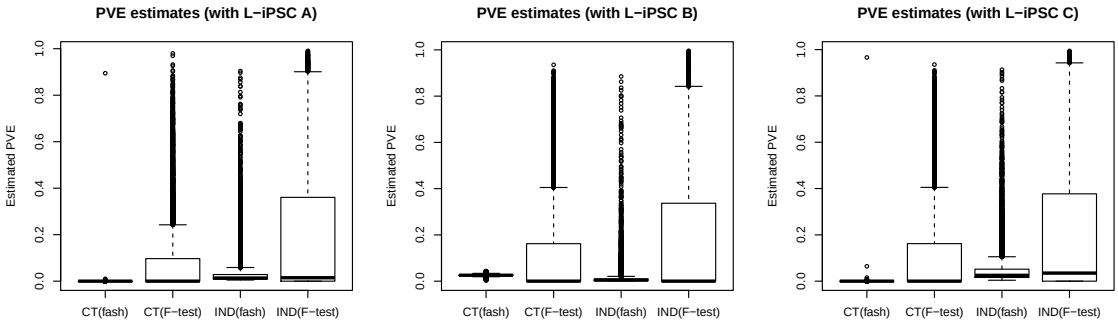

Figure 4.1: Gene-specific PVE estimates of cell-type (CT) or individual (IND), estimated by F-test and *fash* on Burrows data. Each time we only use one of the three L-iPSC replicates to form a balanced dataset.

Burrows et al. (2016) performed the *limma* DE analysis for each pair of cell types and found that $INPP5F$ is the most common DE gene. They also used simple ANOVA R^2 to record the variance explained by cell-types or individuals. The ANOVA R^2 is defined as $R^2 := \text{SST}/(\text{SST}+\text{SSE})$, where SST and SSE are the same as in (4.32). Note that R^2 is different from our defined "PVE" in (4.38). They conclude that "individual genetic background captures a much larger proportion of gene regulatory variation than cell type of origin".

The *fash* results are generally consistent with Burrows et al. (2016): all genes

have almost zero PVE for cell-types, except for gene $INPP5F$ (ENSG00000198825). Compared to the raw F-test PVE estimates (which are substantially noisy), *fash* tends to shrink them towards 0.

4.3.2 Adaptive shrinkage on binomial data (Binomial ash*)*

Table 4.1 gives us the analytical forms of Binomial *ash*, where we have binomial observations $Y_j \sim \text{Binomial}(n_j, p_j)$ $(j = 1, ..., J)$ and n_j's are known. The unknown success probability parameter p_j is of our interest. Binomial *ash* allows us to borrow information across the observations and use the posterior distribution to estimate p_j.

Example: comparison between bulk RNA-Seq and scRNA-Seq data In recent years, single cell RNA-Seq (scRNA-Seq) methods have been more and more frequently used in gene expression analysis. While bulk RNA-Seq data mostly extract gene expression features from millions of cells which have been pooled together, scRNA-Seq can capture expression profile of individual cells. The scRNA-Seq technologies would allow us to fetch more information about the heterogeneity of gene expression across cells. The comparison between bulk RNA-Seq and scRNA-Seq could thus be interesting. If we have both scRNA-Seq data and corresponding bulk RNA-Seq data on the same sample, we might want to quantify the concordance as well as difference between them. Presumably, the difference between bulk RNA-Seq and scRNA-Seq data may rise from various possible sources: effects due to the dynamics of cell transcription, technical differences in sequencing protocols, etc. Hence,

investigating the genes with significant difference might help us better understand the biological mechanism of certain genes as well as the underlying technical features of scRNA-Seq data.

Suppose we have both scRNA-Seq and bulk RNA-Seq data on the same sample. Let X^s_{jg} denote the observed counts of gene g in single cell j. And let X^b_g denote the counts of gene g in the bulk. We first pool the single cell data into a single count, and define $X^s_g := \sum_j X^s_{jg}$. Now we might want to identify the genes that show the most "significant" deviations between X^b_g and X^s_g, and quantify those deviations.

Suppose the bulk and single cell data are independent, then:

$$X^b_g | C_g \sim \text{Binomial}(C_g, p_g), \tag{4.39}$$

where $C_g := X^b_g + X^s_g$ is the total count, p_g is the fraction of all reads that come from bulk at gene g. If the single cell data and bulk data are generally concordant, then the bulk RNA-Seq expression level should be roughly proportional to scRNA-Seq expression level (the ratio relies on sequencing depths). As a result, condition on the total counts C_g, the bulk fraction p_g is supposedly similar across genes. The "outlier" genes where p_g is particularly small or large might be suspicious.

Note that the gene-specific sample bulk fraction $\hat{p}_g := X^b_g / C_g$ is the raw maximum likelihood estimate (MLE) of p_g. We also use the binomial *ash* to estimate p_g, assuming that p_g comes from a unimodal prior with some unknown mode (to be estimated). The posterior mean of p_g (denoted by $\tilde{p}_g$) is thus a shrinkage estimator of p_g, borrowing information across genes. The prior as well as its mode are estimated by the empirical Bayes approach, which makes them adaptive to data.

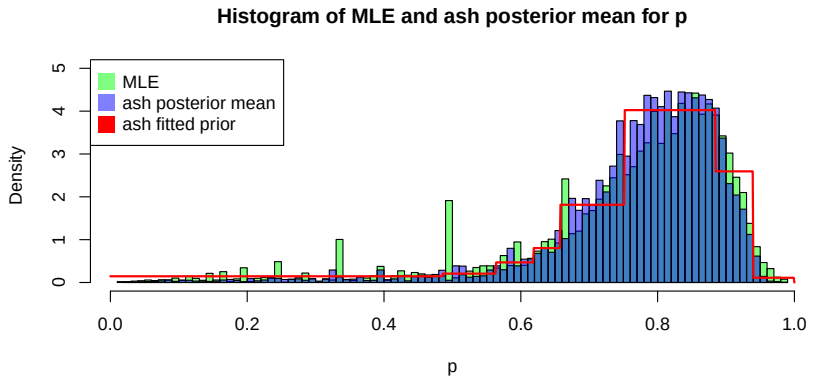

Figure 4.2: Distribution of sample bulk reads fraction $\hat{p}_g = X_g^b/C_g$ and Binomial *ash* posterior estimates on Tung data (NA19091.r1). The red line is the *ash* fitted prior of p_g.

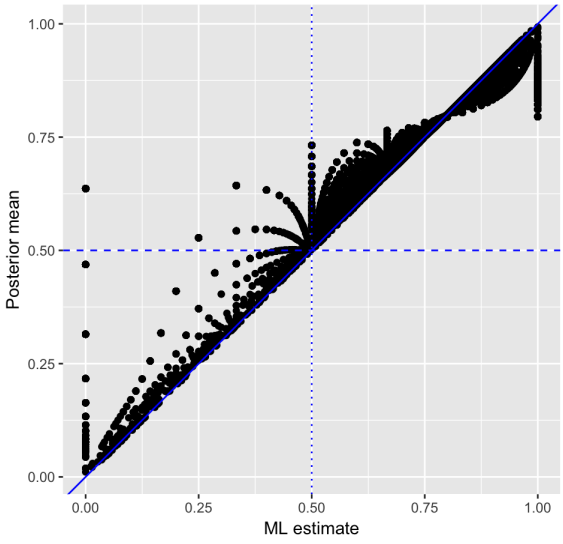

Figure 4.3: Binomial *ash* posterior estimates $\tilde{p}_g$ versus the ML estimates $\hat{p}_g$ on Tung data (NA19091.r1).

Tung et al. (2017) provide both scRNA-Seq and bulk RNA-Seq data for same samples (three individuals and three replicates for each individual). We compare the single cell and bulk data for one replicate NA19091.r1. Genes with both non-zero X_g^b and non-zero X_g^s are selected for our analysis.

Figure 4.2 shows the distribution of sample bulk reads fraction $\hat{p}_g = X_g^b/C_g$ and the binomial *ash* posterior estimates $\tilde{p}_g$. Although most sample fractions $\hat{p}_g$ are over 0.6, there are some extremely small outliers around 0. The binomial *ash* fitted prior is unimodal with mode around 0.8, and the left tail keeps flat from 0 to near 0.5. Figure 4.3 plots the posterior estimates $\tilde{p}_g$ from binomial *ash* versus the sample fraction $\hat{p}_g$. Both figures show that on the left side, quite a few small $\hat{p}_g$'s are pushed higher by binomial *ash*. These genes are further examined and turn out to be low expressed genes, and their bulk reads fractions are highly variable due to the small total count C_g. Thereby, binomial *ash* shrinks these posterior means towards the prior mean, after accounting for the lack of informativeness in low expressed genes.

Table 4.2 lists the genes where the posterior bulk fraction $\tilde{p}_g$ is extremely small or large. We might want to further inspect these genes to investigate the cause of difference between bulk RNA-Seq and scRNA-Seq expression.

Table 4.2: Genes with extremely small or large $\tilde{p}_g$ on Tung data (NA19091.r1).

Gene name	Ensemble ID	X_g^s	X_g^b	$\tilde{p}_g$
TSHZ2	ENSG00000182463	64	1	0.030
HIST1H4L	ENSG00000198558	144	4	0.033
MTRNR2L6	ENSG00000270672	433	16	0.038
BCKDHA	ENSG00000248098	23	1547	0.985
RAB19	ENSG00000146955	3	241	0.978
TUBB3	ENSG00000258947	110	4264	0.975

4.3.3 Adaptive shrinkage on Poisson data (Poisson ash*)*

Table 4.1 provides us the analytical forms of Poisson *ash*, where we have Poisson observations $Y_j \sim \text{Poisson}(c_j\lambda_j)$ $(j = 1, ..., J)$ and c_j's are known scaling factors. The unknown intensity parameter λ_j is of our interest. Poisson *ash* allows us to borrow information across the observations and use the posterior distribution to estimate λ_j.

Example While normal distribution based models have been widely used on classical gene expression data (microarray, bulk RNA-Seq), they have non-negligible limitations when handling single cell RNA-Seq data, which typically have zero inflation and low count level issues. Thereby count distribution based models are generally preferred for scRNA-Seq analysis. In next Chapter, we will discuss the usage of Poisson *ash* on scRNA-Seq data and compare with some existing methods.

