2017

Statistical Methods For Whole Transcriptome Sequencing: From Bulk Tissue To Single Cells

Cheng Jia
University of Pennsylvania, cheng.jia@outlook.com

Statistical Methods For Whole Transcriptome Sequencing: From Bulk Tissue To Single Cells

Abstract
RNA-Sequencing (RNA-Seq) has enabled detailed unbiased profiling of whole transcriptomes with incredible throughput. Recent technological breakthroughs have pushed back the frontiers of RNA expression measurement to single-cell level (scRNA-Seq). With both bulk and single-cell RNA-Seq analyses, modeling of the noise structure embedded in the data is crucial for draw- ing correct inference. In this dissertation, I developed a series of statistical methods to account for the technical variations specific in RNA-Seq experiments in the context of isoform- or gene- level differential expression analyses. In the first part of my dissertation, I developed MetaDiff (https://github.com/jiach/MetaDiff), a random-effects meta-regression model, that allows the incorporation of uncertainty in isoform expression estimation in isoform differential expression anal- ysis. This framework was further extended to detect splicing quantitative trait loci with RNA-Seq data. In the second part of my dissertation, I developed TASC (Toolkit for Analysis of Single-Cell data; https://github.com/scrna-seq/TASC), a hierarchical mixture model, to explicitly adjust for cell-to-cell technical differences in scRNA-Seq analysis using an empirical Bayes approach. This framework can be adapted to perform differential gene expression analysis. In the third part of my dissertation, I developed, TASC-B, a method extended from TASC to model transcriptional bursting- induced zero-inflation. This model can identify and test for the difference in the level of transcrip- tional bursting. Compared to existing methods, these new tools that I developed have been shown to better control the false discovery rate in situations where technical noise cannot be ignored. They also display superior power in both our simulation studies and real world applications.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Epidemiology & Biostatistics

First Advisor
Mingyao Li

Second Advisor
Hongzhe Li

Keywords
differential expression, hierarchical model, RNA-Seq, single-cell, transcriptional bursting

STATISTICAL METHODS FOR WHOLE TRANSCRIPTOME SEQUENCING: FROM BULK TISSUE TO SINGLE CELLS

Cheng Jia

A DISSERTATION

in

Epidemiology and Biostatistics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

Supervisor of Dissertation

_____

Mingyao Li

Associate Professor of Biostatistics

Graduate Group Chairperson

_____

Nandita Mitra, Professor of Biostatistics

Dissertation Committee

Hongzhe Li, Professor of Biostatistics

Nancy Zhang, Associate Professor of Statistics

Blanca Himes, Assistant Professor of Informatics

STATISTICAL METHODS FOR WHOLE TRANSCRIPTOME SEQUENCING:
FROM BULK TISSUE TO SINGLE CELLS

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisor Dr. Mingyao Li for her continuous support of my Ph.D study, for her motivation, guidance and immense knowledge. She has been absolutely indispensable for my research and writing of this thesis, and one cannot ask for a better mentor and advisor.

In addition to my advisor, I would like to thank the rest of my dissertation committee: Dr. Nancy Zhang, Dr. Blanca Himes, and Dr. Hongzhe Li, for their insightful comments and consultation, without which this thesis would not be possible.

Last but not least, I would like to thank my family and friends for their unwavering support throughout my Ph.D. study and my life in general.

ABSTRACT

STATISTICAL METHODS FOR WHOLE TRANSCRIPTOME SEQUENCING:
FROM BULK TISSUE TO SINGLE CELLS

Cheng Jia

Mingyao Li

RNA-Sequencing (RNA-Seq) has enabled detailed unbiased profiling of whole transcriptomes with incredible throughput. Recent technological breakthroughs have pushed back the frontiers of RNA expression measurement to single-cell level (scRNA-Seq). With both bulk and single-cell RNA-Seq analyses, modeling of the noise structure embedded in the data is crucial for drawing correct inference. In this dissertation, I developed a series of statistical methods to account for the technical variations specific in RNA-Seq experiments in the context of isoform- or gene-level differential expression analyses. In the first part of my dissertation, I developed MetaDiff (`https://github.com/jiach/MetaDiff`), a random-effects meta-regression model, that allows the incorporation of uncertainty in isoform expression estimation in isoform differential expression analysis. This framework was further extended to detect splicing quantitative trait loci with RNA-Seq data. In the second part of my dissertation, I developed TASC (Toolkit for Analysis of Single-Cell data; `https://github.com/scrna-seq/TASC`), a hierarchical mixture model, to explicitly adjust for cell-to-cell technical differences in scRNA-Seq analysis using an empirical Bayes approach. This framework can be adapted to perform differential gene expression analysis. In the third part of my dissertation, I developed, TASC-B, a method extended from TASC to model transcriptional bursting-induced zero-inflation. This model can identify and test for the difference in the level of transcriptional bursting. Compared to existing methods, these new tools that I developed have been shown to better control the false discovery rate in situations where technical noise cannot be ignored. They also display superior power in both our simulation studies and real world applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1. RNA

The central dogma of biology describes the flow of genetic information from deoxyribonucleic acid (DNA) to ribonucleic acid (RNA), and subsequently from RNA to protein. As a courier from genomic DNA to cellular protein synthesis machinery (ribosomes), RNA ultimately affects the phenotypes of all cellular organisms on planet earth.

Tremendous discrepancy in complexities and dynamic ranges of genome and protein exists. As an example, human genome contains approximately 21,000 distinct protein-coding genes, much less than previously estimated (Lander, 2011), while a recent survey of the human proteome discovered approximately 293,000 non-redundant peptides (Kim et al., 2014). In addition, human genome is generally diploid, with two copies of every gene present. The level of protein expression on the other hand has a large dynamic range from the most prevalent such as hemoglobins in red blood cells and insulin in pancreatic β cells, to the most rare (*e.g.*, certain transcription factors of which the expression needs to be carefully regulated to avoid malignant growth).

This discrepancy can be partly explained by RNA. Eukaryotic genes are organized on the genome as links of protein coding regions (exons) interwoven with non-coding sequences (introns), which are spliced out during post-transcriptional processing of pre-mRNA (Crick, 1979). Alternative splicing of pre-mRNA molecules can create an arbitrarily large number of isoforms coding for distinct protein products, by including different subsets of exons or even different parts of an exon, in the mature messenger RNA (mRNA), bridging the gap in structural complexities between DNA and protein. Many copies of RNA can be produced from a single genetic locus through transcription, and regulation of its speed directly affects the amount of mRNA present in the cell, thus influencing the level of the particular protein encoded by this mRNA. A large portion of quantitative and sequence complexity of human proteome is attributable to the transcription and alternative splicing (AS) of RNA.

Due to the critical roles played by RNA in cellular processes, mis-regulation of mRNA transcription or post-transcriptional modifications can be extremely detrimental. Therefore, identification of differentially expressed mRNA is a standard part of analysis when comparing controls and cases for any

disease. In addition, mis-regulation of alternative splicing is cause for numerous known diseases, such as β-thalassemia (Cao and Galanello, 2010), spinal muscular atrophy (Cho and Dreyfuss, 2010), etc.

1.2. RNA-Sequencing

The quest for accurately measuring the amount of specific mRNA molecules has witnessed several leaps of technology which has gradually allowed the scientific community to investigate more number of genes simultaneously with greater details and accuracy. Early methods such as Northern blot and real-time quantitative polymerase chain reaction (RT-qPCR) could only detect the expression of several candidate genes with known sequence at a time.

The advent of microarray technology enabled simultaneous quantification of thousands of putative transcripts. This brought about new fields of transcriptomic studies such as non-coding RNA, single nucleotide polymorphisms (SNPs) and alternative splicing events. Despite its popularity, microarray suffers from the following shortcomings. First, it fails to discover novel transcripts that have yet to be annotated. Second, it does not reveal the sequences of the molecules detected outside the hybridization probes. These caveats were eventually addressed by RNA-sequencing technology.

Massive parallel sequencing was first applied to measure mRNA concentration in 2007 (Weber et al., 2007). While early adopters primarily employed pyrosequencing technology (Sugarbaker et al., 2008; Torres et al., 2008; Weber et al., 2007), Illumina sequencing became immediately popular after its introduction (Marioni et al., 2008), and several methods (*e.g.* Bloom et al., 2009; Tang et al., 2009; Wang et al., 2010) were proposed to identify differentially expressed genes using the Illumina platform. Currently, the Illumina platform dominates the market. We henceforth define RNA-Seq (RNA Sequencing) as the technology to profile a cross-sectional snapshot of the presence and quantities of mRNAs in a specific transcriptome with massive parallel sequencing, including but not limited to Illumina sequencing.

1.2.1. Illumina Sequencing

Illumina systems employ a sequencing-by-synthesis method. The basic procedure of an Illumina sequencing experiment involves:

[i] Library preparation.

DNA molecules, either from directly extracted genomic DNA (DNA-Seq) or from cDNA reverse transcribed from a pool of purified mRNA (RNA-Seq), are fragmented and ligated with adapters on both end of the molecule.

[ii] Cluster amplification. The prepared library is loaded onto a chip, and the fragments hybridize to the surface of the chip through the adapters. Each bound fragment is then amplified into a clonal cluster. This step is necessary because current CCD technology is not able to capture the emission from one single fluorescent molecule. A cluster of fragments with identical sequences intensify the signal for it to be detectable by CCD.

[iii] Sequencing. Fluorescently labeled nucleotides are added on to the chip, and throughout the last round of synthesis, the nucleotides are allowed to be incorporated one at a time. High-sensitivity CCD is used to capture the wavelength of the emitted fluorescent signals after incorporation of each base, revealing the underlying sequence of each cluster.

After sequencing, the raw reads obtained will serve as the starting material for downstream computational analysis. Usually, the following steps are needed before any study of substance can be performed on the raw reads (Conesa et al., 2016).

[i] Quality assessment of raw reads. This step mainly aims to determine the overall quality of the sequencing protocol up to this point, by computing parameters such as sequence-specific quality scores, GC content, N content, distribution of sequence length, duplication levels, sequence over-representation and *k*-mer content, *etc* (Andrews, 2010). Reads or samples of lower quality should be excluded from downstream analyses to avoid biases introduced by sequencing artifacts.

[ii] Read mapping/transcriptome assembly. Alignment of reads on a reference genome or transcriptome must be completed before any other statistics can be computed. In the case of RNA-seq, reference transcriptome can be assembled *de novo*, or be derived from an annotated reference genome. After this step, another QC step is usually performed to control the quality of the assembled transcriptome, from parameters such as percentage of mapped reads, percentage of uniquely mapped reads, coverage uniformity, *etc*.

RNA-seq has a number of advantages over microarray:

[i] RNA-Seq is capable of sequencing the genes targeted to single-nucleotide resolution. This has effectively eliminated the cross-hybridization of transcripts with similar sequences, suffered by many array-based assays.

[ii] Microarray technology limits the researcher to detecting transcripts that have already been annotated. RNA-seq is more suitable for exploratory experiments that aim to discover novel transcripts and isoforms.

[iii] RNA-seq delivers higher dynamic range compared to microarray, and significant improvement on signal-to-noise ratio.

1.2.2. Applications of RNA-Seq

The information acquired through RNA-seq can be analyzed to generate complicated insights of the target transcriptomes, either by itself, or combined with corresponding genomic or proteomic data (Han et al., 2015). Some of the basic applications of RNA-seq include:

[i] Gene and transcript quantification. One of the most common applications of RNA-seq is to characterize the expression profile of the entire transcriptome in a sample of interest, by measuring the amount of all genes/transcripts in that sample. This process involves counting the number of reads mapped to each gene or transcript. While gene-level quantification is relatively straightforward, as genes are largely non-overlapping discrete genomic regions, transcript-level quantification on the other hand requires probabilistic modeling, for one read can be mapped to multiple transcripts from the same genetic locus. Tools used for transcript-level gene quantification include Cufflinks (Trapnell et al., 2012) RSEM (Li and Dewey, 2011) and PennSeq (Hu et al., 2013), *etc*.

[ii] Differential expression analysis. The power of RNA-seq can also be harnessed for comparing the transcriptomes of samples from two or more different pharmacological treatments, biological tissues, developmental stages or other grouping factors. Due to over-dispersion introduced in the sequencing protocol and non-canonical mean-variance curves, careful modeling of read counts is required to control the false positive rates in the discovery of differentially expressed genes using RNA-seq data. Many packages were developed in addressing this

issue, and a detailed review for these methods can be found in section 2.3.

[iii] Alternative splicing. RNA-seq provides the detailed sequences of the transcripts detected in addition to their quantities. This has made possible the investigation of the structural differences in transcriptomes, such as exon skipping, intron retention, alternative 3'/5' splice sites, *etc*. A detailed review of available methods for studying alternative splicing using RNA-seq data can be found in section 2.3.

1.2.3. Single-Cell RNA-Sequencing

Traditional bulk RNA-seq measures the average mRNA levels in the target cell populations, which can be heavily influenced by a relatively small proportion of cells that exhibit extreme expression for certain genes (Bengtsson et al., 2005). With the detection threshold of RNA-seq being pushed lower, profiling of entire transcriptomes on the single-cell level has been made possible, thus paving the way to characterizing gene expression heterogeneity among individual cells (Bacher and Kendziorski, 2016; Eberwine et al., 2014; Kolodziejczyk et al., 2015; Sandberg, 2014). Briefly, the protocol for single-cell RNA-seq usually includes the following steps:

[i] Cell capture. The first step of any single-cell RNA-seq protocol invariably involves isolating individual cells from the tissue or *in vitro* culture of interest. Several considerations need to be addressed in order to acquire a suitable sample for downstream procedures:

(a) Cell viability. The isolated cells must be minimally disturbed and highly viable in the final suspension.

(b) Sampling bias. The isolated cells must be a representative sample of the target tissue, without significant bias for any specific subpopulations.

Disassociation can be performed by either enzymatic digestion or mechanical techniques such as laser-capture micro-dissection(Emmert-Buck et al., 1996). Due to the potential difference in disassociation kinetics between different sources of tissues and cell types, biochemical digestion might introduce substantial sampling bias. Laser-capture micro-dissection on the other hand suffers from its low throughput as well as compromised cell viability. After disassociation, several approaches are available to further separate suspended cell clumps into individual cells, including serial dilution (Ham, 1965), micropipetting(Zong et al., 2012),

microwell dilution (Gole et al., 2013), optical tweezers (Landry et al., 2013) and FACS (Navin et al., 2011). Recently, microfluidics-based systems have become mainstream due to their commercial availability (White et al., 2011). Even higher throughput up to hundreds of thousands of cells per assay can be achieved through droplet-based automanipulation methods (Macosko et al., 2015).

[ii] Reverse transcription and pre-amplification. Isolated cells are subsequently lysed with surfactant in their own individual containers. For experiments performed with microfluidics, cell membranes are bursted with surfactant added to the integrated fluidic circuits (IFCs). For droplet-based systems, the cells are lysed immediately after it is injected into a droplet, which contains surfactant in the enclosed solution. Reverse transcription, bar-coding and pre-amplification is then performed *in situ*. Reverse transcription produces complementary DNA from the RNA released from lysed cells. Usually only a minute amount of cDNA can be produced from a single cell, hence the fidelity and efficiency of pre-amplification is vital to the quality of the RNA-seq data.

Many different protocols exist for the procedures following cell lysis, employing distinctive sets of enzymes and varying choices of reaction parameters. These can be roughly categorized into two classes according to their choice of pre-amplification method. SmartSeq, and its updated version SmartSeq2, STRT-seq, the Tang protocol, and SC3-seq use polymerase chain reaction (PCR), which could result in nonlinear amplification. CEL-Seq and MARS-Seq on the other hand choose IVT (in vitro transcription), which in theory linearly amplifies the cDNA, however, IVT could lead to additional 3'-end coverage biases due to the extra reverse transcription step involved.

Protocols like SMART-seq and SMART-seq2 achieve relatively uniform coverage of the entire transcript, which is ideal for discovering novel isoforms and studying structural variants in the transcriptome. Protocols like CEL-seq and STRT-seq focus on tag-counting, generating reads covering only a portion of the transcript. The latter is capable of incorporating bar codes such as unique molecule identifiers (UMIs) to directly measure the number of RNA molecules (Grün and Oudenaarden, 2015).

[iii] Library preparation and sequencing. Amplified cDNA is then subject to fragmentation, sequencing-specific barcoding, and other steps of library preparation. This step is identical to bulk RNA-

seq, except for the multiplexing of cells prior to library preparation in order to increase throughput.

1.3. Transcriptional Bursting

Profiling the cell-to-cell heterogeneity in gene expression has enabled a plethora of new venues of research for molecular biologists. One of the most classic, also the most important areas of study, is the investigation of transcriptional dynamics in cellular systems, which is essential for understanding how gene activity is regulated, thus answering the most fundamental question in molecular biology. From bacteria and yeast to mammalian cells (Blake et al., 2006; Chong et al., 2014; Fukaya, Lim, and Levine, 2016; Suter et al., 2011), transcriptional bursting has been reported in a wide range of organisms spanning the evolutionary tree.

There has been overwhelming evidence that gene transcription is discrete, occurring in bursts of activity between intervals of inactivity. (Chubb et al., 2006; Raj et al., 2006). Two models arise from the observed data to describe these transcriptional fluctuations, one-state and two-state models. In the one-state model, transcription is a Poisson process with a constant mean. This produces a somewhat uniform distribution of transcripts in the cell population (Zenklusen, Larson, and Singer, 2008). While in the two-state model, the cells randomly switch between the “on” and “off” states of transcription for a specific gene. This produces a distribution of mRNA counts with higher variance and inflated zeros.

Characterizing this dynamic process relies on real-time accurate measurement of mRNA *in vivo*. Prior to the popularization of scRNA-seq technology, one must rely upon microscopic imaging to study transcriptional dynamics, using reporter assays (Fiering et al., 1990) or fluorescence in situ hybridization (FISH) (Femino et al., 1998). These technologies, while vital for directly observing the transcriptional dynamics, also suffer from some serious disadvantages.

Reporter assays monitor the expression pattern of artificially constructed protein products with a short half-life such as green fluorescent protein (GFP). Levels of transcription activity is indirectly inferred from the level of enzymatic activity (or fluorescent intensity in the case of GFP) of the reporter protein. This has enabled real-time observation of the transcriptional activity *in vivo*. However, several caveats of this method include:

- Reporter protein levels are affected by factors additional to rate of transcription, such as rate of

translation and protein folding, rate of protein degradation, threshold and sensitivity of imaging *etc*.

- Some reporter protein such as GFP is shown to be toxic to certain cellular functions when over-expressed (Ansari et al., 2016), which could potentially alter the transcriptional activity of the promoters or enhancers under observation.

- Reporter protein is different from the natural product of the targeted promoter or enhancer. It may not perfectly recapitulate the dynamics of transcriptional activity due to the lack of negative feedback from the natural product.

- It is infeasible to scale the reporter assay to tens of thousands of genomic sites simultaneously.

FISH directly observe the amount of transcripts by measuring the intensity of fluorescent signals emitted by probes hybridized to specific target sequences (Raj et al., 2006). Live-cell imaging using alternative methods of probe delivery and live-cell-compatible probes has also been made possible for continuously monitoring the dynamics of gene transcription (Martin and Ephrussi, 2009; Santangelo et al., 2009; Tyagi, 2009). Similar issues of throughput and scale also plague FISH experiments.

Single-cell RNA-seq is a promising new technology to study transcriptional bursting. In 2013, Kim and Marioni, 2013 first investigated the possibility of inferring kinetic parameters of gene transcription by fitting a Beta-Poisson model with a Gibbs sampler. However, this method failed to address the technical noise intrinsic in scRNA-seq data. The authors also failed to incorporate any testing procedures, in order to compare the differential bursting properties across biological conditions.

CHAPTER 2

COMPUTATIONAL TOOLS FOR BULK RNA-SEQ

2.1. Differential Expression: Genes

Methods of detecting differential expression from bulk RNA-Seq data can be roughly classified into the gene-centric and isoform-centric methods. In terms of statistical modeling, two major patterns of approaches arose in the past seven years, count-based methods and read-based methods. Count-based methods model the number of reads mapped to each feature, while read-based methods consider the mapping ambiguity of each read by building a likelihood model that takes the raw reads as input.

2.1.1. Naïve Count-based Methods

Several methods proposed at the early stage of the RNA-Seq technology were naïve count-based methods reliant on certain stringent assumptions of the underlying distribution of the RNA-Seq sampling process. For example, Bloom et al. applied Fisher's exact test on a 2×2 table whose rows represent the experimental conditions, and whose columns correspond to the numbers of reads that fall within and outside the open reading frame of the targeted gene (Bloom et al., 2009). Marioni et al. modeled the number of reads from gene j, sample i and lane k as a Poisson random variable with the rate parameter λ_{ijk} (Marioni et al., 2008), and exploited the LRTs for the standard generalized linear models to test the hypotheses

$$
\begin{aligned}
H_0 &: \lambda_{ijk} = \lambda_j \\
H_1 &: \lambda_{ijk} = \lambda_j^A, \text{ for Group A} \\
&\quad \lambda_{ijk} = \lambda_j^B, \text{ for Group B}
\end{aligned}
$$

DEGSeq

Based on similar assumptions, DEGSeq took a slightly circuitous route, by deriving the conditional

distribution of M given A, defined as

$$M = \log_2 C_1 - \log_2 C_2$$
$$A = \frac{\log_2 C_1 + \log_2 C_2}{2}$$

where C_i is the number of reads that are mapped to a specific gene in sample i. Using $\delta-$method repeatedly, the conditional distribution of M given A can be approximated with a Normal distribution. A simple Z-test was proposed to test the differential expression of a specific gene by testing the null hypothesis $E(M|A) = 0$. DEGSeq requires restrictive, often impractical assumptions, such as the normality of the distribution of $M|A$. Moreover, DEGSeq failed to handle the over-dispersion of the RNA-Seq data.

These early methods suffered from several caveats, the most prominent of which was the assumptions with regards to the distribution of the read counts. All of them assume that the RNA-Seq read counts follow a binomial distribution or Hypergeometric distribution, which is subsequently approximated by a Poisson distribution. These assumptions are rarely satisfied in real-life practice, and ergo can potentially lead to Type I error inflation (Anders and Huber, 2010). Since then, much effort has been invested in building powerful and flexible statistical models that fit the RNA-Seq data more appropriately.

2.1.2. GLM Modeling of Counts

edgeR

GLMs (generalized linear models) incorporating over-dispersion parameters and additional linear covariates are a natural extension to the naïve count-based methods. Many methods were proposed in the framework of negative-binomial modeling, such as edgeR, baySeq, DESeq, ShrinkSeq, etc. In their method edgeR, Robinson and Smyth extended the Poisson model to a negative-binomial model with a common dispersion factor across all genes (Robinson and Smyth, 2008). Define the number of counts mapped to a certain gene in condition i and sample j as Y_{ij}, then this

random variable follows a negative-binomial distribution parametrized as

$$\mu_{ij} = m_{ij}\lambda_i$$
$$Y_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi)$$
$$\mathrm{E}(Y_{ij}) = \mu_{ij}$$
$$\mathrm{Var}(Y_{ij}) = \mu_{ij}(1 + \mu_{ij}\phi)$$

where m_{ij} denotes the library size of sample j, and λ_i denotes the true relative abundance of reads mapped to the target gene. The dispersion parameter ϕ is estimated with conditional maximum likelihood, and testing for differential expression events is equivalent to testing the following hypotheses:

$$H_0 : \lambda_1 = \lambda_2$$
$$H_1 : \lambda_1 \neq \lambda_2$$

McCarthy and Smyth later modified the original edgeR model with the common dispersion parameters, introducing an empirical Bayes shrinkage estimate using a weighted conditional maximum log-likelihood method. This new approach shrinks the dispersion estimates towards a locally common prior instead of setting them to a common value (McCarthy, Chen, and Smyth, 2012). Using Cox-Reid adjusted profile likelihood (APL, which was later used in DESeq2),

$$\mathrm{APL}_g(\phi_g) = \ell\left(\phi_g; \mathbf{y}_g, \hat{\beta}_g\right) - \frac{1}{2}\log\det\mathcal{I}_g$$
$$\mathcal{I}_g = X^T W X$$

McCarthy and Smyth defined the APL of G genes, among which the dispersion parameter is shared.

$$\mathrm{APL}_S(\phi) = \frac{1}{G}\sum_{g=1}^{G}\mathrm{APL}_g(\phi)$$

In order to shrink the entirely individual gene-wise dispersion parameter ϕ_g towards a local shared dispersion parameter by maximizing the following weighted APL.

$$\mathrm{APL}_g(\phi_g) + G_0 \mathrm{APL}_{Sg}(\phi_g) \tag{2.1}$$

The number G_0 is a tuning parameter indicating the number of genes used to generate the prior, *i.e.*, the local shared dispersion parameter.

This trended-by-mean estimate of the dispersion parameter borrows information across a few genes that have similar mean expression level. Later Zhou et al. from the same group robustified the model by iteratively reweighting the Poisson residuals as well as the dispersion estimation by the deviation of an observation from the current fit. The Pearson residuals of an observed count y_{gi} from the NB GLM can be computed at the end of each iteration:

$$r_{gi} = \frac{y_{gi} - \hat{\mu}_{gi}}{\sqrt{\hat{\mu}_{gi}\left(1 + \hat{\phi}_g \hat{\mu}_{gi}\right)}}$$

where $\hat{\mu}_{gi}$ is the fitted value calculated from $\hat{\beta}$, and $\hat{\phi}$ is the estimated dispersion parameter with the above trended-by-mean method. The Pearson residuals are converted to weights using a Huber function:

$$w_{gi} = w(r_{gi}) = \begin{cases} \frac{k}{|r_{gi}|} & \text{for } |r_{gi}| > k \\ 1 & \text{for } |r_{gi}| \leqslant k \end{cases}$$

The new $\hat{\beta}$ can be estimated with the above weights,

$$\beta_g^{\text{new}} = \beta_g^{\text{old}} + \left[X^T (W_g \Sigma_g) X\right]^{-1} X^T W_g z_g$$
$$\mathrm{APL}_g^W(\phi_g) = \sum_i w_{gi} \ell\left(\phi_g; \mathbf{y}_g, \hat{\beta}_g^{\text{old}}\right) - \frac{1}{2} \log\det\left[X^T (W_g \Sigma_g) X\right]$$

And new $\hat{\phi}$ can be estimated by maximizing the linearly weighted APL as in (2.1) based on the weighted APL_g^W. This iterated weighting algorithm increases the robustness of the edgeR method by down-weighting the observations greatly deviant from the fitted model, forfeiting some power to control the false discovery rate in extreme cases.

DESeq

The negative-binomial model was expanded by Anders and Huber in DESeq (Anders and Huber, 2010), by allowing the dispersion parameter to be a smooth function of the mean, effectively allowing variation of the dispersion parameters for genes with different means in different experimental groups.

$$Y_{ij} \sim \mathrm{NB}\left(\mu_{ij}, \sigma^2_{ij}\right)$$

Fitting the above model without constraining the number of parameters will not result in meaningful estimates, due to the usually small sample size of each group in RNA-Seq experiments. Consequently, the following assumptions were proposed by Anders and Huber,

[i] The mean parameter is the product of a condition-dependent value $q_{i,\rho(j)}$ (where $\rho(j)$ is the condition of sample j), and the library size s_j.

$$\mu_{ij} = q_{i,\rho(j)} s_j$$

This is the same as the assumption $\mu_{ij} = m_{ij}\lambda_i$ in Robinson and Smyth's method.

[ii] The variance σ^2_{ij} can be decomposed to the sum of two terms, a *shot noise term* (μ_{ij}) and a *raw variance term* ($s_j^2 v_{i,\rho(j)}$).

$$\sigma^2_{ij} = \mu_{ij} + s_j^2 v_{i,\rho(j)}$$

[iii] The parameter in the raw variance term can be written in the form of a smooth function of $q_{i,\rho(j)}$.

$$v_{i,\rho(j)} = v_\rho(q_{i,\rho(j)})$$

Assumptions [ii] and [iii] have effectively reduced the parameters and enabled pooling of the data from genes with similar expression strength for the variance estimation.

In order to eliminate the influence of a few highly expressed genes on the total number of reads, Anders and Huber suggested a read depth estimator normalized by the geometric mean of the

numbers of reads across all genes,

$$\hat{s}_j = \underset{i}{\text{median}} \left[\frac{k_{ij}}{\left(\prod_{\nu=1}^{m} k_{i\nu} \right)^{\frac{1}{m}}} \right]$$

$q_{i\rho}$ for gene i and condition ρ is estimated by the average of the normalized counts from the samples corresponding to condition ρ:

$$\hat{q}_{i\rho} = \frac{1}{m_\rho} \sum_{j:\rho(j)=\rho} \frac{k_{ij}}{\hat{s}_j}$$

where m_ρ is the number of replicates under condition ρ.

To estimate the smooth function v_ρ that links the $q_{i\rho}$ to the raw variance $s_j^2 v_{i,\rho(j)}$, first find an unbiased estimator of the raw variance parameter $v_{i\rho}$.

$$\begin{aligned}
\hat{v}_{i\rho} &= w_{i\rho} - z_{i\rho} \\
w_{i\rho} &= \frac{1}{m_\rho - 1} \sum_{j:\rho(j)=\rho} \left[\frac{k_{ij}}{\hat{s}_j} - \widehat{q_{i\rho}} \right]^2 \\
z_{i\rho} &= \frac{\hat{q}_{i\rho}}{m_\rho} \sum_{j:\rho(j)=\rho} \frac{1}{\hat{s}_j}
\end{aligned}$$

Local regression with a generalized linear model of the gamma family on $(\hat{q}_{i\rho}, w_{i\rho})$ was used to smooth out the curve and obtain the smooth function $w_\rho(q)$

$$\hat{v}_\rho\left(\hat{q}_{i\rho}\right) = w_\rho(\hat{q}_{i\rho}) - z_{i\rho}$$

The local regression step combines the information across genes to estimate the smooth function, $\hat{v}_\rho(q)$.

In order to test for the differential gene expression, define the joint distribution of $K_{iA} = \sum_{j:\rho(j)=A} K_{ij}$ and $K_{iB} = \sum_{j:\rho(j)=B} K_{ij}$ as $p(K_{iA} = a, K_{iB} = b)$. Denote their sum as $K_{iS} = K_{iA} + K_{iB}$, the author suggested the following p-value,

$$P_i = \frac{\sum_{a+b=k_{iS}, p(a,b) \leqslant p(k_{iA}, k_{iB})} p(a,b)}{\sum_{a+b=k_{iS}} p(a,b)}$$

Note: The joint distribution $p(a, b)$ is calculated as the product of two independent NB variables K_{iA} and K_{iB}, whose probability distribution is in turn obtained by matching the moments (mean and coefficient of variation),

$$\hat{q}_{i0} = \sum_{j:\rho(j)\in\{A,B\}} \frac{k_{ij}}{\hat{s}_j}$$

$$\hat{\mu}_{iA} = \sum_{j\in A} \hat{s}_j \hat{q}_{i0}$$

$$\hat{\mu}_{iB} = \sum_{j\in B} \hat{s}_j \hat{q}_{i0}$$

$$\hat{\sigma}^2_{iA} = \sum_{j\in A} \hat{s}_j \hat{q}_{i0} + \hat{s}^2_j \hat{v}_A(\hat{q}_{i0})$$

$$\hat{\sigma}^2_{iB} = \sum_{j\in B} \hat{s}_j \hat{q}_{i0} + \hat{s}^2_j \hat{v}_B(\hat{q}_{i0})$$

The approach employed by DESeq has several limitations in practice:

[i] The number of experimental groups is limited, because the joint distribution of $p(G_1, G_2, \ldots, G_n)$ is required to calculate the p-values. With the number of groups increasing, the complexity of estimating this joint distribution increases exponentially.

[ii] The model cannot incorporate continuous covariates. Estimates of $q_{i\rho}$ and certain other parameters are performed in a group-wise fashion, which requires the size of each group be larger than 1. Continuous variables need to be binned before they can be incorporated into this model.

[iii] For genes with the same mean expression level, the variance estimated in DESeq is identical. This assumption almost never holds in real-life practice. This problem was later rectified by an update in DESeq, in which the greater value between the empirical gene-wise dispersion and the mean-dependent fitted value was used as the final dispersion parameter. However, this approach introduced bias towards larger variance estimate, and as a consequence rendered DESeq too conservative (Love, Huber, and Anders, 2014).

DESeq2

To rectify some of the issues faced by DESeq, in 2014 Anders and Huber published DESeq2, which exploited an empirical Bayes method to estimate the dispersion parameters, and expanded

the original model using GLMs with a logarithmic link:

$$\log_2 q_{ij} = \sum_r x_{jr}\beta_{ir}$$

DESeq2 abandoned the practice of estimating $q_{i\rho}$ in a group-wise fashion, instead computing the q_{ij} for every sample and every gene. Consequently, an assortment of covariates can be integrated into this linear term, both discrete and continuous. In DESeq, the variance estimation is done in three steps. First, calculate the residual *raw variance* term for every gene in each group; second, for each group, fit a local regression with the raw variance $w_{i\rho}$ as the dependent variable and $q_{i\rho}$ as the independent variable; third, compute the raw variance of a gene in a certain group from the above curve, and add it to the *shot noise term* (a variance term linearly dependent on the mean value $q_i\rho$) to obtain the final variance estimate. This arbitrary approach does not model the stochasticity of the raw variance of each gene. In DESeq2, the above group-wise method is no longer adaptable. Therefore, a curve is fitted with $(\hat{q}i, \hat{v}_i)$, with $\hat{q}_i$ and $\hat{v}_i$ denoting the naïve MLE estimates of the mean and variance of gene i respectively. The final variance estimate is obtained by shrinking the naïve estimates of the variance towards the fitted curve with an empirical Bayes algorithm.

$$\alpha_i^{\text{MAP}} = \arg\max_\alpha \left[\ell_{\text{CR}}\left(\alpha; \vec{\mu}_{i\cdot}, \vec{K}_{i\cdot}\right) + \Lambda_i(\alpha)\right]$$

$$\Lambda_i(\alpha) = \frac{-\left[\log\alpha - \log\alpha_{\text{tr}}\left(\vec{\mu}_{i\cdot}\right)\right]^2}{2\sigma_d^2}$$

$$\sigma_d^2 = \max\{s_{lr}^2 - \psi_1\left[\frac{m-p}{2}\right], 0.25\}$$

$$s_{lr} = \underset{i}{\text{mad}}\{\log\alpha_i^{gw} - \log\alpha_{\text{tr}}(\overline{\mu}_i)\}$$

$$\alpha_i^{gw} = \arg\max_\alpha \ell_{\text{CR}}\left[\alpha; \vec{\mu}_{i\cdots}, \vec{K}_{i\cdots}\right]$$

$$\ell_{\text{CR}}\left[\alpha; \vec{\mu}, \vec{K}\right] = \ell(\alpha) - \frac{1}{2}\log\det\left[X^T W X\right]$$

$$\ell(\alpha) = \sum_j \log f_{\text{NB}}(K_j; \mu_j, \alpha)$$

$$\alpha_{\text{tr}}(\overline{\mu}) = \frac{a_1}{\overline{\mu}} + \alpha_0$$

where ℓ_{CR} is the Cox-Reid adjusted profile likelihood, as used in updated edgeR (McCarthy, Chen, and Smyth, 2012). Optimizing the sum of the logarithm of ℓ_{CR} and the prior log-likelihood (Λ_i) results in the Bayesian shrinkage of the dispersion parameter estimation to the prior.