Chapter 5

GENE EXPRESSION DISTRIBUTION DECONVOLUTION OF SINGLE CELL RNA-SEQ DATA

5.1 Introduction

In recent years, single cell RNA-Seq (scRNA-Seq) methods have gained substantial popularity in analyzing gene expression data. While more traditional methods like microarray and bulk RNA-Seq technologies mostly depend on estimating the average gene expression level from millions of cells, individual cells often vary in their gene expression levels, which captures the dynamics of cell transcription. Typical scRNA-Seq data contains the expression profile of individual cells, thereby allowing for the quantification of a much richer set of features that can capture the heterogeneity of gene expression across cells. Apart from the mean expression level, measures like dispersion (e.g. coefficient of variation) and nonzero fraction are also useful in various contexts. For instance, dispersion can provide additional information about biological states that are not captured by the population mean alone (Shalek et al., 2014; Klein et al., 2015; Zeisel et al., 2015; Shaffer et al., 2017). Burstiness, on the other hand, can help us better understand transcriptional regulation at the single cell level (Kærn et al., 2005; Shalek et al., 2014).

Nevertheless, isolating technical noise (henceforth simply called noise) from useful signals in scRNA-Seq data and accurately estimating the relevant statistics (mean, dispersion, burstiness, etc) of true gene expression distribution is a challenging task. Compared to bulk RNA-Seq, scRNA-Seq techniques require more complicated pro-

cedures, resulting in elevated noise levels in the subsequent data pipeline. Unique Molecular Identifiers (UMI) (Kivioja et al., 2012) were introduced as a barcoding technique to reduce amplification noise, but the distribution of observed UMI counts is still insufficient for inferring the true expression distribution in many cases. Moreover, scRNA-Seq data often prove challenging for classical bulk RNA-Seq analysis methods to tackle primarily because of the prevalence of burstiness, which can lead to zero-inflated data.

Recently, Wang et al. (2017) developed *DESCEND*, a statistical method that deconvolves the true cross-cell gene expression distribution from observed scRNA-Seq UMI counts. *DESCEND* adaptively fits the true gene expression distribution using the "G-modeling" empirical Bayes distribution deconvolution approach (Efron, 2016). The "G-modeling" technique only assumes the distribution to be a general exponential family distribution (with natural spline basis), which is highly flexible and avoids specifying parametric assumptions.

Inspired by *DESCEND*, we further propose a general deconvolution framework for scRNA-Seq data, which decouples the technical sampling errors from UMI counts and then recover the true gene expression distribution. In addition, we also introduce three methods with different distributional assumptions: *ZINB*, Poisson *ash* and nonparametric deconvolution. For the true expression distribution, *ZINB* makes the zero-inflated gamma distribution assumption, Poisson *ash* only requires the assumption of unimodality for the non-zero part, and nonparametric deconvolution is assumption-free. Along with *DESCEND*, the four methods assumes standard Poisson sampling errors for scRNA-Seq data, but retain different levels of flexibility and

adaptivity due to their distributional assumptions.

We compare the estimated expression distribution, corresponding statistics and goodness of fit of the four methods on three real scRNA-Seq datasets, Zeisel data (Zeisel et al., 2015), Tung data (Tung et al., 2017) and Buettner data (Buettner et al., 2015). We show that the methods often provide similar mean and dispersion statistics despite of the discrepancies in shape of fitted expression distribution. However, our proposed method Poisson *ash* normally achieves better fit in terms of higher likelihood, and better highlights the sub-population structure preserved in Zeisel data.

The R code and analysis results are available from http://github.com/mengyin/general_ash.

5.2 Methods

Suppose we observe UMI counts from an scRNA-Seq experiment. Define observed count Y_{cg} for gene g in cell c to be the true gene expression level μ_{cg} plus additional noise. Since the underlying structure of true gene expression levels is of interest, we would like to deconvolve the variability of Y_{cg} into two parts, noise and the variability of true gene expression:

$$Y_{cg} \sim F_{cg}(\mu_{cg}), \quad \mu_{cg} \sim G_g(\cdot), \tag{5.1}$$

where F_{cg} captures the noise and G_g represents the true expression distribution of gene g across cells. After deconvolving G_g from the noisy observed counts Y_{cg}, we can

further estimate other distribution-related quantities of interest (mean, CV, etc.).

Several studies (Brennecke et al., 2013; Kim and Marioni, 2013; Kim et al., 2015; Grün et al., 2014) have examined the public scRNA-Seq datasets and proposed Poisson distribution to capture the noise in UMI counts after accounting for cross-cell differences in library size. Hence, the Poisson distribution is a suitable choice for F_{cg}:

$$Y_{cg} \sim \text{Poisson}(\alpha_c \lambda_{cg}), \quad \lambda_{cg} \sim G_g(\cdot), \tag{5.2}$$

where α_c is a cell specific scaling constant. A straightforward way to set α_c is to simply use the library size (total UMI count of cell c). Other moderated scaling factors (e.g, robust normalized scaling factors, efficiency of cell if spike-ins are available) may be suitable alternatives, depending on context. Here λ_{cg} represents the relative gene expression level for gene g in cell c. In genomic contexts, developing models for relative gene expression is typically more sensible than directly working on the absolute expression μ_{cg}.

A crucial property of scRNA-Seq data is the zero-inflated pattern. The zeros could be caused by factors like the variation in expression across cells, or the bursty process of gene transcription, where periods of RNA synthesis is followed by periods of inactivity (Chubb et al., 2006; Raj et al., 2006; Dar et al., 2012). The UMI counts of scRNA-Seq typically yield a zero-inflated gene expression distribution: zero counts in "inactivate" cells and non-zero counts in "active" cells. Therefore a point mass at zero should be included in G_g to incorporate this pattern:

$$G_g = \pi_g \delta_0 + (1 - \pi_g) H_g, \tag{5.3}$$

where π_g is the zero fraction (inactive cell fraction), δ_0 is the point mass at zero, and H_g is the gene expression distribution corresponding to active cells.

We then fit the deconvolution distribution G_g using the observed counts. The fitted distribution $\hat{G}_g$ captures the typical expression properties in single cell data we described above: average expression level ($\mathrm{E}(\hat{G}_g)$); zero-inflation (zero fraction π_g, nonzero mean $\mathrm{E}(\hat{H}_g)$), dispersion ($\mathrm{CV}(\hat{G}_g)$), etc.

*5.2.1 Zero Inflated Negative Bimomial (*ZINB*)*

Since the conjugate prior for Poisson distribution is the gamma distribution, a simple choice for H_g is $\mathrm{Gamma}(a_g, b_g)$ which results in a simple analytical form for the likelihood: after integrating out the prior distribution H_g, the marginal distribution of Y_{cg} is the mixture of a point mass at zero and a negative binomial distribution,

$$p(Y_{cg}) = \int p(Y_{cg}|\lambda_{cg})p(\lambda_{cg}) \tag{5.4}$$

$$= \pi_g\delta_0 + (1-\pi_g)\mathrm{NB}\left(\frac{Y_{cg}}{\alpha_c}; a_g, \frac{1}{b_g+1}\right), \tag{5.5}$$

where $\mathrm{NB}(x; r, p)$ is the probability density function for negative binomial distribution at x with parameters r (number of failures until end of experiment) and p (success probability).

For each gene, we estimate the parameters π_g, a_g, b_g by maximizing the log-likelihood L_g:

$$L_g(\pi_g, a_g, b_g) = \sum_c \log(p(Y_{cg}|\pi_g, a_g, b_g)). \tag{5.6}$$

The *ZINB* method is computationally efficient due to the simplicity of its statis-

tical model. However, *ZINB* places a relatively strong assumption on the expression distribution, which is that the expression of active cells follows a gamma distribution that is by definition unimodal, and also has a certain tail decay rate. Due to this lack of flexibility, *ZINB* may fail to capture the true expression distribution if there exists complicated sub-population structures.

The *ZINB* method was implemented by Abhishek Sarkar, and the code is available from https://github.com/aksarkar/singlecell-qtl.

5.2.2 DESCEND

Recently Wang et al. (2017) proposed *DESCEND*, a method for scRNA-Seq expression distribution deconvolution that adoptsg the G-modeling empirical Bayes distribution deconvolution technique in Efron (2016). *DESCEND* takes an exponential family distribution (Poisson, log-normal and gamma distributions being special cases) as H_g and estimates its shape adaptively from the observed counts using natural cubic splines.

The density of H_g has the following form:

$$p_H(x) = \exp(Q(x)^T\theta - \phi(\theta)), \tag{5.7}$$

where θ is a vector of parameters and $\phi(\theta)$ is the normalization factor. Specific forms of $Q(x)$ corresponds to specific parametric models (e.g. gamma, log-normal). In practice, *DESCEND* sets $Q(x)^T$ as a five-degree natural cubic spline function so that the model can learn $Q(x)$ adaptively from the data.

To estimate the parameters θ_g, DESEND maximizes the penalized likelihood

$$L_g^*(\theta_g) = \sum_c \log(p(Y_{cg}|\theta_g)) - c_0||\theta_g||^2, \tag{5.8}$$

where c_0 is an adaptively chosen regularization constant. Since the natural cubic spline based exponential family is highly flexible, *DESCEND* uses this regularization term to avoid over-fitting.

The *DESCEND* method is implemented in an R package `descend`, which is publicly available from https://github.com/jingshuw/descend.

5.2.3 Poisson ash

Stephens (2016) proposed *ash* (Adaptive SHrinkage) to model a list of normal observations which share a common underlying prior distribution. In the context of gene expression studies, *ash* suggests using a unimodal distribution as the prior for the true expression levels, and estimates the unimodal prior adaptively with empirical Bayes. The unimodal assumption of *ash* provides more flexibility than any specific parametric distribution assumption, but also preserves robustness against over-fitting. Here we extend the *ash* framework to tackle the gene expression distribution deconvolution problem for scRNA-Seq data.

We now assume that H_g is a unimodal distribution. In practice, a mixture of uniform distributions can be used to approximate the unimodal distribution:

$$H_g = \sum_{k=1}^{K_g} p_{gk}\text{Uniform}[a_{gk}, c_g] + \sum_{l=1}^{L_g} q_{gl}\text{Uniform}[c_g, b_{gl}], \tag{5.9}$$

where c_g is the mode (non-negative in our setting), $\mathbf{p}_g, \mathbf{q}_g$ are mixture proportions (sum to 1), and a_{gk}, b_{gl} ($0 \leq a_{gk} < c_g$, $c_g < b_{gl}$) are pre-selected grids that cover a sufficiently wide range of values. We estimate the mode c_g and mixture proportions $\mathbf{p}_g, \mathbf{q}_g$ by maximizing the likelihood:

$$L_g(\mathbf{p}_g, \mathbf{q}_g, c_g) = \sum_c \log(p(Y_{cg}|\mathbf{p}_g, \mathbf{q}_g, c_g)). \tag{5.10}$$

In *ash* framework, given a fixed mode c_g, optimizing L_g over all possible $\mathbf{p}_g, \mathbf{q}_g$'s is a convex optimization problem, and we denote the optimized log-likelihood (given c_g) as $\hat{L}_g(c_g) = \arg\max_{\mathbf{p}_g, \mathbf{q}_g} L_g(\mathbf{p}_g, \mathbf{q}_g, c_g)$. Since $\hat{L}_g(c_g)$ is a function of c_g, we further use 1-d numerical optimization method to fit the mode:

$$\hat{c}_g = \arg\max_{c_g} \hat{L}_g(c_g). \tag{5.11}$$

With a large enough number of mixture components, any general unimodal distribution can be well approximated by the uniform mixture distribution in (5.9). In our applications 30-50 mixture components generally suffice.

A special case for H_g would be the non-negative unimodal distribution with mode at 0, which implies that $K_g = 0$ and all other uniform mixture components have lower limits at 0. Since the optimization procedure can be numerically unstable in such a corner case, in practice we separately fit this Poisson *ash* model with a single mode at 0, and compare its log-likelihood with the optimized log-likelihood in (5.10). The model with higher log-likelihood is then recommended.

Note that for scRNA-Seq data, we use the "identity link" which assumes λ_{cg}

unimodal distributed, instead of the "log link" which assumes $\log(\lambda_{cg})$ unimodal. Due to the zero-inflated nature of scRNA-Seq, fitting the log link model is numerically unstable with extremely small grid lower bound.

The Poisson *ash* method is implemented in R package *ashr*, which is publicly available from https://github.com/stephens999/ashr.

5.2.4 Nonparametric deconvolution

Note that all above mentioned methods make assumptions on the expression distribution H_g for active cells: *ZINB* assumes a single gamma distribution; *DESCEND* assumes an exponential family distribution; Poisson *ash* assumes a unimodal distribution. While the distributional constraints lessens the possibility of over-fitting due to noise present in data, we still propose the fully nonparametric deconvolution method as an alternative approach.

Specifically, any distribution H_g can be approximated by a mixture of uniforms on a sufficiently dense grid (Koenker and Mizera, 2014):

$$H_g = \sum_{k=1}^{K_g} p_{gk} \text{Uniform}[(k-1)a_g, ka_g], \tag{5.12}$$

where $\mathbf{p}_g$ are mixture proportions (sum to 1), and a_g is the pre-selected grid step-size. To ensure the goodness of nonparametric approximation, the number of mixture components K_g should be sufficiently large, and $[0, K_g a_g]$ covers a sufficiently wide range.

The computations necessary are essentially identical to those for general ash

(Section 4), so we reuse them here to estimate $\mathbf{p}_g$.

5.3 Applications

5.3.1 Zeisel data

Zeisel et al. (2015) described a scRNA-Seq dataset of mouse hippocampal region. The dataset has read counts of 12234 genes in 3005 cells from the mouse somatosensory cortex and hippocampus CA1 region. The 3005 cells have been clustered into 7 major cell types: Astrocytes-Ependymal, Endothelial-Mural, Interneurons, Microglia, Oligodendrocytes, CA1 pyramidal and S1 pyramidal, and the number cells in each cell type are 224, 235, 290, 98, 820, 939 and 399 respectively. The dataset is publicly available from https://hemberg-lab.github.io/scRNA.seq.datasets/mouse/esc/.

We run *ZINB*, *DESCEND*, Poisson *ash* and the nonparametric deconvolution method on a subset of this scRNA-Seq dataset and compare their deconvolution results. This subset of data consists of cell types Astrocytes-Ependymal, Endothelial-Mural and Microglia (557 cells in total). We use the normalized library sizes as the scaling factors α_c for cells:

$$\alpha_c = \frac{\sum_g Y_{cg}}{\sum_{c,g} Y_{cg}/C}, \tag{5.13}$$

where C is the total number of cells. The numerator in (5.13) is the raw library size for cell c (total counts), and the denominator is the average library size across all cells.

In some applications, the properties of G_g may be further used as genetic variant

specific features. For instance, the mean and spread information extracted from the expression distribution can be used in eQTL analysis. In general, we would also like to investigate if the different deconvolution methods would produce $\hat{G}_g$ with significantly distinct shapes. Hence we compare the deconvolution results in the following aspects.

Likelihood To evaluate the goodness of fit of different models, we can compare the log-likelihood of the probability models:

$$L_g = \sum_c \log p(Y_{cg}). \tag{5.14}$$

The *ZINB*, Poisson *ash* and nonparametric deconvolution directly provide the log-likelihood in (5.14) when fitting the hyperparameters with empirical Bayes method. However, *DESCEND* optimizes the penalized log-likelihood L_g^* in (5.8) instead. Fortunately, the *DESCEND* package also gives the optimum value for unpenalized likelihood L_g (without providing the corresponding fitted deconvolution model).

Even though *ZINB*, *DESCEND*, Poisson *ash* and nonparametric deconvolution have the same Poisson likelihood, their models for G_g are not nested. Hence, using likelihood ratio tests to compare the models is statistically unsound. However, the difference in log-likelihoods of different methods still reveals the goodness of fit for scRNA-Seq data. We consider the following scenarios for the difference in log-likelihood between two methods: within ± 2 (insignificant difference); over ± 2 but within ± 5 (moderate significant difference); over ± 5 (very significant difference).

We start with the restricted version of Poisson *ash* which only allows unimodal

G_g: the uni-mode is either 0 or some non-zero positive value, and in the former case the density of G_g monotonically decays. We denote this model as Poisson *ash* (unimodal), which essentially assumes that one unimodal distribution is sufficient to capture the expression distribution across cells. We compare its log-likelihood with that of the the most flexible approach, fully nonparametric deconvolution. For only 0.80% genes, the nonparametric deconvolution has significantly higher log-likelihood than that of Poisson *ash* unimodal model (with difference being at least 2). For the rest over 99% genes, the log-likelihood difference is insignificant, which means the Poisson *ash* unimodal model is quite sufficient to capture the underlying gene expression distribution.

For genes where the nonparametric deconvolution achieves significantly higher log-likelihood, we further inspect those genes with largest log-likelihood difference to see if more complicated models are needed to fit the data. Figure 5.1 shows the scaled expression Y_{cg}/α_c distribution for six genes *Atp1a2*, *C1qa*, *Eif4a1*, *Ccl4*, *Cdc42* and *Klhl9*, where the log-likelihood difference between Poisson *ash* unimodal model and nonparametric deconvolution is 5.61, 5.25, 4.92, 4.63, 4.50 and 4.43 respectively. The six genes display a common pattern that, the scaled expression histogram split the cells into zero expression and non-zero expression parts, with noticeable gap between them. This suggests us to consider a more flexible expression distribution in form (5.3), with a point mass at zero δ_0 and some distribution H_g for the non-zero part.

Hence, we try the models with δ_0 but different assumptions for H_g: *ZINB* (H_g is gamma distribution); *DESCEND* (H_g belongs to exponential family) and Poisson *ash* (H_g is unimodal) and check their log-likelihood improvements compared to the

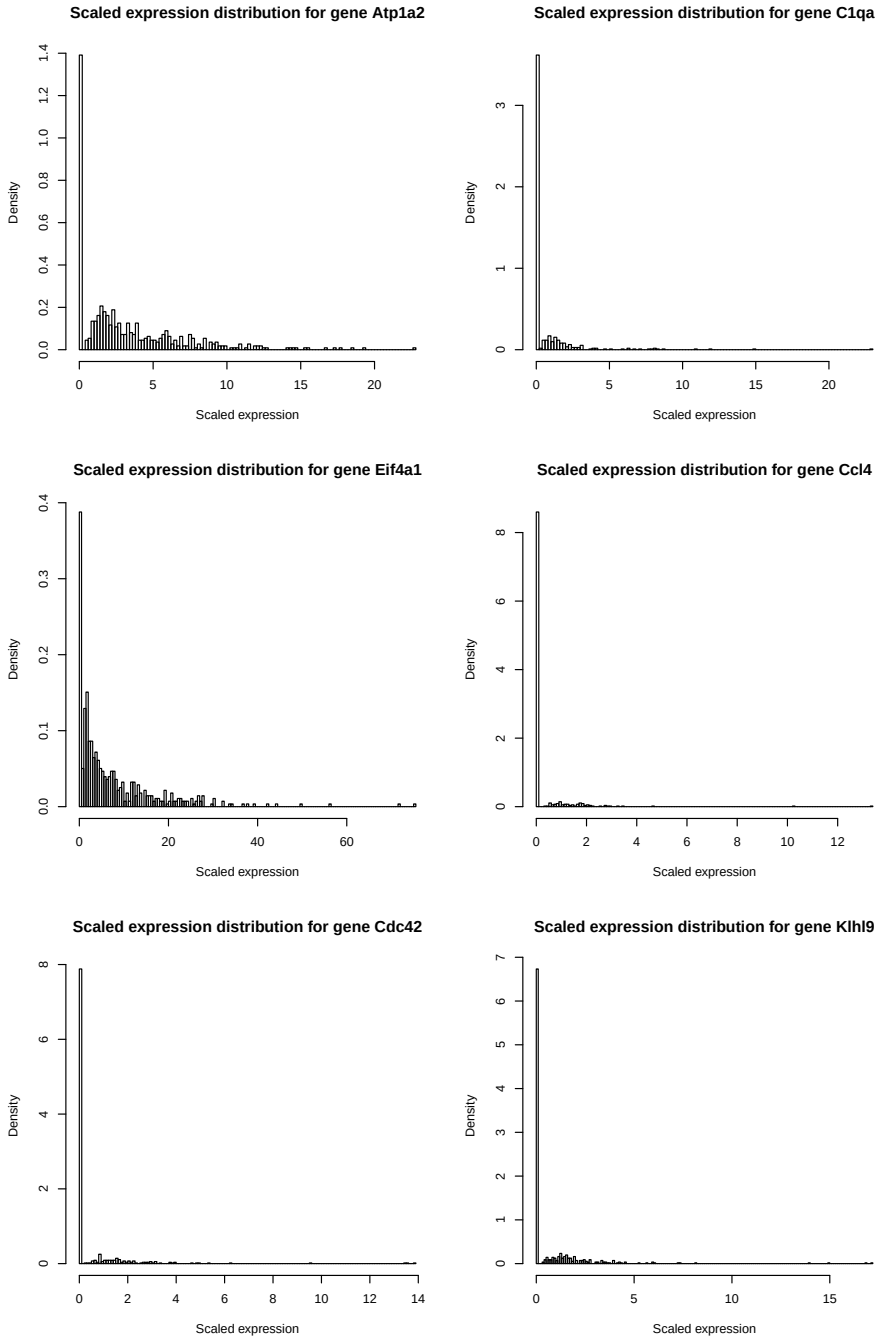

Figure 5.1: Distribution of scaled expression Y_{cg}/α_c for genes *Atp1a2*, *C1qa*, *Eif4a1*, *Ccl4*, *Cdc42* and *Klhl9*, where the nonparametric deconvolution model has significantly higher log-likelihood than that of Poisson *ash* unimodal model.