Similar method is used to shrink the logarithmic fold change estimates:

$$\vec{\beta}_i = \arg\max_{\vec{\beta}} \left[\sum_j \log f_{NB} \left[K_{ij}; \mu_j \left(\vec{\beta} \right), \alpha_i \right] + \Lambda \left(\vec{\beta} \right) \right]$$

$$\mu_j \left(\vec{\beta} \right) = s_{ij} \exp \left[\sum_r x_{jr} \beta_r \right]$$

$$\Lambda \left(\vec{\beta} \right) = \sum_r -\frac{\beta_r^2}{2\sigma_r^2}$$

Compared to DESeq, DESeq2 is more focused on estimating the fold change between different conditions rather than the presence or absence of said change. Empirical Bayes shrinkage is also used to shrink the logarithmic fold change (LFC) associated with the given covariates. This results in MAP LFCs that are biased towards zero, which effectively removes the inflated LFCs for genes with low counts. The strength of the shrinkage depends on the mean count as well as the information available for the LFC estimation. This approach offered a more reproducible estimator than the naïve MLEs.

DSS

The modeling of over-dispersion is at the core of construction of a GLM model for RNA-Seq data. In addition to the approaches implemented in edgeR and DESeq, DSS (Wu, Wang, and Wu, 2013) provided a log-normal prior for dispersion parameter in the Gamma distribution of the Poisson-Gamma mixture. Wu *et al.* noted that there was no conjugate prior that would facilitate the computation of the posterior distribution, and a log-normal distribution better approximated the estimated ϕ_g in real RNA-Seq data.

$$Y_{gi} | \theta_{gi} \sim \text{Poisson} \left(\theta_{gi} s_i \right)$$

$$\theta_{gi} | \phi_{gi} \sim \text{Gamma} \left(\mu_{g,k(i)}, \phi_g \right)$$

$$\phi_g \sim \log - \text{normal} \left(m_0, \tau^2 \right)$$

Wu et al. arrived at the posterior distribution of ϕ_g given the data,

$$\begin{aligned}\log\left[p\left(\phi_g|Y_{gi},\nu_{gi},i=1,\ldots,n\right)\right] \propto & \sum_i \psi\left[\phi_g^{-1}+Y_{gi}\right]-n\psi\left[\phi_g^{-1}\right]-\phi_g^{-1}\sum_i \log\left[1+\nu_{gi}\phi_g\right] \\ & +\sum_i Y_{gi}\left[\log\left(\nu_{gi}\phi_g\right)-\log\left(1+\nu_{gi}\phi_g\right)\right] \\ & -\frac{\left[\log\left(\phi_g\right)-m_0\right]^2}{2\tau^2}-\log\left(\phi_g\right)-\log\left(\tau\right)\end{aligned}$$

where $\psi\left[\cdot\right]$ is the log gamma function and n is the number of samples. An MAP estimator $\tilde{\phi}_g$ can be computed by maximizing the above posterior distribution with the Newton-Raphson method, after replacing ν_{gi} with $\hat{\mu}_{g,k(i)}=\frac{\sum_{j:k(j)=k(i)}\frac{Y_{gj}}{s_j}}{n_{k(i)}}$, where $k(i)$ indicates the experimental group of the i^{th} sample, and $n_{k(i)}$ indicates the number of samples in that group. The hyperparameters m_0 and τ are estimated using an empirical Bayes method, from the MOM estimates of $\hat{\phi}_g$.

$$\begin{aligned}z_{gi} &\equiv \frac{Y_{gi}^2-Y_{gi}}{s_i^2} \\ \hat{\phi}_g &= \frac{\sum_i z_{gi}}{\sum_i \hat{\mu}_{g,k(i)}^2}-1 \\ \hat{m}_0 &= \underset{g}{\text{median}}\left[\log\left(\hat{\phi}_g\right)\right]\end{aligned}$$

Individually estimated $\hat{\phi}_g$ is crude for estimating the true ϕ_g, which is why it is only used in estimating the hyperparameters m_0 and τ. The MAP estimator $\tilde{\phi}_g$ achieves shrinkage and sharing of information across genes, by taking advantage of the estimated prior. After the model has been fitted, Wald test was chosen to test the null hypothesis, $\mu_{g,1}=\mu_{g,2}$.

BBSeq

Other than negative binomial distribution, a similar beta-binomial distribution was used by Zhou et al. in their BBSeq package (Zhou, Xia, and Wright, 2011). The linker function is a logit function connecting the covariates with θ in the binomial model.

$$\text{logit}\left[E\theta_{i\cdot}\right]=X\beta_i$$

And θ_{ij} follows a β-distribution parametrized such that its variance is $\phi_i E\left(\theta_{i\cdot}\right)\left[1-E\left(\theta_{i\cdot}\right)\right]$. Zhou et al. proposed two methods of estimating the dispersion parameters similar to edgeR and DESeq.

One method is to estimate each dispersion parameter individually for each gene. The other is to use a fitted curve to compute the dispersion parameter from the mean of the gene. BBSeq adopted a polynomial relationship,

$$\psi = \text{logit}\,(\phi) = \sum_{k=0}^{K} \gamma_k \left[\frac{1}{n} \sum_{i=1}^{n} XB_i \right]^k$$

In this curve fitting, Zhou et al. used the mean of the n sample XB_i as input, and the logit-transformed ϕ as output. This is similar to the treatment by DESeq, except that in DESeq the dispersion parameter is estimated in a group-wise fashion, where XB is identical for each group.

2.1.3. Read-based method

Cuffdiff

The methods so far are count-based methods, which assume that read alignment is completed, and the number of reads is fixed for all the regions under investigation. This assumption usually does not hold, as many reads cannot be unambiguously mapped to a certain region. Read-based method such as Cufflinks (Trapnell et al., 2010) took a different approach by building a likelihood model that incorporates the uncertainty of read assignment. With the notations listed in Table 2.1, the likelihood can be written as,

$$L(\rho|R) = \prod_{r \in R} \sum_{t \in T} \alpha_t \left[\frac{F(It(r))}{l(t) - I_t(r) + 1} \right]$$

where

$$\alpha_t = \frac{\rho_t \tilde{l}(t)}{\sum_{u \in T} \rho_u \tilde{l}(t)}$$

In order to decompose the likelihood into manageable components, define the following probabilities:

Notation	Parameter
T	All transcripts in a transcriptome
G	A maximal partial of transcripts into loci
R	Set of sequenced fragment from T
ρ_t	Proportion of transcript $t \in T$
σ_g	Proportion of transcripts in each locus $\sigma_g = \sum_{t \in g} \rho_t$
τ_t	Proportion of each transcript in each locus $\tau_t = \frac{\rho_t}{\sigma_g}$
$l(t)$	Length of transcript t
$l(S)$	Length of a collection of transcripts $S \subset T$ $l(S) = \frac{\sum_{t \in S} \rho_t l(t)}{\sum_{t \in S} \rho_t}$
$F(\cdot)$	PMF for the distribution of a fragment length. $\sum_{i=1}^{\infty} F(i) = 1$. Assumed to be normal.
$\tilde{l}(t)$	Adjusted length for transcript. $\tilde{l}(t) = \sum_{i=1}^{l(t)} F(i)\,[l(t) - i + 1]$
$\tilde{l}(S)$	Adjusted length for a group of transcripts. $\tilde{l}(S) = \frac{\sum_{t \in S} \rho_t \tilde{l}(t)}{\sum_{t \in S} \rho_t}$
$A_{R,T}$	$M \times \lvert T \rvert$ matrix with $A(r,t) = 1$ if r is compatible with t, and 0 otherwise
$I_t(r)$	If $A(r,t) = 1$, $I_t(r)$ is the length of fragment r implied by the map to t If $A(r,t) = 0$, $I_t(r) = \infty$ and $F(It(r)) = 0$

Table 2.1: Notations and Parameters in Trapnell et al., 2010

[i] The probability that a fragment originates from a transcript within a given locus g.

$$\beta_g = \frac{\sigma_g \tilde{l}(g)}{\sum_{h \in G} \sigma_h \tilde{l}(h)}$$

[ii] The probability of selecting a fragment from a single transcript t conditioned on selecting a transcript from the locus g in which t is contained.

$$\gamma_t = \frac{\tau_t \tilde{l}(t)}{\sum_{u \in g} \tau_u \tilde{l}(u)}$$

And by conditional probability, the above likelihood is rewritten as,

$$L(\rho|R) = \left[\prod_{g \in G} \beta_g^{X_g}\right] \left[\prod_{g \in G} \left[\prod_{r \in R : r \in g} \sum_{t \in g} \gamma_t \cdot \frac{F(I_t(r))}{l(t) - I_t(r) + 1}\right]\right]$$

The testing of differential expression for a specific locus g under conditions $C1$ and $C2$ can then be performed by testing whether the log-ratio of the MLE FPKM .

$$\log\left\{\left[\frac{10^9 X_g^{C1}\hat{\gamma}_t^{C1}}{\tilde{l}(t)M^{C1}}\right] / \left[\frac{10^9 X_g^{C2}\hat{\gamma}_t^{C2}}{\tilde{l}(t)M^{C2}}\right]\right\}$$

is different from 0. The above test statistic can be approximated by a normal distribution, whose variance is calculated with δ-method.

2.2. Differential Expression: Isoforms

Methods introduced above are focused on testing the differential expression on the gene level. This approach is relatively straight forward as genes are generally non-overlapping and assigning a read to a gene is usually more reliable than determining the source of a read from a specific isoform of said gene. Therefore, methods of detecting DE on the isoform level is quite different from the methods we've across so far.

EBSeq

Focused on testing the differential expression of an isoform, Leng et al. proposed an empirical Bayes hierarchical model EBSeq, which correlated the dispersion of the NB distribution parameters with the number of isoforms of the gene (Leng et al., 2013). Denote the number of reads mapped to isoform i of gene g in sample s under condition C as $X_{gi,s}^C$,

$$X_{gi,s}^C | r_{gi,0} l_s, q_{gi}^C \sim NB\left(r_{gi,0} l_s, q_{gi^C}\right)$$
$$q_{gi}^C | \alpha, \beta^{I_g} \sim Beta\left(\alpha, \beta^{I_g}\right)$$

Denote the prior predictive probability under the hypothesis of equivalent expression as $f_0^{I_g}(X)$, which is a form of NB density with hyper-parameters.

$$f_0^{I_g}(X_{gi}^{C1,C2}) = \left[\prod_{s=1}^{S}\binom{X_{gi,s}+r_{gi,s}-1}{X_{gi,s}}\right]\frac{Beta\left(\alpha+\sum_{s=1}^{S} r_{gi,s}, \beta^{I_g}+\sum_{s=1}^{S} X_{gi,s}\right)}{Beta\left(\alpha,\beta^{I_g}\right)}$$

And denote the PPP under the hypothesis of differential expression as $f_1^{I_g}(X)$, which is equal to,

$$f_1^{I_g}(X_{gi}^{C1,C2}) = f_0^{I_g}\left(X_{gi}^{C1}\right) f_0^{I_g}\left(X_{gi}^{C2}\right)$$

And let the latent Bernoulli variable z denotes the status of the isoform (EE vs DE), and let the prior of z be

$$z \sim \text{Bernoulli}(p)$$

The authors claim that the posterior probability of the isoform being DE is

$$\frac{p f_1^{I_g}(X_{gi}^{C1,C2})}{(1-p) f_0^{I_g}(X_{gi}^{C1,C2}) + p f_1^{I_g}(X_{gi}^{C1,C2})}$$

which can then be used to assess the significance of DE of an isoform. The means and variance of the above functions are estimated individually with MOM estimators, while hyper-parameters, α, β^{I_g} and p are estimated by maximizing the likelihood with EM algorithm.

One of the caveats of EBSeq is that it failed to incorporate estimation uncertainty for each gene/isoform in their model, despite the claims from the authors. The only parameter remotely related to estimation uncertainty is the β parameter in the prior, which can take on three different values depending on whether the target gene has 1, 2 or $\geqslant 3$ exons. This approach does not solve the problem of estimation uncertainty, and methods that explicitly model this phenomenon perform much better than EBSeq (Jia et al., 2015).

MMDIFF

In addition to directly model the counts like what EBSeq did, one could also take a two step approach. First, estimate the abundance of isoforms with a software package such as Cufflinks, MMSEQ, IsoEM, RSEM, etc. Second, perform the testing of isoform DE with the isoform expression known. One issue with this approach is that it requires the downstream model to account for the estimation uncertainty of isoform expression. For this purpose, Turro et al. implemented a Bayesian linear mixed effects model in their MMDIFF package (Turro, Astle, and Tavaré, 2014). The idea is to introduce a random component in the linear mixed effects model to represent the variation of the estimator for the mean expression of each isoform. MMDIFF takes the output of MMSEQ and runs an MCMC algorithm for Bayesian model selection (Carlin and Chib, 1995) to

select between the null and alternative model as well as estimate model parameters:

$$\text{Unsaturated: } y = \alpha^{(0)} + \mathbf{M}\beta^{(0)} + \mathbf{P}^{(0)}\eta^{(0)} + \nu^{(0)} + \epsilon^{(0)}$$
$$\text{Saturated: } y = \alpha^{(1)} + \mathbf{M}\beta^{(1)} + \mathbf{P}^{(1)}\eta^{(1)} + \nu^{(1)} + \epsilon^{(1)}$$

y denotes the log-transformed expression of a feature (gene, isoform, etc). $\mathbf{M}$ and β denotes the design matrix and the model-invariant parameters for which all models are required to adjust for, *e.g.* age, sex, etc. $\mathbf{P}$ and η denotes the design matrix and the model-dependent parameters. For example, if we are to compare the two models with the null excluding the group parameter, $P(0)$ would be set to 0, while $P(1)$ would be the group indicator. ν is the parameter describing the estimation uncertainty of y, which is proposed to follow a normal distribution with variance estimated by MMSEQ. ϵ is the biological variance of y, which can be parametrized to be either common across all samples, or common within experimental groups. In their simulations, the inclusion of the estimation variance ν improves both PPV and NPV, thus rendering the method more powerful with better controlled Type I error.

However, MMDIFF has several caveats that make it difficult to use. First, the use of Bayesian inference, while a statistically sound choice, complicates the comparison with other methods. In real-life practice, it outputs the posterior probability of the alternative model, without explicit control for the false discovery rate. Second, it only takes input from MMSEQ, therefore, the inference made by MMDIFF is limited by the accuracy of the isoform estimation by MMSEQ, resulting in failure to take advantage of more accurate estimation methods. Third, the use of MCMC severely affects its computational performance. In our simulation studies, MMDIFF could take a day to run a moderately sized sample.

2.3. Differential Alternative Splicing

Gene expression is not the whole story of the molecular regulation of a eukaryotic cell. It is known that different subset of exons from the same genetic loci can be concatenated to form different mRNAs (isoforms) from the same genomic loci, through a process named alternative splicing (AS). This process is biologically relevant, as well as highly regulated in response to the inter- and intra-cellular environment. Therefore, identification of differential alternative splicing (DAS) events was an essential part of a complete molecular profile. Fortunately, it has been made massively accessible

by the advent of RNA-Seq technology. RNA-Seq provides more detailed information regarding the sequences of the RNA molecules, which allows it to distinguish the signals from different isoforms of a gene.

The methods we've covered so far are mostly focused on whole-gene quantification, instead of DAS detection. The statistical models for these two tasks are generally quite different despite using the identical source of data. As noted in Shi and Jiang, 2013, the testing of DAS of a gene is mathematically a two-layer nested question. Assume the gene of interest has p isoforms, and under condition C the expression for isoform i is μ_i^C, and the proportion of this isoform among all transcripts is $\theta_i^C = \frac{\mu_i^C}{\sum_{i=1}^{p} \mu_i^C}$. The first layer is to test for the differential expression of all the isoforms of this gene.

$$H_0 : \mu_i^{C_1} = \mu_i^{C_2} \text{ for all } i = 1, \ldots, p$$
$$H_1 : \mu_i^{C_1} \neq \mu_i^{C_2} \text{ for at least one } i = 1, \ldots, p$$

If the first test is rejected, *i.e.*, there is at least one isoform in this gene that has shown patterns of differential expression, then we invoke the second layer, to test for the differential alternative splicing of this gene.

$$H_0 : \theta_i^{C_1} = \theta_i^{C_2} \text{ for all } i = 1, \ldots, p$$
$$H_1 : \theta_i^{C_1} \neq \theta_i^{C_2} \text{ for at least one } i = 1, \ldots, p$$

If a gene is tested positive for DAS, then at least one of its isoforms must show DE, which is why these two tests are nested. Due to this complexity of the problem, the mathematical strategies for detecting DAS are much more diverse compared to DE analysis. Roughly, they can be categorized into the following three classes:

[i] Testing of differential exon usage using a Bayesian hierarchical model, *e.g.* MISO (Katz et al., 2010) and MATS (Shen et al., 2012);

[ii] Direct modeling of the counts data with GLM, *e.g.* DEXSeq (Anders, Reyes, and Huber, 2012) and rSeqDiff (Shi and Jiang, 2013);

[iii] Other miscellaneous methods, *e.g.* DSGSeq (Wang et al., 2013), SplicingCompass (Aschoff

et al., 2013) and IUTA (Niu et al., 2014).

2.3.1. Testing of differential exon usage

MISO

MISO (Katz et al., 2010) is the pioneering method in Bayesian hierarchical modeling for testing of DAS events from RNA-Seq data. From the same framework, MISO devised two types of analyses, exon-centric, which tests for the differential usage of one exon at a time, and isoform-centric, which tests for changes of isoform proportions. Denote the relative abundance of a genes isoforms as a vector Ψ. Note that $\sum_k \Psi_k = 1$, where Ψ_k denotes the relative abundance of isoform k. Assuming uniform sampling of the RNA-Seq process, then the probability of a non-junction read being sampled from isoform k can be written in terms Ψ weighted by the mappable length $c_k = l_k - RL + 1$,

$$\Psi_{fk} = \frac{c_k \Psi_k}{\sum_i c_i \Psi_i}$$

Denote the number of mappable positions of isoform I_k in an experiment with read length RL as $m(RL, I_k)$. The probability of read R_n being sampled from isoform k can be written as,

$$P(R_n|k, \Theta) = \frac{R_n^k}{m(RL, I_k)}$$

where R_n^k denotes whether read R_n can be generated from isoform k. And the posterior distribution can be evaluated as,

$$\begin{aligned} P(\Psi|R_{1:N}) \propto & P(R_{1:N}|\Psi)P(\Psi) \\ = & \sum_{I_1=1}^{K} \cdots \sum_{I_N=1}^{K} \prod_{n=1}^{N} P(R_n|I_{1:N}, \Theta) P(I_{1:N}|\Psi) P(\Psi) \end{aligned}$$

The exon-centric analysis only contains two isoforms at a time. Using an uniform prior for Ψ, and the MAP (also the MLE) estimator for this analysis when single-ended reads are used can be derived. For isoform-centric analysis, the following choices of distributions are used in this hierarchical

model,

$$\Psi \sim \text{Dirichlet}(\alpha) \text{ for every gene g}$$

$$I_n|\Psi \sim Multinomial(1, \Psi) \text{ for every read } n \text{ mapped to gene } g$$

$$R_n \sim P(R_n|I_n, \Theta) \text{ uniform distribution}$$

However, there is no analytic solution for the isoform-centric analysis, in which case a posterior distribution of the vector Ψ is calculated with a hybrid MH-Gibbs sampling scheme. Bayes factors derived from the posterior distributions of the null and alternative model are used to test for differential isoform expression.

MATS

Another method employing Bayesian theory of inference to test for differential exon usage is MATS (Shen et al., 2012), whose model is summarized below. Denote $\Psi = (\psi_1, \psi_2)$ as the exon inclusion levels for an exon in two samples, $I_{i\cdot}$ as the counts of the exon including isoform of exon i, and $S_{i\cdot}$ as the counts of the exon skipping isoform of exon i. The Gibbs sampler setup is as follows:

$$(\psi_1, \psi_2) \sim MultiVarUniform(0, 1, cor = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix})$$

$$\rho \sim Uniform(0, 1)$$

$$I_{i1}|\psi_{i1} \sim Binomial(n = I_{i1} + S_{i1}, p = \psi_{i1})$$

$$I_{i2}|\psi_{i2} \sim Binomial(n = I_{i2} + S_{i2}, p = \psi_{i2})$$

Shen et al. adopted a complicated sampling scheme for the calculation of p-values to test for the DAS events. In summary, it involves four steps:

Step-1: Estimating constrained MLE under the null ($|\psi_1 - \psi_2| < c$);

Step-2: Using the constrained MLE to simulate M sets of new data, and perform the above MCMC procedure for each set to obtain a posterior distribution of (ψ_1, ψ_2) under the null;

Step-3: Compute a probability for each set of the newly simulated data the probability $P(|\psi_1 - \psi_2| > c)$, denoted as $P_j^{sim}, j = 1, \ldots, M$;

Step-4: Compare the posterior distribution to the observed probability of $P(|\psi_1 - \psi_2| > c)$, denoted as P^{obs}, to each P_j^{sim}, and compute the proportion of P_j^{sim} that is greater than P^{obs} as the empirical

p-value for this test.

MATS suffers from a sleuth of caveats. In real-life practice, it's extremely slow due to the multi-layer sampling scheme required for an unnecessarily convoluted statistical model. Most often than not, its results cannot be replicated by confirmatory experiments, implying invalid assumptions and incorrect modeling, *e.g*, there is no explicit modeling of over-dispersion. In addition, MATS does not support biological replicates, instead, the software pools all the counts together from different samples of the same group and treats them as one sample. This approach dramatically reduces the information embedded in the replicates, and subsequently drastically affects its power. The same group published an updated version of MATS, rMATS to rectify some of the caveats above (Shen et al., 2014).

rMATS

The first improvement of rMATS over MATS is the support of replicates in each sample. Shen et al. modeled the logit of the exon inclusion level of each sample to be generated from a normal distribution, and given the exon inclusion level, in each sample, the read counts of the exon including isoform were modeled like in MATS. For exon i ($i = 1, \ldots, N$), replicate j in sample 1 ($j = 1, \ldots, M_1$) or replicate k in sample 2 ($k = 1, \ldots, M_2$).

$$
\begin{aligned}
&\text{logit}(\psi_{i1j}) \sim \text{Normal}(\mu = \text{logit}(\psi_{i1}), \sigma^2 = \sigma^2_{i1}) \\
&\text{logit}(\psi_{i2k}) \sim \text{Normal}(\mu = \text{logit}(\psi_{i2}), \sigma^2 = \sigma^2_{i2}) \\
&I_{i1j}|\psi_{i1j} \sim \text{Binom}\left(n = I_{i1j} + S_{i1j}, p = \frac{l_{iI}\psi_{i1j}}{l_{iI}\psi_{i1j} + l_{iS}(1-\psi_{i1j})}\right), j = 1, \ldots, M_1 \\
&I_{i1k}|\psi_{i1k} \sim \text{Binom}\left(n = I_{i1k} + S_{i1k}, p = \frac{l_{iI}\psi_{i1k}}{l_{iI}\psi_{i1k} + l_{iS}(1-\psi_{i1k})}\right), k = 1, \ldots, M_2
\end{aligned}
$$

Parameters can be estimated by iteratively calculating the mean and variance of the exon inclusion levels ψ_{i1}, ψ_{i2}, and the latent variables denoting the individual exon inclusion levels in each sample. The former is achieved by maximizing a Laplace-approximated marginal likelihood of the group-level exon inclusion levels, using the latent variable estimates from the last iteration. And the latent variables can be computed by maximizing the full likelihood, with the group-level parameters fixed. Testing of DAS exons in unpaired experiments can thus be performed by LRT tests. The authors presented a more conservative tests compared to the one used in MATS due to computational

conveniences.

$$H_0 : |\psi_1 - \psi_2| = c$$

$$H_0 : |\psi_1 - \psi_2| > c$$

This LRT statistic follows a χ^2 distribution with one degree of freedom.

Shen et al. also proposed an algorithm to test for DAS exons in paired experiments. The statistical model of the paired experiments is similar to the unpaired setup, except for the distributions of ψ.

$$\begin{pmatrix} \text{logit}(\psi_{i1j}) \\ \text{logit}(\psi_{i2j}) \end{pmatrix} \sim \text{Normal}\left(\mu = \begin{pmatrix} \text{logit}(\psi_{i1}) \\ \text{logit}(\psi_{i2}) \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_{i1}^2 & \rho_i \sigma_{i1} \sigma_{i2} \\ \rho_i \sigma_{i1} \sigma_{i2} & \sigma_{i2}^2 \end{pmatrix} \right)$$

Similar procedures were used to estimate the parameters in this paired model and test for DAS exons.

Compared to MATS, rMATS has much more extended functionalities, but some of the caveats of the original MATS remain. MATS and rMATS both assume a binomial distribution of the counts given ψ, completely ignoring the over-dispersion. Only two experimental conditions are allowed, multi-group tests are not well-supported. Computing time is still quite hefty even with rMATS, which abandoned the complicated sampling scheme.

2.3.2. NB-based methods

The differential alternative splicing of exons does not directly reflect the relative abundance of isoforms, which is of equal if not greater interests to researchers biologically. MATS only provides DAS analysis on the exon level, but not on the isoform level. Methods based on direct modeling of read counts with a negative-binomial distribution inspired by the methods identifying DE genes could potentially solve this problem.

DEXSeq

In 2012, the team that developed DESeq authored a new package DEXSeq, that utilized NB-based GLM to test isoform DAS events. They introduced 'counting bins', the biggest unit that is either present or absent from all the isoforms. This concept was necessary, because many exons have alternative 3' or 5' start site, which renders its boundary variable across different isoforms. It is similar to the 'mathematical exons' used in DSGSeq (Wang et al., 2013).

Denote k_{ijl} as the number of reads overlapping counting bin l of gene i in sample j, $j = 1, \ldots, m$. Denote the expected value of the concentration of cDNA fragments contributing to counting bin l of gene i as μ_{ijl}.

$$E(K_{ijl}) = s_j \mu_{ijl}$$

where s_j is the size factor for sample j. And the model can be written as a log-linear model with NB distribution,

$$K_{ijl} \sim NB(\mu = s_j \mu_{ijl}, \sigma^2 = \alpha_{il})$$
$$\log \mu_{ijl} = \beta_i^G + \beta_{il}^E + \beta_{ij}^S + \beta_{i\rho_j l}^{EC}$$

And covariates in the log-linear model are explained in Table 2.2. The dispersion parameters are

Covariate	Explanation
β_i^G	Logarithm of baseline gene expression
β_{il}^E	Logarithm of the fraction of reads mapped to bin l among reads mapped to gene i
β_{ij}^S	Logarithm of the overall fold change of gene i in sample j
$\beta_{i\rho_j l}^{EC}$	Effect of the condition ρ_j on the fraction of reads mapped to bin l among reads mapped to gene i
ρ_j	Condition of sample j

Table 2.2: Covariates and explanations for Anders, Reyes, and Huber, 2012

fitted for each counting bin using methods similar to Love, Huber, and Anders, 2014, using the above model. And testing of differential usage of each counting bin l' can be achieved with LRT by fitting the following two models,

$$\text{Unsaturated: } \log \mu_{ijl} = \beta_i^G + \beta_{il}^E + \beta_{ij}^S$$
$$\text{Saturated: } \log \mu_{ijl} = \beta_i^G + \beta_{il}^E + \beta_{ij}^S + \beta_{i\rho_j l}^{EC} \delta_{ll'}$$

where $\delta_{ll'} = 1$, if $l = l'$, and $\delta_{ll'} = 0$ otherwise. The null hypothesis states that all conditions have equal usage of the counting bin, and if one or more of the conditions violate this hypothesis the null will be rejected. The model can be slightly changed to test if there is also an overall change of gene expression among different conditions, by replacing the β_{ij}^S with a term $\beta_{i\rho_j}^C$, representing the effect of conditions rather than samples on the overall gene expression.

rSeqDiff

EBSeq is focused on testing for the differential expression of isoforms, which does not necessarily lead to differential alternative splicing. Methods such as DEXSeq are focused on testing the differential use of exons, which does not directly quantify the differences in relative abundances of isoforms under different conditions. As mentioned above, Shi et al. proposed a nested testing paradigm in their rSeqDiff method, which effectively illustrated the logical relationship between testing for DE isoforms and DAS genes.

In their implementation, a linear Poisson model with the following parametrization is used for inference.

$$f_{\theta_k}(N_k) = \prod_{j=1}^{J} \frac{(\theta_k a_{kj})^{n_{kj}} \exp[-\theta_k a_{kj}]}{n_{kj}!}$$

where θ_k is a vector of the isoform abundance values under k^{th} condition, j indexes the sample, and a_{kj} denotes the vector of the sampling rates from all isoforms for read type j in condition k. A matrix also doubles as a compatibility matrix where the elements are set to 0 if the read type cannot be sampled from a specific isoform.

For testing purposes, three models with varying degrees of freedom for the parameter θ are devised. Index conditions by k.

Model-0 [No differential expression]: this model assumes a constant θ across all k conditions.

$$\mathcal{L}(\theta_0|N_k) = \prod_{k=1}^{K} f_{\theta_0}(N_k)$$

Model-1 [Differential expression without DAS]: this model assumes the θ_k for all conditions to be completely linearly dependent. Hence, denote the multiplicative factor for condition k as $\tau_k > 0$.

$$\mathcal{L}(\theta_0, \tau|N_k) = \prod_{k=1}^{K} f_{\tau_k\theta_0}(N_k)$$

Model-2 [DAS]: this model assumes a different θ_k for each condition k.

$$\mathcal{L}(\theta_1, \ldots, \theta_k|N_k) = \prod_{k=1}^{K} f_{\theta_k}(N_k)$$

MLEs of the above models are computed with an EM algorithm, and a hierarchical LRT is used for model selection. Briefly, the first step include two tests that compare Model 0 vs Model 1 and Model 0 vs Model 2, each at significance level $\alpha/2$. If both tests fail to reject the null hypothesis, then Model 0 is the correct model, and there is no DE or DAS with the target gene. If one of the tests rejects the null, then the corresponding model is selected as correct. If both tests reject the null, then proceed to the second step of testing, which compares Model 1 vs Model 2 at significance level α. If the null is rejected then Model 2 is correct, otherwise Model 1 is correct.

The greatest contribution of rSeqDiff is that it elucidated the logical relationship between testing for DE and DAS of isoforms. With a hierarchical likelihood approach, users can now distinguish between the two types of differentiation of isoform usage. One caveat of rSeqDiff is that it utilizes a linear Poisson model, which is incapable of modeling over-dispersion. Subsequently, no treatment of over-dispersion was implemented, and the method could potentially have inflated Type I error.

2.3.3. Other models

In addition to the methods above, some other interesting methods are available that do not fall into any of the categories above. This is in no way implying that they are less impressive, rather, some of them are hard to categorize simply because of their uniqueness and novelty. Most of these methods are specialized in testing of DAS exons from two experimental groups with replicates.

DSGSeq

One of such methods is DSGSeq (Wang et al., 2013) written by the developer of DEGSeq. If we denote the abundance of transcripts from all the 'mathematical exons' (same as the 'counting bins' in DEXSeq) in the vector $\mathbf{k}$, and the probability of a read falling into exon j as p_j. Let $\mathbf{p}$ be the vector whose elements are p_j, then the following relationship holds,

$$\mathbf{p} = \frac{1}{\Xi} BLA^T \mathbf{k}$$

where B, L are diagonal matrices representing sequencing preference of exons and exon length, Ξ is a normalization factor, and A is the exon-isoform compatibility matrix, in which a_{ij} is a variable indicating the presence or absence of exon j in isoform i. The authors of DSGSeq noted that a one-to-one correspondence between $\mathbf{p}$ and $\mathbf{k}$ exists if A is of full row rank, *i.e.*, no isoform can be a linear combinations of other isoforms in this gene. This is an easily satisfied condition in the human

genome, and it provides a shortcut for testing of differential alternative splicing of a target gene. The most direct hypothesis of testing for DAS in genes is the following,

$$H_0 : \mathbf{k}_1 = \mathbf{k}_2$$
$$H_1 : \mathbf{k}_1 \neq \mathbf{k}_2$$

where $\mathbf{k}_1$ and $\mathbf{k}_2$ represents the transcript abundance vectors under conditions 1 and 2. However, due to the bijective relationship between $\mathbf{k}$ and $\mathbf{p}$, we can instead test the following hypotheses:

$$H_0 : \mathbf{p}_1 = \mathbf{p}_2$$
$$H_1 : \mathbf{p}_1 \neq \mathbf{p}_2$$

The rationale for this change of variables is that $\mathbf{p}$ is much more readily estimable than $\mathbf{k}$, which is essentially a latent variable. Denote the number of samples as n, the reads mapped to exon j in sample i as Y_{ij}, and the total number of reads mapped to the gene in sample i as M_i. An unbiased estimator for p_j is simply the mean,

$$\hat{p}_j = \frac{1}{n} \sum_{i=1}^{n} \frac{Y_{ij}}{M_i}$$

In order to more accurately estimate $\hat{p}_j$, DSGSeq used a weighted mean and estimated the weights using a variance-minimizing estimator, which is still very simple to compute.

After estimating the $\hat{p}_j$ and $\mathrm{Var}[\hat{p}_j]$ for all exons, the authors use a NB-statistic to test for the presence of a differentially used exons. Denote the number of exons as m,

$$\mathrm{NB} - \mathrm{stat} = \frac{1}{m} \sum_{i=1}^{m} \frac{(\hat{p}_{j1} - \hat{p}_{j2})^2}{\widehat{\mathrm{Var}}[\hat{p}_{j1}] + \widehat{\mathrm{Var}}[\hat{p}_{j1}]}$$

And genes are ranked by the NB-statistic with the larger values being more significant for differential alternative splicing. In addition, the NB-statistic for the gene can be easily decomposed into m components, each corresponding to an exon in the gene. The method is interesting for its simplicity, and great performance in real-life practice. One caveat of DSGSeq is that it does not provide a p-value due to the lack of null distribution of the NB-statistic. Therefore, the cutoff for the significance of the DAS events is somewhat *ad hoc*.

IUTA

Another intuitive method for testing for DAS is to test for difference in the relative abundance vector of isoforms. Denote $\{\theta_k\}_{k=1}^K$ as the relative abundance for isoform k, and $\theta = (\theta_1, \ldots, \theta_K)$. Testing for DAS is equivalent to,

$$H_0 : \theta_0 = \theta_1$$

$$H_1 : \theta_0 \neq \theta_1$$

The fact that θ is compositional complicates the above tests. One major contribution of IUTA is the proposal of using an isometric log-ratio (ilr) transformation to transform the compositional data from the open simplex to the real space $\mathbb{R}^{K-1}$ with Euclidean geometry. Now to test for the DAS isoforms, one needs to test,

$$H_0 : ilr(\theta_0) = ilr(\theta_1)$$

$$H_1 : ilr(\theta_0) \neq ilr(\theta_1)$$

The transformed relative abundance follows a multivariate normal distribution,

$$ilr(\boldsymbol{\Theta}_{ij}) \sim N\left(ilr(\theta_i), \boldsymbol{\Sigma}_i + y_{ij}\right)$$

where θ_i denotes the shared relative abundance vector of a specific group, and the variance can be decomposed into two factors using the hierarchical interpretation. To test the equality of mean with different variance-covariance matrices, KY, SKK or CQ tests can be performed. The parameters θ are estimated from a likelihood similar to the one employed by MISO using MLE method.

Instead of relying upon analysis of exon usage, IUTA directly tests for the differences in the compositional vector of isoform relative abundances. This approach is more appropriate for analyses of genes with complicated isoform structure, compared to exon-centric methods.

SplicingCompass

Another unique approach to test for differential alternative splicing events is SplicingCompass (Aschoff et al., 2013), which takes advantage of a measurement aptly named 'splicing angle', which represents the geometric angle between two vectors of exon usage. For a specific gene from sample i, SplicingCompass computes a vector containing the reads mapped to each unique exon $\bar{v}_i$.