Poisson *ash* unimodal model (which is used as baseline model). Table 5.1 shows their log-likelihood improvements for the above six genes *Atp1a2*, *C1qa*, *Eif4a1*, *Ccl4*, *Cdc42* and *Klhl9*. Figure 5.2 shows the fitted G_g by Poisson *ash* and *DESCEND* for these six genes.

Table 5.1: Log-likelihood improvement upon the Poisson *ash* unimodal model for genes *Atp1a2*, *C1qa*, *Eif4a1*, *Ccl4*, *Cdc42* and *Klhl9*.

Gene	Nonparametric	Poisson *ash*	*DESCEND*	*ZINB*
$Atp1a2$	5.61	0.00	-28.12	-18.23
$C1qa$	5.25	0.00	-8.97	-7.33
$Eif4a1$	4.92	4.35	3.78	-1.98
$Ccl4$	4.63	0.00	-214.15	-1.63
$Cdc42$	4.50	2.19	3.78	1.42
$Klhl9$	4.43	4.16	1.20	-7.64

The results suggest that for genes *Atp1a2*, *C1qa* and *Ccl4*, the Poisson *ash* model with δ_0 plus unimodal H_g does not make difference from the baseline unimodal model, since the fitted H_g gets highest probability for the mixture uniform components with left limit 0, and thus makes G_g unimodal with mode at 0. On the other hand, *ZINB* and *DESCEND* yield even lower log-likelihoods than baseline. For genes *Eif4a1*, *Cdc42* and *Klhl9*, Poisson *ash* significantly increment the baseline log-likelihood of unimodal model. The *DESCEND* model also substantially improves the baseline log-likelihood for *Eif4a1* and *Cdc42*. Among these six genes, *Cdc42* is the only one where *DESCEND* achieves slightly higher log-likelihood than Poisson *ash*. In Figure 5.1, the scaled expression histogram for *Cdc42* has a more than one relatively eye-catching bumps (e.g., around 0.8 and 1.6), which might be the reason why *DESCEND* gives slightly higher log-likelihood.

The gene $Cdc42$ is an example where the distinct assumptions on H_g lead to

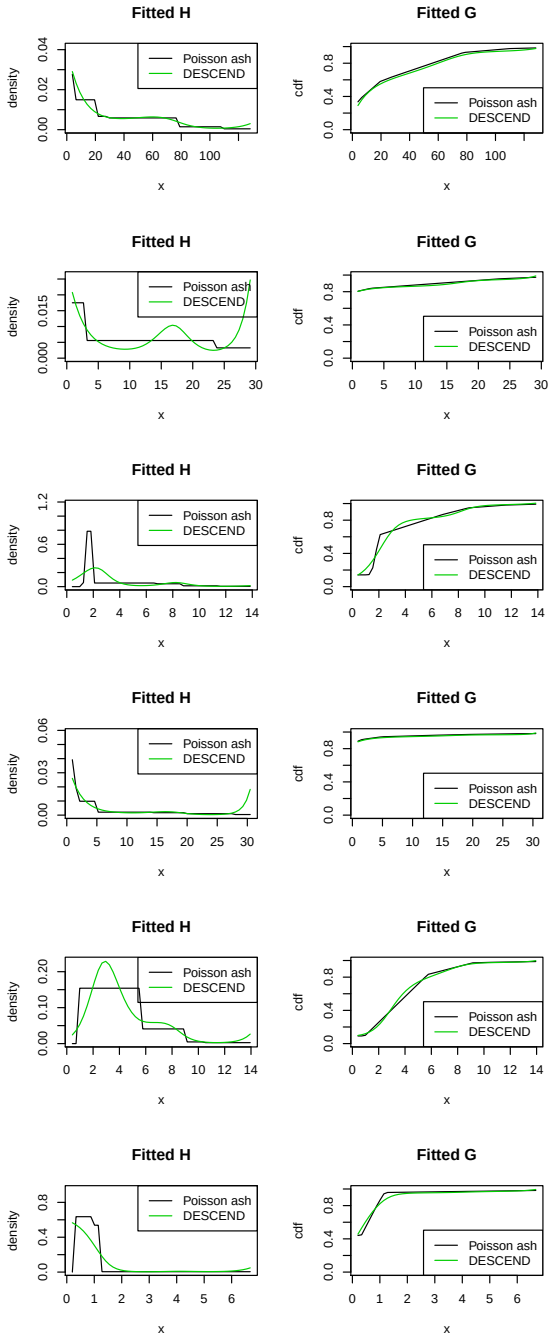

Figure 5.2: Fitted G_g by Poisson *ash* and *DESCEND* for genes *Atp1a2*, *C1qa*, *Eif4a1*, *Ccl4*, *Cdc42* and *Klhl9* (from top to bottom). Each row shows the pdf and cdf function of $\hat{G}_g$ for that gene.

different model fits. We further investigate if the unimodal assumption of Poisson *ash* is sufficient to capture the non-zero expression distribution in contrast to *DESCEND*. Table 5.2 shows the distribution of the differences in log-likelihood. For each method (*DESCEND*, *ZINB*, nonparametric), we subtract the log-likelihood of Poisson *ash* from that of the target method for each gene, and then summarize the percentage of genes with log-likelihood difference falling into intervals $(-\infty, 5]$, (-5,-2], (-2,2], (2,5] and $(5, \infty)$. The five categories corresponds to the cases where: the target method gives much lower log-likelihood; significantly lower log-likelihood; similar log-likelihood; significantly higher log-likelihood; much higher log-likelihood than that of Poisson *ash*.

Table 5.2: Distribution (%) of log-likelihood difference between deconvolution methods and Poisson *ash*.

	$(-\infty, 5]$	(-5,-2]	(-2,2]	(2,5]	$(5, \infty)$
ZINB	2.43	12.09	85.48	0.01	0.00
DESCEND	6.26	7.71	85.93	0.09	0.01
Nonparametric	0.13	0.11	99.03	0.72	0.01

From Table 5.2, we see that for most genes (over 99%) , the nonparametric deconvolution and Poisson *ash* don't have significant differences in terms of the log-likelihood difference. Presumably, nonparametric deconvolution is the most flexible method and should achieve higher log-likelihood than Poisson *ash* in any scenario. However, due to numerical errors and the limitation of uniform mixtures, there are less than 1% genes where the former has noticeably higher log-likelihood. The majority of the genes result in similar log-likelihoods for Poisson *ash* and nonparametric deconvolution, so the extra flexibility provided by nonparametric deconvolution seems unnecessary here.

For *ZINB*, which puts stronger assumption on G_g (H_g is single gamma distribution) than Poisson *ash* (H_g is unimodal distribution), it is natural that Poisson *ash* gives higher log-likelihood. From Table 5.2, Poisson *ash* has much higher (> 5) log-likelihood compared to *ZINB* for 2.43% genes, and has significantly higher (> 2) log-likelihood for another 12.09% genes. For these genes, the zero-inflated gamma prior assumption of *ZINB* may not suffice in fitting this specific scRNA-Seq dataset. The rest 85.48% genes have similar log-likelihoods for Poisson *ash* and *ZINB*, and there are only 0.01% outlier genes where the *ZINB* model gives a better fit due to numerical instability.

The comparison between *DESCEND* and Poisson *ash* is an interesting one. The prior H_g for *DESCEND* belongs to the general exponential family, so its shape is not restricted to any unimodal distribution. Nevertheless, the exponential family assumption and the limited number of basis used in g-modeling would constrain the shape of distribution. As a result, it is hard to directly compare the theoretical flexibility of the two models. Moreover, even though *DESCEND* gives the optimized unpenalized likelihood, it actually fits the model by optimizing the penalized likelihood to avoid over-fitting.

We still first compare the (unpenalized) log-likelihood of *DESCEND* with that of Poisson *ash*. Table 5.2 shows that Poisson *ash* has much higher (> 5) log-likelihood compared to *DESCEND* for 6.26% genes, and has significantly higher (> 2) log-likelihood for another 7.71% genes. There are 85.93% genes with similar log-likelihoods for Poisson *ash* and *DESCEND*, yet 0.1% genes have significantly higher log-likelihood for *DESCEND*. We checked the genes where *DESCEND* gives

significantly higher log-likelihood: the top five genes with the highest log-likelihood differences are *Nek7*, *Agpat3*, *Fam216b*, *Tdrp* and *Gak*, with log-likelihood differences over 3.

The scaled expression histograms for these five genes are given in Figure 5.3. We see that these genes typically have few large outliers which make the histogram tail longer. Furthermore, the frequency of cells does not simply decrease over scaled expression level, but rather has some small bumps in the middle.

We plot the fitted prior G_g from *DESCEND* and Poisson *ash* for these 5 genes in Figure 5.4. The left column shows the pdf function of the fitted prior $\hat{G}_g$ (point mass at zero already absorbed) and the right column shows the cdf function of $\hat{G}_g$. The visualization indeed reveals the discrepancy between *DESCEND* fitted prior and Poisson *ash* fitted prior. Unfortunately *DESCEND* only provides density within this x-axis range so we are unable to plot the densities outside the current range in Figure 5.4. However, the cdf plots already indicate that for genes $Nek7$, $Agpat3$, $Tdrp$ and Gak, *DESCEND* puts a much heavier probability mass on the right tail outside the x-axis range than Poisson *ash*, which is actually unconvincing according to the original scaled expression histogram (Figure 5.3).

Even though *DESCEND* provides extra flexibility by allowing multimodality for H_g and hence achieves higher log-likelihood for these genes, its goodness of fit is still questionable. Looking back at Figure 5.3, it is insufficient to conclude that the bumps in frequency are due to underlying true expression distribution rather than Poisson noise, since the frequency bumps are not too large either. Moreover, the extremely long tailed $\hat{G}_g$ fitted by *DESCEND* seems suspicious, which suggests

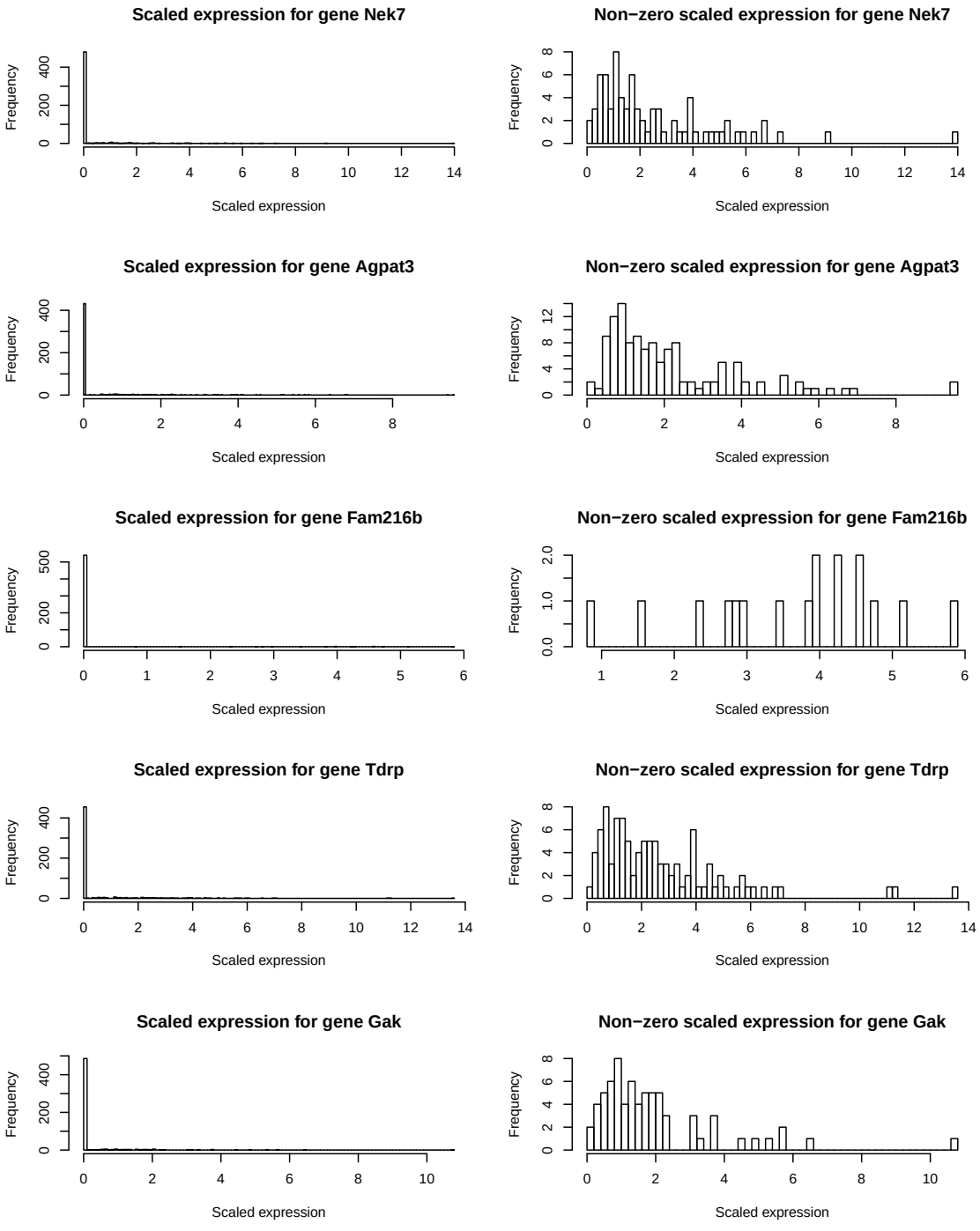

Figure 5.3: Distribution of scaled expression Y_{cg}/α_c for genes *Nek7*, *Agpat3*, *Fam216b*, *Tdrp* and *Gak*.

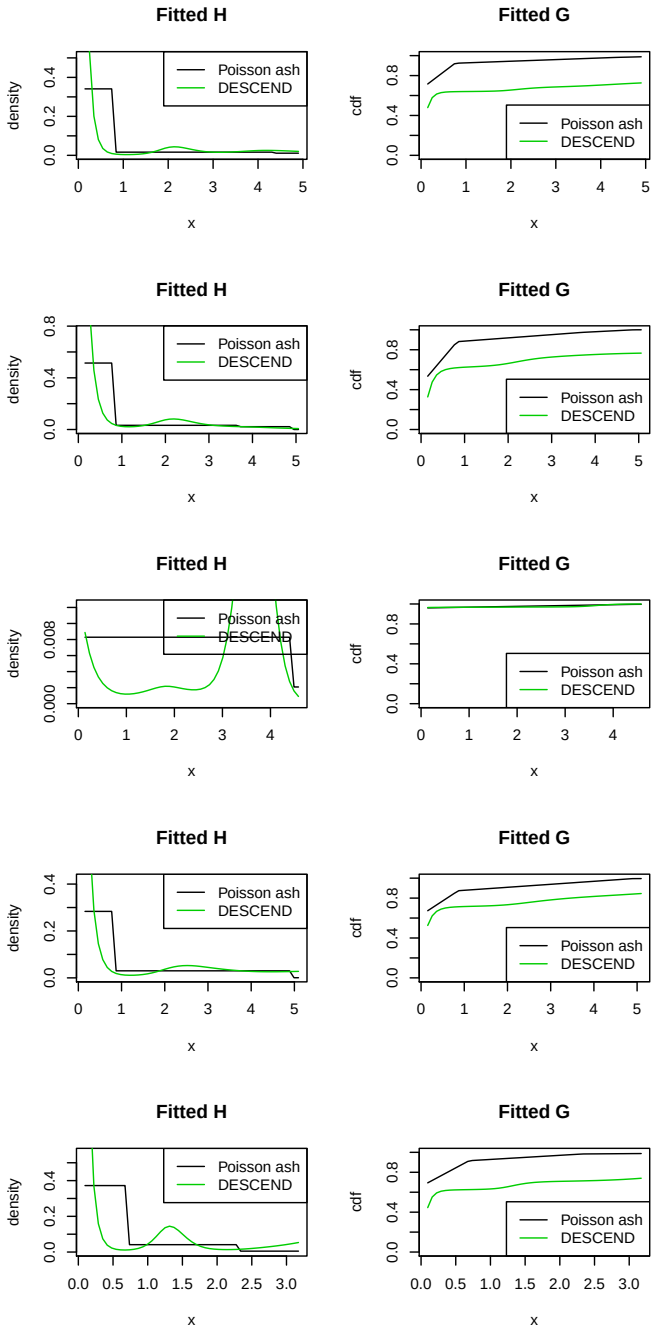

Figure 5.4: Fitted G_g for genes *Nek7*, *Agpat3*, *Fam216b*, *Tdrp* and *Gak*, where *DESCEND* gives much higher log-likelihood than Poisson *ash*. Each row shows the pdf and cdf function of $\hat{G}_g$ for that gene.

DESCEND might be over-sensitive to large outliers.

One of the original purposes of having a flexible prior G_g is to deal with the potential sub-populations in scRNA-Seq data. In Zeisel data, we use the expression counts for cell types Astrocytes-Ependymal, Endothelial-Mural and Microglia. To investigate if this sub-population structure of cell types could result in multimodal $\hat{G}_g$, we further check the raw count distribution grouped by cell types (here we split Astrocytes-Ependymal into Astrocytes and Ependymal; split Endothelial-Mural into Endothelial and Mural). For gene *Agpat3* where multiple bumps exist in scaled expression distribution (Figure 5.3), we visualize the scaled expression distribution by cell types in Figure 5.5. The most notable difference in scaled expression distribution among different cell types is in the tail shape. The scaled expression distribution of Astrocytes cells yields significantly longer tail than that of Ependymal and Microglia, and also shows a rough bimodal pattern with two major parts split at 3. On the other hand, the scaled expression levels for Microglia and Ependymal cells are quite low and concentrated, and the scaled expression of Mural cells is either smaller than 3 or bigger than 9.

This example shows that the scaled expression seems to have different spreads for different cell types, rather than yielding different mode/average levels. The bimodal pattern within Astrocytes mainly contributes to the overall "bimodal" scaled expression distribution, while the differences in expression distribution among the other cell types seem to be determined mainly by spreads.

Note that in Poisson *ash*, H_g consists of many uniform mixture components with different widths (anchored at same mode). Even though the mixture com-

ponents were not specifically designed for clustering sub-populations, their various spreads seem to corroborate the phenomenon described above, in that different sub-populations have tails with differing lengths.

Mean The mean of deconvolution distribution $\mathrm{E}(G_g)$ essentially represents for the expected relative expression level for gene g, after removing the technical sampling error (Poisson noise) in scRNA-Seq data:

$$\mathrm{E}(G_g) = (1 - \pi_g)\mathrm{E}(H_g). \tag{5.15}$$

Figure 5.6 plots the gene-specific deconvolution distribution means for *ZINB*, *DESCEND* and nonparametric deconvolution against that of Poisson *ash* (on log-log scale). We see that the four methods result in very similar means for G_g. This is expected, since the average expression level for each gene after removing the Poisson

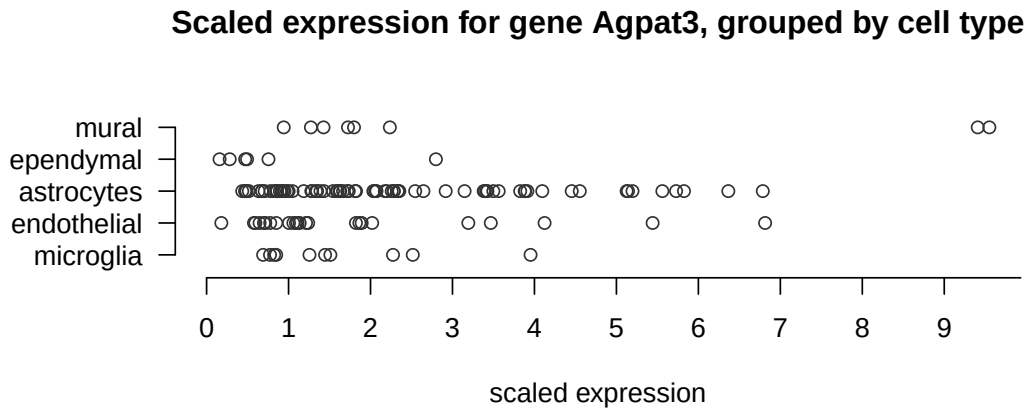

Figure 5.5: Non-zero scaled expression Y_{cg}/α_c for gene *Agpat3*, grouped by cell types. Each point in the figure represents for the scaled expression level of one single cell.

noise should be relatively consistent across the deconvolution methods, even though the specific shape of G_g can be different.

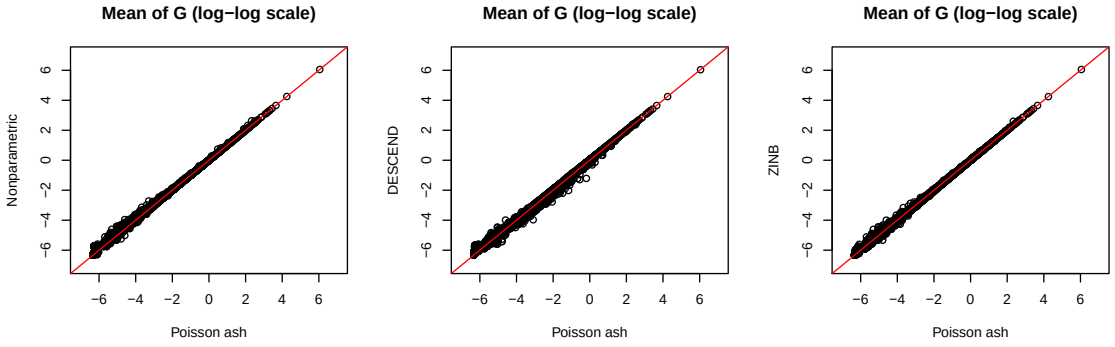

Figure 5.6: Mean of the deconvolution distribution G_g for methods nonparametric, *DESCEND*, *ZINB* against that of Poisson *ash*.