For n samples, it then computes $\binom{n}{2}$ angles in all the pairs of samples.

$$\Phi_{ij} = \arccos\left[\frac{\bar{v}_i \cdot \bar{v}_j}{\|\bar{v}_i\|\|\bar{v}_j\|}\right] \cdot \frac{180}{\pi}$$

Then a one-sided t-test is used to test whether angles of within-condition sample pairs are significantly smaller than those of between-condition sample pairs.

CHAPTER 3

COMPUTATIONAL TOOLS FOR SINGLE-CELL RNA-SEQ

Due to the complexities of single-cell RNA-seq protocols, a series of novel methods have been developed specifically for this new platform. Briefly, these methods can be categorized into several functional classes (Bacher and Kendziorski, 2016): normalization, noise reduction, Identifying highly variable genes, identification of subpopulation, pseudotemporal ordering, and differential expression analysis. In this dissertation, we focus on the methods for differential expression, and by extension, normalization.

3.1. Normalization

The protocols of scRNA-seq are carried out in individual chambers or droplets after the cells are isolated and separated. It is technologically infeasible to strictly control the environment of reaction to absolute uniformity. Cell-to-cell heterogeneity can cause severe differences in the rate of reaction and subsequently influence the read counts recovered from the assay. It is therefore critical to carefully normalize the raw reads before any analysis is carried out using data from scRNA-seq experiments. Several methods have since been developed.

3.1.1. SAMStrt

The first attempt at normalizing scRNA-seq data is from Katayama et al., 2013. The authors adapted SAMseq (Li and Tibshirani, 2013) which is a differential expression method designed for bulk RNA-seq, with modified sequencing depth estimation by assuming equivalent spike-in-molecules/cell in each experimental set. This method of normalization is later adopted by Lun, Bach, and Marioni, 2016, when the sample size of any biological condition fails to meet the minimum requirement for their SCRAN algorithm.

3.1.2. GRM

A slightly less naïve method was proposed in Ding et al., 2015. In their work, spike-in ERCC (Baker et al., 2005) molecules are used to construct a gamma regression model. Briefly, log-transformed concentration $\mathrm{y} = \log(\mathrm{concentration})$ $(\log - \mathrm{C})$ and FPKM $\mathrm{x} = \log(\mathrm{FPKM})$ $(\log - \mathrm{R})$ are modeled

with a gamma distribution with a polynomial, to account for the non-linearity in the relationship between x and y. The model is written as:

$$y \sim Gamma(y; \mu(x), \varphi) \tag{3.1}$$

$$\mu(x) = \sum_{i=0}^{n} \beta_i x^i \tag{3.2}$$

$$f(y) = \frac{1}{y\Gamma(\sigma)} \left(\frac{\varphi y}{\mu(x)}\right)^{\varphi} \exp\left(-\frac{\varphi y}{\mu(x)}\right) \tag{3.3}$$

The parameters β_i and φ are determined with MLE. The degrees of $\mu(x)$, n, is determined by fitting four models with $n = 1$ to $n = 4$ and selecting the n that corresponds to the smallest average technical noise of ERCCs. Once the model is fitted, the true expression of a gene can be estimated from its FPKM as,

$$\hat{y}_{gene} = E[y_{gene}] = \hat{\mu}(x_{gene}) = \sum_{i=0}^{n} \hat{\beta}_i x_{gene}^i \tag{3.4}$$

GRM takes advantage of the fact that the concentration of ERCC spike-ins is known, and by linking the read counts to the actual concentration, one can fit a model that maps the raw FPKM to the true gene expression. The method is straightforward and intuitive to understand. Although the performance can be more thoroughly assessed and benchmarked against other normalization methods, including bulk RNA-seq tools, GRM is an interesting addition to the scRNA-seq computational toolbox.

3.1.3. SCRAN

A robust method of normalization has been developed in Lun, Bach, and Marioni, 2016 by pooling cells into carefully designed clusters. The idea involves estimating the true size factors of many different pools ($E[R_{ik}]$) of cells first across genes, and subsequently, solving for the cell-specific size factors by fitting a series of over-represented linear systems.

The authors propose that $E[R_{ik}] = \sum_{S_k} \theta_j$, where θ_j is the cell specific size factors for cell j in S_k, which is a subset of the cells in the population of interest. R_{ik} is the random variable representing the true size factor of subset S_k for gene i, whose expectation is the true size factor for the whole set S_k. Now $E[R_{ik}]$ can be robustly estimated by averaging r_{ik} over i, and this will serve as the response variable in the linear systems.

Now $E[R_{ik}]$ can also be represented as the sum of the true size factors across the cells in the selected pool. Therefore, by selecting multiple pools, a linear system can be constructed. Denote the $E[R_{ik}]$ estimated by averaging across genes as $\lambda_1, \cdots, \lambda_K$, and the true size factor for each cell i as θ_i. Denote x_{jk} as the design matrix indicating the present of a cell j in pool k. Then the linear systems can be written as,

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 1 & 0 & 1 & \cdots & 0 & 1 & 0 \\ \vdots & & & & & & \\ 1 & 1 & 0 & \cdots & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_J \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_K \end{bmatrix} \tag{3.5}$$

Select enough pools of cells, and solve Equation 3.5, we will get a set of robust estimates for θ_j, thus achieving the goal of normalization for the single-cell RNA-seq counts.

One caveat of the SCRAN method is that it requires large enough number of cells in every single group compared, therefore, it is limited by the size of the smallest group. In our experience, if one group contains 20-30 cells, the algorithm will fail, regardless of the sample size of the other group.

3.2. Differential Expression: Genes

3.2.1. SCDE

One of the very first papers on detecting gene-level differential expression was published by the Kharchenko group at Harvard (Kharchenko, Silberstein, and Scadden, 2014). Due to the blatant abuse of notations and complete lack of statistical rigor in their manuscript, I have yet to achieve the goal of understanding the mathematics in their model. But for the completeness of this dissertation, I will make an attempt at describing their work. The words in quotes are copied from their paper (Kharchenko, Silberstein, and Scadden, 2014) verbatim. I did not write or alter any of these sentences.

The first step of their procedure is to fit individual error models, which I can only guess is on a per-cell basis. “All pairs of individual cells belonging to a given subpopulation (for example, all MEF cells) were analyzed with a three-component mixture model.” Now The following “components”

were directly copied from their published manuscript.

$$\begin{cases} r_1 \approx \text{Poisson}(\lambda_0)\ \text{Dropout in } c_1 \\ \begin{cases} r_1 \approx \text{NB}(r_2) \\ r_2 \approx \text{NB}(r_1) \end{cases} \quad \text{Amplified} \\ r_2 \approx \text{Poisson}(\lambda_0)\ \text{Dropout in } c_2 \end{cases} \tag{3.6}$$

If one were to treat the above math as a set of statistical notations, then the following questions are begging to be asked:

- What does it mean for a random variable to be approximately equal ($\approx$) to a distribution?
- What happens to the three-component mixture model? What are the three components in the above brackets?
- What does $r_2 \approx \text{NB}(r_1)$ mean? Is r_1 a random variable or a parameter? Even if we consider the possibility of Bayesian inference where both r_1 and r_2 are random variables, the definition of the distribution of a parameter should involve nothing but hyper-parameters.
- Does the center "amplified" bracket count as one component? Is it a two-dimension random variable? If so how does one mix two one-dimension random variables and one two-dimension random variable?

The next step would be to fit an "individual error model Ω_C", Dr. Kharchenko and team suggest that "The RPM level r_c observed for a gene in cell c was modeled as a mixture of a dropout and amplified components, as a function of an expected expression magnitude e, as"

$$\begin{cases} r_c \approx \text{NB}(e) & \text{Amplified} \\ r_c \approx \text{Poisson}(\lambda_0) & \text{Dropout} \end{cases} \tag{3.7}$$

"with the mixing parameter $m = \log(e)$."

Up till now, the use of statistical notations in this paper does not conform to the conventions by which they are usually applied. Differential expression analysis is performed "with a Bayesian approach". They define "the posterior probability of a gene being expressed at an average level x

in a subpopulation of cells S" according to

$$p_S(x) = E\left[\prod_{c \in B} p\left(x|r_c, \Omega_c\right)\right] \tag{3.8}$$

where B is a bootstrap sample of S. I would like to point out that the expectation of a probability again is a blatant misuse of notation. They also define "$p(x|r_c, \Omega_c)$" as "the posterior probability for a given cell c", according to

$$p(x|r_c, \Omega_c) = p_d(x)p_{Poisson}(x) + (1 - p_d(x))p_{NB}(x|r_c) \tag{3.9}$$

"where p_d is the probability of observing a dropout event in cell c for a gene expressed at an average level x in S, $p_{Poisson}(x)$ and $p_{NB}(x|r_c)$ are the probabilities of observing expression magnitude of r_c in case of a dropout (Poisson) or successful amplification (NB) of a gene expressed at level x in cell c, with the parameters of the distributions determined by the Ω_c fit. For the differential expression analysis, the posterior probability that the gene shows a fold expression difference of f between subpopulations S and G was evaluated as"

$$p(f) = \sum_{x \in X} p_S\left(x\right) p_G(f_x) \tag{3.10}$$

"where x is the valid range of expression levels." The authors further claimed that "the posterior distributions were renormalized to unity, and an empirical P-value was determined to test for significance of expression difference." Terms such as "renormalized to unity" does not make much sense in statistics without further clarification.

I cannot comment on the mathematical validity of this model, due to their unfortunately unintelligible notations. Nor could I get the software to run properly without modifying their code, which misuses the parallel computing functions so dangerously, that it quickly consumes the entire RAM of the machine once it starts. I would caution whoever wishes to give it a try do so after fully auditing their published code and consulting with an expert in **R**, which the original authors do not appear to be.

3.2.2. MAST

MAST (Finak et al., 2015) is a first statistically sensible attempt at directly model the bimodality of the single-cell gene expression. In order to correct for the inflated zeros in scRNA-seq data,

Finak, McDavid, Yajim *et al.* proposed a Hurdle model with two parts, the rate of expression over background (probability of non-zero), and the mean of the positive expression (mean of the non-zero component). The authors use *cellular detection rate* (CDR), the proportion of genes detected in each cell, account for cell-to-cell technical differences such as dropout, amplification efficiency, cell size, cell cycle, *etc*, that affect the overall gene expression in individual cells. CDR is treated as an additional covariate in the regression model, to control for the aforementioned biases.
MAST models the variation in $\log_2(\mathrm{TPM}+1)$ expression matrix instead of raw read counts, as a two-part generalized linear regression model. Denote Z_{ig} as an indicator that gene g is expressed in cell i.

- CDR is modeled using logistic regression.

$$\operatorname{logit}\left[P(Z_{ig}=1)\right] = X_i\beta_g^D \tag{3.11}$$

- the mean positive expression is modeled as a conditionally normal distribution given that $Z_{ig}=1$, *i.e.*, gene g is expressed.

$$\Pr\left[Y_{ig}=y|Z_{ig}=1\right] = N(X_i\beta_g^C, \sigma_g^2) \tag{3.12}$$

Bayesian GLM was used to regularize the coefficients for the discrete regression in case of complete separation, and additional regularization is performed on the variance parameter of the continuous model, in order to increase the robustness of the differential analysis when a gene is only expressed in a small number of cells. MAST outputs three p-values:

- "disc": the p-values generated with logistic regression described in Equation 3.11, testing for significant correlation between levels of zero inflation and the given covariates.
- "cont": the p-values generated with linear regression described in Equation 3.12, testing for significant correlation between levels of positive mean expression and the given covariates.
- "hurdle": the p-values generated by adding the χ^2-statistics computed for the above two tests, and combining the degrees of freedom, testing for significant correlation in either of the two scenarios.

One of the advantages of MAST is its computational efficiency, due to their clever use of existing models and software packages. The biggest caveat of MAST lies in the manner with which the technical noises are incorporated. While CDR is an important summary statistic to use for normalizing cell-to-cell heterogeneity, it is hardly sufficient for the scRNA-seq data, as is shown in Jia et al., 2017. Briefly, CDR is computed on a per-cell basis, which reflects the baseline differences of gene detection rates. However, this probability is also related to the expression level of the gene, which has not been accounted for in the discrete model. In addition, the relationship between the mean positive component and the true expression of a gene can also be different due to varying factors affecting efficiencies of reverse transcription and PCR reactions in individual chambers, such as rates of dissolution, amount of reactants, enzymatic activity, *etc*. These technical differences are not incorporated in any of the models. Lastly, the input for MAST is TPM, which is estimated from read or fragment counts. Therefore, in order to apply MAST model efficiently, a preprocessing step is required to compute the TPM values for each gene in each cell. This might not be accurate due to the extremely low starting material of scRNA-seq, especially for technologies such as SMART-seq, which is not designed specifically for tag counting for its lack of compatibility with UMIs. This can significantly limit the use case of MAST, especially considering SMART-seq is a popular and established scRNA-seq protocol.

3.2.3. scDD

Another attempt at modeling the multi-modality of RNA-seq data came from the Kendziorski group at University of Wisconsin (Korthauer et al., 2016). The method is motivated by the observation that the distribution of the log-transformed non-zero expression measurements of single-cell RNA-seq data is usually multi-modal for a specific gene. Therefore, testing for the positive mean alone might not disclose the existing differences. Therefore, a mixture model framework is employed to describe the read counts of Y_g of a gene g from scRNA-seq. Specifically, assume Y_g follows a conjugate Dirichlet process mixture (DPM) of normals. A Bayes factor comparing two models

- "DD": the data arises from two independent condition-specific models
- "ED": the data arises from one overall model regardless of condition

Let $\mathcal{M}$ denotes the model, the Bayes factor is computed as,

$$\mathrm{BF}_g = \frac{f(Y_g|\mathcal{M}_{\mathrm{DD}})}{f(Y_g|\mathcal{M}_{\mathrm{ED}})} \tag{3.13}$$

In order to fit the above model, the authors took a multi-step approach taking advantage of the product partition model (PPM) formulation.

- estimate partition membership $\hat{Z}$ that maximizes the *maximum a posteriori* (MAP) by optimizing the Bayesian information criterion (BIC) of the marginal density $f(Y|Z)$ using R package Mclust.
- estimate the component-specific parameters with the closed form solutions obtained due to conjugacy.
- estimate the MAP of the joint predictive distribution of data Y and partition Z.
- compute the bayes factor, and if needed, permute the condition labels to calculate an empirical p-value

The scDD method is theoretically sound, and has provided some novel insight on testing for distributional differences of gene expression measurements in scRNA-seq data. In addition to testing for shifts of mean, it is capable of characterizing the distributional patterns of gene expression, and testing for differences in these patterns between biological conditions. However, adjustment for additional covariates is quite limited in scDD, and it is inadvisable to compare patterns between more than 2 biological conditions. It would be very valuable if the model could be extended to incorporate a regression component, thus allowing scDD to account for confounding factors. And it would be interesting to see the minimum sample size it requires to efficiently compare the differences in patterns between 3 or more biological conditions.

CHAPTER 4

GENERALIZED LINEAR MIXED-EFFECTS MODELS FOR DETECTING DE ISOFORMS AND SPLICING QUANTITATIVE TRAIT LOCI (SQTLS) FROM BULK RNA-SEQ DATA

4.1. MetaDiff

4.1.1. Motivation

So far we have reviewed more than two dozen methods for detecting gene differential expression, isoform differential expression and differential alternative splicing using bulk RNA-seq. Despite the sheer number of models available, there is still ample room for improvement. For example, in terms of detecting isoform differential expression, ideally a method should satisfy the following criteria:

[i] Account for isoform expression estimation uncertainty.

[ii] Account for variation in the estimation uncertainty across features and biological replicates.

[iii] Flexibility to adjustment for covariates and confounding factors, discrete and continuous.

Methods such as DESeq, DESeq2 and edgeR are not designed to model counts with estimation uncertainty. Methods such as Cuffdiff, BitSeq and EBSeq do a terrible job accounting for it. Moreover, they cannot include covariates and other confounders in their model. MMDIFF in theory is able to satisfy the above criteria, but their choice of Bayesian modeling makes it difficult to compare its performance to other methods. More importantly, MMDIFF only works with results from MMSEQ. Since the result of isoform DE detection is highly reliant upon the accurate estimation of the isoform expression, this severely limits the chance of MMDIFF improving its accuracy by switching out the upstream method. MMDIFF outputs the posterior probability of the alternative model, without inferential information on the included covariates, making it impossible for users to control for a fixed threshold of false discovery rate. Lastly, the MCMC sampling scheme in MMDIFF method makes the program computationally inefficient, without the chance of parallelizing the procedures. This results in terrible run time, up to a day for a moderately sized group of samples.

As a result, we have developed MetaDiff, a meta-regression-based general framework for identifying differentially expressed isoforms, accounting for estimation uncertainty of the upstream pack-

ages. It can take input from any software packages as long as the variance (or confidence interval) of the isoform expression estimate is included. This flexibility allows the user to choose any source of estimation package they prefer, facilitating the improvement in inferential accuracy by upgrading the upstream estimating software. It is a frequentist method which outputs raw and FDR-adjusted p-values for each feature, which can then be used to control for a specific false discovery rate in the identified samples. Due to the use of a very efficient R package $\mathrm{metatest}$, the program can be parallelized, utilizing the full capacity of a multi-core CPU. It is also extremely efficient, with run time up to less than an hour for the same sample tested on MMDIFF.

4.1.2. Model and estimation

A random effects model was used to include the variance of the estimates of the isoform expression.

$$\log[Y_i] = \beta_0 + \beta_1 X_i + \beta_2 Z_i + \mu_i + \epsilon_i$$

The interpretation of the components of the above model can be found in Table 4.1. In addition to

Parameter	Interpretation
Y_i	Random variable, the estimated isoform expression
β_0	Baseline log-expression
β_1	Coefficients associated with the variables of interest
X_i	Design matrix of the variables of interest
β_2	Coefficients associated with the additional covariates and confounding factors
Z_i	Design matrix of the additional covariates and confounding factors
μ_i	Estimation uncertainty for $\log[Y_i]$. $\mu_i \sim N(0, \mathrm{Var}[\log[Y_i]])$
ϵ_i	Random error. $\epsilon_i \sim N(0, \tau^2)$

Table 4.1: Components and Interpretations in Jia et al., 2015

the assumptions listed in Table 4.1, we also assume $\mathrm{Cov}(\mu_i, \epsilon_i) = 0$, and the n observations are statistically independent. Estimation of this model is performed with R package metatest, and the the input of this method can be in the form of both raw FPKM (Trapnell et al., 2010) or log-FPKM (Turro, Astle, and Tavaré, 2014). Log-FPKMs can be directly fed into the program after parsing. However, raw FPKMs need to be log-transformed. Assume the FPKM prior to transform follows a normal distribution,

$$Y_i \sim N(\mu_i, \sigma_i^2)$$

Using δ-method, log-FPKMs will follow a normal distribution with mean and variance,

$$\log[Y_i] \sim N(\log[\mu_i], \frac{\sigma_i^2}{\mu_i^2})$$

Hence, the mean and variance of the transformed FPKMs can be approximated by,

$$E \log[Y_i] = \log[\mu_i]$$
$$\text{Var}[\log[Y_i]] = \frac{\sigma_i^2}{\mu_i^2}$$

4.1.3. Testing for DE isoforms

The null hypothesis of no differential expression between different groups (or other covariates of interest) is equivalent to,

$$H_0 : \beta_1 = 0$$
$$H_1 : \beta_1 \neq 0$$

Two types of tests are devised to test for these hypotheses: *t*-test and Bartlett corrected likelihood-ratio test (BcLR). The classical *t*-test statistic for t-test is

$$T = \frac{\hat{\beta}_1}{\widehat{SE}\left[\hat{\beta}_1\right]}$$

Under the null hypothesis, this statistic follows a Student-*t* distribution with $n - D_{\beta_1} - D_{\beta_2} - 1$ degrees of freedom, where D_{β_1} and D_{β_2} denote the dimensions of the parameter vector β_1 and β_2 respectively.

Compared to Wald-test, Student-*t* distribution does not rely on the restrictive asymptotic assumptions, therefore, it is applicable when sample size is small, which is quite common in the case of RNA-Seq experiments.

Alternatively, we can formulate this problem as a selection problem for nested linear models, and use a likelihood ratio test corrected with Bartlett's method for small sample inference. The BcLR

test statistic is,

$$\mathrm{BcLR} = \mathrm{BCF} \times 2(\ell - \ell_0)$$

where BCF denotes the Bartlett correction factor (Huizenga, Visser, and Dolan, 2011a), ℓ denotes the log-likelihood under the alternative, and ℓ_0 denotes the log-likelihood under the null.

4.1.4. Simulation

16 cases and 16 controls were simulated with Flux Simulator (Griebel et al., 2012) with the annotated human genome hg19, each with a read depth uniformly distributed between 8 million to 10 million. In order to assess the performance of the method in the absence and presence of covariates and confounding factors, three repetitions of simulations were performed with three different transcriptomic profiles. These three scenarios are illustrated in Figure 4.1.

Scenario I is the basic scenario where there is no covariate to be adjusted for and no confounding. In this case, 30% of the transcripts are DE between cases and control, half (15%) of which are up-regulated by 1.25 fold in cases compared to controls, and the other half are down-regulated by 1.25 fold in cases compared to controls.

Scenario II introduces a covariate by which the expression of some transcripts are influenced. 10% of the transcripts are now influenced by the age of the subject. Age is a random variable that follows the same uniform distribution $\mathrm{Uniform}(18, 60)$ in both cases and controls. The expression of these transcripts increases by 1.35 fold with every one standard deviation increase in age, which is equivalent to 2.5% increased expression for 1 year increase in age.

Scenario III introduces confounding on top of Scenario II. With the rest of the simulation setup identical, we now allow the age variable to follow two different uniform distributions in cases and controls. In cases, age follows $\mathrm{Uniform}(40, 85)$, while in controls, age still follows $\mathrm{Uniform}(18, 60)$.

4.1.5. Results

[i] Empirical FDR

Packages designed for testing gene-level DE, *e.g.*, DESeq, DESeq2 and edgeR, do not explicitly model the estimation uncertainty of the isoform expression. They tend to under-estimate the variance of the expression when used for isoform DE detection, and subsequently render

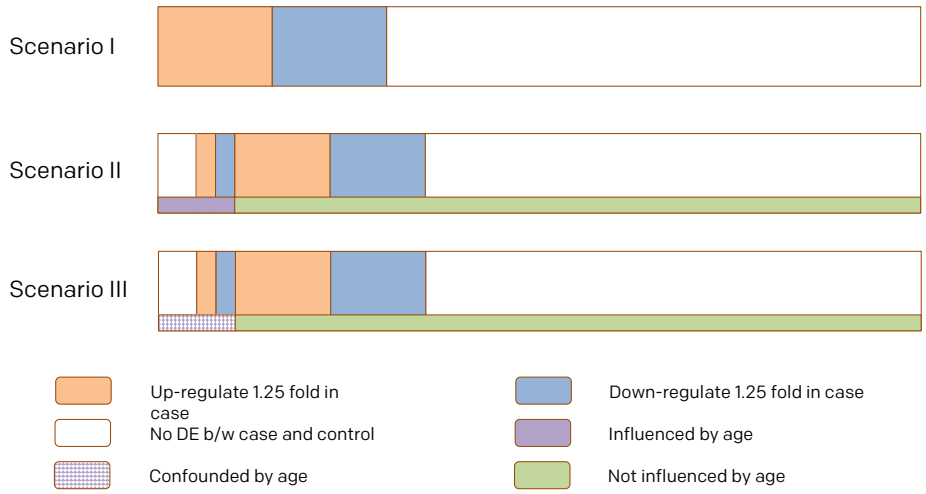


Figure 4.1: 3 Simulation Scenarios. Scenario I: 15% up-regulated in cases; 15% down-regulated in cases; 70% non-DE. Scenario II: 5% non-DE but influenced by age; 2.5% up-regulated in cases and influenced by age; 2.5% down-regulated in cases and influenced by age; 12.5% up-regulated in cases but not influenced by age; 12.5% down-regulated in cases but not influenced by age; 65% non-DE and not influenced by age. Scenario III: same as Scenario II, except that age follows different distributions between cases and control.

the FDR uncontrolled. We want to know if this problem persists in our package, despite our direct modeling of the estimation uncertainty. Empirical FDR, calculated as the fraction of true non-DE features among those labeled as DE, was plotted against the nominal FDR, the threshold given to the software package to label DE features. Figure 4.2 shows the curves for all the methods under all 3 simulation scenarios.

When no covariate or confounding is present (Scenario I), only DESeq and DESeq2 show slightly inflated FDR when the sample size is small. This is consistent with the fact that DESeq and DESeq2 tend to under-estimate the dispersion parameter when the sample size is small, with the trended-by-mean estimate used by DESeq more severe than the empirical Bayes shrinkage method used by DESeq2. When the sample is of sufficient size, this inflation disappears. All the other methods have empirical FDR under control in this scenario.

When the expression of a portion of the features (10%) is influenced by a non-confounding covariate (age, Scenario II), methods that do not allow covariates (EBSeq) or do not allow true continuous covariates (DESeq) start to have trouble keeping the empirical FDR under control. When the sample size is 4+4 or 8+8, the empirical FDR for EBSeq is around 0.6 when the nominal value is in fact 0.05, a more than 10-fold inflation is observed. Regardless of the sample size, the empirical FDR of both EBSeq and DESeq are severely inflated. Despite the

fact that the DESeq package accepts any type of covariates, it does not intrinsically support continuous variables like age. In fact, they will be binned to form discrete groups prior to running the estimation algorithm. This partly explained why DESeq is showing severe inflation when the continuous variable age is present. MetaDiff, edgeR and DESeq2 have properly handled covariates, hence the empirical FDR is still lower than the nominal values. Surprisingly, although Cuffdiff does not support covariates, its empirical FDR is also under control. But later in Figure 4.4, we observe that Cuffdiff in general has lower power than methods who have accounted for the age variable. This conservativeness might in part explain the controlled FDR of Cuffdiff.

When confounding is present (Scenario III), Cuffdiff, EBSeq, and DESeq have all yielded severely inflated FDRs, due to their inability to treat confounding variables. While both tests of MetaDiff (BcLR and Student-*t* test), DESeq2 and edgeR have FDRs well under control. These results have shown that the MetaDiff does not suffer from the problem of inflated FDRs other methods face when covariates or confounding factors are present. In addition, one of the indispensable components of a model testing for gene/isoform differential expression is the treatment for confounding factors, lack of which can lead to inferential mistakes such as inflated FDR.

[ii] Performance under the null

The observation of inflated FDR could be due to a variety of causes, one of which is uncontrolled Type I error. In order to illustrate the performance of the models under the null hypothesis, we generated quantile-quantile (Q-Q) plot for each method under all three scenarios. The raw p-values of true non-DE features are extracted from the exported result of each method, log-transformed, sorted and plotted against a log-transformed expected value of the same quantile from a $\mathrm{Uniform}(0,1)$ distribution. A well-balanced method should have p-values from the non-DE transcripts fall on the diagonal line of this plot.

In Scenario I where there is no covariate to be adjusted for and no confounding factors, edgeR, DESeq and DESeq2 all have deviated from the diagonal line. This pattern is not surprising for DESeq and DESeq2 due to their inflated FDR. However, even with FDR under control, edgeR has shown clear deviation from the diagonal line. This usually signals some type of violation of the model assumption. One suspect is the lack of direct modeling of estimation uncertainty in these models, which could lead to unusual behavior of the algorithm.

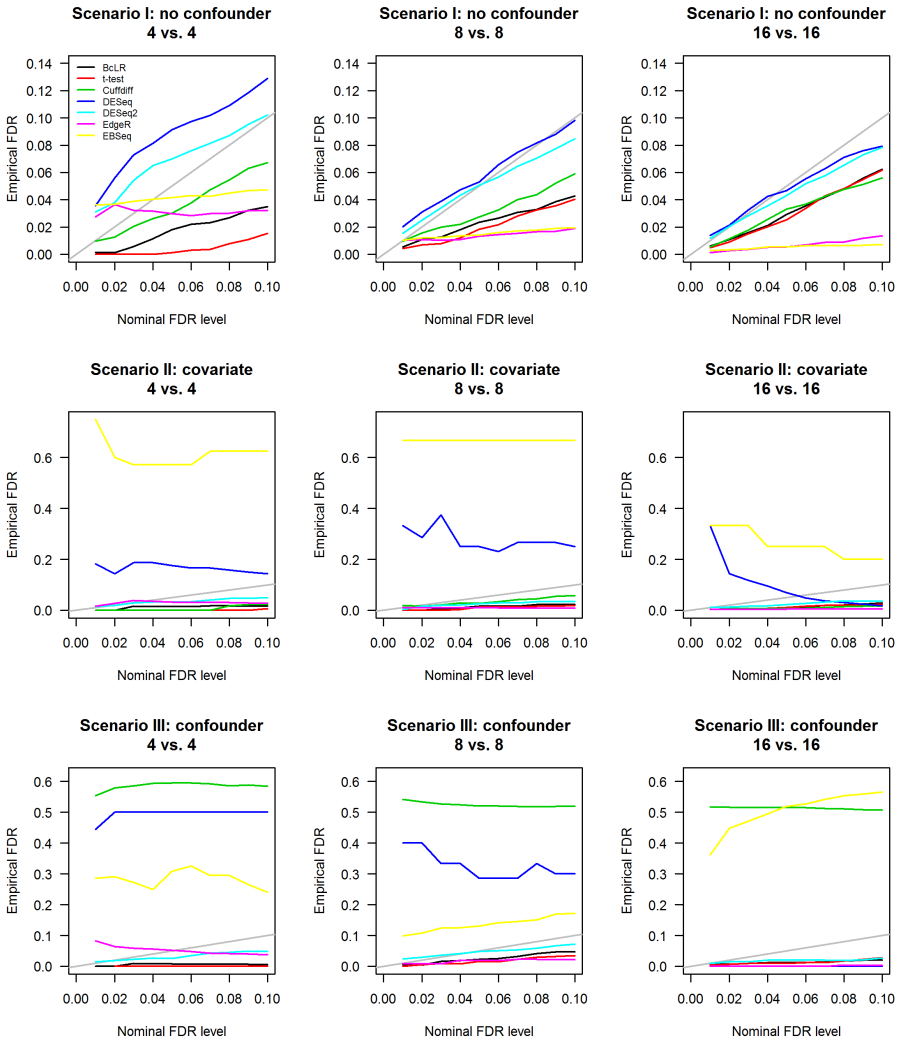


Figure 4.2: Empirical FDR vs nominal FDR. Empirical FDR was computed as the fraction of the true non-DE features among those declared to be DE by the specified software package. Nominal FDR level was the FDR threshold given to the specified package to determine the set of DE features.

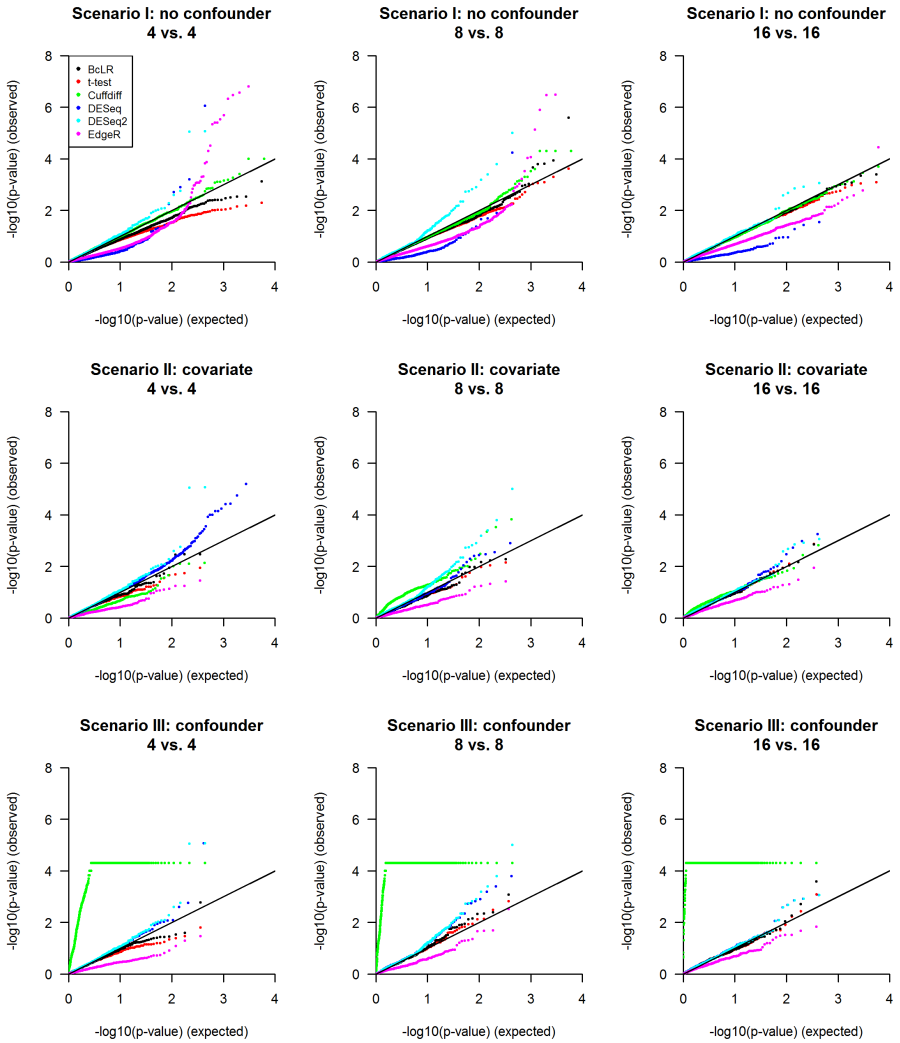


Figure 4.3: Q-Q plot of log-transformed raw p-values of true non-DE transcripts under the null hypothesis. The raw p-values exported by each method for transcripts that are not differentially expressed in each scenario are log-transformed, and then plotted against a log-transformed $\mathtt{Uniform}(0, 1)$ distribution.

In Scenario II where the expression of some genes are influenced by a non-confounding variable, DESeq adn DESeq2 continue to deviate from the diagonal line, albeit in a less pronounced manner. So does edgeR, but in a different direction compared to its pattern in Scenario I. Inclusion of a covariate seems to dampen the level of deviation of the p-values from the uniform distribution.

In Scenario III where a confounding covariate is influencing the expression of a subset of genes, DESeq, DESeq2 and edgeR maintain their deviation from the diagonal line, but on an even lower scale. Cuffdiff shows severe deviation from the diagonal line, which is consistent with its grossly inflated FDR in Figure 4.2. It's also apparent that the p-value plateaus below a certain point for Cuffdiff, and this is due to the fact that the sampling scheme used by Cuffdiff caps the smallest p-values at a fixed point, resulting in an accumulation of identical p-values for the first few hundred most significant features.

In summary, the methods without treatment for estimation uncertainty all show some level of deviation from the diagonal line of the Q-Q plot, indicating violation of model assumptions. More importantly, both tests used by MetaDiff, BcLR and Student-*t* test are close to the diagonal line in all three scenarios, suggesting superior performance under the null hypothesis.

[iii] Power comparison

Results so far have shown that the tests used in MetaDiff have empirical FDR under control and are well-behaved under the null hypothesis. Next, we wish to compare the power of these methods in detecting DE features with the usual FDR threshold levels. A range of nominal FDR levels $[0.01 - 0.1]$ are used to identify the DE features using the FDR-adjusted p-values from all methods with three different sample size setting under all three scenarios. The numbers of DE features are counted and divided by the total number of true DE transcripts under their respective scenarios to arrive at the empirical power, which is subsequently plotted against the nominal FDR level used in Figure 4.4. Between the BcLR test and Student-*t* test used by MetaDiff, the former clearly has higher power when sample size is 4+4. The two tests show almost identical power when sample size is 8+8 or 16+16.

In Scenario I, for medium- or large-sized experiments, BcLR clearly has the best power among all methods compared. When the sample size is small, DESeq and DESeq2 have better power in comparison to BcLR. But their results should be taken with more caution since in this setting, they also exhibit inflated FDR, as is shown in Figure 4.2.

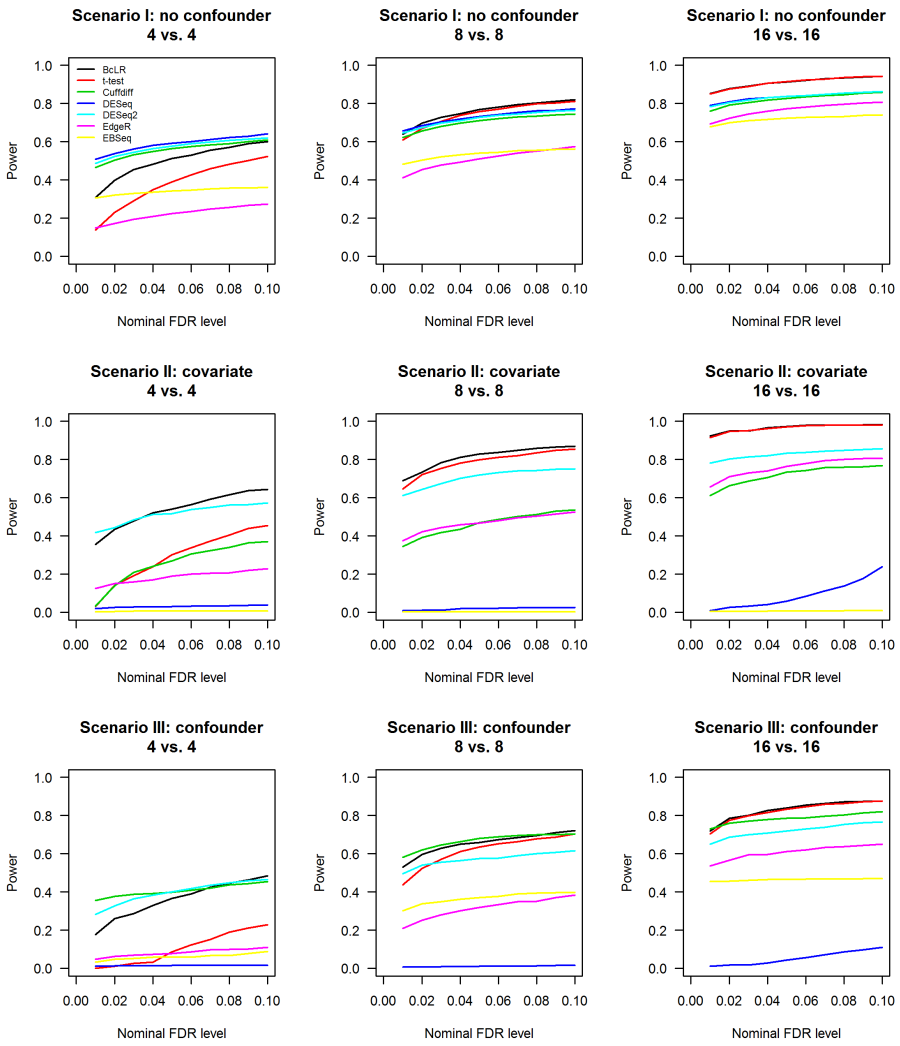


Figure 4.4: Power comparison. Power is calculated as the fraction of the correctly identified DE features among all true DE features. FDR-adjusted p-values from each method are subject to filtering with various nominal FDR thresholds, the features passing each threshold will be counted, and divided by the total number of true DE features to arrive at the estimated power for this method at this threshold. Estimated power is plotted against the nominal FDR threshold level for each method with three different sample size settings in all three scenarios.

In Scenario II, BcLR has the best power among all methods regardless of sample size. Student-t test has almost identical power as BcLR in medium and large size settings, but its power in small sample size experiments is limited. DESeq2 comes in second in comparison to BcLR overall, followed by edgeR and CuffDiff. It's interesting to notice that EBSeq and DESeq have almost no power in Scenario II. As a sanity check, the power of these two methods on genes that are not confounded by age is calculated, and they have similar power to that observed in Scenario I on these genes. This implies that DESeq and EBSeq are simply not robust to genes with expression influenced by a continuous covariate.

In Scenario III, BcLR continues to have the best power of all methods except for CuffDiff. CuffDiff seemingly performs better when the sample size is relatively small, however, it also has severely inflated FDR in Scenario III. Combining these two facts, we postulate CuffDiff cannot efficiently distinguish the true DE features from the non-DE ones, but simply label the majority of them DE indiscriminately. DESeq2 comes after BcLR, followed by edgeR and EBSeq. DESeq again has no power in this scenario.

[iv] Application in heart failure data

We apply MetaDiff and the other methods assessed above on an RNA-Seq dataset from a study on human heart failure. It is a relatively small dataset with 3 controls and 4 cases. And among these 7 subjects, 4 are male and 3 are female. In addition, the participants have a wide range of age at the time of the study. Left ventricular free-wall tissue was harvested from each heart and snap frozen until RNA-Seq sample preparation and sequencing, which was performed at the High-Throughput Sequencing Facility of Penn Genome Frontiers Institute following standard protocols. On average, the sequencing yielded 43 million 2×101-bp paired-end reads. The raw FASTQ data was mapped to hg19 human genome with TopHat, and isoform expression was estimated with Cufflinks.

The results are summarized in Table 4.2. Consistent with our simulation studies, when the sample size is small, BcLR detects fewer DE isoforms compared to DESeq, DESeq2 and EBSeq. It is mainly because in this scenario these three methods tend to have severely inflated FDR for experiments with small sample size. Hence these results shall be taken with extreme caution. Interestingly after adjustment by age and sex, BcLR test discovered significantly more genes than the rest of the methods. This is also consistent with our simulation studies, in which BcLR displayed the highest power for two of the three scenarios when sample size

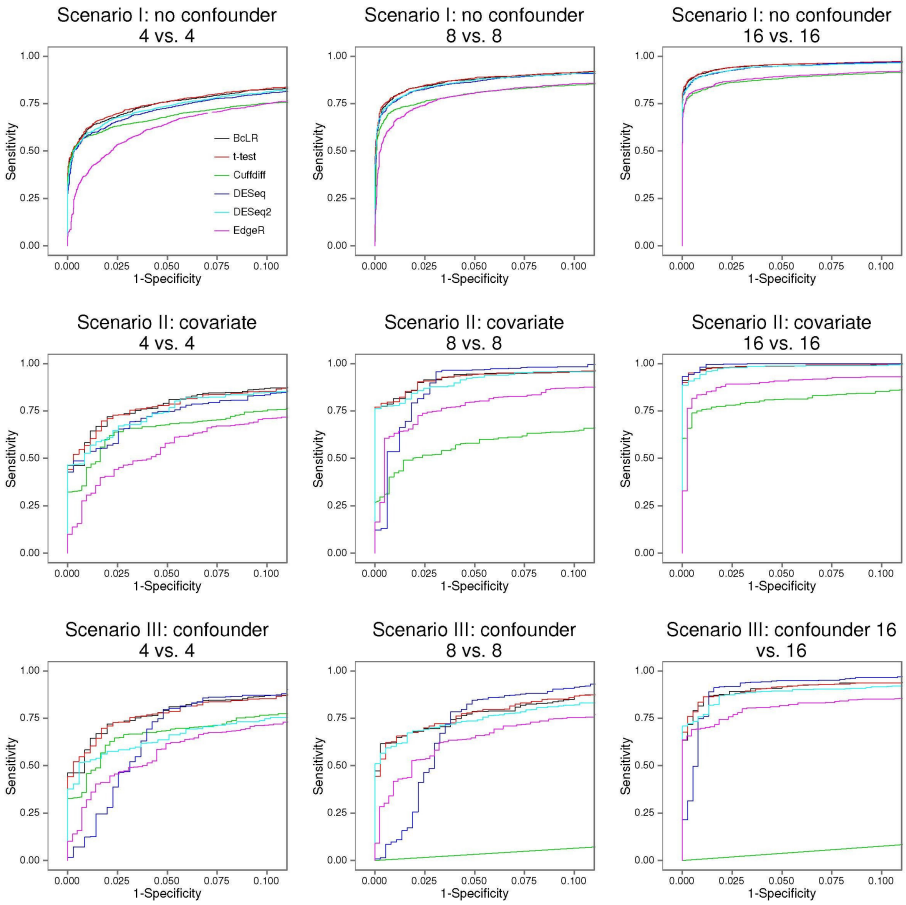


Figure 4.5: Zoomed in ROC curves.

is 4+4. In addition, with experiments with small sample size, BcLR showed consistently controlled FDR, which gave us confidence that the majority of the genes detected should be true DE. Among the transcripts labeled as DE by BcLR, the majority (71/94) showed significant covariates (p¡0.05) associated with either age or sex, indicating that the expression of the majority of these transcripts are in fact influenced by age or sex.

	Unadjusted	Age-sex-adjusted	Overlap
BcLR	6	95	1
t-test	1	0	0
DESeq	106	77	56
DeSeq2	102	49	31
EdgeR	3	0	0
Cuffdiff	7	-	-
EBSeq	256	-	-

Table 4.2: Number of isoforms detected in heart failure data.

4.2. Splicing QTL

Misregulation of alternative splicing could potentially trigger the onset of a variety of diseases, such as β-thalassemia (Treisman, Orkinl, and Maniatis, 1983; Treisman et al., 1982), spinal muscular dystrophy (Cartegni and Krainer, 2002; Kashima and Manley, 2003), amyotrophic lateral sclerosis (Kim et al., 2013) and cancer (Hahn and Scott, 2012; Imielinski et al., 2012). Therefore, identifying regulatory elements of alternative splicing is pivotal in illustrating the mechanisms of a large number of diseases.

The regulation of alternative splicing is primarily mediated by *cis*-regulatory elements or *trans*-acting factors. *Cis*-regulatory elements reside in close proximity to the ORF of the gene, and regulate gene expression through direct promoter activation or silencing. On the other hand, *trans*-acting factors need not be located in the vicinity of the gene. Coupled with genotyping microarrays, RNA-Seq technology has provided an unprecedented opportunity to identify the single nucleotide polymorphisms (SNPs) to which the alternative splicing of certain genes is associated, *i.e.*, splicing quantitative trait loci (sQTL).

Pioneering studies to identify sQTLs used simple linear regressions with the percentage of exon read counts over total gene read counts as the dependent variables, and the SNP genotypes as the covariates (Montgomery et al., 2010; Pickrell et al., 2010). These methods failed to include a

large portion of relevant information in the RNA-Seq studies, most importantly, the variability of the read counts from RNA-Seq data, which leads to inflated Type I errors and false positive results.

To overcome these difficulties, Zhao et al. developed GLiMMPS, a generalized linear model for identifying sQTLs that accounts for the variability of the exon-specific and overall read counts (Zhao et al., 2013). Denote the number of the junction reads corresponding to the exon-including isoform as $y = IJ$, and that of those corresponding to the exon-skipping isoform as SJ. The number of junction reads representing the total expression of these two isoforms is $n = SJ + \frac{IJ}{2}$. The model computes the estimated exon inclusion level $\hat{\psi}$ as,

$$\hat{\psi} = \frac{y}{n} = \frac{\frac{IJ}{2}}{SJ + \frac{IJ}{2}}$$

Given n, the exon-including junction read count follows a binomial distribution,

$$y|n \sim Binom(n, \psi)$$

As in the case of detecting DE genes and isoforms, a simple binomial model is insufficient to describe the over-dispersion of the read counts from the RNA-Seq data. To model the extra variability, one could use a beta-binomial model, a negative binomial model, or simply add a multiplicative scale factor to the variance of the response. Zhao et al. instead chose the method developed by Browne et al. (Browne et al., 2005), which added a normally distributed random effect to each individual $\text{logit}(\psi_{ij})$. The variance of this error term is different for each SNP.

$$\mu_{ij} \sim N(0, \sigma^2_{uj})$$
$$\text{logit}(\psi_{ij}) = \beta_0 + \beta_j g_{ij} + \mu_{ij}$$

The random error term μ_{ij} is a combination of two different sources of variation, the overall read-depth variability as well as variation of the exon inclusion level for the same SNP. Laplace approximation was used to estimate the parameters and a likelihood ratio test was used to compute the p-values for the fixed effect β_j for each SNP j.

Despite the significant improvement of GLiMMPS over the simple linear regression methods, it still bears several caveats.

[i] GLiMMPS only used the junction reads, discarding all the information contained in the reads mapped to the body of the exons. This reduces the accuracy of the estimation for the exon inclusion levels, thus affecting the power of the test.

[ii] GLiMMPS cannot utilize the extra information contained in paired-end data. As paired-end RNA-Seq experiments become increasingly accessible and popular, the capability of including paired-end experiments in the method design is highly desirable.

[iii] GLiMMPS treats the variation in the estimated exon inclusion level from the same genotype group as a random effect. This effectively forces sharing of a common variation parameter across exon inclusion levels from different samples, while in actuality this assumption might not hold due to variable library sequencing depths and coverage of the specific genomic area.

[iv] GLiMMPS assumes uniform sampling along the transcripts. Several studies have shown that this does not hold for RNA-Seq experiments, and the ignorance of the non-uniformity leads to biased estimation of isoform expression (Hu et al., 2014).

Our approach involves a two-step procedure, the first step is to estimate the exon-inclusion levels with PennSeq (Hu et al., 2014), a recently developed read-based method corrected for non-uniform RNA-Seq sampling. The PennSeq algorithm considers all reads mapped to a given exon trio, including junction and non-junction reads. It intrinsically supports paired-read sequencing data, and allows unique non-uniform distributions for each isoform. Using an expectation-maximization (EM) algorithm, PennSeq outputs the estimated mean and variance of the relative abundance of the exon-including isoform, which can then be fed into downstream models as the response variable.

4.2.1. Random effects meta-regression

Several choices arose when searching for a suitable downstream models to compute the associations between the genotype of a SNP and the exon inclusion level of a specific exon trio. In order to account for estimation uncertainty of the isoform expression, a random effects meta-regression model is considered with the following model setup. Denote the estimated isoform relative abundance as Y_i and its standard deviation σ_{1i}. The random effects meta-regression model can be

written as,

$$\begin{aligned}
&\text{logit}(Y_i) = \beta_0 + \beta_1 G_i + \mu_i + \epsilon_i \\
&\mu_i \sim N(0, \sigma^2_{1i}) \\
&\epsilon_i \sim N(0, \sigma^2_e)
\end{aligned}$$

Assume μ_i and ϵ_i are uncorrelated, and n observations are independent, this model can be estimated with standard meta-regression packages such as metafor in R.

4.2.2. Beta regression

Since the exon inclusion levels are random variables in the range $(0, 1)$, β-distribution is an intuitive choice for modeling this variable. Unlike meta-regression, beta regression allows for direct usage of the exon inclusion levels as the dependent variable without transformation (Ferrari and Cribari-Neto, 2004). Beta regression is based on an alternative parametrization of the β-distribution.

$$\begin{aligned}
&Y_i \sim \mathcal{B}[\mu, \phi] \\
&E(Y_i) = \mu \\
&\text{Var}(Y_i) = \frac{\mu(1-\mu)}{1+\phi} \\
&f(Y_i; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma[(1-\mu)\phi]} Y_i^{\mu\phi-1}(1-Y_i)^{(1-\mu)\phi-1}
\end{aligned}$$

where $\mu \in (0, 1)$ and $\phi > 0$. Now the beta regression model can be written down,

$$\text{logit}(\mu_i) = \beta_0 + \beta_1 G_i$$

Fitting of the beta-regression model is performed with betareg package in R. For identifying significant sQTLs, Wald tests are performed to test the null hypothesis $H_0 : \beta_1 = 0$.

4.2.3. Generalized linear mixed effects model

Inspired by GLiMMPS, a generalized linear mixed effects model was fitted with the estimated exon inclusion levels as the response variables. Since we used a much more accurate approach to estimate the exon-inclusion levels, We expect our GLMM method to have greater power than

GLiMMPS. For sample i, denote the total number of reads mapped to an exon trio as M_i, and the exon inclusion level as Y_i. Then the estimated number of reads from a certain exon $R_i = M_i Y_i$, where $R_i \sim Binom(M_i, Y_i)$. With this setup, the GLMM can be written as,

$$\text{logit}\,[E(Y_i)] = \beta_0 + \beta_1 G_i + \epsilon_i$$
$$\epsilon_i \sim N(0, \tau^2)$$

Similar to GLiMMPS, we also use the $lme4$ package in R to fit this mixed-effects model.

4.2.4. Data simulation

In order to assess the performance of the above three methods as well as the GLiMMPS, we simulated paired-end RNA-Seq data with Flux Simulator (Griebel et al., 2012) from the annotated human genome version 19. RefSeq genes were filtered to select those non-overlapping with at least two isoforms and three exons. For each gene, we chose the longest isoform and generated a shorter isoform by randomly removing one of the interior exons, resulting in 4,710 exon trios in the final list. The SNP genotypes were assumed to follow the Hardy-Weinberg equilibrium with a minor allele frequency (MAF) of 0.4. The exon inclusion levels are determined by,

$$Y_i = \text{expit}(-0.35 + \beta_1 G_i + \epsilon_i)$$
$$\epsilon_i \sim N(0, 0.05^2)$$

β_1 was set to $\log(1.2)$ for half of the exon trios (true sQTL), and 0 for the other half (true non-sQTL). We simulated the raw reads of 120 individuals in FASTQ format with 10 million 76 bp paired-end reads per sample. Each simulated data set is aligned to the hg19 genome with TopHat, and exon inclusion levels were estimated with the PennSeq algorithm. Testing of sQTLs were done with GLiMMPS, meta-regression (PSMeta), generalized linear mixed effect model (PSGLMM) and beta regression (PSBeta).

4.2.5. Results

[i] Comparison of exon inclusion level estimates.

GLiMMPS and PennSeq use different methods in estimating the exon inclusion levels, which

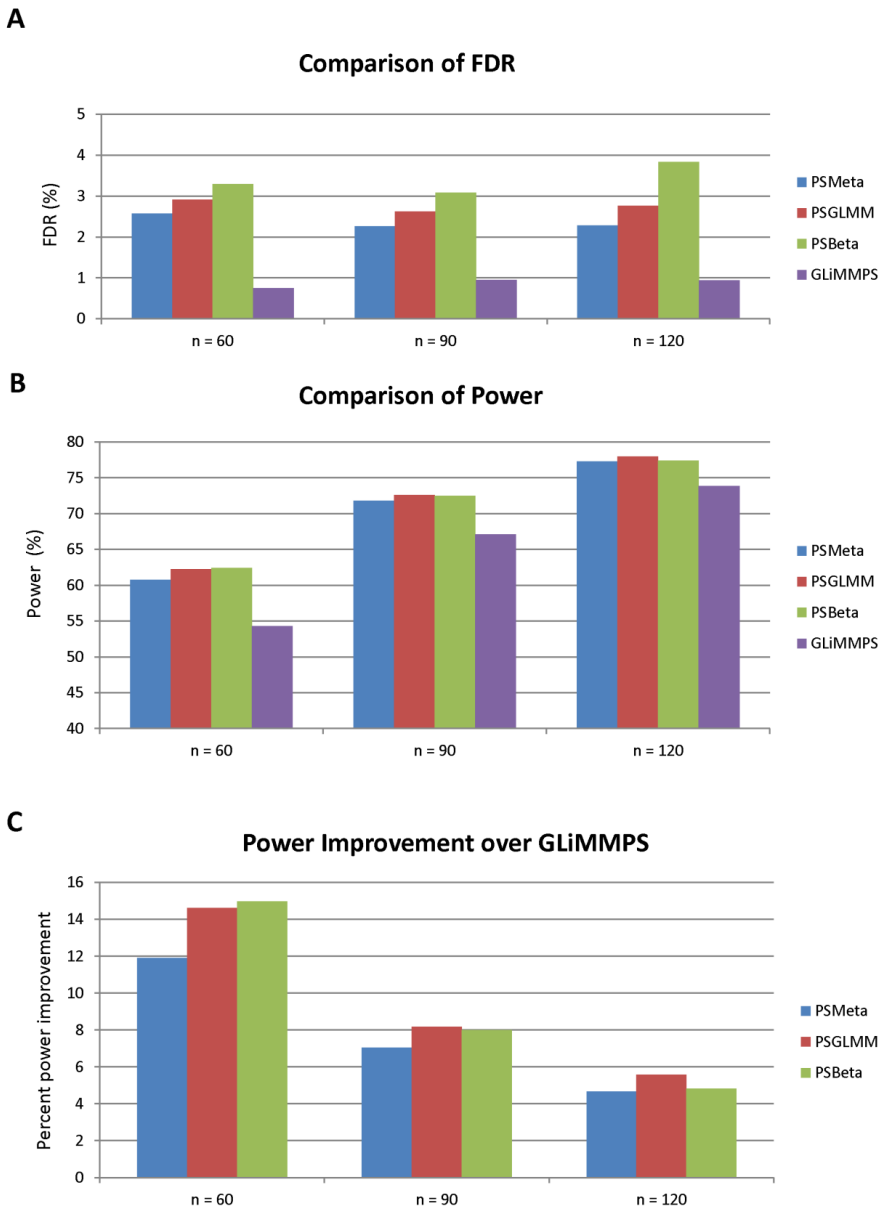


Figure 4.6: FDR and Power of PSMeta, PSGLMM, PSBeta and GLiMMPS. 60 and 90 subjects were randomly chosen from the pool of 120 subjects to form the experiment groups with smaller sample size. From each experiment, PSMeta, PSGLMM, PSBeta and GLiMMPS were used to test for significant association between the given genotype and the exon inclusion level estimates. P-values exported by these methods are FDR-adjusted using the BenjaminiHochberg procedure. Genes with FDR smaller than the threshold level 0.05 are labeled as significant. FDR is computed as the fraction of the true non-significant genes among genes labeled "significant" by each method. Power is computed as the fraction of the genes labeled "significant" by each method among all the true significant genes. Power improvement is computed as the percent improvement for the power of the specified method over the that of GLiMMPS.

can be pivotal in determining the final power of the sQTL testing. We computed the Pearson's correlation coefficients for the genes with $\beta_1 = \log(1.2)$ between the estimated exon inclusion levels and true value. PennSeq yielded significantly more accurate estimate than GLiMMPS. Out of 120 subjects, 102 (85%) showed higher correlation between PennSeq estimates and the true value than that between GLiMMPS estimates and the true value.

[ii] Comparison of false discovery rate and power

For our simulation studies, the FDR of all the methods are less than the nominal threshold (Figure 4.6A). Therefore, at level 0.05, all methods have their false discovery rate under control. We also note that GLiMMPS is overly conservative on this metric, with its FDR generally under 1%, which is less than $1/5$ of the nominal FDR. This conservativeness in turn affects the power of GLiMMPS in identifying true sQTLs. As is shown in Figure 4.6B, all the rest of the methods have superior power in comparison to GLiMMPS at the nominal FDR 0.05. When sample size is small at $\mathrm{n} = 60$, PSBeta has the higest power, followed closely by PSGLMM and PSMeta. All of these methods had at least 12% improvement in power over GLiMMPS. When sample size increases, PSGLMM outperformed PSBeta. When $\mathrm{n} = 90$, both PSGLMM and PSBeta achieved 8% improvement in power over GLiMMPS. When $\mathrm{n} = 120$, the power improvement reduces to 5%. This is intuitive due to the fact that as the number of sample increases, more information is available for GLiMMPS to estimate the exon inclusion levels with, while the relative benefit of utilizing constitutive reads in the flanking exons dwindles. Interestingly, our methods have similar power with 90 samples to GLiMMPS with 120 samples. This effectively saves significant time and resources for users, as 33% less samples are needed with our methods to achieve the same power.

[iii] Impact of non-uniformity

PennSeq accounts for the non-uniformity of the sampling process of RNA-Seq, while GLiMMPS assumes simple uniform sampling. This will potentially cause reduced power for GLiMMPS when used on transcripts experiencing severe non-uniformity. To test this hypothesis, we took advantage of the simulation metric, fraction of coverage, defined as the fraction of the transcript covered by reads. Using this metric internally calculated by Flux, we picked the genes ranked at the bottom $1/3$ in terms of mean fraction of coverage across samples in our simulation study to recompute the FDR and power of all the methods. These genes tend to

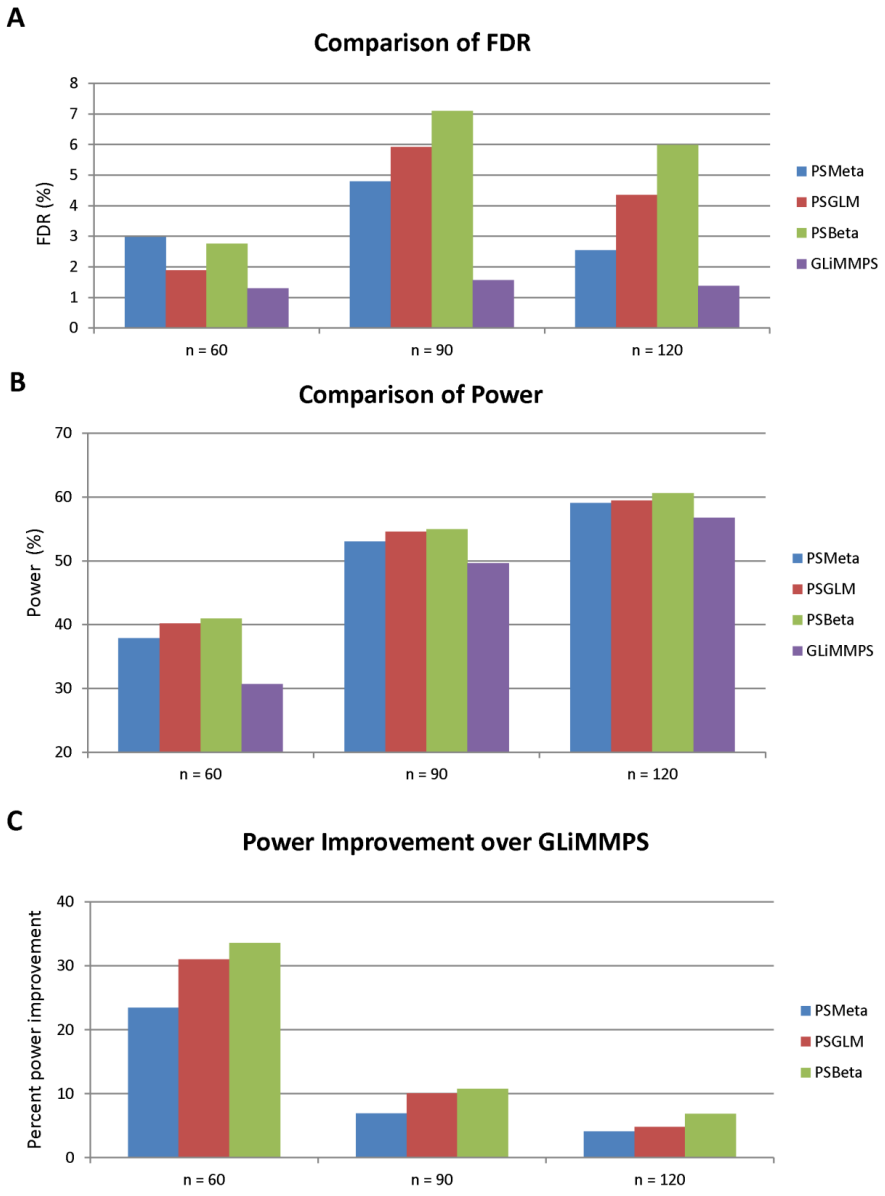


Figure 4.7: FDR and Power of PSMeta, PSGLMM, PSBeta and GLiMMPS for low-coverage genes. 60 and 90 subjects were randomly chosen from the pool of 120 subjects to form the experiment groups with smaller sample size. Only genes ranked at the bottom third in terms of sequencing coverage are included. From each experiment, PSMeta, PSGLMM, PSBeta and GLiMMPS were used to test for significant association between the given genotype and the exon inclusion level estimates. P-values exported by these methods are FDR-adjusted using the BenjaminiHochberg procedure. Genes with FDR smaller than the threshold level 0.05 are labeled as significant. FDR is computed as the fraction of the true non-significant genes among genes labeled "significant" by each method. Power is computed as the fraction of the genes labeled "significant" by each method among all the true significant genes. Power improvement is computed as the percent improvement for the power of the specified method over the that of GLiMMPS.

have (1) less coverage, *i.e.*, less information to infer the isoform expression and (2) severe non-uniformity due to the limited coverage and the stochasticity of the sampling process. The results have corroborated our hypothesis, as is shown in Figure 4.7. Unsurprisingly, the difference between our methods and GLiMMPS is more pronounced when we focus on the genes more deviated from the uniform sampling assumption. We notice that PSGLMM and PSBeta have slightly inflated false discovery rate when the sample size is relatively large. But the FDR of PSMeta is under control for all sample sizes. All methods had their power reduced to a certain extent, with GLiMMPS affected more severely than others. Compared to the power obtained from all simulated exon-trios, PSMeta, PSBeta, and PSGLMM experienced a power loss between 34-38%, whereas GLiMMPS experienced a power loss of 44%. In addition, the power improvement of our methods over GLiMMPS is more significant, especially when the sample size is small. For example, when $\mathrm{n} = 60$, PSMeta achieves a 23% improvement in power over GLiMMPS, which is twice of the improvement with all genes considered. This analysis implies that sampling non-uniformity can cause serious power loss for GLiMMPS, especially when sample size is small. And this loss of power can be effectively rescued by using PSMeta.

[iv] Real data application.

We tested the performance of our methods on a real RNA-Seq data of 91 lymphoblastoid B cell lines from the CEU population (Utah residents with ancestry from northern and western Europe) generated by Lappalainen et al. (Lappalainen et al., 2013) for the International HapMap Project. Each sequenced sample contains approximately 10 million 75 bp paired-end reads, mapped to the hg19 reference genome using the JIP pipeline. The Phase 1 genotyping dataset contains only 79 CEU samples, which reduces the number of samples with both RNA-Seq and DNA genotype data available to 78.

We focused our search on cis-sQTLs on chromosome 22, due to the limited sample size. The search for sQTLs of a specific exon trio is restricted to the genomic area from 200kb upstream of the trio to 200kb downstream. Only SNPs with Hardy-Weinberg p-value ¿ 0.0001 and MAF ¿ 0.1 were included. The final list contains 132 exon trios in 72 genes, 29,878 SNPs, and 80,074 exon-trio-SNP pairs. Benjamini-Hochberg procedure was used to adjust for multiple testing, and a exon-trio-SNP pair was declared significant if the FDR-adjusted p-value was smaller than 0.05. If we assume the majority of the exon-trio-SNP pairs were null, *i.e.*, there

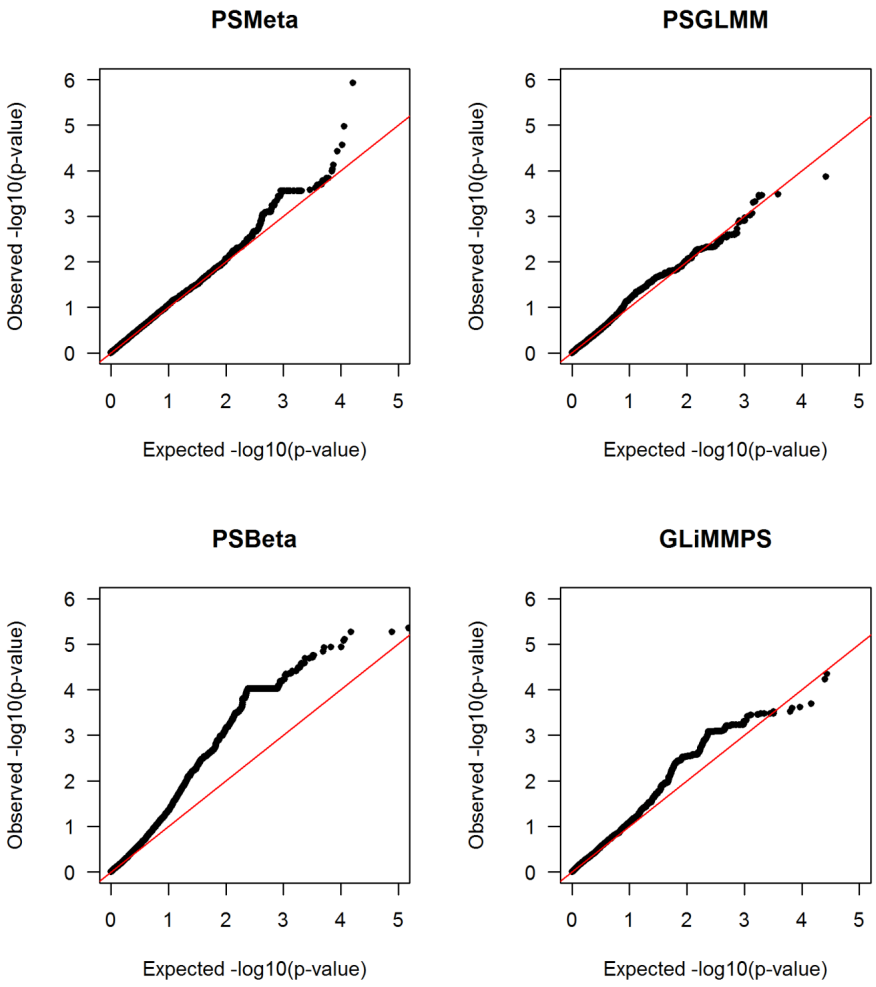


Figure 4.8: Quantile-quantile (Q-Q) plot for the negative log10 transformed raw p-values of each method. The raw p-values generated from the CEU population were transformed with a negative log function with base 10. The transformed p-values were sorted and plotted against the negative log10 transformed expected value of the same quantile from a $\mathrm{Uniform}(0, 1)$ distribution.

is no significant association between the exon inclusion level of the trio and the genotype of the SNP, we can use this real data to gauge the performance of the methods under the null. In Figure 4.8, we plotted the quantile-quantile (Q-Q) plot of the negative log10 transformed raw p-values from each method. PSBeta shows early deviation from the diagonal line, implying violation of its model assumption. We suspect it's due to the fact that the exon inclusion levels might not be distributed according to a β-distribution, and its function form could be potentially overly restrictive, causing inflated FDRs and early deviation from the null. GLiMMPS also displayed early deviation from the null, while PSMeta and PSGLMM had most of its points fall on the diagonal line, with deviation occurring when the p-values are extremely small. This result is consistent with the previous analyses of the behavior under the null for these methods, and with real RNA-Seq data, we have shown that the two best methods among the four tested are PSMeta and PSGLMM.