CV (Coefficient of Variation) Apart from the mean of G_g, the spread of the expression distribution can also useful be in some contexts (e.g. eQTL analysis). We also compute the CV (coefficient of variation) for fitted G_g:

$$\mathrm{CV}_g = \frac{\mathrm{SD}(G_g)}{\mathrm{E}(G_g)}, \tag{5.16}$$

where the variance of G_g is given by

$$\mathrm{Var}(G_g) = (1 - \pi_g)(\mathrm{E}(H_g)^2 + \mathrm{Var}(H_g)) - \mathrm{E}(G_g)^2. \tag{5.17}$$

Figure 5.7 shows the gene-specific CV of fitted deconvolution distribution $\hat{G}_g$ for *ZINB*, *DESCEND* and nonparametric deconvolution against that of Poisson *ash*. In general, Poisson *ash* and nonparametric deconvolution have quite similar CV's.

This is consistent with our previous results, where Poisson *ash* and nonparametric deconvolution have very similar performance on the Zeisel dataset .

However, the difference in CV between *DESCEND* and Poisson *ash* is much bigger. Though the CV of *DESCEND* is often comparable with that of Poisson *ash*, there are cases where *DESCEND* only results in a CV that is half that of Poisson *ash*. Note that the left bottom corner of middle plot in Figure 5.7 is an example where the two methods return dramatically different results. In particular, Poisson *ash* results a very small CV (close to 0) but *DESCEND* CV is almost as large as 10. The reason for this is that these outlier genes barely have any non-zero counts. For example, genes *Gm19557*, *2310002J15Rik*, *Chrdl2 Mei1* and *Speer4e* only have one active cell out of the 557 cells and the only non-zero expression count is 1. For these genes, Poisson *ash* returns $\hat{G}_g$ CV as 0.64, while *DESCEND* returns a CV over 10. We visualize the fitted distribution $\hat{G}_g$ for gene *Gm19557* to illustrate the issue. Figure 5.8 shows that the fitted distribution of Poisson *ash* is extremely short tailed and ends at a very small value near 0, whereas $\hat{G}_g$ of *DESCEND* has a much heavier tail and decay very slowly. For this gene with 556 zeros and 1 one, the mean of $\hat{G}_g$ for *DESCEND* and Poisson *ash* are both very small (around 0.002) but still within the same order of magnitude. However, the big difference in CVs for the two methods reveals sensitivity to model assumptions in such a corner case: a mixture of uniforms can directly cut the tail at some point, but the tail of an exponential family distribution can expand all the way to infinity. With only one non-zero count, it is extremely hard to fit the non-zero prior part H_g. The method *ZINB* shares the problem in that its CV is often much bigger than that of Poisson *ash*. The lack

of flexibility of *ZINB* is more amplified here, since the fitted gamma distribution H_g has a relatively fixed decay rate and results in a heavier tail than many other exponential family distributions.

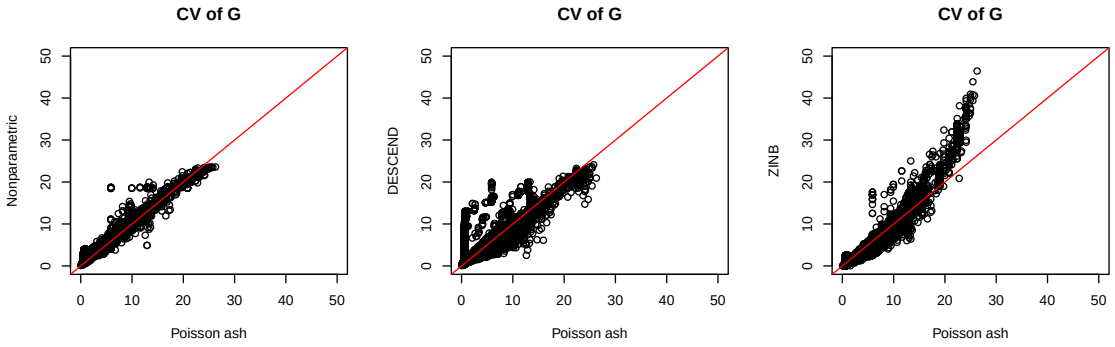

Figure 5.7: CV of the deconvolution distribution G_g for methods nonparametric, *DESCEND*, *ZINB* against that of Poisson *ash* on Zeisel data.

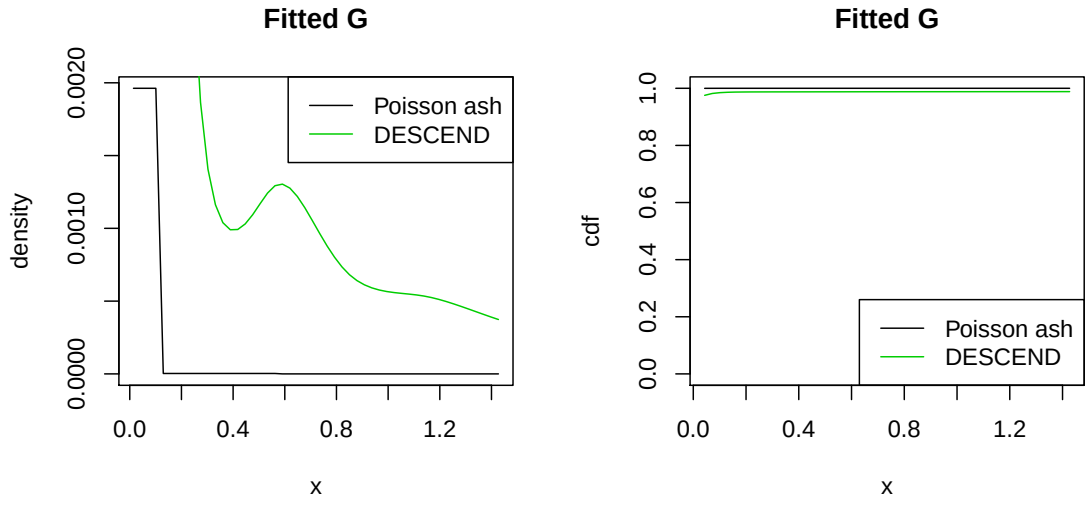

Figure 5.8: Fitted G_g for gene *Gm19557*, where *DESCEND* gives much higher CV than Poisson *ash*.

We now throw out the genes with less than 5 non-zero counts and compare the

CVs using the filtered data. Figure 5.9 shows the gene-specific CV of $\hat{G}_g$ for *ZINB*, *DESCEND* and nonparametric deconvolution against that of Poisson *ash*. Now the CVs remain more consistent across the methods, and there are hardly any large outliers in CVs (bigger than 20).

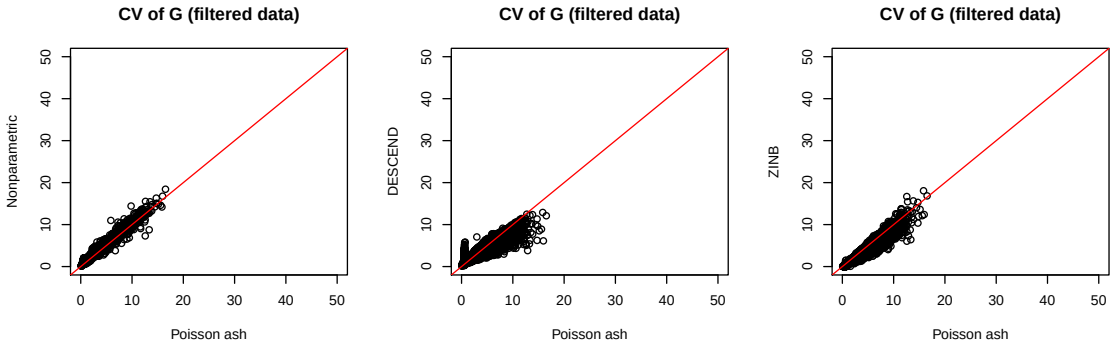

Figure 5.9: CV of the deconvolution distribution G_g for methods nonparametric, *DESCEND*, *ZINB* against that of Poisson *ash* on filtered Zeisel data (remove genes with less than 5 non-zero counts).

Null proportion Another important aspect of scRNA-Seq expression analysis is to capture the nonzero fraction. Specifically, the nonzero fraction $1-\pi_g$ represents for the fraction of cells where the gene is expressed. Figure 5.10 shows the gene-specific fitted null proportion $\hat{\pi}_g$ for *ZINB*, *DESCEND* and nonparametric deconvolution against that of Poisson *ash*. Unlike the mean and CV, there are no strong correlations in $\hat{\pi}_g$ among different methods. Nevertheless, this is not surprising considering the identifiability issue when estimating the null proportion: it is hard to distinguish if the low counts are due to pure Poisson noise or due to very small positive expression signal. To preserve adaptivity, none of the methods forces H_g to be far isolated from the point mass at zero, thus making it difficult to precisely capture the null proportion.

Shape For the shape of distribution G_g, one major difference in distribution assumptions between *DESCEND* and Poisson *ash* is the ability to handle multimodality. Poisson *ash* only allows for unimodal H_g, but *DESCEND* does not put any constraints on its number of modes, and sometimes does result in multimodal $\hat{H}_g$ in practice. However, whether or not the multimodality assumption is necessary in practice would be up for debate. For example, as we discussed in Section 5.3.1, the extra flexibility of allowing multimodality could lead to over-fitting issues instead of improving the fitted expression distribution when large outliers are present in scRNA-Seq data.

To detect potential multimodal pattern for scRNA-Seq expression distribution, Bacher and Kendziorski (2016) proposed the following approach: genes for which at least 75% of cells showed non-zero expression are selected. For each gene, zeros were removed and the R package *Mclust* was applied to log expression to estimate the number of modes (because zeros were removed prior to *Mclust*, a mode at zero will not contribute to the total number of modes). The *Mclust* package fit a Gaussian

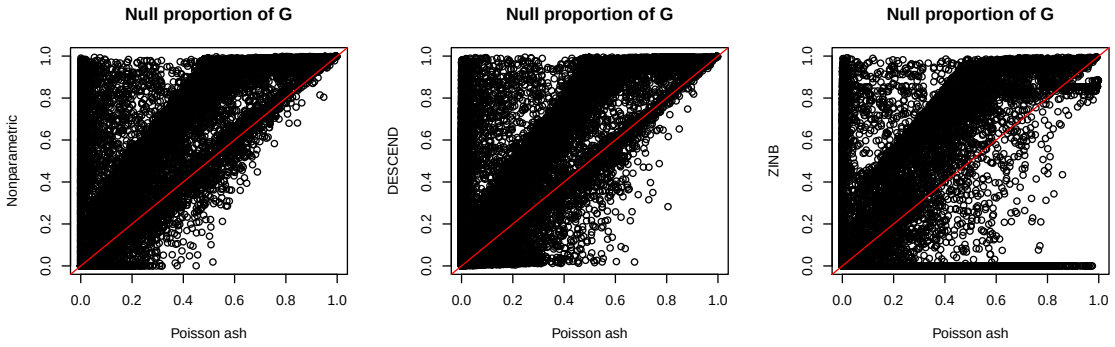

Figure 5.10: Null proportion π_g of the deconvolution distribution G_g for methods nonparametric, *DESCEND*, *ZINB* against that of Poisson *ash* on Zeisel data.

mixture model, and the output optimal number of mixture components was used as the estimate of the number of modes. Note that this approach is particular different from our previous discussed deconvolution methods: it directly fits a Gaussian mixture model for the observed expression (on log scale), but does not involve any deconvolution step which estimates the true expression distribution from the noisy observations. Bacher and Kendziorski (2016) analyzed three scRNA-Seq datasets and visualized the number of modes in Figure 1c of their paper, showing that at least 60% genes have more than two modes.

We applied the same *Mclust* method on Zeisel data, among the 3215 genes with at least 75% non-zero expressed cells, 1666 (51.82%) genes just have one mode, 681 (21.18%) and 61 (1.9%) genes have 2 and 3 modes respectively, and only 1 gene, *Gpr37l1*, has 4 modes. We visualize the scaled expression distribution and the fitted deconvolution distribution of *DESCEND* and Poisson *ash* for gene *Gpr37l1* in Figure 5.11. The log-likelihood of nonparametric deconvolution, Poisson *ash* and *DESCEND* models are given by -1328.454, -1329.898 and -1338.806 respectively. For this gene, Poisson *ash* is sufficient to capture the underlying expression distribution since its log-likelihood is very close to that of the nonparametric deconvolution. Figure 5.11 also indicates that the Poisson *ash* fitted distribution seems more sensible compared to *DESCEND*, that the density substantially drops beyond 20. Therefore, even though the observed expression histogram has quite a few bumps and results in 4 modes in the *Mclust* fitted model, it is nevertheless convincing that the underlying true expression distribution is multimodal.

5.3.2 *Tung data*

The data in Tung et al. (2017) have three C1 replicates from three human induced pluripotent stem cell lines and UMI were added to all samples. One replicate of the individual NA19098.r2 was removed from the data due to low quality and 564 cells are kept after filtering. The dataset is publicly available from https://github.com/jdblischak/singleCellSeq. Each replicate is a batch with less than 100 cells. Unlike the Zeisel data, the sample size of Tung data is relatively small. We run *DESCEND* and Poisson *ash* on one replicate NA19091.r2 (96 cells in total) and compare their results on this dataset with a limited number of cells.

Figure 5.12 shows the mean, CV and null proportion $\hat{\pi}_g$ of the fitted distribution $\hat{G}_g$ given by *DESCEND* and Poisson *ash*. For the CV plot we also filter out the genes with less than 5 non-zero counts. The patterns of $\hat{G}_g$ mean/CV for the two methods are similar to what we observed from Zeisel data: the means are almost the same (Pearson correlation 99.99%), the CVs are very similar (Pearson correlation 96.66%), but the null proportions are less correlated due to the identifiability issue.

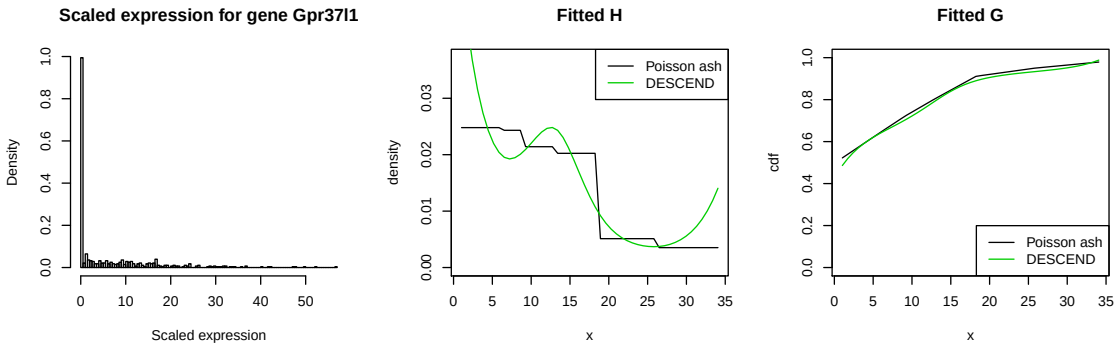

Figure 5.11: Scaled expression distribution, fitted expression distribution $\hat{G}_g$ by *DESCEND* and Poisson *ash* of gene *Gpr37l1*.

For Tung data, most genes have very small zero fractions, which actually makes the null proportion easier to estimate compared to Zeisel data.

5.3.3 Buettner data

The data in Buettner et al. (2015) were used in Bacher and Kendziorski (2016) to estimate the number of modes of observed log expression distribution. Individual *Mus musculus* embryonic stem cells were sorted using fluorescence-activated cell sorting (FACS) for cell-cycle stage, then single cell RNA-Seq was performed using the C1 Single Cell Auto Prep System (Fluidigm). The scRNA-Seq dataset is consisted of 96 *Mus musculus* embryonic stem cells in the G2M stage of the cell cycle. This dataset is publicly available at https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-2805/.

Bacher and Kendziorski (2016) applied *Mclust* on 1000 genes with at least 25% non-zero expressed cells and thus estimate the number of modes for log expression. There are 343, 614, 41 and 2 genes with 1, 2, 3 and 4 modes respectively.

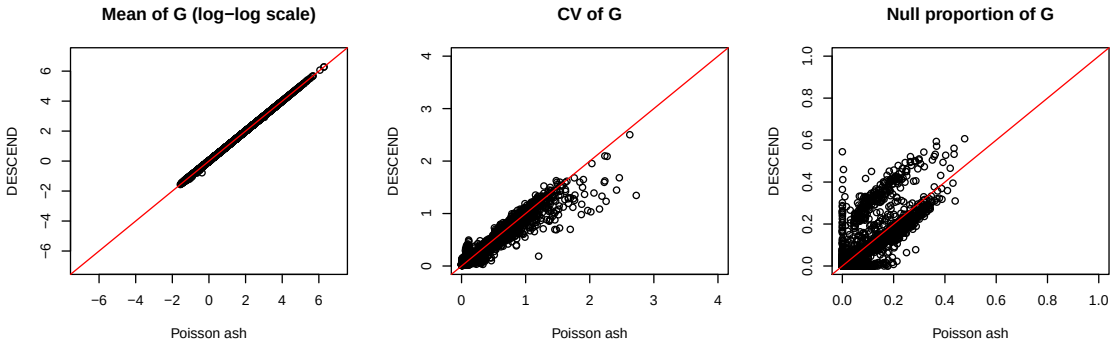

Figure 5.12: Mean, CV and null proportion of the fitted distribution $\hat{G}_g$ for methods *DESCEND* and Poisson *ash* on Tung data.

Gene *Dyrk1a* (ENSMUSG00000022897) has 4 *Mclust* modes. We apply *DESCEND*, Poisson *ash* and the nonparametric deconvolution on this gene and the log-likelihoods are -638.36, -630.71 and -621.51 respectively. The notable difference between log-likelihoods of Poisson *ash* and nonparametric deconvolution implies that a unimodal distribution might be insufficient to capture the non-zero expression distribution. Figure 5.13 shows the scaled expression distribution and fitted G_g by *DESCEND* and Poisson *ash* for *Dyrk1a*. In this case, a multimodal H_g might be needed.

5.4 Discussion

For scRNA-Seq data, Wang et al. (2017) proposed *DESCEND*, a method that deconvolves the true cross-cell gene expression distribution from observed UMI counts. In this project, we generalize a deconvolution framework for scRNA-Seq data that removes the noise from UMI counts and then estimate the true gene expression distribution. The method *DESCEND* is a special case which assumes the expression

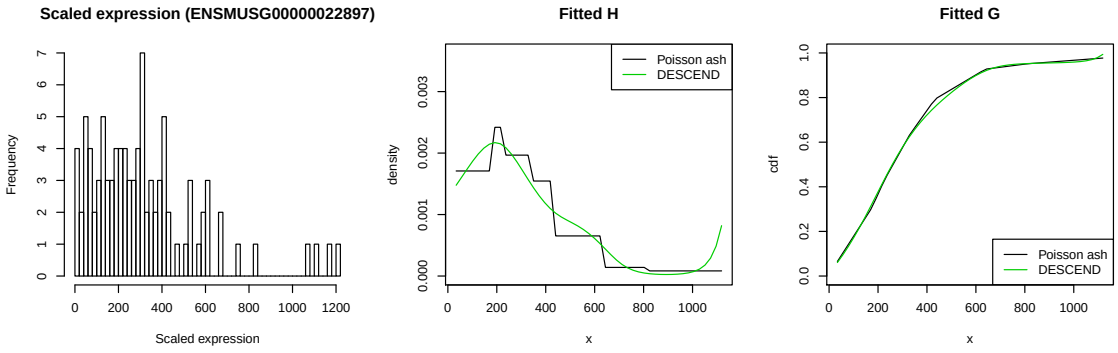

Figure 5.13: Scaled expression distribution, fitted expression distribution $\hat{G}_g$ by *DESCEND* and Poisson *ash* of gene *Dyrk1a*, which has 4 *Mclust* modes for log expression.

distribution to be an flexible exponential family distribution (with natural spline basis). We also propose the methods *ZINB*, Poisson *ash* and nonparametric deconvolution with different assumptions on the expression distribution: besides the point mass at zero to account for burstiness in scRNA-Seq data, *ZINB* assumes the active cell expression distribution to be a single gamma distribution; Poisson *ash* only requires the active cell expression distribution to be unimodal; and nonparametric deconvolution does not even have specific distributional assumptions. The nonparametric deconvolution method is the most flexible one among the four methods, while *ZINB* is the method with the most strict constraints. On the other hand, an overly flexible method may have overfitting issues due to the noise present in data. The ideal method would strike a balance between adaptivity and robustness. The relative flexible assumptions of *DESCEND* or Poisson *ash* make them adaptive to a wide range of data, but the constraints (exponential family for the former, unimodality for the latter) also prevent them from overfitting to some extent.