CHAPTER 5

ACCOUNTING FOR TECHNICAL NOISE IN SINGLE-CELL RNA SEQUENCING ANALYSIS

5.1. Motivation

As we have reviewed above, current scRNA-seq protocols are complex, often introducing technical biases that vary across cells (Hicks, Teng, and Irizarry, 2015), which, if not properly removed, can lead to severe type I error inflation in differential expression analysis. Compared to bulk RNA sequencing, in scRNA-seq the reverse transcription and preamplification steps lead to dropout events and amplification bias, the former describing the scenario in which a transcript expressed in the cell is lost during library preparation and is thus undetectable at any sequencing depth. In particular, due to the high prevalence of dropout events in scRNA-seq, it is crucial to account for them in data analysis, especially if conclusions involving low to moderately expressed genes are being drawn Pierson and Yau, 2015. In handling dropout events, existing studies take varying approaches: some ignore dropouts by focusing only on highly expressed genes (Shalek et al., 2013, 2014), some model dropouts in a cell-specific manner (Finak et al., 2015; Kharchenko, Silberstein, and Scadden, 2014; Kim et al., 2015; Vallejos, Marioni, and Richardson, 2015), while others use a global zero-inflation parameter to account for dropouts Pierson and Yau, 2015. Since each cell is processed individually within its own compartment during the key initial steps of library preparation, technical parameters that describe amplification bias and dropout rates should be cell-specific in order to adjust for the possible presence of systematic differences across cells. One way to quantify these biases, adopted by existing noise models (Finak et al., 2015; Kharchenko, Silberstein, and Scadden, 2014; Kim et al., 2015; Vallejos, Marioni, and Richardson, 2015), is to make use of spike-in molecules that comprise a set of external RNA sequences such as the commonly used external RNA Controls Consortium (ERCC) spike-ins (Baker et al., 2005), which are added to the cell lysis buffer at known concentrations (Bacher and Kendziorski, 2016; Stegle, Teichmann, and Marioni, 2015).

In the review above, we have looked at existing methods in great detail. To reiterate, an ideal method for modeling technical noise in scRNA-seq should

- model the inflated zeros in a cell-specific fashion;
- model the probability of zero inflation with consideration of the gene expression level
- model the amplification bias in a cell-specific manner
- allow incorporation of information from ERCC controls
- allow adjustment for additional covariates

Existing methods fail in at least one of the above criteria. Therefore, in this dissertation, we propose a new statistical framework that allows a more robust utilization of spike-ins to account for cell-specific technical noise. To obtain reliable estimates of cell-specific dropout parameters, we develop an empirical Bayes procedure that borrows information across cells. This is motivated by the observation that, although each cell has its own set of parameters for characterizing its technical noise, these parameters share a common distribution across cells which can be used to make the cell-specific estimates more stable. We demonstrate an application of this general framework by a likelihood-based test for differential expression. An advantage of the proposed framework over the existing approaches is that it can flexibly and efficiently adjust for cell-specific covariates, such as cell cycle stage or cell size, which may confound differential expression analysis.

5.2. Generative model of single-cell RNA sequencing

In scRNA-seq data, we have observed that the relationship between the mapped read count for a gene and its true expression level in a cell can be characterized using two functions, shown in Figure 5.1. Figure 5.1 shows examples of the relationships depicted in Equations-5.1 and -5.2 in the Zeisel et al. data (Zeisel et al., 2015). This scRNA-seq dataset is from murine brain cells acquired from Zeisel et al (Zeisel et al., 2015). It contains counts of 19,972 endogenous genes and 57 ERCC spike-ins of 3,005 cells from various regions of mouse brain, counted with UMIs. The cells are categorized into nine level-1 classes and 48 level-2 classes, with the level-2 classes considered relatively homogenous. In this paper, we focus our analyses on two level-2 classes, CA1Pyr1 and CA1Pyr2, which respectively contain 447 and 380 cells. The counts are preprocessed by selecting the top 25% of genes in total read account across the 827 cells, resulting in 6,405 genes in real

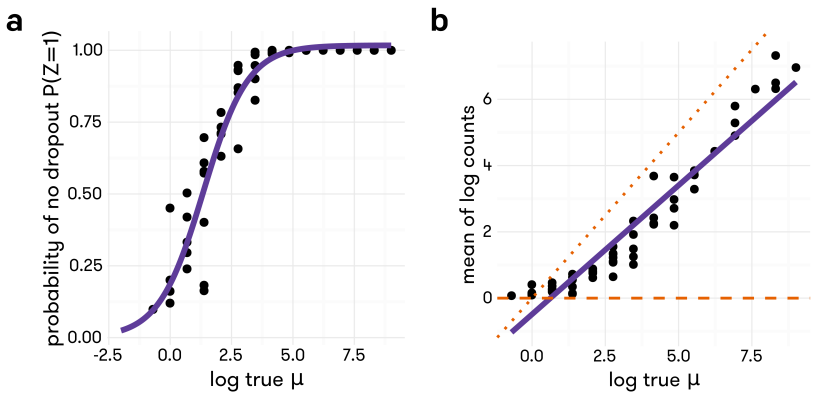


Figure 5.1: Proportion of cells with non-zero read count (in **a**) and mean across cells of log read count (in **b**) versus log true molecule count for ERCC spike-ins in Zeisel et al. data. Included in the plot are the best logistic curve fit (in **a**) and the best linear fit (in **b**).

data two-group comparison analysis. For studies involving class CA1Pyr2 only, selection of the top 25% of genes in the 447 cells yield 5,018 genes in the data set.

These relationships in Figure 5.1 have also been seen in other studies (Kharchenko, Silberstein, and Scadden, 2014; Kim et al., 2015). Note that the intercept α_c is negative, indicating incomplete capture efficiency of reverse transcription, and that the slope, β_c, when deviating from 1, reflects what is often called amplification bias. In experiments that use unique molecular identifiers (UMIs) (Islam et al., 2014), Y_{cg} is the molecule count, and β_c should be approximately 1. Together, Equations-5.1 and -5.2 characterize the technical noise specific to each cell.

5.2.1. Modelling spike-ins

In greater detail, let Y_{cg} be the observed number of reads or transcripts (if UMI is used) for the spiked-in molecule g in cell c. Let μ_g be the true number of molecules of g added to each cell lysate. Given the cell-specific technical parameters $(\alpha_c, \beta_c, \kappa_c, \tau_c)$, the distribution of Y_{cg} can be modelled with the following steps:

Step.*a* Let Z_{cg} be the indicator that dropout does not occur, i.e. the gene is captured in the library. The probability of $Z_{cg} = 1$ (π_{cg}) depends on the amount of added spike-in g, μ_g. A logistic model can be used to describe this relationship.

$$\begin{aligned} \pi_{cg} &= \text{expit}\left[\kappa_c + \tau_c \log \mu_g\right] \\ Z_{cg} &\sim \text{Bernoulli}\left(\pi_{cg}\right) \end{aligned} \tag{5.1}$$

Step.*b* Let λ_{cg} be the expected value for the read count of spike-in g in cell c.

$$\log \lambda_{cg} = \alpha_c + \beta_c \log \mu_g. \tag{5.2}$$

Step.*c* Given the status of Z_{cg}, the observed count for spike-in g in cell c Y_{cg} can be modelled as,

$$Y_{cg}|Z_{cg} \sim \begin{cases} \text{Poisson}(\lambda_{cg}), & \text{if } Z_{cg} = 1 \\ 0, & \text{if } Z_{cg} = 0 \end{cases}. \tag{5.3}$$

The conditional probability density function of Y_{cg} given Z_{cg} is,

$$\Pr[Y_{cg}|Z_{cg} = 0] = \begin{cases} 1, & \text{if } Y_{cg} = 0 \\ 0, & \text{if } Y_{cg} > 0 \end{cases} \tag{5.4}$$

$$\Pr[Y_{cg}|Z_{cg} = 1] = \frac{\lambda_{cg}^{Y_{cg}} e^{-\lambda_{cg}}}{y_{cg}!}. \tag{5.5}$$

Step.*d* We can arrive at the marginal likelihood of Y_{cg} by summing over the support of Z_{cg},

$$\begin{aligned} \Pr[Y_{cg}] &= \sum_{Z_{cg}} \Pr[Y_{cg}, Z_{cg}] \\ &= \sum_{Z_{cg}} \Pr[Y_{cg}|Z_{cg}] \Pr[Z_{cg}] \\ &= \Pr[Y_{cg}|Z_{cg} = 0] \Pr[Z_{cg} = 0] + \Pr[Y_{cg}|Z_{cg} = 1] \Pr[Z_{cg} = 1] \\ &= \begin{cases} 1 \cdot (1 - \pi_{cg}) + e^{-\lambda_{cg}} \pi_{cg}, & \text{if } Y_{cg} = 0 \\ 0 \cdot (1 - \pi_{cg}) + \dfrac{\lambda_{cg}^{Y_{cg}} e^{-\lambda_{cg}}}{y_{cg}!} \pi_{cg}, & \text{if } Y_{cg} > 0 \end{cases} \\ &= \begin{cases} 1 + \pi_{cg} \left(e^{-\lambda_{cg}} - 1 \right), & \text{if } Y_{cg} = 0 \\ \dfrac{\pi_{cg} \lambda_{cg}^{Y_{cg}} e^{-\lambda_{cg}}}{y_{cg}!}, & \text{if } Y_{cg} = 1 \end{cases}. \end{aligned} \tag{5.6}$$

Step.*e* Plug (5.1) and (5.2) into (5.6), the full likelihood of the spike-in RNA molecules can be expressed using the technical parameters $(\alpha_c, \beta_c, \kappa_c, \tau_c)$ and the amount of spike-in for

g, μ_g,

$$\Pr[Y_{cg}] = \begin{cases} 1 + \text{expit}(\kappa_c + \tau_c \log \mu_g)\left(e^{-e^{\alpha_c + \beta_c \log \mu_g}} - 1\right), & \text{if } Y_{cg} = 0 \\ \dfrac{\text{expit}(\kappa_c + \tau_c \log \mu_g)\left[e^{\alpha_c + \beta_c \log \mu_g}\right]^{Y_{cg}} e^{-e^{\alpha_c + \beta_c \log \mu_g}}}{y_{cg}!}, & \text{if } Y_{cg} = 1 \end{cases}, \quad (5.7)$$

where

$$\text{expit}[x] = \frac{1}{1 + \exp[-x]}. \quad (5.8)$$

5.2.2. Modelling biological genes

The above observations have motivated the model shown in Figure 5.2, where the true but unobserved absolute expression level μ_{cg} follows distribution F_g, the specification of which depends on the analysis objective. For example, for the common task of detecting differentially expressed (DE) genes between groups, we assume μ_{cg} follows a log-Normal distribution with mean θ_{gj} and variance σ_{gj}, where j is the group identifier. The log-Normal distribution has been demonstrated previously to be a useful model for single cell gene expression (Bengtsson et al., 2005), and lends computational simplicity to the estimation procedure. The technical noise in the cell is captured by the intermediate variables Z_{cg}, characterized by Equation 5.10, and λ_{cg}, characterized by Equation 5.11. Given Z_{cg} and λ_{cg}, the distribution of Y_{cg} is shown in Equation 5.12. Assuming F_g is in the form of log-normal distribution, the count of reads or transcripts for a biological gene g in cell c can be modeled with the following steps:

Step.*a* Given the cell-specific technical parameters, we assume the actual expression of gene g in cell c follows (Bengtsson et al., 2005),

$$\mu_{cg} \sim \text{LogNormal}\left(\theta_g, \sigma_g^2\right) \quad (5.9)$$

where θ_g and σ_g are the gene-specific parameters characterizing the mean and standard deviation of the log-normal distribution.

Step.*b* Let Z_{cg} be the indicator that dropout does not occur. The probability of $Z_{cg} = 1$ (π_{cg}) depends on the gene's true absolute expression in the cell, μ_{cg}. A logistic model can be

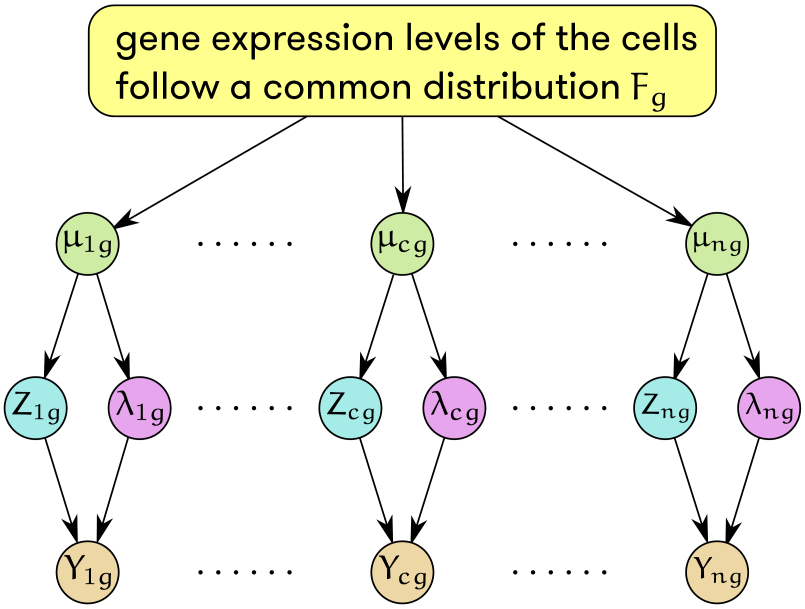


Figure 5.2: Schematic of TASC model for a single gene g across n cells, with μ_{cg} being true absolute expression, Y_{cg} being observed read count, and Z_{cg}, λ_{cg} being intermediate variables that model dropout and amplification, capture, and sequencing biases.

used to describe this relationship.

$$\pi_{cg} = \text{expit}\left[\kappa_c + \tau_c \log \mu_{cg}\right] \tag{5.10}$$
$$Z_{cg}|\mu_{cg} \sim \text{Bernoulli}\left(\pi_{cg}\right)$$

Step.*c* Given the cell-specific technical parameters $(\alpha_c, \beta_c, \kappa_c, \tau_c)$, let λ_{cg} be the expected value for the read count of spike-in g in cell c.

$$\log \lambda_{cg} = \alpha_c + \beta_c \log \mu_{cg} \tag{5.11}$$

Step.*d* Similar to the case of spike-in molecules, given the status of Z_{cg}, the observed count for gene g in cell c, Y_{cg}, can be modeled as,

$$Y_{cg}|Z_{cg}, \mu_{cg} = \begin{cases} \text{Poisson}(\lambda_{cg}), & \text{if } Z_{cg} = 1 \\ 0, & \text{if } Z_{cg} = 0 \end{cases} \tag{5.12}$$

And the conditional probability density function is,

$$\begin{aligned}
\Pr\left[Y_{cg}|Z_{cg}=0,\mu_{cg}\right] =& \begin{cases} 1, & \text{if } Y_{cg}=0 \\ 0, & \text{if } Y_{cg}>0 \end{cases} \\
\Pr\left[Y_{cg}|Z_{cg}=1,\mu_{cg}\right] =& \frac{\lambda_{cg}^{Y_{cg}} e^{-\lambda_{cg}}}{y_{cg}!} \\
=& \frac{\left[e^{\alpha_c+\beta_c \log \mu_{cg}}\right]^{Y_{cg}} e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}}}{y_{cg}!}.
\end{aligned}$$

Step.*e* The joint probability of Y_{cg}, Z_{cg} and μ_{cg} can be subsequently expressed as,

$$\begin{aligned}
\Pr\left[Y_{cg},Z_{cg},\mu_{cg}\right] =& \Pr\left[Y_{cg}|Z_{cg},\mu_{cg}\right]\Pr\left[Z_{cg},\mu_{cg}\right] \\
=& \Pr\left[Y_{cg}|Z_{cg},\mu_{cg}\right]\Pr\left[Z_{cg}|\mu_{cg}\right]\Pr\left[\mu_{cg}\right].
\end{aligned}$$

The marginal likelihood of Y_{cg},μ_{cg} can be computed by summing over the support of Z_{cg},

$$\begin{aligned}
&\Pr\left[Y_{cg},\mu_{cg}\right] \\
=&\Pr\left[Y_{cg},\mu_{cg},Z_{cg}=0\right]+\Pr\left[Y_{cg},\mu_{cg},Z_{cg}=1\right] \\
=&\Pr\left[Y_{cg}|Z_{cg}=0,\mu_{cg}\right]\Pr\left[Z_{cg}=0|\mu_{cg}\right]\Pr\left[\mu_{cg}\right] \\
&+\Pr\left[Y_{cg}|Z_{cg}=1,\mu_{cg}\right]\Pr\left[Z_{cg}=1|\mu_{cg}\right]\Pr\left[\mu_{cg}\right] \\
=&\begin{cases} (1-\pi_{cg})f_{LN}\left(\mu_{cg}|\theta_g,\sigma_g^2\right)+e^{-e^{\alpha_c+\beta_c\log\mu_{cg}}}\pi_{cg}f_{LN}\left(\mu_{cg}|\theta_g,\sigma_g^2\right), & \text{if } Y_{cg}=0 \\ \frac{\left[e^{\alpha_c+\beta_c\log\mu_{cg}}\right]^{Y_{cg}}e^{-e^{\alpha_c+\beta_c\log\mu_{cg}}}}{y_{cg}!}\pi_{cg}f_{LN}\left(\mu_{cg}|\theta_g,\sigma_g^2\right), & \text{if } Y_{cg}>0 \end{cases},
\end{aligned} \tag{5.13}$$

where

$$f_{LN}\left(\mu_{cg}|\theta_g,\sigma_g^2\right)=\frac{1}{\mu_{cg}\sigma_g\sqrt{2\pi}}\,e^{-\frac{\left(\ln\mu_{cg}-\theta_g\right)^2}{2\sigma_g^2}}. \tag{5.14}$$

Therefore, the marginal likelihood for Y_{cg} can be computed by integrating out μ_{cg},

$$\Pr\left[Y_{cg}\right]=\int_{\mu_{cg}}\Pr\left[Y_{cg},\mu_{cg}\right]d\mu_{cg}.$$

Assuming independence between cells, then the marginal distribution of $\mathbf{Y}_g = (Y_{1g}, \cdots, Y_{cg}, \cdots, Y_{Ng})$ can be expressed as,

$$\Pr[\mathbf{Y}_g] = \prod_{c=1}^{N} \int_{\mu_{cg}} \Pr[Y_{cg}, \mu_{cg}] \, d\mu_{cg}. \tag{5.15}$$

The parameters θ_g and σ_g^2 can therefore be estimated by maximizing the above marginal likelihood.

5.2.3. Empirical Bayes estimation of cell-specific technical parameters

Cell-specific technical parameters include

- α_c and β_c, characterizing capture and amplification efficiencies for any gene in cell c.
- κ_c and τ_c, characterizing the probability of any gene to be detected, *i.e.* not undetected due to technical dropout, in cell c.

They are estimated using ERCC spike-ins. From the generative model, we have arrived at the full marginal likelihood for Y_{cg} given the technical parameters $\Psi_c = (\alpha_c, \beta_c, \kappa_c, \tau_c)$. Maximum likelihood estimates (MLEs) can be obtained by optimizing the complete likelihood over the support of Ψ_c for cell c. However, in our simulations, naïve MLEs suffer from numerical instability and lack of convergence for κ_c and τ_c (Figure 5.3), which prompts us to derive a better strategy for estimating κ_c and τ_c.

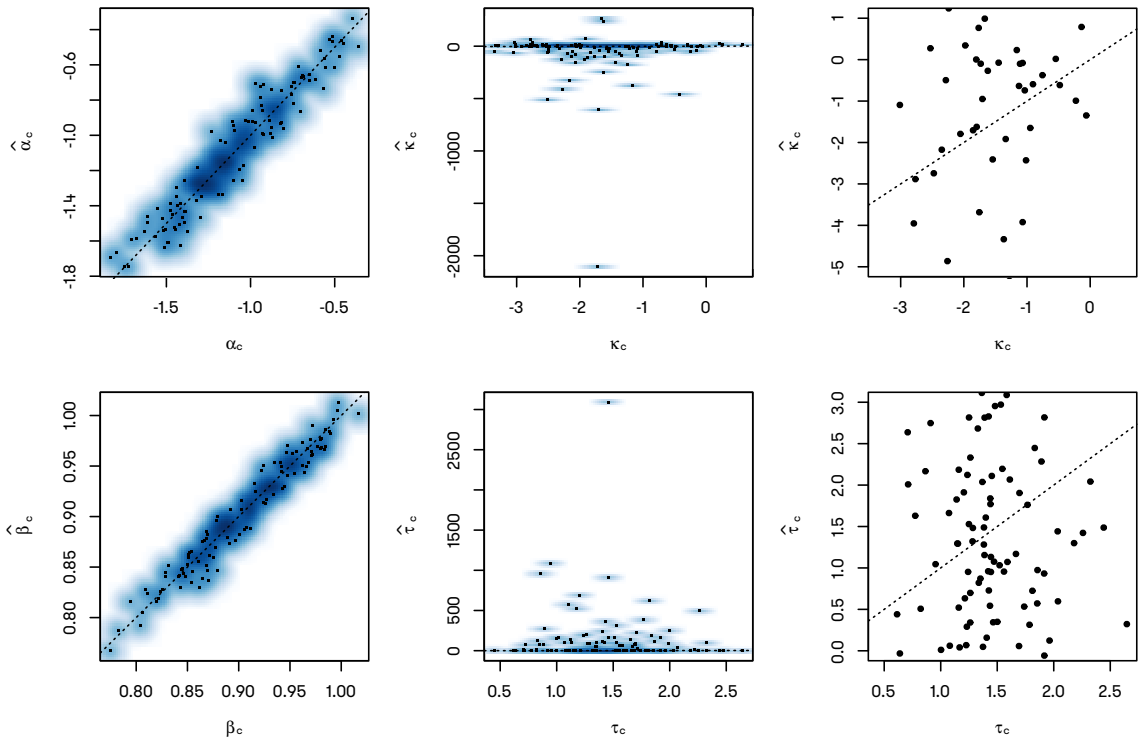


Figure 5.3: Comparing the maximum likelihood estimators of cell-specific technical parameters $\mathbf{\Psi}_c$ with their true values. Left panel: scatter plot comparing α_c (upper) and β_c (lower) estimated with maximum likelihood methods (y axis) to their true values (x axis). Middle panel: scatter plot comparing κ_c (upper) and τ_c (lower) estimated with maximum likelihood methods (y axis) to their true values (x axis). Right panel: scatter plot comparing κ_c (upper) and τ_c (lower) estimated with maximum likelihood methods (y axis) to their true values (x axis), zoomed in view. Identity line (dotted) is plotted for ease of comparison.

Upon further investigation, we have pinpointed the issues with likelihood estimators:

- due to the limitations of ERCC spike-ins, each cell contains little information w.r.t the drop-out probabilities due to paucity of spike-ins with low concentrations, thus necessitating borrowing information across cells if we wish to estimate the dropout-related parameters with better stability;

- we have observed that the κ_c and τ_c are negatively correlated, and similar relationships are observed for α_c and β_c as well (Figure 5.4). The estimating procedure can take advantage of this knowledge to model the correlation among the parameters.

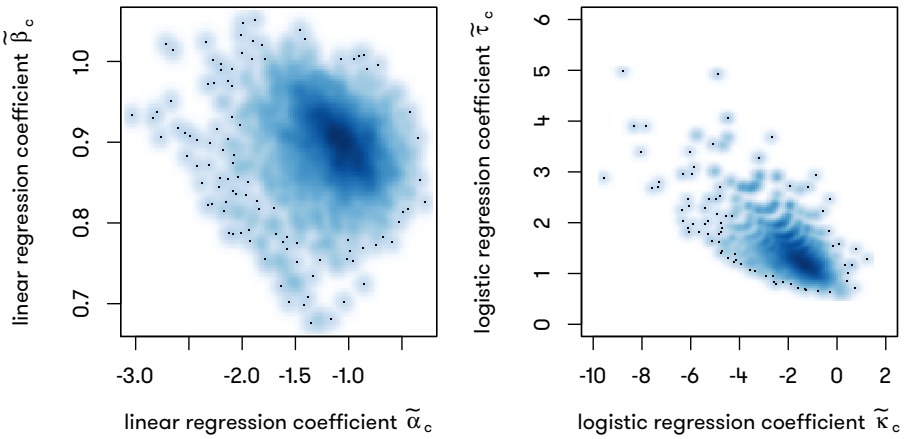


Figure 5.4: Scatter plot describing the correlation between α_c and β_c, and κ_c and τ_c. Left panel: $\tilde{\alpha}_c$ (y axis) compared to $\tilde{\beta}_c$ (x axis); both estimated from linear regressions. Right panel: $\tilde{\kappa}_c$ (y axis) compared to $\tilde{\tau}_c$ (x axis); both estimated from logistic regressions.

Taking the above observations into consideration, we propose an empirical Bayesian approach in which we assume the vector $\boldsymbol{\Psi}_c = (\alpha_c, \beta_c, \kappa_c, \tau_c)$ follows a multivariate normal distribution with mean ψ and covariance matrix $\Sigma_{\boldsymbol{\Psi}}$

$$\boldsymbol{\Psi}_c \sim \mathcal{N}(\psi, \Sigma_{\boldsymbol{\Psi}}). \tag{5.16}$$

Denote the observed read counts for the spike-in molecules as $\mathbf{Y}_c = \{Y_{cg}, g = 1, \cdots, G\}$, with G being the number of synthetic mRNA molecules added to the cell lysates. Assuming independence $Y_{c_1 g} \perp\!\!\!\perp Y_{c_2 g}$ for $c_1 \neq c_2$, the full likelihood for the observed Y_{cg} across cells can be written as,

$$\begin{aligned}\mathcal{L}[\psi, \Sigma_{\boldsymbol{\Psi}} | \mathbf{Y}] &= \prod_c \Pr[\mathbf{Y_c} | \psi, \Sigma_{\boldsymbol{\Psi}}] \\ &= \prod_c \int \Pr[\mathbf{Y_c} | \boldsymbol{\Psi}_c] \Pr[\boldsymbol{\Psi}_c | \psi, \Sigma_{\boldsymbol{\Psi}}] \, d\boldsymbol{\Psi}_c. \end{aligned} \tag{5.17}$$

Conditional on $\boldsymbol{\Psi}_c$, the probability density function of $\mathbf{Y_c}$ is just the likelihood in (5.7), and $\Pr[\boldsymbol{\Psi}_c | \psi, \Sigma_{\boldsymbol{\Psi}}]$ is the bivariate normal density function per our assumptions. To estimate the expected values of $\boldsymbol{\Psi}_c$ in the above models, we need to first compute the hyper-parameters $(\psi, \Sigma_{\boldsymbol{\Psi}})$ by maximizing the above likelihood. Due to the lack of closed form solutions, this calls for the numerical maximization of a numerically integrated function. The integration would be evaluated over 4 variables, and the maximization over 14 (4 for the mean and 10 for the covariance matrix) with a positive-definite re-

straint on Σ_{Ψ}. This numerical problem has turned out to be unsolvable for the current computational infrastructure accessible by the majority of our users.

We propose a computationally efficient approach to estimate the required parameters Ψ_c for all cells. We recognize that constraints on the covariance structure Σ_{Ψ} are necessary to reduce the dimensionality of our optimization. We assume Σ_{Ψ} is a diagonal block matrix by imposing independence between the vectors (α_c, β_c) and (κ_c, τ_c). Then we estimate these two vectors separately.

(α_c, β_c) can be estimated efficiently by fitting the linear regression with $\log[Y_{cg}]$ as the response variable and the amount of spiked-in ERCC molecules as the predictor variable, using only genes that are detected ($\{g, \text{s.t. } Y_{cg} > 0\}$),

$$\log \mathrm{E}[Y_{cg}] = \alpha_c + \beta_c \log \mu_g. \tag{5.18}$$

We recognize that this estimator is biased as a result of the data missing not at random (MNAR). However, in our simulation studies, this estimator does not show any discernible bias when compared to the truth (Figure 5.5), indicating the bias incurred by MNAR is under control.

On the other hand, the alternative estimators for (κ_c, τ_c) have proven to be a bit more elusive since the indicator of dropout is latent, *i.e.* we do not directly observe which zeros in our read counts are caused by technical dropouts *versus* Poisson sampling during sequencing. One approach is to assume all zeros are technical dropouts, and use logistic regression to estimate (κ_c, τ_c),

$$\operatorname{logit}(\Pr[Y_{cg} > 0 | \mu_g]) = \kappa_c + \tau_c \log \mu_g. \tag{5.19}$$

However, this has two drawbacks. First, this estimator is highly biased, since not all zeros are effects of technical dropout, and some of these zeros are due to the low expression of gene g in cell c. Second, since those genes with lower expression have a higher probability of dropping out, naïve logistic regressions could fail from complete or quasi-complete separation. Complete and quasi-complete separation happens when the outcome variable (in this case the event of being observed) separates a predictor (in this case $\log \mu_{cg}$) completely (complete) or very well to a certain extent (quasi-complete). In both cases, the coefficients associated with the affected covariates cannot be estimated. Our model requires that all of the cell-specific technical parameters be known for the

downstream computations, failure to estimate (κ_c, τ_c) will result in cell c being removed from the sample pool, thus causing unnecessary loss of data. The root of this issue is identical to that of the simple MLEs (Figure 5.3), therefore similarly some form of shrinkage is the key to stably estimating these two dropout-related parameters.

We propose the following steps to compute the cell-specific dropout parameters (κ_c, τ_c). Let $\boldsymbol{\delta}_c = (\kappa_c, \tau_c)$.

Step.*a* perform logistic regression of (5.19) and obtain $\hat{\boldsymbol{\delta}}_c$ for cells that do not exhibit complete or quasi-complete separations.

Step.*b* estimate prior of $\boldsymbol{\delta}_c$ by fitting a bivariate normal distribution using the estimated $\hat{\boldsymbol{\delta}}_c$ to compute the mean $\mathrm{E}[\boldsymbol{\delta}_c]$ and covariance matrix $\Sigma_{\boldsymbol{\delta}_c}$.

Step.*c* use the estimated mean and covariance matrix of $\boldsymbol{\delta}_c$ to compute the posterior mean of κ_c and τ_c. The complete probability density function for $\boldsymbol{\delta}_c$ and $\mathbf{Y}_c$ is

$$\begin{aligned}\Pr[\boldsymbol{\delta}_c, \mathbf{Y}_c] =& \Pr[\mathbf{Y}_c|\boldsymbol{\delta}_c] \Pr[\boldsymbol{\delta}_c] \\ =& \Pr[\boldsymbol{\delta}_c] \prod_g \Pr[Y_{cg}|\boldsymbol{\delta}_c] \\ =& f_N(\boldsymbol{\delta}_c|\mathrm{E}[\boldsymbol{\delta}_c], \Sigma_{\boldsymbol{\delta}_c}) \cdot \prod_g \Pr[Y_{cg}|\boldsymbol{\delta}_c]. \end{aligned} \tag{5.20}$$

The posterior distribution of δ_c is

$$\begin{aligned}\Pr[\boldsymbol{\delta}_c|\mathbf{Y}_c] =& \frac{\Pr[\boldsymbol{\delta}_c, \mathbf{Y}_c]}{\Pr[\mathbf{Y}_c]} \\ =& \frac{\Pr[\boldsymbol{\delta}_c, \mathbf{Y}_c]}{\int \Pr[\boldsymbol{\delta}_c, \mathbf{Y}_c] \, d\boldsymbol{\delta}_c}, \end{aligned} \tag{5.21}$$

with f_N being the PDF of a bivariate normal density and $\Pr[Y_{cg}|\boldsymbol{\delta}_c]$ is equal to (5.7) in form. The posterior mean of $\boldsymbol{\delta}_c$ can then be computed by integrating the PDF over the

support of bivariate random variable δ_c, *i.e.* $\mathbb{R}^2$.

$$\begin{aligned}
\mathrm{E}\left[\kappa_c|\mathbf{Y}_c\right] &= \int \kappa_c \Pr\left[\delta_c|\mathbf{Y}_c\right] \mathrm{d}\delta_c \\
&= \int \frac{\kappa_c \Pr\left[\delta_c, \mathbf{Y}_c\right]}{\int \Pr\left[\delta_c, \mathbf{Y}_c\right] \mathrm{d}\delta_c} \mathrm{d}\delta_c \\
&= \frac{\int \kappa_c \Pr\left[\delta_c, \mathbf{Y}_c\right] \mathrm{d}\delta_c}{\int \Pr\left[\delta_c, \mathbf{Y}_c\right] \mathrm{d}\delta_c} \qquad (5.22) \\
\mathrm{E}\left[\tau_c|\mathbf{Y}_c\right] &= \frac{\int \tau_c \Pr\left[\delta_c, \mathbf{Y}_c\right] \mathrm{d}\delta_c}{\int \Pr\left[\delta_c, \mathbf{Y}_c\right] \mathrm{d}\delta_c} \qquad (5.23)
\end{aligned}$$

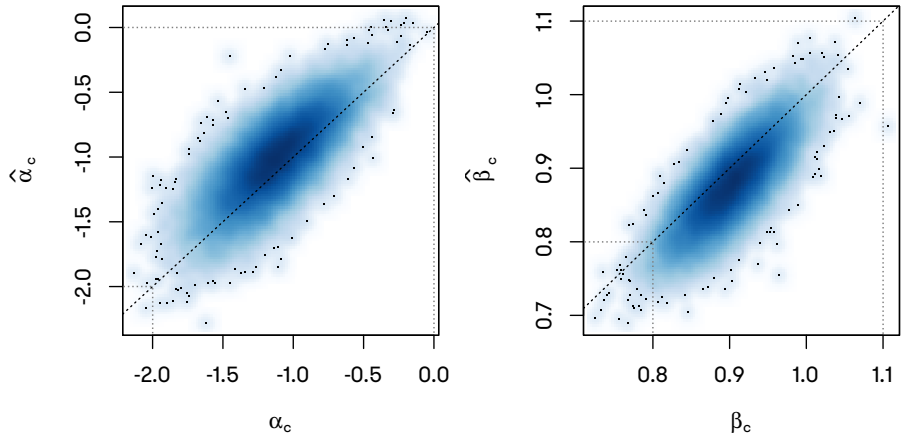


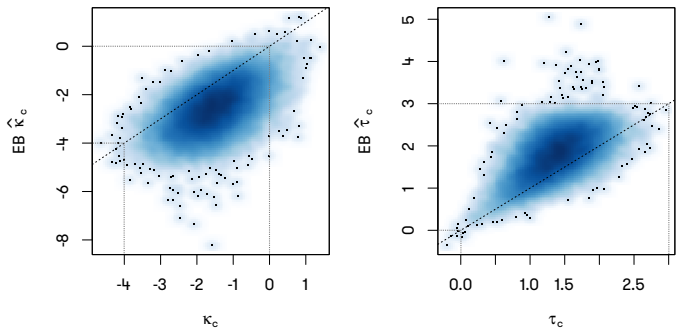
Figure 5.5: Comparing the estimated $\hat{\alpha}_c$, $\hat{\beta}_c$ to the true values of α_c and β_c. Left panel: $\hat{\alpha}$ estimated from linear regressions (y axis) compared to their true values (x axis). Right panel: $\hat{\beta}$ estimated from linear regressions (y axis) compared to their true values (x axis). Both panels: dotted lines represent the unit lines with intercept equal to 0, and slope equal to 1.

We have performed a series of simulation studies to assess the performance of the aforementioned estimators of α_c, β_c, κ_c and τ_c. Using the largest level 2 class in the Zeisel data (Zeisel et al., 2015), we have estimated the cell-specific parameters (α_c, β_c) and (κ_c, τ_c) using the method described above. Denote $\zeta_c = (\alpha_c, \beta_c)$. Two bivariate normal distributions are fitted to the estimated parameters $\hat{\zeta}_c$ and $\hat{\delta}_c$ to get the mean and covariance matrices of these two vectors, $\mathrm{E}\left[\zeta_c\right]$, Σ_ζ, $\mathrm{E}\left[\delta_c\right]$, and Σ_δ. New technical parameters are then sampled from the bivariate normal distributions $\mathcal{N}\left(\mathrm{E}\left[\zeta_c\right], \Sigma_\zeta\right)$ and $\mathcal{N}\left(\mathrm{E}\left[\delta_c\right], \Sigma_\delta\right)$. From these new technical parameters, counts of 57 ERCC spike-ins present in the Zeisel data (Zeisel et al., 2015) in 200 cells are generated according to the hierarchical model described above. The simulation is repeated 100 times to get a more compre-

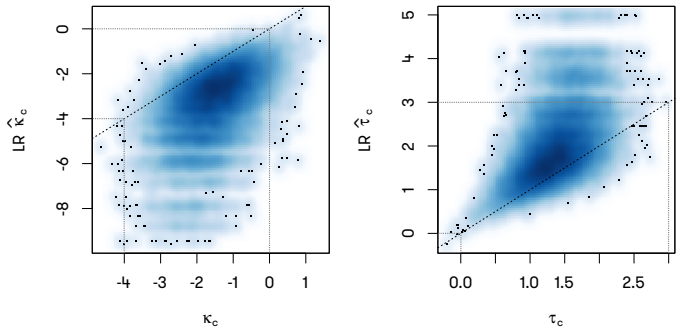
hensive picture of the performance of these estimators.

Despite being biased, the linear estimator for ζ_c has performed fairly well showing high concordance with the truth (Figure 5.5). As expected, estimated β_c is slightly lower than true β_c, and estimated α_c is slightly lower than true α_c. However, in general the true value can be efficiently recovered even in the presence of this minor yet discernible bias.

The empirical Bayesian estimator for δ_c has also displayed decent concordance with the truth (Figure 5.6). More importantly, when compared to the naïve logistic regressions, our empirical Bayesian estimators show dramatic improvement in terms of accuracy. Estimates from naïve logistic regressions show a much larger spread, and this is after we have filtered out a significant portion of the cells showing complete or quasi-complete separation, in which case the estimates cannot be obtained at all. These samples would need to be discarded in downstream analyses if no shrinkage is implemented.



(a) $\hat{\kappa}_c$ (left panel) and $\hat{\kappa}_c$ (right panel) estimated using the empirical Bayes approach compared to their true values.



(b) $\hat{\kappa}_c$ (left panel) and $\hat{\kappa}_c$ (right panel) estimated using simple logistic regressions compared to their true values.

Figure 5.6: Comparing the estimated $\hat{\kappa}_c$, $\hat{\tau}_c$ to the true values of κ_c and τ_c. Dotted line represents the unit line with intercept being 0, and slope equal to 1.

Figure 5.7 shows the distribution of estimated $(\hat{\alpha}_c, \hat{\beta}_c)$ and $(\hat{\kappa}_c, \hat{\tau}_c)$ across cells for the Zeisel data (Zeisel et al., 2015). The mean function, determined by $(\hat{\alpha}_c, \hat{\beta}_c)$, and the non-dropout rate function, determined by $(\hat{\alpha}_c, \hat{\beta}_c)$, are shown for four cells chosen to represent the middle and extremes of these distributions.

5.2.4. Differential expression analysis

Previous studies have shown that cells vary in size, with larger cells having more RNA molecules to attain similar concentration levels to smaller cells (Padovan-Merhar et al., 2015). This indicates that to detect DE genes, it is more appropriate to test for concentration difference between groups. To allow this, we include cell size, which can be estimated by the ratio of reads from endogenous RNA to reads from spike-in sequences, as a covariate. Other potential covariates, such as cell

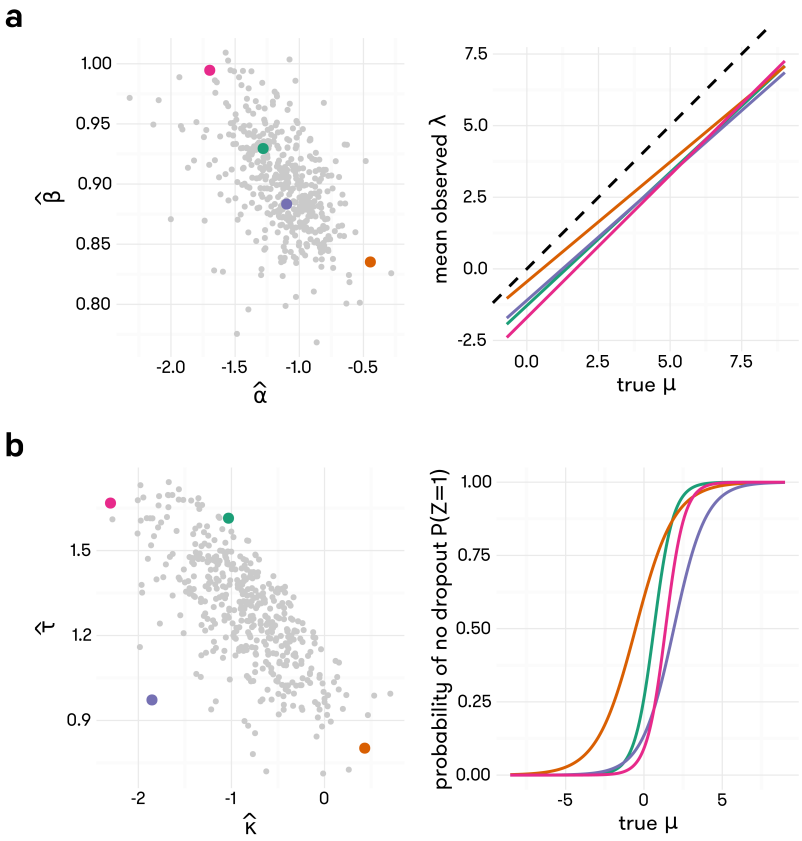


Figure 5.7: Distributions of empirically estimated values of $(\hat{\alpha}_c, \hat{\beta}_c)$ and $(\hat{\kappa}_c, \hat{\tau}_c)$ across all cells in Zeisel data. Four cells are selected from each plot to represent the distribution, and the line (in **a**) and logistic curve (in **b**) corresponding to the technical parameters estimated for these cells are shown in matching colors.

cycle stage, can also be included in the model to avoid spurious association. For cell cycle, we add as covariate the expression of a curated set of marker genes, such as the set from (Tirosh et al., 2016), or a latent factor representing cell cycle, as in (Buettner et al., 2015). A likelihood-ratio test is developed to detect DE genes.

Based on the hierarchical model, testing for differentially expressed (DE) genes is straightforward. In our model, for cells within the same group, the true expression level of gene g, μ_{cg} follows a log-normal distribution with mean θ_g and variance σ_g^2. Testing for differential expression involves comparing the means from different groups on a gene-by-gene basis. We propose a likelihood ratio test for this purpose. Let θ_g be expressed as a linear combination of the covariates for which one wishes to test or adjust, $\theta_g = X\Gamma_g$, $\Gamma_g = (\gamma_1, \cdots, \gamma_i, \cdots, \gamma_p)$ with γ_i denoting the coefficient for predictor x_i in the design matrix $X = (x_1, \cdots, x_i, \cdots, x_p)$. Testing for each covariate involves fitting a full model $\theta_g = X\Gamma_g$ and a reduced model with target covariate x_i removed from the design matrix X. Denote the reduced design matrix and coefficient vector to be $\tilde{X}$ and $\tilde{\Gamma}_g$ respectively. Denote the biological variance of the full and reduced model as σ_g^2 and $\tilde{\sigma}_g^2$. Formally the question of whether x_i is significantly associated with the gene expression can be formulated as the following hypothesis test,

$$H_0 : \gamma_i = 0$$
$$H_1 : \gamma_i \neq 0.$$

The likelihood ratio test statistic for this above test, T_i can be constructed as,

$$T_{ig} = 2\left[\log\left(\hat{\mathcal{L}}_1\right) - \log\left(\hat{\mathcal{L}}_0\right)\right],$$

where $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_0$ are the likelihoods maximized under H_1 and H_0, respectively. Asymptotically, $T_{ig}i$ follows a χ^2-distribution with 1 degree of freedom under the null hypothesis (γ_i=0). Raw p-values can subsequently be adjusted for multiple comparisons with false discovery rate controlling procedures such as the Benjamini-Hochberg procedure or the Holm-Bonferroni procedure.

5.2.5. Expectation-Maximization algorithm

When the number of covariates is small, the parameters can be estimated using the Simplex algorithm, which does not involve the calculation of derivatives. However, the Simplex algorithm is

not suitable when the number of covariates is large. To circumvent this problem, we have also developed an expectation-maximization (EM) algorithm to estimate the biological mean (θ_g) and variance σ_g^2. Briefly, the log likelihood for gene g can be written as,

$$\begin{aligned}
\ell\left[\mathbf{Y}_g, \boldsymbol{\mu}_g | \theta_g, \sigma_g^2\right] &= \sum_c \ell\left[Y_{cg}, \mu_{cg} | \theta_g, \sigma_g^2\right] \\
&= \sum_c \left\{\ell\left[Y_{cg} | \mu_{cg}, \theta_g, \sigma_g^2\right] + \ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right]\right\} \\
&= \sum_c \left\{\ell\left[Y_{cg} | \mu_{cg}\right] + \ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right]\right\} \\
&= \sum_c \ell\left[Y_{cg} | \mu_{cg}\right] + \sum_c \ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right].
\end{aligned}$$

E-step:

$$\begin{aligned}
&E\left[\ell\left[\mathbf{Y}_g, \boldsymbol{\mu}_g \middle| \theta_g, \sigma_g^2\right] | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] \\
=&\sum_c E\left[\ell\left[Y_{cg} | \mu_{cg}\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] + \sum_c E\left[\ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] \\
=&C(\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}) + \sum_c E\left[\ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right]
\end{aligned}$$

The first term is ignorable since it is a constant function of $\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}$, over which the maximization is to be performed. So in order to evaluate this expectation in the E-step, we only need to compute $E\left[\ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right]$. Due to the assumptions we have made for the functional form of F_g, it follows a log normal distribution.

$$\begin{aligned}
&E\left[\ell\left[\mu_{cg} | \theta_g, \sigma_g^2\right] | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] \\
=&E\left\{\left[-\frac{\left[\log\left(\mu_{cg}\right) - \theta_g\right]^2}{2\sigma_g^2} - \frac{1}{2}\log\left(2\pi\sigma_g^2\right)\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right\} \\
=&E\left\{\left[-\frac{\log\left(\mu_{cg}\right)^2 - 2\theta_g\log\left(\mu_{cg}\right) + \theta_g^2}{2\sigma_g^2} - \frac{1}{2}\log\left(2\pi\sigma_g^2\right)\right] \middle| \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right\} \\
=&-\frac{E\left[\log\left(\mu_{cg}\right)^2 | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] - 2\hat{\theta}_g^{(t)} E\left[\log\left(\mu_{cg}\right) | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] + \left(\hat{\theta}_t^{(t)}\right)^2}{2\left(\hat{\sigma}_g^{(t)}\right)^2} - \frac{1}{2}\log\left(2\pi\left(\hat{\sigma}_g^{(t)}\right)^2\right)
\end{aligned}$$

Two expectations need to be evaluated in order to compute the above value. Briefly,

$$E\left[\log(\mu_{cg})^2 | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] = \frac{\int_0^\infty \log[\mu_{cg}]^2 \Pr[Y_{cg}|\mu_{cg}] \Pr\left[\mu_{cg}|\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}\right] d\mu_{cg}}{\int_0^\infty \Pr[Y_{cg}|\mu_{cg}] \Pr\left[\mu_{cg}|\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}\right] d\mu_{cg}}$$

$$E\left[\log(\mu_{cg}) | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] = \frac{\int_0^\infty \log[\mu_{cg}] \Pr[Y_{cg}|\mu_{cg}] \Pr\left[\mu_{cg}|\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}\right] d\mu_{cg}}{\int_0^\infty \Pr[Y_{cg}|\mu_{cg}] \Pr\left[\mu_{cg}|\hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}\right] d\mu_{cg}}$$

M-step:

The M-step involves maximizing the above expected log-likelihood w.r.t the parameters $\hat{\theta}_g^{(t)}$ and $\hat{\sigma}_g^{(t)}$, in the case of simple quantification,

$$\hat{\theta}_g^{(t+1)} = \frac{1}{N}\sum_{c=1}^{N} E\left[\log(\mu_{cg}) | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right]^{(t)}$$

$$\hat{\sigma}_g = \sqrt{\frac{1}{N}\sum_{c=1}^{N} E\left[\log(\mu_{cg})^2 | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] - 2\hat{\theta}_g^{(t+1)} E\left[\log(\mu_{cg}) | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right] + \left(\hat{\theta}_t^{(t+1)}\right)^2}.$$

In the case of $\theta_g = X\Gamma_g$, the above E-step is the same, after substituting $\hat{\theta}_g^{(t)} = X\hat{\Gamma}_g^{(t)}$. The M-step for $\hat{\Gamma}_g^{(t+1)}$ is replaced by a linear regression with $E\left[\log(\mu_{cg}) | \hat{\theta}_g^{(t)}, \hat{\sigma}_g^{(t)}, \mathbf{Y}_g\right]$ as the response variable, and X as the predictor. The M-step for $\hat{\sigma}_g^{(t+1)}$ is unchanged, after substituting $\hat{\theta}_g^{(t+1)} = X\hat{\Gamma}_g^{(t+1)}$.

5.2.6. Estimation of cell size factor

Single-cell RNA-seq requires normalization on cell size because larger cells tend to have more RNA molecules. To estimate the cell size S_c ($c = 1, \cdots, N$, N being the number of cells), we take advantage of the spike-ins as well. Denote the read count for biological gene b in cell c as $\xi_{cb}, b = 1, \cdots, B$, where B is the total number of biological genes after filtering. Also denote the counts of the spike-in molecule e as $\xi_{ce}, e = 1, \cdots, E$, where E is the total number of spike-in molecules. The cell size factor can be computed as,

$$S_c = \frac{\sum_{b=1}^{B} \xi_{cb}}{\sum_{e=1}^{E} \xi_{ce}}.$$

In our software implementation, this cell size factor is computed and automatically used as a covariate to adjust for any possible confounding incurred due to different cell sizes unless the users explicitly disable it. In order to compare the detected DE genes with and without adjustment for the cell size factors, we have looked at the genes called significantly differentially expressed when comparing the two level-2 classes CA1Pyr1 and CA1Pyr2. 1604 genes are uniquely detected with adjustment for the cell size factors, while 663 genes are uniquely detected without. 3346 genes are differentially expressed regardless of the adjustment. Considering the fact that this dataset contains cells of relatively homogeneous sizes (Figure 5.33), it is highly possible that the difference will be more pronounced in samples of more heterogeneous sizes.

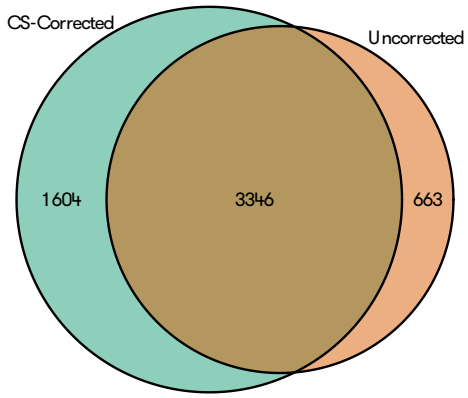


Figure 5.8: Venn diagram showing the overlapping of genes detected to be differentially expressed between comparisons with and without cell size adjustment.

5.3. Evaluation of Performance and Comparison with Other Methods

In this section, we evaluate the performance of TASC on both simulated and two real scRNA-seq data sets and compare it with four existing methods, including SCDE (Kharchenko, Silberstein, and Scadden, 2014), MAST (Finak et al., 2015), and DESeq2 (Love, Huber, and Anders, 2014), and SCRAN (Lun, Bach, and Marioni, 2016). As SCRAN only provides normalized read counts, we perform differential expression analysis using DESeq2 with SCRAN normalized read counts. We include two versions of SCRAN in our evaluation, the original SCRAN, and SCRAN.SP that utilizes ERCC spike-ins in normalization. These methods are rated in terms of type I error rate and power in detecting DE genes, and their results on a real data set with genuine gene expression difference.

5.3.1. Type I error rates in the absence of batch effects

To assess the accuracy of type I error control of TASC and other existing methods, 447 cells from the level-2 class CA1Pyr2 from the Zeisel et al. data (Zeisel et al., 2015), which is the largest level-2 class, are randomly split into two groups of roughly equal size. Therefore, no gene should be differentially expressed when one group is compared with the other. Differential expression analyses are performed with TASC, SCDE, MAST, DESeq2, SCRAN and SCRAN.SP. Raw p-values are extracted from each method, and the performance of each method is assessed by histograms and quantile-quantile plots of the corresponding p-values, shown in Figure 5.9. Our results show that TASC, DESeq2, SCRAN and SCRAN.SP have p-values that are uniformly distributed as expected under the null, whereas SCDE is overly conservative with enrichment of p-values near one, and MAST is severely anti-conservative with enrichment of p-values near zero.

5.3.2. Type I error rates in the presence of batch effects

Batch effects are common in scRNA-seq data (Hicks, Teng, and Irizarry, 2015). As we have discussed above, four technical parameters dictate the relationship between the true expression of a gene and the observed counts in a specific cell in scRNA-seq experiments. In our framework, these four parameters are modeled in groups of two. The first two parameters are α_c and β_c, which represent the efficiency of capture and amplification, relating the log mean of the Poisson distribution to the true log expression of the gene in cell c. The last two parameters are κ_c and τ_c, which are influenced by the propensity of a gene being observed in the final sequencing, *i.e.* not a dropout. Both of these parameters vary across cells, and directly affect our estimates for the true expression of the gene g in cell c. Therefore, it is of great interest to see whether adjustment for these cell-specific technical parameters or failure to do so has an effect on the specificity for calling significant differentially expressed genes from scRNA-seq data.

To evaluate effectiveness in type I error control in the presence of batch effects, we have generated a data set that contains batch effects as characterized by systematic differences in the technical parameters $(\alpha_c, \beta_c, \kappa_c, \tau_c)$ between groups. To introduce batch differences between the two groups under comparison, cell-specific technical parameters (α_c, β_c) and (κ_c, τ_c), are estimated from the cells in CA1Pyr2 class and a bivariate normal distribution is fit separately to (α_c, β_c) and (κ_c, τ_c). One group in the simulated data draws its cell-specific technical parameters from these empiri-

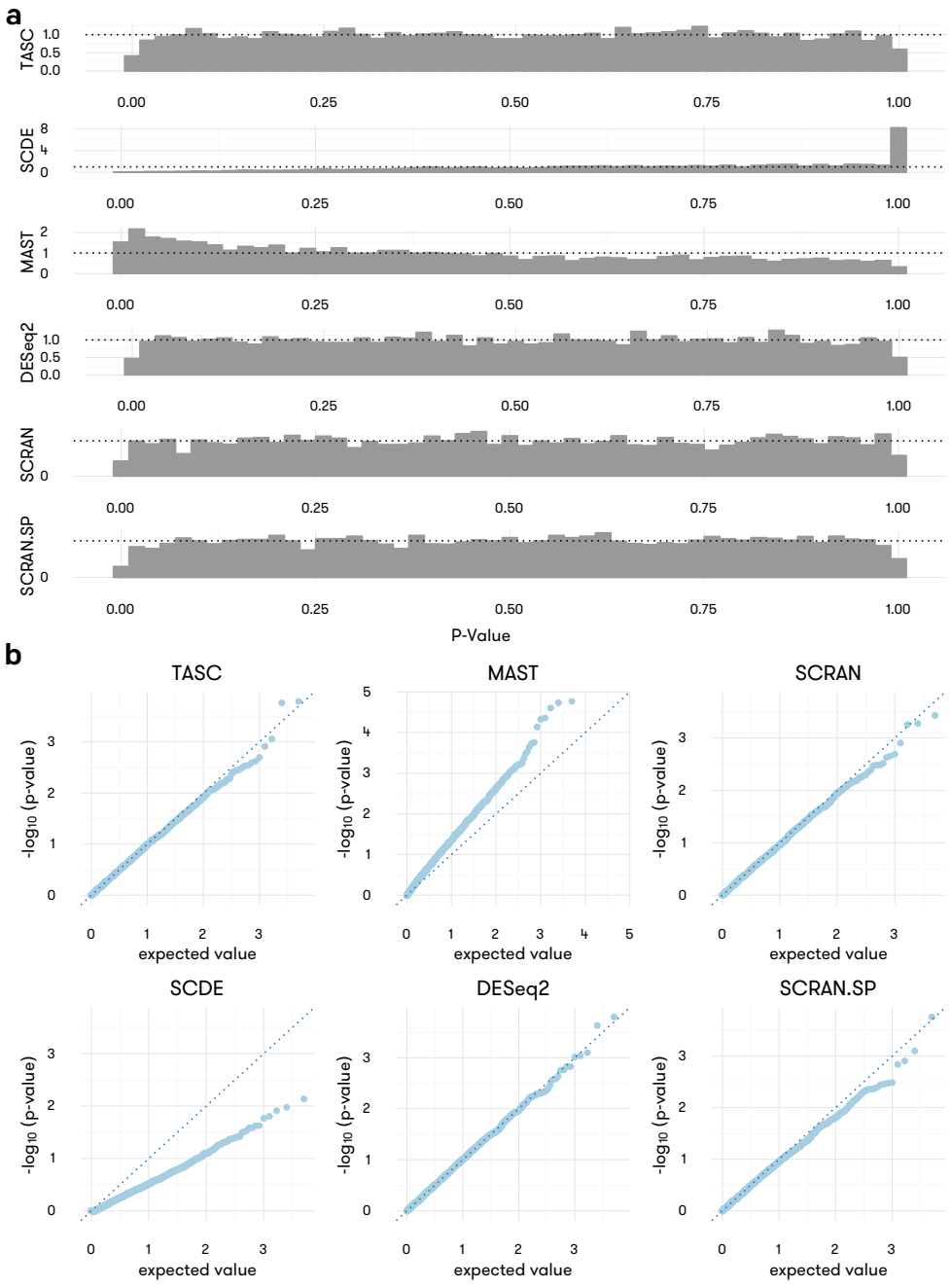


Figure 5.9: Distribution of achieved p-values (in **a**) and the corresponding quantile-quantile plots (in **b**) for four methods applied to CA1Pyr2 cells from Zeisel et al. data, split randomly into two groups, thus emulating a case where all p-values should be uniformly sampled from $[0, 1]$.

cal distributions, and the other group draws its technical parameters from distributions where the mean(s) of combinations of technical parameters are shifted by amounts shown on the axes of the heatmaps in Figure 5.10. The magnitude of the shift represents the severity of batch effect difference between the two groups. The rest of the parameters controlling the expression of genes are the same for the two groups and are derived from estimates from the CA1Pyr2 class. Simulations are performed to generate the counts of 5,018 genes in 100 cells (50 in each group). Differential expression analyses are performed and the raw p-values are used to estimate the false positive rate (FPR). The deviation of the estimated FPR from the expected value is plotted on heatmaps to reflect the type I error rates under varying severity of batch effects. Figure 5.10 shows that TASC has well controlled type I error rates across a wide range of batch effect severity, whereas SCDE appears to be conservative overall, and MAST, DESeq2, SCRAN and SCRAN.SP are anti-conservative and susceptible to batch effects.

The detailed steps of data simulation are as follows:

- Cell-specific parameters, $\mathbf{\Psi}_c$, as well as gene-specific parameters, the biological mean (θ_g) and variance (σ_g^2) are estimated from the "CA1Pyr2" class in the Zeisel data(Zeisel et al., 2015) using our model.

- Two bivariate normal distributions for $\delta_c = (\kappa_c, \tau_c)$ and $\zeta_c = (\alpha_c, \beta_c)$ are fitted with the estimated parameters.

- The sample is randomly divided into two groups of roughly equal sizes. A difference is then added to the mean of δ_c for cells from one of the groups, resulting in two sets of cells whose δ_c can be characterized as:

$$\begin{aligned} E[\kappa_c]_2 &= E[\kappa_c]_1 + \Delta E[\kappa_c] \\ E[\tau_c]_2 &= E[\tau_c]_1 + \Delta E[\tau_c] \end{aligned}$$

  The magnitudes of $\Delta E[\kappa_c]$ and $\Delta E[\tau_c]$ determine the degree of batch effects. We have generated combinations of $\Delta E[\kappa_c]$ and $\Delta E[\tau_c]$, with both values ranging from -0.4 to 0.8.

- The generative model is used to simulate the counts with δ_c sampled from the corresponding bivariate normal distribution. For each combination of $\Delta E[\kappa_c]$ and $\Delta E[\tau_c]$, approximately 4000

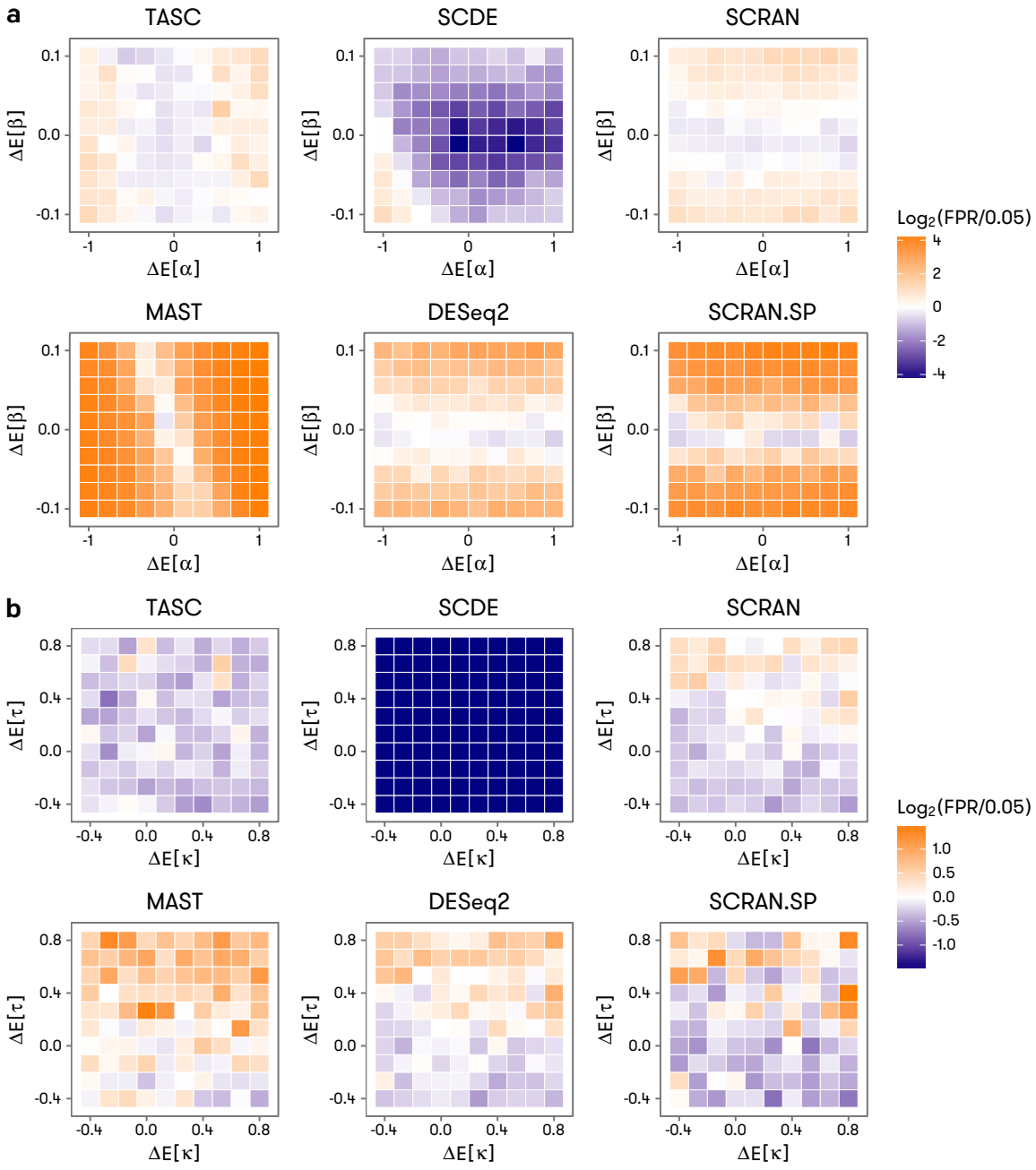


Figure 5.10: Accuracy of false positive rate control under mild to severe batch effects for TASC, SCDE, MAST, and DESeq2. The batch effect severity takes the form of between-group difference in the expected values of the technical parameters, denoted by $\Delta E\,[\kappa]$ and $\Delta E\,[\tau]$ (in **a**), and $\Delta E\,[\alpha]$ and $\Delta E\,[\beta]$ (in **b**) in the axes of the heatmaps. The color scale of the heatmaps reflects deviation of achieved false positive rate from the target value of 0.05 used in the tests.

genes are generated, and p-values are calculated by running differential expression analyses with each tested method.

- The p-values are subsequently used to compute the false positive rates (FPRs), *i.e.* the proportion of DE genes called ($p < 0.05$) among all genes tested (since all of them are not differentially expressed). The FPR is then compared with the desired significance level (0.05) and a heat map is generated by plotting $\log_{10}(FPR/0.05)$ with varying colours on a grid representing the combinations of $\Delta E\left[\kappa_c\right]$ and $\Delta E\left[\tau_c\right]$.

- Similar simulations are performed for α_c and β_c, with the only change being the range of $\Delta E\left[\alpha_c\right]$ ($[-1, 1]$) and $\Delta E\left[\beta_c\right]$ ($[-0.1, 0.1]$).

5.3.3. Power

In addition to controlling type I error, an ideal statistical method should also be sensitive, *i.e.* exhibiting extraordinary power when compared to existing algorithms. To investigate the power of the methods under realistic scenarios, we continue to utilize the 5,018 genes from the CA1Pyr2 class in Zeisel et al. data set. Among them, 4,018 genes are designated as true non-DE, whose counts are directly extracted from the Zeisel et al. data set after group membership randomization. The remaining 1,000 are designated as true DE, whose counts are simulated from parameters estimated with real data, with an induced between-group fold change that is randomly sampled from a distribution that generates more genes with weak to moderate expression difference than strong difference. The detailed steps as as follows:

- The simulation scenario is the classic two-group comparison. Let the true expression of gene g from group 1 follow a log-normal distribution $\mu_{cg} \sim \mathrm{LogNormal}\left(\theta_{g1}, \sigma_g^2\right)$, and the same gene from group 2 follow a log-normal distribution with a different mean $\mu_{cg} \sim \mathrm{LogNormal}\left(\theta_{g2}, \sigma_g^2\right)$. For simplicity, in this simulation we assume g display similar biological variance across groups. This assumption is purely for simplicity, and our model can easily handle situations where this is not true. In our current iteration of implementation, the biological variance of the two groups is assumed to be identical.

- From cells in the level 2 class "CA1Pyr2" in Zeisel data set(Zeisel et al., 2015), we estimate the technical parameters $\mathbf{\Psi}_c$ for each cell c, the mean (θ_g) and standard deviation (σ_g) of log

gene expression for reach gene g using TASC. Genes with extremely low total read counts are removed, leaving a total of 5018 genes in the final pool.

- 1000 genes are randomly picked to be differentially expressed. The effect size, *i.e.* fold change between the two groups, $\eta_g = \exp(|\theta_{g1} - \theta_{g2}|)$ ranges from 1.05 to 2.5, and is assigned so that the majority of DE genes only exhibit mild difference in expression (Figure 5.12). This distribution of η_g dovetails with the overall experience from two-group comparison experiments.

- Counts of the 1000 DE genes are sampled from our generative model using the technical parameters $\mathbf{\Psi}_c$ and σ_g^2 estimated in previous steps. More specifically, θ_{g1} is directly from the mean estimated in previous steps, and $\theta_{g2} = \theta_{g1} \pm \log \eta_g$, where η_g is the fold change for gene g. The sign of $\log \eta_g$ is randomly assigned.

- Counts of the 4018 non-DE genes are equal to the Zeisel data(Zeisel et al., 2015). Since our group membership is randomly assigned, none of these genes should be differentially expressed.

- The above steps are repeated 100 times and each dataset consists of 5018 genes (1000 DE genes and 4018 non-DE genes). The 447 cells are then down-sampled into various sample sizes for 5 different simulations, 20 vs 20 (20 cells in group 1 and 20 cells in group 2, same hereinafter), 50 vs 50, 100 vs 100, 150 vs 150 and 200 vs 200.

- In each simulation, TASC, MAST(Finak et al., 2015) and DESeq2(Love, Huber, and Anders, 2014) are used to call DE genes. For each DE gene, the power can be estimated by dividing the number of datasets in which it is called significant (p is less than or equal to the pre-set significant level) by the total number of simulations (100).

The scheme of simulation is illustrated in Figure 5.11.

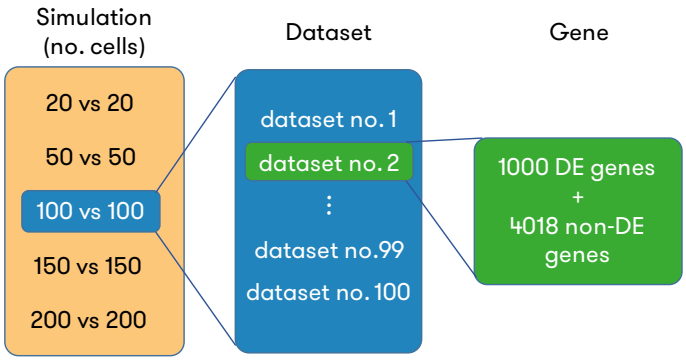


Figure 5.11: The scheme of simulation for power comparisons. Simulations differ by their sample sizes, *i.e.* the number of cells in each group. This is achieved by downsampling each group to the desired number of cells from the complete data (447 cells in total). One simulation contains 100 datasets, generated by repeating the sampling process from the same parameters. Each dataset contains the counts of 5018 genes in specified number of cells. 1000 genes are differentially expressed while the rest are not.

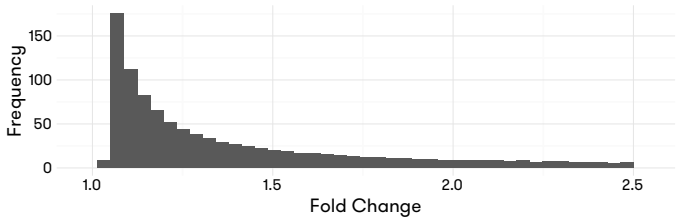


Figure 5.12: Distribution of η_g in the simulation study.

We have made sure that our simulated datasets are visually indistinguishable when counts from a random pair of cells are compared. In Figure 5.13, 9 pairs of cells from the 447 cells are randomly selected and plotted. In a specific pair, each dot represents a gene with its count in one cell plotted on the x axis and that in the other cell on the y axis. These plots closely resemble similar plots reported before generated from various scRNA-seq experiments, which suggests that our simulation scheme can largely recapitulate the between cell variability in scRNA-seq data.