We compared the performances of the four methods on real scRNA-Seq datasets Zeisel data and Tung data, and saw that the four deconvolution methods *ZINB*, *DESCEND*, Poisson *ash* and nonparametric deconvolution typically produce nearly identical means for the fitted expression distribution $\hat{G}_g$. The coefficient of variations of $\hat{G}_g$ can differ at times, but are generally similar across the methods when there are sufficient active cells to well estimate the expression distribution. Nevertheless, it is difficult to accurately estimate the null proportion (fraction of inactive cells) no matter which method is used, because accurately separating out noise from true signals on extremely low expressed genes is nearly infeasible.

Even though some summary properties (mean, CV) are relatively consistent across the methods, the fitted expression distribution $\hat{G}_g$ itself can look noticeably different. In theory, the nonparametric deconvolution model should always achieve the highest log-likelihood. In practice however, we discover that the log-likelihood of Poisson *ash* model is often comparable to that of nonparametric deconvolution, and occasionally much higher than that of *ZINB* or *DESCEND*. Although likelihood is an intuitive criterion to evaluate the goodness of fit of the model, it neglects the potential overfitting possibility and the justification of model based on domain knowledge. In applications, we should carefully examine the fitted distributions by different methods and choose the one with proper distributional assumptions and reasonable results, depending on context. For example, there are some cases in Zeisel data where *DESCEND* achieves higher log-likelihood than Poisson *ash*, yet the expression distribution does not tend to be multimodal.

Note that for the expression distribution of active cells H_g, the four methods do not make strong assumptions that rely on the biological nature of scRNA-Seq data. *ZINB* uses the conjugate prior gamma distribution for computational convenience, while *DESCEND* and Poisson *ash* use flexible distribution families (exponential family, unimodal family) to guarantee adaptivity. However, the biological backgrounds and properties of the scRNA-Seq dataset can be considered during the model selection procedure. For example, on Zeisel data we find that the sub-population structure due to different cell types actually contributes to differences in distribution tail lengths, rather than multimodality corresponding to cell type groups. In this case, the Poisson *ash* model with uniform mixture components in various widths

would naturally be suited, and also has better interpretability.

References

Anders, S. and W. Huber (2010). Differential expression analysis for sequence count data. *Genome biol 11*(10), R106. 2, 26, 32

Bacher, R. and C. Kendziorski (2016). Design and computational analysis of single-cell RNA-sequencing experiments. *Genome biology 17*(1), 63. 105, 106, 108

Baldi, P. and A. D. Long (2001). A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics 17*(6), 509–519. 2, 5

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 289–300. 14, 26, 42

Brennecke, P., S. Anders, J. K. Kim, A. A. Kołodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni, et al. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature methods 10*(11), 1093. 82

Broberg, P. et al. (2003). Statistical methods for ranking differentially expressed genes. *Genome Biol 4*(6), R41. 5

Buettner, F., K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology 33*(2), 155. 81, 108

Burrows, C. K., N. E. Banovich, B. J. Pavlovic, K. Patterson, I. G. Romero, J. K. Pritchard, and Y. Gilad (2016). Genetic variation, not cell type of origin, underlies the majority of identifiable regulatory differences in iPSCs. *PLoS genetics 12*(1), e1005793. 71, 72, 73

Chubb, J. R., T. Trcek, S. M. Shenoy, and R. H. Singer (2006). Transcriptional pulsing of a developmental gene. *Current biology 16*(10), 1018–1025. 82

Dar, R. D., B. S. Razooky, A. Singh, T. V. Trimeloni, J. M. McCollum, C. D. Cox, M. L. Simpson, and L. S. Weinberger (2012). Transcriptional burst frequency and burst size are equally modulated across the human genome. *Proceedings of the National Academy of Sciences 109*(43), 17454–17459. 82

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38. 10

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association 99*(465), 96–104. 28, 34, 35

Efron, B. (2007). Correlation and large-scale simultaneous significance testing. *Journal of the American Statistical Association 102*(477), 93–103. 34

Efron, B. (2010). Correlated z-values and the accuracy of large-scale statistical estimates. *Journal of the American Statistical Association 105*(491), 1042–1055. 34

Efron, B. (2016). Empirical Bayes deconvolution estimates. *Biometrika 103*(1), 1–20. 80, 84

Efron, B., R. Tibshirani, J. D. Storey, and V. Tusher (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American statistical association 96*(456), 1151–1160. 2, 5

Gagnon-Bartsch, J. A. and T. P. Speed (2012). Using control genes to correct for unwanted variation in microarray data. *Biostatistics 13*(3), 539–552. 28, 34

Grün, D., L. Kester, and A. Van Oudenaarden (2014). Validation of noise models for single-cell transcriptomics. *Nature methods 11*(6), 637. 82

James, W. and C. Stein (1961). Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 361–379. 1

Kærn, M., T. C. Elston, W. J. Blake, and J. J. Collins (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics 6*(6), 451. 79

Kim, J. K., A. A. Kolodziejczyk, T. Ilicic, S. A. Teichmann, and J. C. Marioni (2015). Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature communications 6*, 8687. 82

Kim, J. K. and J. C. Marioni (2013). Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome biology 14*(1), R7. 82

Kivioja, T., A. Vähärautio, K. Karlsson, M. Bonke, M. Enge, S. Linnarsson, and J. Taipale (2012). Counting absolute numbers of molecules using unique molecular identifiers. *Nature methods 9*(1), 72. 80

Klein, A. M., L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell 161*(5), 1187–1201. 79

Koenker, R. and I. Mizera (2014). Convex optimization, shape constraints, compound decisions, and empirical Bayes rules. *Journal of the American Statistical Association 109*(506), 674–685. 87

Law, C. W., Y. Chen, W. Shi, and G. K. Smyth (2014). Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol 15*(2), R29. 22, 27, 32, 39

Leek, J. T., W. E. Johnson, H. S. Parker, A. E. Jaffe, and J. D. Storey (2012). The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics 28*(6), 882–883. 28, 34

Leek, J. T. and J. D. Storey (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet 3*(9), e161. 28, 34

Lönnstedt, I. and T. Speed (2002). Replicated microarray data. *Statistica sinica*, 31–46. 2, 5

Lonsdale, J., J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young, et al. (2013). The genotype-tissue expression (GTEx) project. *Nature genetics 45*(6), 580–585. 22, 38

Love, M. I., W. Huber, and S. Anders (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology 15*(12), 1–21. 26

Lu, M. and M. Stephens (2016). Variance adaptive shrinkage (vash): flexible empirical Bayes estimation of variances. *Bioinformatics*, btw483. 26, 27, 31

Marioni, J. C., C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research 18*(9), 1509–1517. 32

Murie, C., O. Woody, A. Y. Lee, and R. Nadon (2009). Comparison of small n statistical tests of differential expression applied to microarrays. *BMC bioinformatics 10*(1), 1. 5

Phipson, B., S. Lee, I. J. Majewski, W. S. Alexander, and G. K. Smyth (2016). Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression. *The annals of applied statistics 10*(2), 946. 6

Raj, A., C. S. Peskin, D. Tranchina, D. Y. Vargas, and S. Tyagi (2006). Stochastic mRNA synthesis in mammalian cells. *PLoS biology 4*(10), e309. 82

Risso, D., J. Ngai, T. P. Speed, and S. Dudoit (2014). Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology 32*(9), 896–902. 28, 34

Robbins, H. (1985). An empirical Bayes approach to statistics. In *Herbert Robbins Selected Papers*, pp. 41–47. Springer. 1

Robinson, M. D., D. J. McCarthy, and G. K. Smyth (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics 26*(1), 139–140. 2, 26, 32

Rocke, D. M., L. Ruan, J. J. Gossett, B. Durbin-Johnson, and S. Aviran (2015). Controlling false positive rates in methods for differential gene expression analysis using RNA-Seq data. *bioRxiv*, 018739. 4, 33, 36, 38

Schwartzman, A. (2010). Comment. *Journal of the American Statistical Association 105*(491), 1059–1063. 35

Shaffer, S. M., M. C. Dunagin, S. R. Torborg, E. A. Torre, B. Emert, C. Krepler, M. Beqiri, K. Sproesser, P. A. Brafford, M. Xiao, et al. (2017). Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature 546*(7658), 431. 79

Shalek, A. K., R. Satija, J. Shuga, J. J. Trombetta, D. Gennert, D. Lu, P. Chen, R. S. Gertner, J. T. Gaublomme, N. Yosef, et al. (2014). Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature 510*(7505), 363. 79

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology 3*(1), 3. 2, 5, 6, 7, 9, 11, 12, 22, 26, 31, 68

Stein, C. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Technical report, STANFORD UNIVERSITY STANFORD United States. 1

Stephens, M. (2016). False discovery rates: a new deal. *Biostatistics*, kxw041. 2, 6, 8, 27, 28, 30, 31, 43, 48, 62, 63, 66, 85

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(3), 479–498. 14, 26, 42

Tung, P.-Y., J. D. Blischak, C. J. Hsiao, D. A. Knowles, J. E. Burnett, J. K. Pritchard, and Y. Gilad (2017). Batch effects and the effective design of single-cell gene expression studies. *Scientific reports 7*, 39921. 75, 81, 107

Tusher, V. G., R. Tibshirani, and G. Chu (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences 98*(9), 5116–5121. 5

Varadhan, R. (2010). SQUAREM: Squared extrapolation methods for accelerating fixed-point iterations. *URL http://CRAN. R-project. org/package= SQUAREM, R package version 1*, 12. 10

Wang, J., M. Huang, E. Torre, H. Dueck, S. Shaffer, J. Murray, A. Raj, M. Li, and N. R. Zhang (2017). Gene expression distribution deconvolution in single cell RNA sequencing. *bioRxiv*, 227033. 80, 84, 109

Wu, H., C. Wang, and Z. Wu (2013). A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. *Biostatistics 14*(2), 232–243. 26

Zeisel, A., A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science 347*(6226), 1138–1142. 79, 81, 88

Appendix A

A.1 Algorithm to estimate hyper-parameters in *vash*

Details of the algorithm used to estimate hyper-parameters c and π:

To maximize the likelihood (2.12) we iteratively update c and π using the following steps until they converge:

$$c^{new} = \arg\max_{c} \log L(c, a_1^{old}, ...a_K^{old}, \pi_1^{old}, ..., \pi_K^{old}) \tag{A.1}$$

$$(\pi_1^{new}, ..., \pi_K^{new}) = \left(\frac{\sum_g \tilde{\pi}_{g1}^{old}}{\sum_{g'} \sum_{k'} \tilde{\pi}_{g'k'}^{old}}, ..., \frac{\sum_g \tilde{\pi}_{gK}^{old}}{\sum_{g'} \sum_{k'} \tilde{\pi}_{g'k'}^{old}} \right). \tag{A.2}$$