5.3.4. Overall Power Performance

The average power curves in Figure 5.14a are obtained by smoothing the estimated power across genes with similar fold change. Our results demonstrate that TASC has the highest power, followed by SCRAN.SP, SCRAN, DESeq2, MAST, and SCDE. Figure 5.14b shows that the higher sensitivity of TASC is more pronounced when fold change is moderate; for example, when fold change is

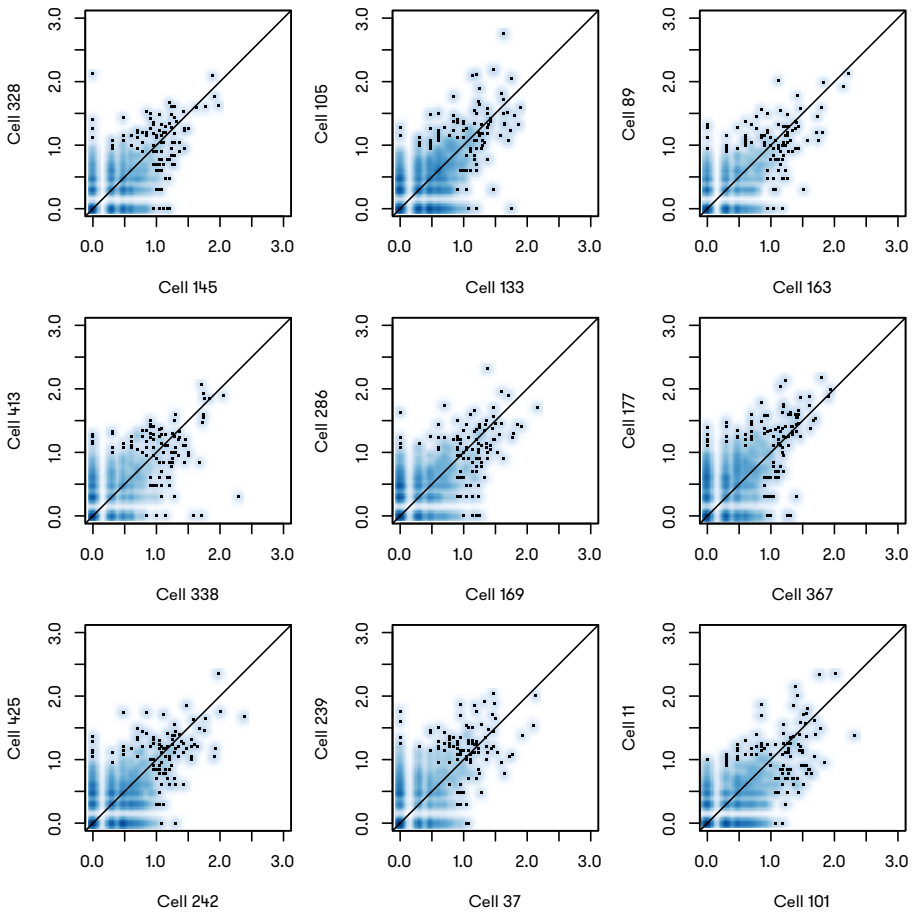


Figure 5.13: Scatter plots for 9 randomly picked pairs of cells in simulated data. For each panel, two cells are randomly chosen from the a total of 447. With two cells indexed as i and j, $\log(Y_{\mathrm{ig}}+1)$ is plotted against $\log(Y_{\mathrm{jg}}+1)$.

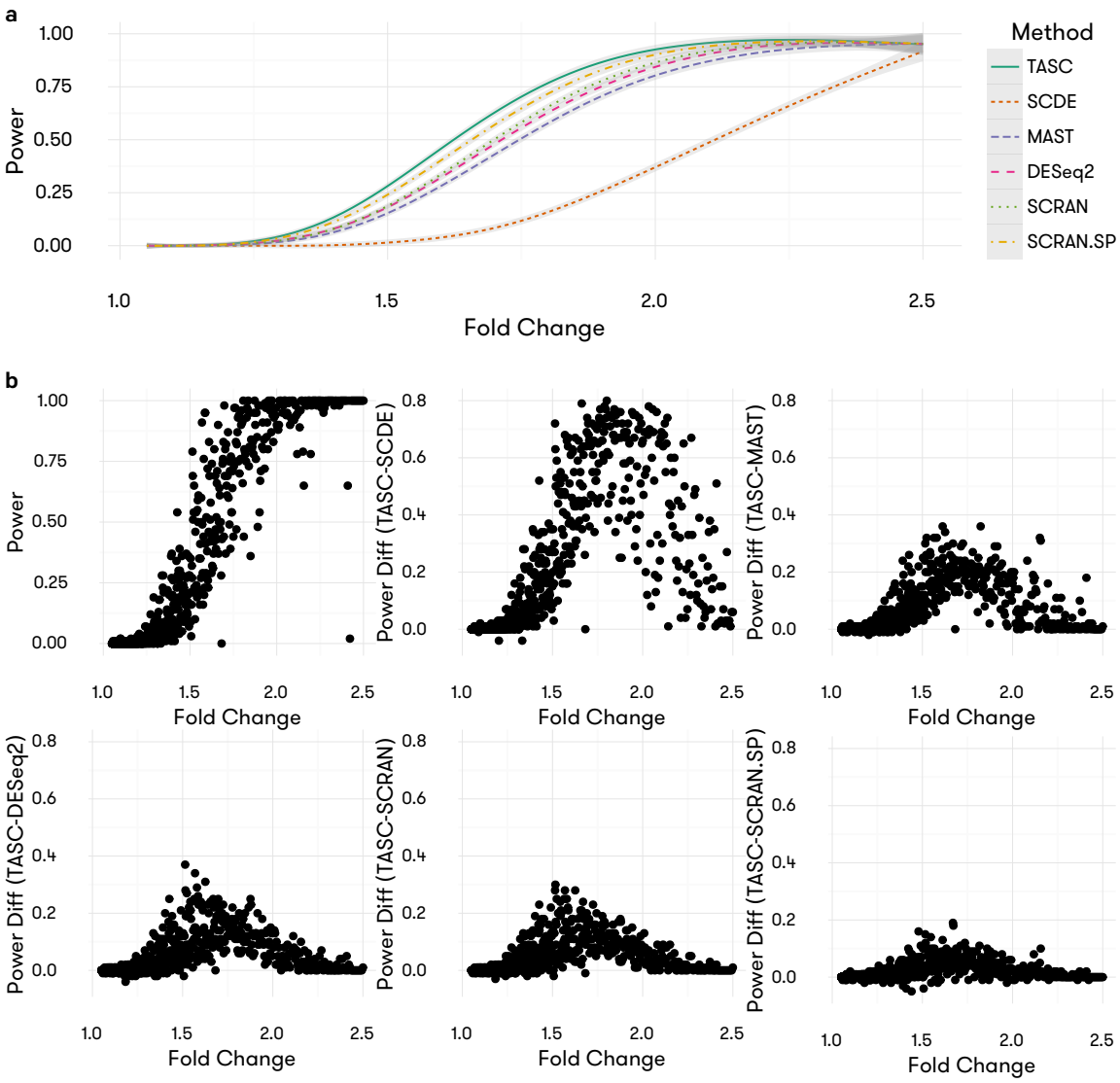


Figure 5.14: **a**. Achieved power of TASC, SCDE, MAST, DESeq2, SCRAN, and SCRAN.SP for detecting varying fold changes in mean in the simulated data set within 100 cells in each group. Results both with (SCRAN.SP) and without (SCRAN) the use of ERCC are included for SCRAN. **b**. Power differences between TASC and the other methods in the simulated data set.

1.75, at the 0.0001 significance level, the average power of TASC is 8%, 20%, 25%, 37%, and 428% higher than SCRAN.SP, SCRAN, DESeq2, MAST, and SCDE, respectively.

Power and effect size

Since we have simulated 1000 DE genes with varying effect size, it is straightforward to investigate how the η_g influences the power of our method. In Figure 5.15, estimated power ($\omega_g = n_{Sg}/n_{Tg}$,

where n_{Sg} is the number of datasets in which the p-value of TASC is less than or equal to the specified significance level, and $n_{Tg} = 100$ is the total number of datasets in each simulation) is plotted against η_g. Due to the differences in other parameters such as θ_{g1}, θ_{g2} and σ_g, genes with similar η_g can be detected with dramatically different power. This leads to a spread in our power-effect size curve. For example, when we pick the significance level to be 10^{-4}, genes that display approximately 2-fold change between the two groups can be detected from less than 40% of the time to over 80% depending on specific properties of the gene. This closely resembles the actual analyses and speaks to the importance of simulating data based on real data. Figure 5.15 is plotted from the simulation with 100 cells in each group.

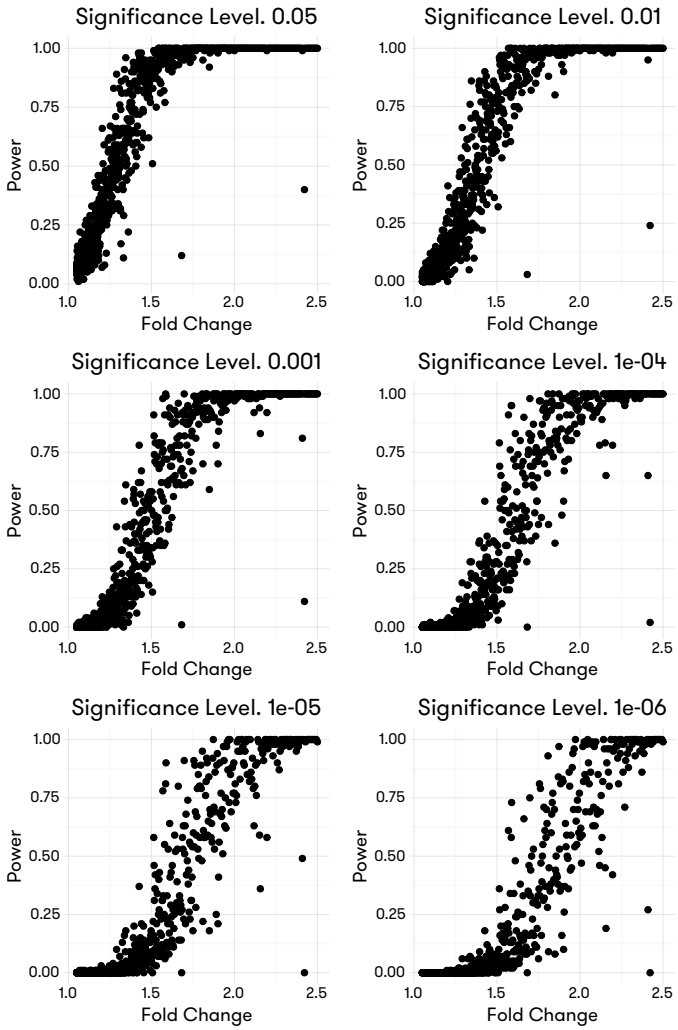


Figure 5.15: Relationship between the estimated power and the effect size. Each DE gene is plotted with the x axis indicating their η_g. Y axis represents the proportion of datasets in which TASC has called this gene significantly differentially expressed (p is less than or equal to the specified significance level). The sample size of this simulation is 100 vs 100.

SCDE performs quite conservatively in our studies on the type I error. Unsurprisingly, when compared to TASC, has dramatically attenuated power. Figure 5.16 and Figure 5.17 illustrate the relationship between the power of and the effect size of gene. With significance levels set at all values (10^{-6} to 0.05) TASC overpowers by a considerable margin. The difference is particularly prominent when the significance level is set below 10^{-4}, which is the common in scRNA-seq analyses due to the preference of controlling for false positives. When the significance level is set to be 10^{-4}, genes with $\eta_g \approx 1.75$ can be detected over 75% of the time by TASC, but less than 25% of the time by

SCDE. This is translated into a difference of power between 40% to 80%, a 4-fold improvement in most cases.

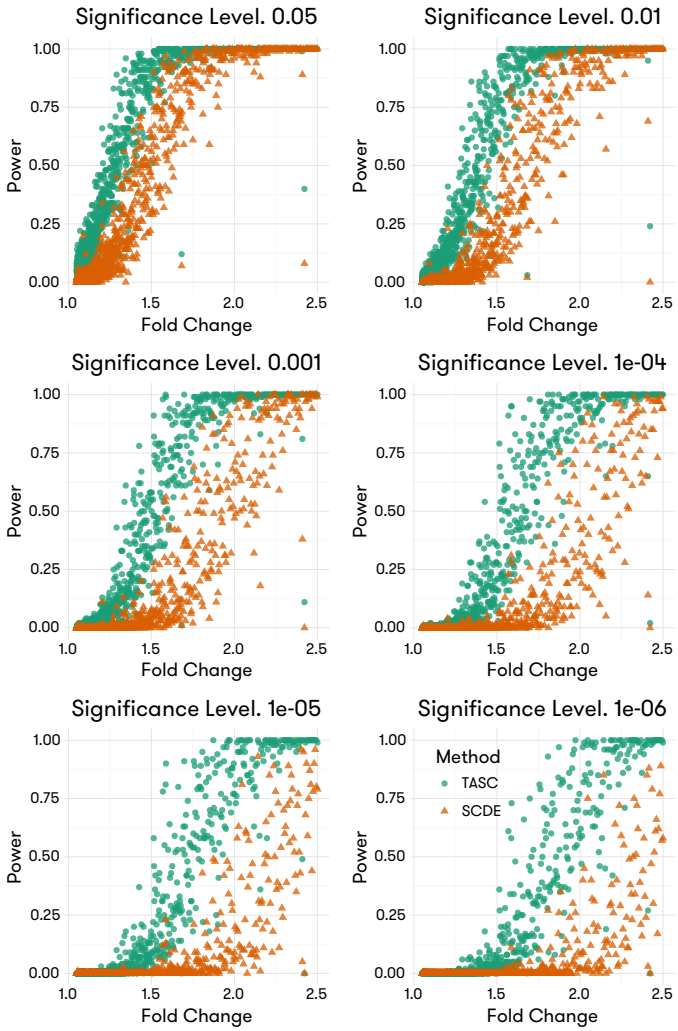


Figure 5.16: Power comparison between TASC and with various effect sizes. Each panel contains the power curve of TASC and under the specified significance level. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

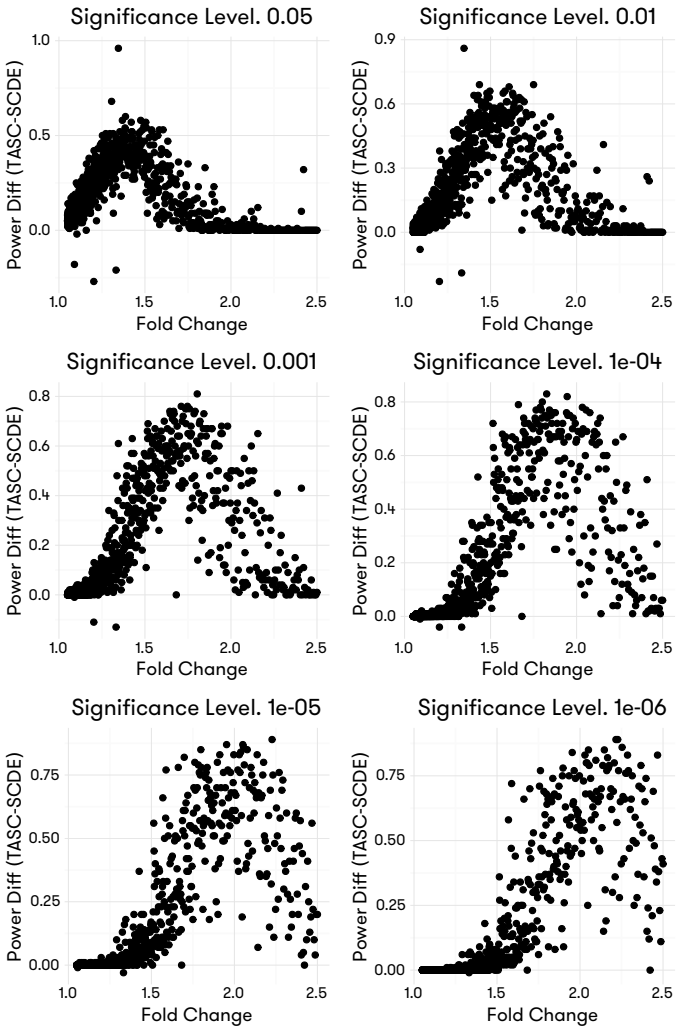


Figure 5.17: Power improvement of TASC over with various effect sizes. Each panel contains the power improvement curve of TASC and under the specified significance level. Y axis represents the difference in absolute not relative values in estimated power between TASC and , *i.e.* $\omega_g^{TASC} - \omega_g$. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

Another method specifically designed for scRNA-seq is MAST(Finak et al., 2015), which shows inflated type I error in our studies based on real data even in the absence of batch effects (Figure 5.9). Among all four methods tested, MAST(Finak et al., 2015) has the most difficult controlling the type I error rate when batch effects are present in the dataset (Figure 5.10). In terms of power, MAST(Finak et al., 2015) has also performed poorly compared to TASC (Figure 5.18 and Figure 5.19). Using genes with $\eta_g \approx 1.75$ as an example, the power difference between TASC and MAST(Finak et al., 2015) is 10% to over 30%. This suggests that MAST(Finak et al., 2015) has

a tendency to mislabel non-DE genes as DE and DE genes as non-DE, and the results produced by MAST(Finak et al., 2015) should be validated by other methods to reduce the number of false positives.

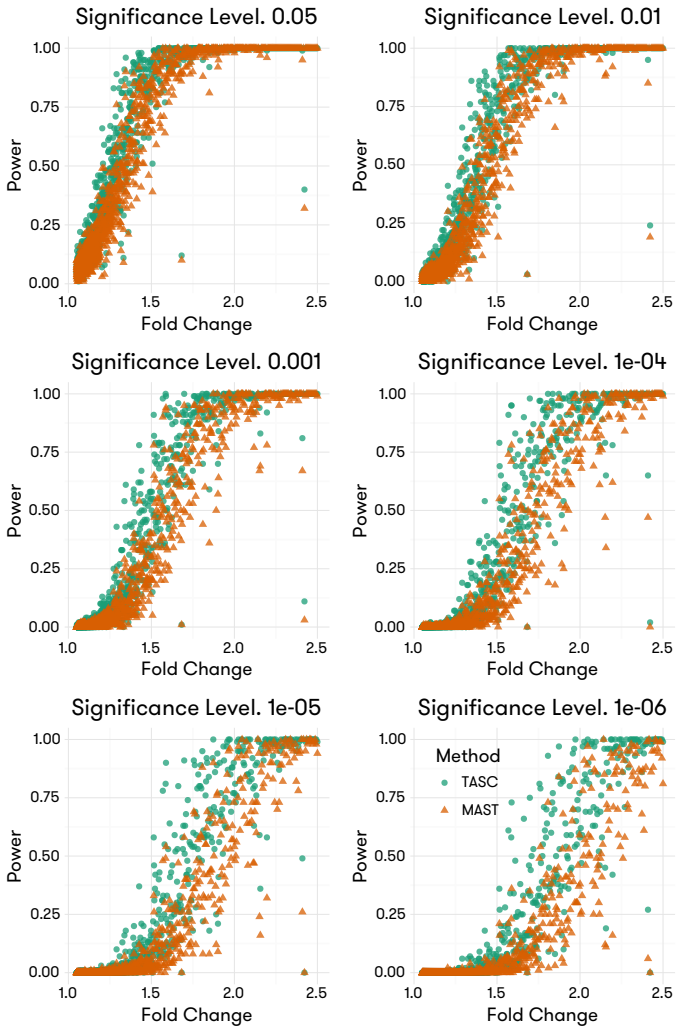


Figure 5.18: Compare power between TASC and MAST(Finak et al., 2015) with various effect sizes. Each panel contains the power curve of TASC and MAST(Finak et al., 2015) under the specified significance level. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

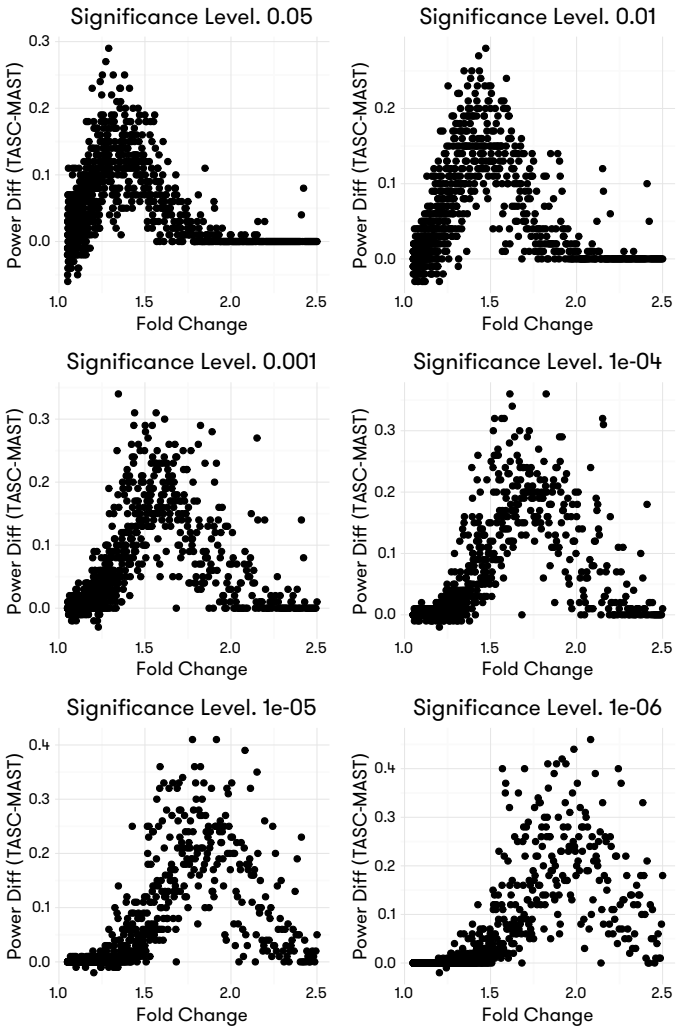


Figure 5.19: Power improvement of TASC over MAST(Finak et al., 2015) with various effect sizes. Each panel contains the power improvement curve of TASC and MAST(Finak et al., 2015) under the specified significance level. Y axis represents the difference in absolute not relative values in estimated power between TASC and MAST(Finak et al., 2015), *i.e.* $\omega_g^{TASC} - \omega_g^{MAST}$. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

DESeq2(Love, Huber, and Anders, 2014) is a popular method for differential expression analysis. Although developed for bulk RNA-seq data, our simulation study suggests that DESeq2 has decent overall performances such as type I error rate in the absence of batch effects (Figure 5.9). In terms of power, however, DESeq2 is outperformed by TASC just like the other methods. Using genes with a fold change near 1.75 as an example, TASC represents a difference of 10% to 30% on the absolute not relative scale over DESeq2. A more troubling issue is that DESeq2(Love, Huber, and Anders, 2014) lacks the ability to adjust for batch effects and can display serious type I inflation in

the presence of batch effects (Figure 5.10).

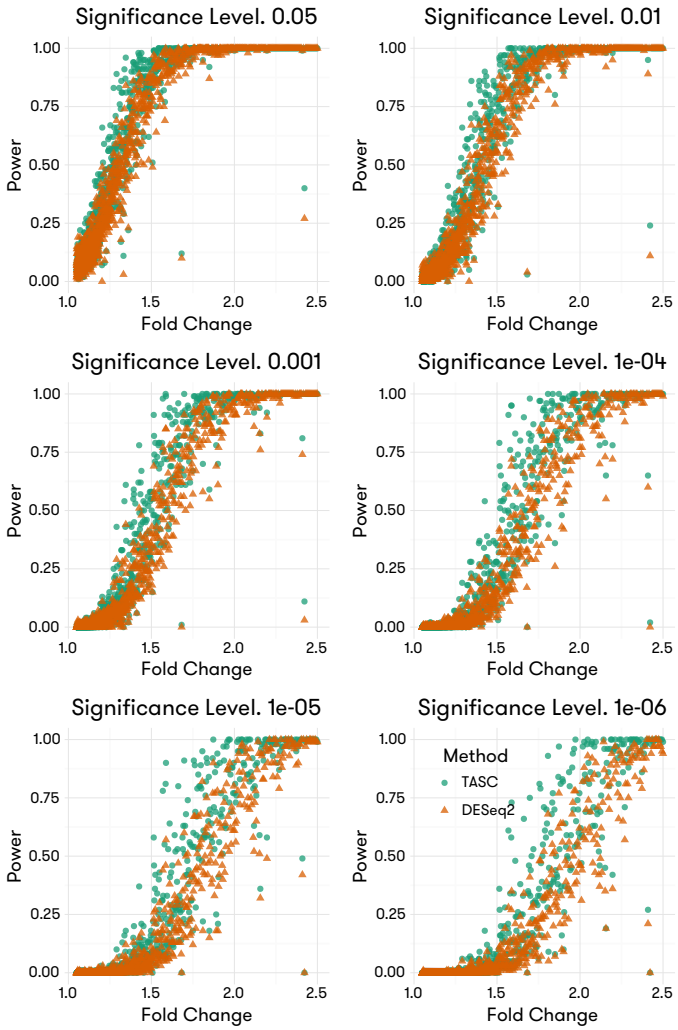


Figure 5.20: Compare power between TASC and DESeq2(Love, Huber, and Anders, 2014) with various effect sizes. Each panel contains the power curve of TASC and MAST(Finak et al., 2015) under the specified significance level. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

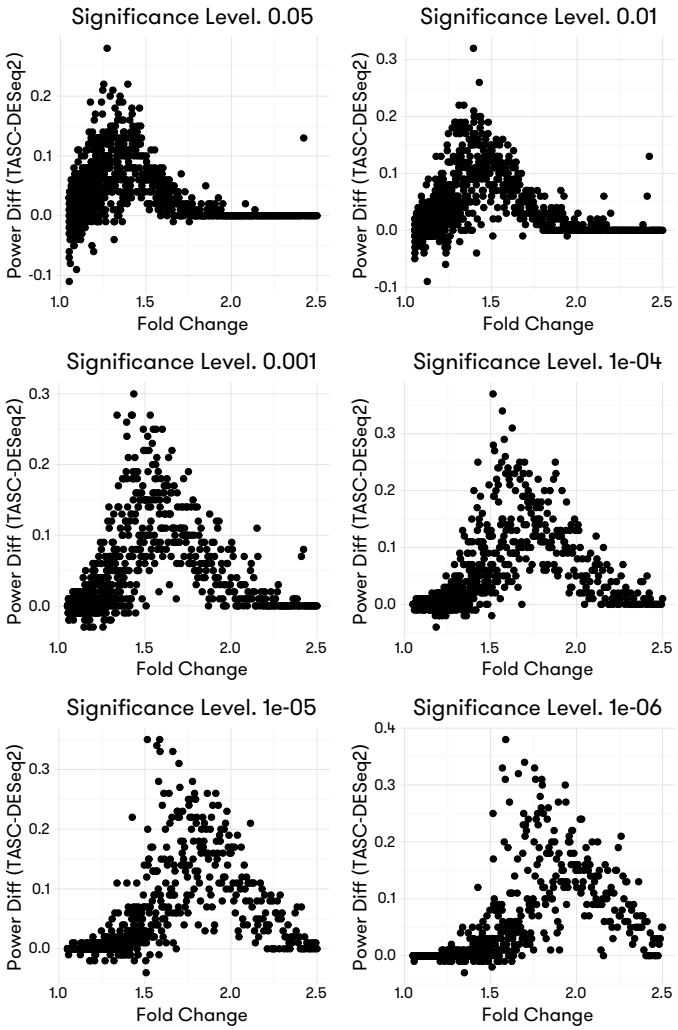


Figure 5.21: Power improvement of TASC over DESeq2(Love, Huber, and Anders, 2014) with various effect sizes. Each panel contains the power improvement curve of TASC and DESeq2(Love, Huber, and Anders, 2014) under the specified significance level. Y axis represents the difference in absolute not relative values in estimated power between TASC and DESeq2(Love, Huber, and Anders, 2014), *i.e.* $\omega_g^{TASC} - \omega_g^{DESeq2}$. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

SCRAN(Lun, Bach, and Marioni, 2016) is a recently developed method for normalizing scRNA-seq data using cell-specific deconvolved pool-based size factors. As a normalization scheme, its performance is highly dependent on the downstream method of analysis. We have tested SCRAN in the scenario of two-group comparison coupled with DESeq2 and it has shown improved performance over using DESeq2 alone. Since the SCRAN package can also take advantage of the counts for spike-ins to derive the normalization factors, we have looked at both naïve SCRAN (without spike-

ins) and SCRAN.SP (SCRAN run with spike-ins). In some cases, due to the limitations of the sample size available, only results from SCRAN.SP are presented.

In terms of power, naïve SCRAN coupled with DESeq2 shows performance similar to DESeq2. In all significance levels tested, TASC overpowers SCRAN+DESeq2 by up to 30%, especially for moderately differentially expressed genes with fold change around 1.75.

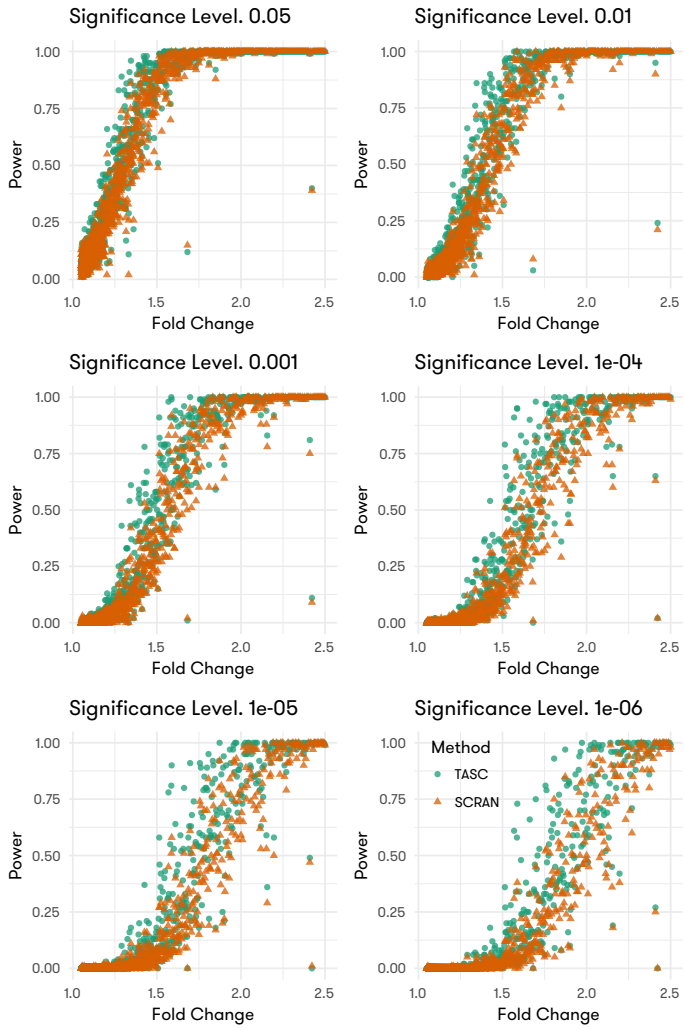


Figure 5.22: Compare power between TASC and SCRAN(Lun, Bach, and Marioni, 2016) with various effect sizes. Each panel contains the power curve of TASC and SCRAN(Lun, Bach, and Marioni, 2016) under the specified significance level. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

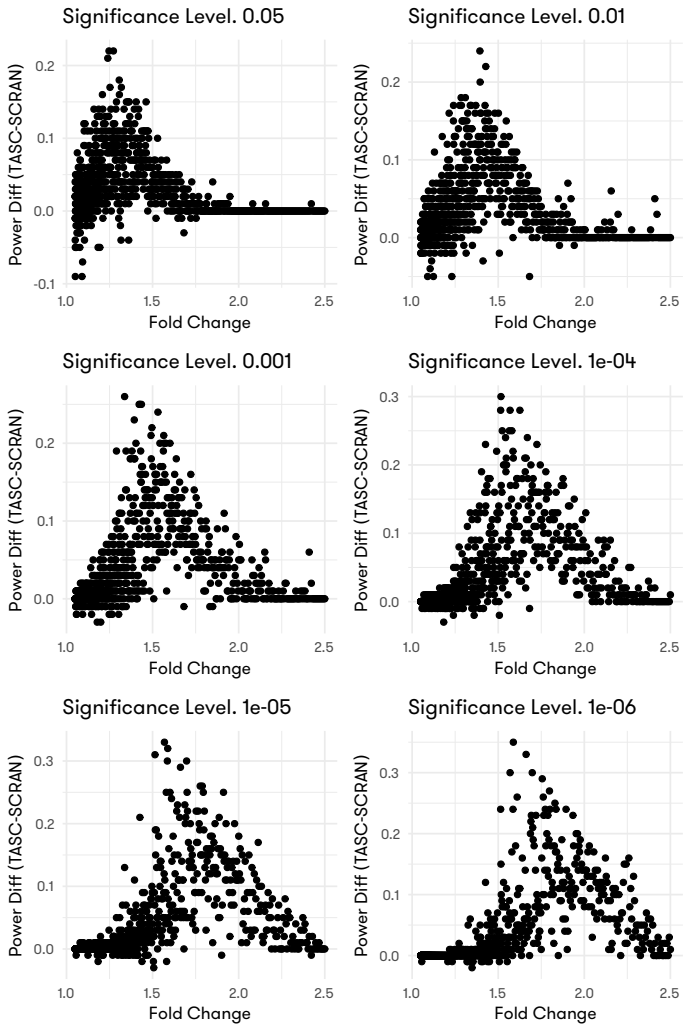


Figure 5.23: Power improvement of TASC over SCRAN(Lun, Bach, and Marioni, 2016) with various effect sizes. Each panel contains the power improvement curve of TASC and SCRAN(Lun, Bach, and Marioni, 2016) under the specified significance level. Y axis represents the difference in absolute not relative values in estimated power between TASC and SCRAN(Lun, Bach, and Marioni, 2016), *i.e.* $\omega_g^{\mathrm{TASC}} - \omega_g^{\mathrm{SCRAN}}$. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

Due to the incorporation of spike-in information, SCRAN.SP coupled with DESeq2 shows profound improvement of power over DESeq2. When compared to TASC, SCRAN.SP is only moderately disadvantaged by up to about 10-20%, the best performer among all methods tested.

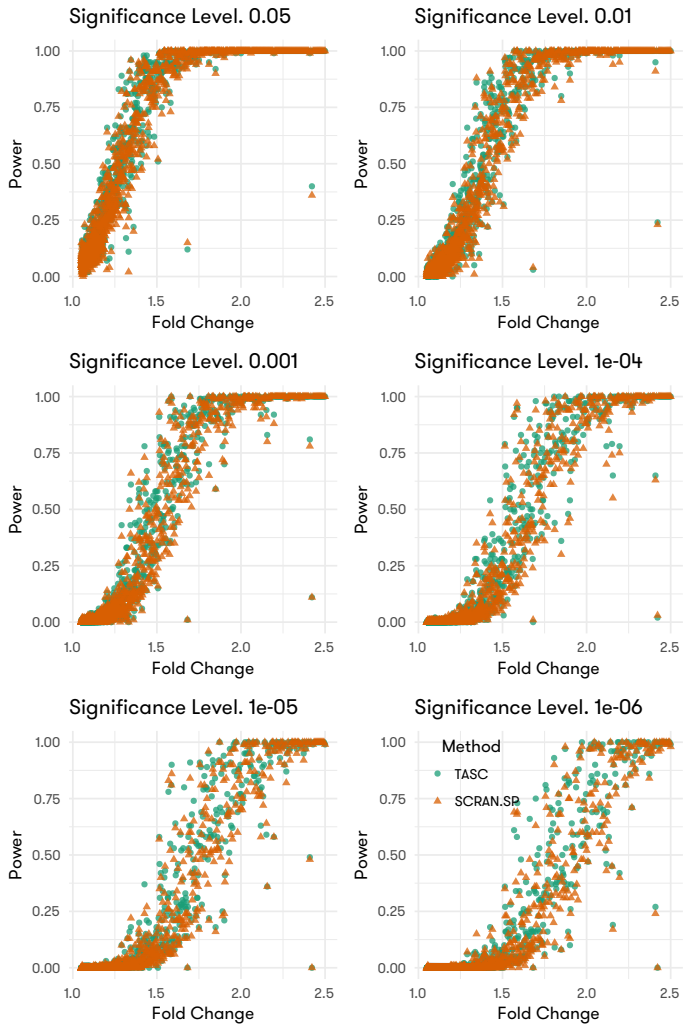


Figure 5.24: Compare power between TASC and SCRAN(Lun, Bach, and Marioni, 2016) with various effect sizes. Each panel contains the power curve of TASC and SCRAN(Lun, Bach, and Marioni, 2016) under the specified significance level. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

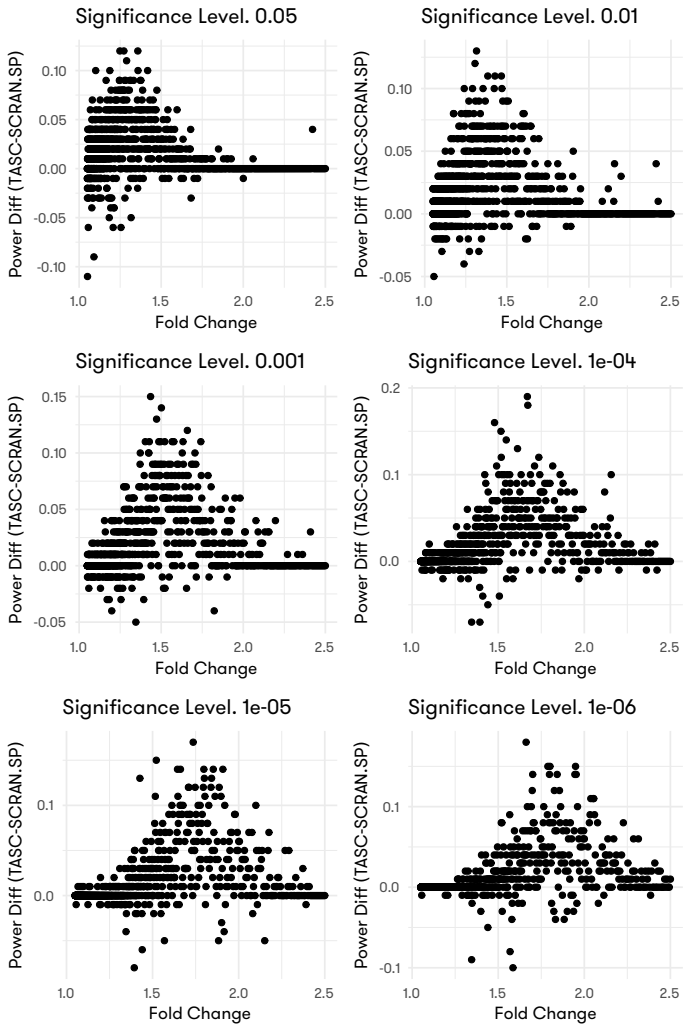


Figure 5.25: Power improvement of TASC over SCRAN(Lun, Bach, and Marioni, 2016) with various effect sizes. Each panel contains the power improvement curve of TASC and SCRAN(Lun, Bach, and Marioni, 2016) under the specified significance level. Y axis represents the difference in absolute not relative values in estimated power between TASC and SCRAN(Lun, Bach, and Marioni, 2016), *i.e.* $\omega_g^{TASC} - \omega_g^{SCRAN}$. This plot is generated from the simulation 100 vs 100 (Figure 5.11).

Power and sample size

To investigate the relationship between power achieved by a method and the sample size required, we have down-sampled the complete dataset into varying sizes in different simulations (Figure 5.11). This has allowed us to look into the behaviour of TASC under different sample size with greater detail.

As the sample size increases, TASC becomes more powerful in detecting small changes in gene expression (Figure 5.26). When the sample size is only 20 vs 20, TASC has virtually no power except for genes that are highly differentially expressed ($\eta_g \geqslant 2.5$). These genes however can be detected by TASC with almost 100% power when the sample size is equal to or greater than 50 vs 50. For moderately differentially expressed genes ($1.5 < \eta_g < 2$), TASC would require at least 100 vs 100 to achieve considerable power. For genes with small changes in its expression ($\eta_g < 1.3$), TASC shows no power when the sample size is smaller than or equal to 200 vs 200. However, it is extremely difficult to detect these with significant power without sacrificing the false positive rate.

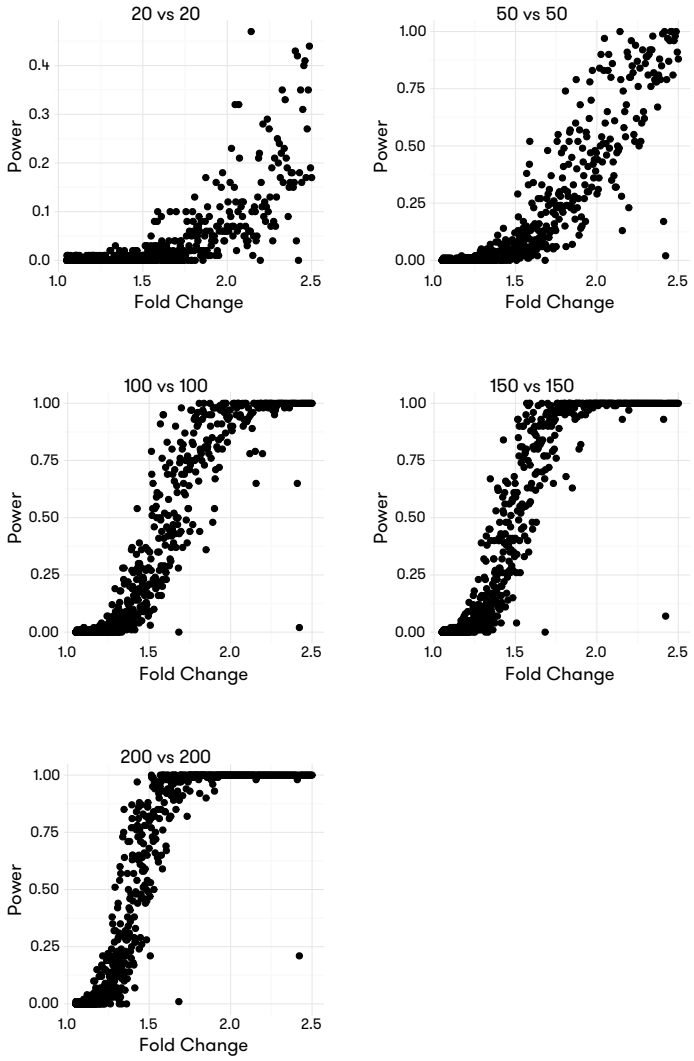


Figure 5.26: Power curves for TASC from simulations with different sample sizes. In each panel, the estimated power ω_g of each gene for TASC is plotted against the effect size (fold change) assigned for this gene simulated at the specified sample size.

In order to assess the average power for genes of specific effect size, we have used the generalized additive model (GAM) to smooth out the power curve. Briefly, the relationship between estimated power of a gene (ω_g) is regressed onto the fold change assigned to this gene (η_g) using GAM with smooth terms $\mathrm{df} = 4$ and $\mathrm{spar} = 1$ for the spline. Resulting smoothed curves are then plotted for each method under various sample sizes for comparison (Figure 5.14a and 5.27).

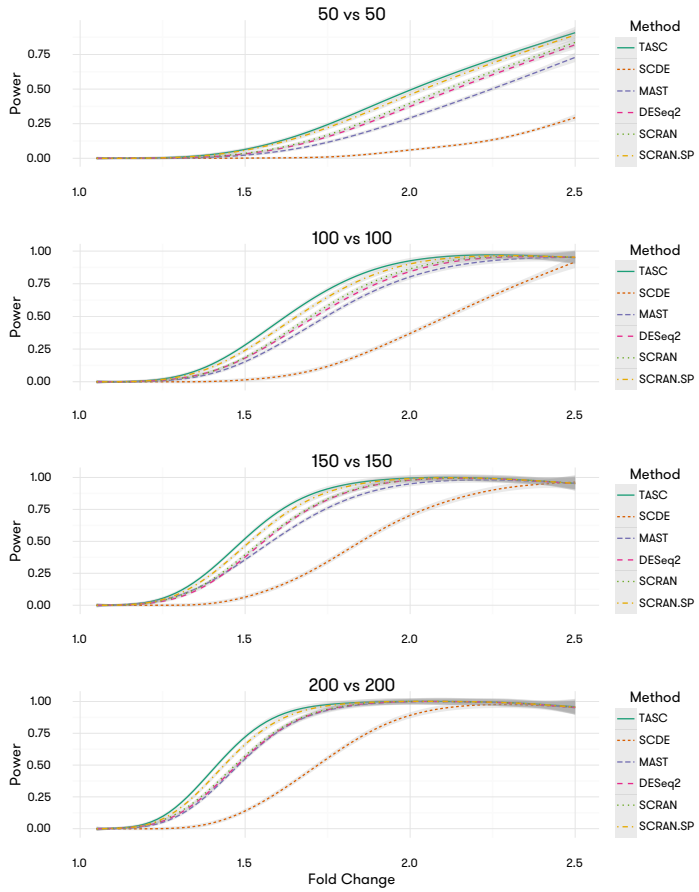


Figure 5.27: Power curves for TASC, SCDE, MAST, DESeq2, SCRAN and SCRAN.SP for sample sizes of 50 vs 50 and above. In each panel the smoothed power curves for all methods from specified sample size are plotted. X axis indicates the fold change η_g for each gene. Y axis represents the average power for each method after smoothing with GAM as described.

From all simulations of varying sample sizes, TASC has the best power among the four methods tested. TASC is particularly powerful when the genes are only moderately differentially expressed ($\eta \approx 1.75$). This improvement is more dramatic when the sample size is relatively modest (50 vs 50). As the sample size goes up, almost all methods can reliably detect the highly DE genes ($\eta > 2$) with 100% power, which suggests the importance of decently large sample size in single-cell

experiments.

5.3.5. Differential Expression analysis on real data

Zeisel et al. data

To gauge the performance of our method in real use case scenarios, we have performed differential gene expression analyses using the two largest level 2 classes of the Zeisel data ("CA1Pyr2" and "CA1Pyr1"). Since these two level-2 classes represent different cell type groups, we expect genuine gene expression differences between them. To evaluate the impact of sample size, the two groups are subsampled to $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ of their original size, as shown in Table 5.1, and differential expression analyses are performed on each subsampled data set. The raw p-values are used to detect DE genes at the 0.0001 significance level, and the number of detected DE genes is plotted against the sample size for each method. The numbers of detected DE genes are shown in Table 5.1. Consistent with our simulations, SCDE finds the least number of DE genes, followed by MAST, whereas SCRAN.SP detects the most number of DE genes when is greater than 100. TASC, SCRAN, and DESeq2 detect similar number of DE genes across most sample sizes.

CA1Pyr2	CA1Pyr1	Numerical Label	Text Label
380	380	380	S32
190	190	190	S16
95	95	95	S8
48	48	48	S4
24	24	24	S2
12	12	12	S1

Table 5.1: Sample sizes of the sub-sampled Zeisel data(Zeisel et al., 2015) sets for two group comparison. Numerical labels are used to approximate the sample sizes in plotting. Text labels are used to distinguish analyses during discussion.

In order to assess the biological relevance of the differentially expressed genes discovered by each method, a gene ontology study has been performed, with results summarized in Tables 5.2,5.3,5.4,5.5,5.6,5.7. All genes used are called by each method with p-values smaller than 10^{-8}. This significance level was chosen in order to find the strongest DE genes, while preserving enough

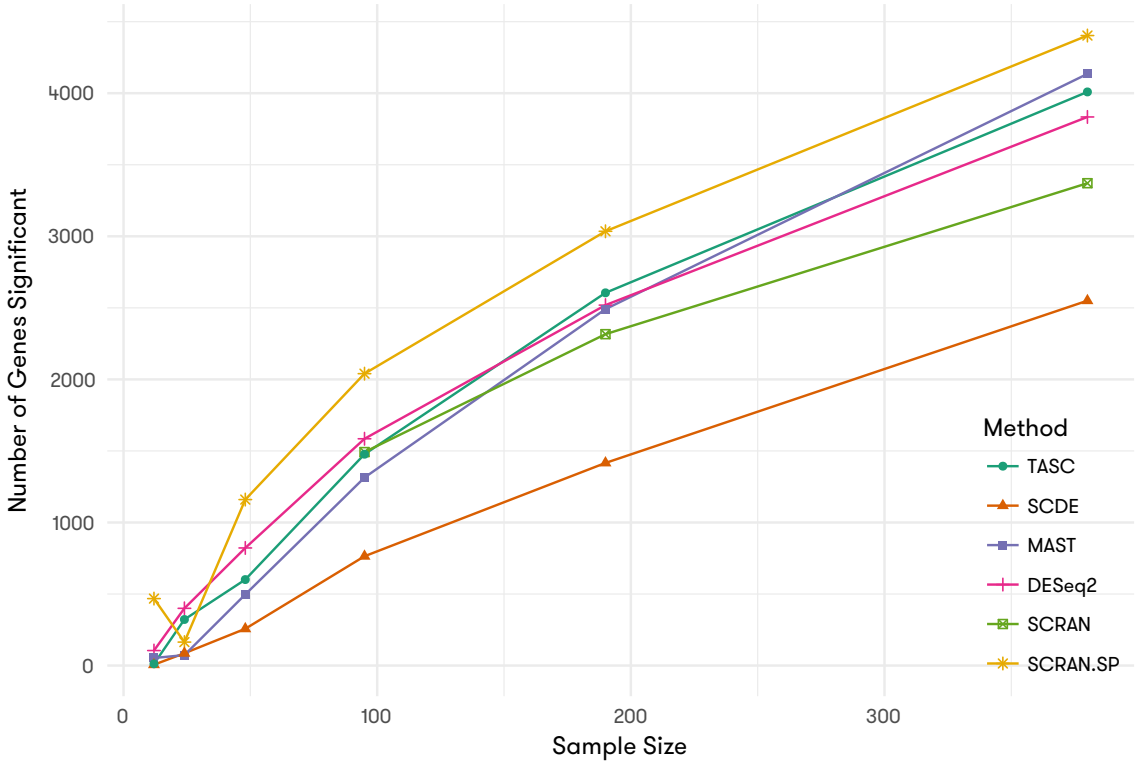


Figure 5.28: Number of DE genes identified by each method between two level-2 classes in Zeisel et al. data at the 0.0001 significance level, under varying sample sizes.

genes for meaningful ontology analysis.

SCAP-T data

In order to test the performance of our model using noisier non-UMI data, we have taken advantage of the SCAP-T dataset, which is an scRNA-seq data from murine brain cells acquired from the SCAP-T study (dbGaP Study Accession phs000835.v4.p1). This data set, which does not have UMIs, contains counts of 46,422 endogenous genes and 87 ERCC spike-ins of 198 neurons and 26 astrocytes from mouse brain. The counts are preprocessed by two filtering procedures: Filter 1 keeps the top 25% of genes in total read account across all the cells. Filter 2 keeps all the genes with non-zero counts in 5 cells or more. Since neurons and astrocytes are processed on different days, this allows us to evaluate whether our model is able to capture and control batch effect. Unlike the Zeisel et al data, SCAP-T data is much noisier, and the cells are much more heterogeneous. In Figure 5.29, a wide range of values for the parameters $(\alpha, \beta, \kappa, \tau)$ can be observed for these samples, and some significant difference exists within the same tissue type as well.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_MF_FAT	GO:0044822 poly(A) RNA binding	319	14.35644	8.13E-59	1802	1197	16866	1.42E-55	1.38E-55
GOTERM_MF_FAT	GO:0003723 RNA binding	400	18.0018	2.47E-58	1802	1707	16866	4.33E-55	4.18E-55
GOTERM_CC_FAT	GO:0098800 inner mitochondrial membrane protein complex	77	3.465347	3.15E-37	1839	124	14413	3.41E-34	5.03E-34
GOTERM_CC_FAT	GO:0098798 mitochondrial protein complex	87	3.915392	1.43E-35	1839	161	14413	1.55E-32	2.28E-32
GOTERM_BP_FAT	GO:0051641 cellular localization	426	19.17192	1.56E-33	1914	2310	17911	1.27E-29	3.13E-30
GOTERM_CC_FAT	GO:0097458 neuron part	371	16.69667	1.88E-33	1839	1620	14413	2.04E-30	3.01E-30
GOTERM_MF_FAT	GO:0003735 structural constituent of ribosome	102	4.590459	1.39E-32	1802	264	16866	2.44E-29	2.36E-29
GOTERM_CC_FAT	GO:0030529 intracellular ribonucleoprotein complex	249	11.20612	2.22E-32	1839	934	14413	2.40E-29	3.54E-29
GOTERM_CC_FAT	GO:1990904 ribonucleoprotein complex	249	11.20612	2.66E-32	1839	935	14413	2.89E-29	4.25E-29
GOTERM_BP_FAT	GO:0043933 macromolecular complex subunit organization	410	18.45185	6.97E-32	1914	2229	17911	5.68E-28	1.39E-28
GOTERM_BP_FAT	GO:0034622 cellular macromolecular complex assembly	211	9.49595	2.19E-30	1914	887	17911	1.78E-26	4.38E-27
GOTERM_CC_FAT	GO:0005840 ribosome	113	5.085509	3.70E-30	1839	286	14413	4.01E-27	5.91E-27
GOTERM_CC_FAT	GO:0044391 ribosomal subunit	95	4.275428	1.16E-29	1839	216	14413	1.26E-26	1.85E-26
GOTERM_BP_FAT	GO:0044085 cellular component biogenesis	459	20.65707	1.51E-29	1914	2654	17911	1.23E-25	3.01E-26
GOTERM_CC_FAT	GO:0044455 mitochondrial membrane part	89	4.005401	1.83E-28	1839	199	14413	1.98E-25	2.92E-25
GOTERM_CC_FAT	GO:0044429 mitochondrial part	228	10.26103	1.97E-28	1839	870	14413	2.13E-25	3.14E-25
GOTERM_CC_FAT	GO:0043005 neuron projection	293	13.18632	1.41E-27	1839	1253	14413	1.53E-24	2.25E-24
GOTERM_CC_FAT	GO:0005743 mitochondrial inner membrane	137	6.165617	3.13E-27	1839	415	14413	3.39E-24	5.00E-24
GOTERM_BP_FAT	GO:0022607 cellular component assembly	419	18.85689	7.11E-27	1914	2418	17911	5.79E-23	1.42E-23

Table 5.2: Top 20 GO terms discovered for differentially expressed genes called by TASC.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_CC_FAT	GO:0098800 inner mitochondrial membrane protein complex	72	5.930807	3.52E-49	1027	124	14413	3.13E-46	5.48E-46
GOTERM_CC_FAT	GO:0098798 mitochondrial protein complex	81	6.672158	5.84E-49	1027	161	14413	5.21E-46	9.10E-46
GOTERM_CC_FAT	GO:0044455 mitochondrial membrane part	82	6.75453	2.42E-41	1027	199	14413	2.16E-38	3.77E-38
GOTERM_CC_FAT	GO:0070469 respiratory chain	53	4.365733	3.06E-37	1027	88	14413	2.72E-34	4.76E-34
GOTERM_CC_FAT	GO:0005746 mitochondrial respiratory chain	50	4.118616	3.19E-36	1027	80	14413	2.84E-33	4.96E-33
GOTERM_CC_FAT	GO:0044429 mitochondrial part	170	14.00329	1.92E-35	1027	870	14413	1.71E-32	2.99E-32
GOTERM_CC_FAT	GO:0005743 mitochondrial inner membrane	111	9.143328	2.08E-35	1027	415	14413	1.86E-32	3.24E-32
GOTERM_CC_FAT	GO:0005740 mitochondrial envelope	143	11.77924	2.79E-35	1027	652	14413	2.49E-32	4.35E-32
GOTERM_CC_FAT	GO:0098803 respiratory chain complex	48	3.953871	5.85E-34	1027	79	14413	5.21E-31	9.11E-31
GOTERM_CC_FAT	GO:0031966 mitochondrial membrane	135	11.12026	6.40E-34	1027	607	14413	5.70E-31	9.96E-31
GOTERM_CC_FAT	GO:0019866 organelle inner membrane	112	9.2257	4.20E-31	1027	467	14413	3.74E-28	6.54E-28
GOTERM_CC_FAT	GO:0005739 mitochondrion	254	20.92257	2.18E-29	1027	1792	14413	1.94E-26	3.39E-26
GOTERM_CC_FAT	GO:0031967 organelle envelope	178	14.66227	3.89E-28	1027	1069	14413	3.46E-25	6.05E-25
GOTERM_CC_FAT	GO:0031975 envelope	178	14.66227	6.90E-28	1027	1074	14413	6.14E-25	1.07E-24
GOTERM_CC_FAT	GO:0030964 NADH dehydrogenase complex	34	2.800659	1.04E-27	1027	47	14413	9.27E-25	1.62E-24
GOTERM_CC_FAT	GO:0045271 respiratory chain complex I	34	2.800659	1.04E-27	1027	47	14413	9.27E-25	1.62E-24
GOTERM_CC_FAT	GO:0005747 mitochondrial respiratory chain complex I	34	2.800659	1.04E-27	1027	47	14413	9.27E-25	1.62E-24
GOTERM_CC_FAT	GO:0097458 neuron part	233	19.19275	1.22E-27	1027	1620	14413	1.09E-24	1.90E-24
GOTERM_CC_FAT	GO:0043005 neuron projection	193	15.89786	3.11E-26	1027	1253	14413	2.77E-23	4.85E-23

Table 5.3: Top 20 GO terms discovered for differentially expressed genes called by SCDE.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_MF_FAT	GO:0044822 poly(A) RNA binding	301	13.87097	1.51E-51	1763	1197	16866	2.61E-48	2.55E-48
GOTERM_MF_FAT	GO:0003723 RNA binding	376	17.32719	7.54E-50	1763	1707	16866	1.30E-46	1.27E-46
GOTERM_CC_FAT	GO:0098798 mitochondrial protein complex	88	4.0553	2.26E-37	1792	161	14413	2.43E-34	3.61E-34
GOTERM_CC_FAT	GO:0098800 inner mitochondrial membrane protein complex	76	3.502304	5.74E-37	1792	124	14413	6.15E-34	9.14E-34
GOTERM_CC_FAT	GO:0097458 neuron part	368	16.95853	7.42E-35	1792	1620	14413	7.96E-32	1.18E-31
GOTERM_BP_FAT	GO:0051641 cellular localization	418	19.26267	8.36E-33	1879	2310	17911	6.72E-29	1.67E-29
GOTERM_BP_FAT	GO:0043933 macromolecular complex subunit organization	404	18.61751	9.54E-32	1879	2229	17911	7.66E-28	1.91E-28
GOTERM_CC_FAT	GO:0044455 mitochondrial membrane part	90	4.147465	4.56E-30	1792	199	14413	4.89E-27	7.26E-27
GOTERM_BP_FAT	GO:0034622 cellular macromolecular complex assembly	207	9.539171	1.03E-29	1879	887	17911	8.29E-26	2.06E-26
GOTERM_CC_FAT	GO:0030529 intracellular ribonucleoprotein complex	238	10.96774	1.75E-29	1792	934	14413	1.88E-26	2.79E-26
GOTERM_CC_FAT	GO:1990904 ribonucleoprotein complex	238	10.96774	2.07E-29	1792	935	14413	2.22E-26	3.30E-26
GOTERM_CC_FAT	GO:0043005 neuron projection	289	13.31797	4.62E-28	1792	1253	14413	4.96E-25	7.37E-25
GOTERM_BP_FAT	GO:0044085 cellular component biogenesis	447	20.59908	6.43E-28	1879	2654	17911	5.16E-24	1.28E-24
GOTERM_CC_FAT	GO:0070469 respiratory chain	55	2.534562	2.07E-27	1792	88	14413	2.22E-24	3.30E-24
GOTERM_CC_FAT	GO:0005746 mitochondrial respiratory chain	52	2.396313	4.24E-27	1792	80	14413	4.55E-24	6.75E-24
GOTERM_MF_FAT	GO:0003735 structural constituent of ribosome	93	4.285714	5.82E-27	1763	264	16866	1.01E-23	9.83E-24
GOTERM_CC_FAT	GO:0044429 mitochondrial part	220	10.13825	1.00E-26	1792	870	14413	1.08E-23	1.60E-23
GOTERM_CC_FAT	GO:0005743 mitochondrial inner membrane	133	6.129032	3.69E-26	1792	415	14413	3.96E-23	5.89E-23
GOTERM_CC_FAT	GO:0044456 synapse part	187	8.617512	6.23E-26	1792	697	14413	6.69E-23	9.94E-23

Table 5.4: Top 20 GO terms discovered for differentially expressed genes called by MAST.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_MF_FAT	GO:0003723 RNA binding	403	18.16133	2.82E-62	1767	1707	16866	4.91E-59	4.77E-59
GOTERM_MF_FAT	GO:0044822 poly(A) RNA binding	320	14.42091	1.93E-61	1767	1197	16866	3.37E-58	3.27E-58
GOTERM_CC_FAT	GO:0030529 intracellular ribonucleoprotein complex	274	12.3479	3.36E-44	1835	934	14413	3.64E-41	5.36E-41
GOTERM_CC_FAT	GO:1990904 ribonucleoprotein complex	274	12.3479	4.19E-44	1835	935	14413	4.55E-41	6.70E-41
GOTERM_CC_FAT	GO:0098800 inner mitochondrial membrane protein complex	80	3.605228	1.63E-40	1835	124	14413	1.77E-37	2.60E-37
GOTERM_MF_FAT	GO:0003735 structural constituent of ribosome	110	4.957188	1.69E-39	1767	264	16866	2.94E-36	2.86E-36
GOTERM_CC_FAT	GO:0005840 ribosome	126	5.678233	1.82E-39	1835	286	14413	1.98E-36	2.91E-36
GOTERM_CC_FAT	GO:0098798 mitochondrial protein complex	90	4.055881	1.96E-38	1835	161	14413	2.12E-35	3.13E-35
GOTERM_CC_FAT	GO:0044391 ribosomal subunit	103	4.641731	6.10E-36	1835	216	14413	6.62E-33	9.73E-33
GOTERM_CC_FAT	GO:0044429 mitochondrial part	244	10.99594	1.21E-35	1835	870	14413	1.32E-32	1.94E-32
GOTERM_CC_FAT	GO:0044455 mitochondrial membrane part	96	4.326273	4.99E-34	1835	199	14413	5.41E-31	7.96E-31
GOTERM_CC_FAT	GO:0005739 mitochondrion	400	18.02614	6.55E-34	1835	1792	14413	7.10E-31	1.04E-30
GOTERM_BP_FAT	GO:0051641 cellular localization	423	19.06264	6.31E-33	1907	2310	17911	5.08E-29	1.26E-29
GOTERM_BP_FAT	GO:0006518 peptide metabolic process	209	9.418657	8.62E-31	1907	872	17911	6.94E-27	1.72E-27
GOTERM_BP_FAT	GO:0034622 cellular macromolecular complex assembly	211	9.508788	1.31E-30	1907	887	17911	1.06E-26	2.62E-27
GOTERM_BP_FAT	GO:0006412 translation	182	8.201893	2.91E-30	1907	714	17911	2.34E-26	5.81E-27
GOTERM_CC_FAT	GO:0005743 mitochondrial inner membrane	142	6.399279	3.94E-30	1835	415	14413	4.27E-27	6.29E-27
GOTERM_BP_FAT	GO:0043043 peptide biosynthetic process	185	8.337089	5.04E-30	1907	735	17911	4.05E-26	1.01E-26
GOTERM_BP_FAT	GO:0044085 cellular component biogenesis	459	20.68499	6.21E-30	1907	2654	17911	5.00E-26	1.24E-26

Table 5.5: Top 20 GO terms discovered for differentially expressed genes called by DESeq2.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_MF_FAT	GO:0003723 RNA binding	386	18.60241	9.88E-62	1660	1707	16866	1.67E-58	1.66E-58
GOTERM_MF_FAT	GO:0044822 poly(A) RNA binding	308	14.84337	2.49E-61	1660	1197	16866	4.22E-58	4.20E-58
GOTERM_CC_FAT	GO:0030529 intracellular ribonucleoprotein complex	268	12.91566	2.49E-46	1727	934	14413	2.64E-43	3.96E-43
GOTERM_CC_FAT	GO:1990904 ribonucleoprotein complex	268	12.91566	3.10E-46	1727	935	14413	3.30E-43	4.94E-43
GOTERM_CC_FAT	GO:0098800 inner mitochondrial membrane protein complex	80	3.855422	1.78E-42	1727	124	14413	1.89E-39	2.83E-39
GOTERM_CC_FAT	GO:0005840 ribosome	125	6.024096	1.69E-41	1727	286	14413	1.79E-38	2.69E-38
GOTERM_CC_FAT	GO:0098798 mitochondrial protein complex	90	4.337349	1.43E-40	1727	161	14413	1.51E-37	2.27E-37
GOTERM_MF_FAT	GO:0003735 structural constituent of ribosome	108	5.204819	2.02E-40	1660	264	16866	3.42E-37	3.40E-37
GOTERM_CC_FAT	GO:0044429 mitochondrial part	243	11.71084	1.21E-39	1727	870	14413	1.29E-36	1.93E-36
GOTERM_CC_FAT	GO:0044391 ribosomal subunit	102	4.915663	1.90E-37	1727	216	14413	2.02E-34	3.03E-34
GOTERM_CC_FAT	GO:0005739 mitochondrion	391	18.84337	6.36E-37	1727	1792	14413	6.76E-34	1.01E-33
GOTERM_CC_FAT	GO:0044455 mitochondrial membrane part	96	4.626506	3.26E-36	1727	199	14413	3.46E-33	5.19E-33
GOTERM_CC_FAT	GO:0005743 mitochondrial inner membrane	141	6.795181	2.27E-32	1727	415	14413	2.41E-29	3.62E-29
GOTERM_BP_FAT	GO:0006518 peptide metabolic process	203	9.783133	1.50E-31	1799	872	17911	1.18E-27	3.00E-28
GOTERM_CC_FAT	GO:0005740 mitochondrial envelope	186	8.963855	2.10E-31	1727	652	14413	2.23E-28	3.35E-28
GOTERM_BP_FAT	GO:0034622 cellular macromolecular complex assembly	203	9.783133	1.72E-30	1799	887	17911	1.35E-26	3.43E-27
GOTERM_BP_FAT	GO:0006412 translation	176	8.481928	1.79E-30	1799	714	17911	1.41E-26	3.56E-27
GOTERM_BP_FAT	GO:0043043 peptide biosynthetic process	179	8.626506	2.63E-30	1799	735	17911	2.07E-26	5.25E-27
GOTERM_BP_FAT	GO:0051641 cellular localization	394	18.98795	2.57E-29	1799	2310	17911	2.02E-25	5.12E-26

Table 5.6: Top 20 GO terms discovered for differentially expressed genes called by SCRAN.

Category	Term	Count	%	PValue	List Total	Pop Hits	Pop Total	Bonferroni	FDR
GOTERM_CC_FAT	GO:0097458 neuron part	496	18.30934	1.42E-61	2217	1620	14413	1.62E-58	2.28E-58
GOTERM_BP_FAT	GO:0051641 cellular localization	561	20.70875	3.69E-55	2363	2310	17911	3.26E-51	7.44E-52
GOTERM_MF_FAT	GO:0044822 poly(A) RNA binding	355	13.10447	1.81E-54	2229	1197	16866	3.51E-51	3.10E-51
GOTERM_CC_FAT	GO:0043005 neuron projection	395	14.58103	1.86E-51	2217	1253	14413	2.12E-48	2.98E-48
GOTERM_BP_FAT	GO:0007399 nervous system development	540	19.93355	2.37E-50	2363	2261	17911	2.09E-46	4.77E-47
GOTERM_CC_FAT	GO:0044456 synapse part	264	9.745293	6.06E-50	2217	697	14413	6.91E-47	9.74E-47
GOTERM_BP_FAT	GO:0031175 neuron projection development	289	10.66814	3.41E-49	2363	921	17911	3.01E-45	6.87E-46
GOTERM_MF_FAT	GO:0003723 RNA binding	438	16.16833	1.04E-48	2229	1707	16866	2.01E-45	1.78E-45
GOTERM_CC_FAT	GO:0045202 synapse	304	11.22185	2.01E-48	2217	876	14413	2.29E-45	3.23E-45
GOTERM_BP_FAT	GO:0048666 neuron development	318	11.73865	1.54E-46	2363	1092	17911	1.36E-42	3.11E-43
GOTERM_BP_FAT	GO:0030182 neuron differentiation	367	13.54743	5.47E-44	2363	1379	17911	4.83E-40	1.10E-40
GOTERM_BP_FAT	GO:0048699 generation of neurons	394	14.54411	5.48E-44	2363	1526	17911	4.84E-40	1.10E-40
GOTERM_BP_FAT	GO:0051649 establishment of localization in cell	429	15.8361	6.74E-44	2363	1722	17911	5.95E-40	1.36E-40
GOTERM_CC_FAT	GO:0036477 somatodendritic compartment	301	11.11111	5.80E-42	2217	921	14413	6.62E-39	9.32E-39
GOTERM_BP_FAT	GO:0022008 neurogenesis	409	15.09782	6.04E-42	2363	1638	17911	5.34E-38	1.22E-38
GOTERM_BP_FAT	GO:0051128 regulation of cellular component organization	536	19.7859	1.07E-40	2363	2399	17911	9.41E-37	2.15E-37
GOTERM_BP_FAT	GO:0048812 neuron projection morphogenesis	196	7.235142	2.51E-39	2363	570	17911	2.21E-35	5.05E-36
GOTERM_BP_FAT	GO:0043933 macromolecular complex subunit organization	501	18.49391	1.83E-38	2363	2229	17911	1.61E-34	3.68E-35
GOTERM_BP_FAT	GO:0046907 intracellular transport	337	12.44001	2.28E-37	2363	1304	17911	2.02E-33	4.61E-34

Table 5.7: Top 20 GO terms discovered for differentially expressed genes called by SCRAN.SP.

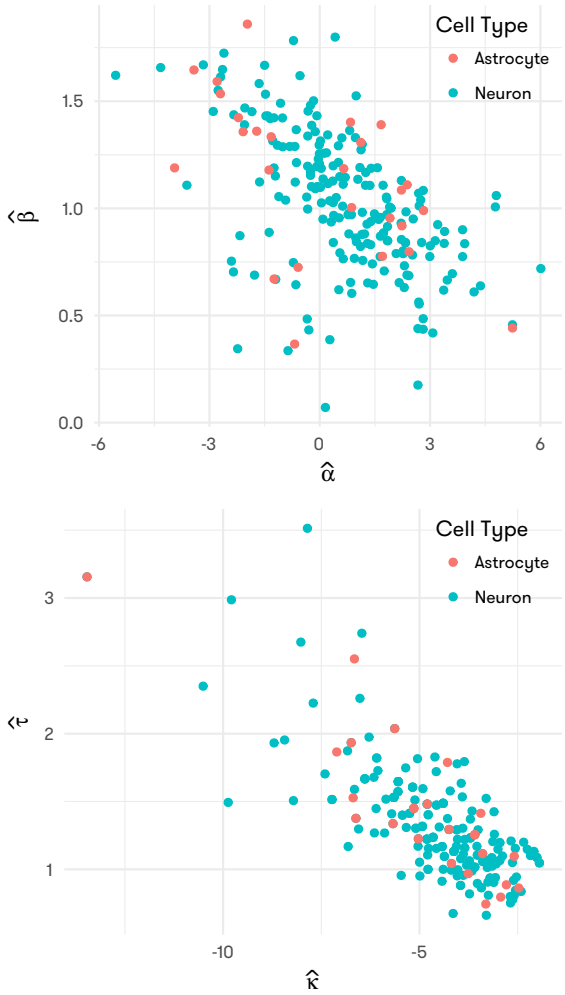


Figure 5.29: Scatter plots describing the distribution of Ψ_c of the SCAP-T data.

Before we can use ERCC spike-ins in the SCAP-T data to model the technical noise, necessary pre-processing is required to tease out the cells that are of low quality. One can achieve this by looking at the R^2 values from the linear regression with the log counts as the response variable, and the log true concentration of the ERCC as the input covariate. SCAP-T data obviously has much wider range of R^2 (Figure 5.30) compared to Zeisel et al. data (Figure 5.33), suggesting some trimming might be necessary to remove those cells with really low R^2 if TASC is to be used.

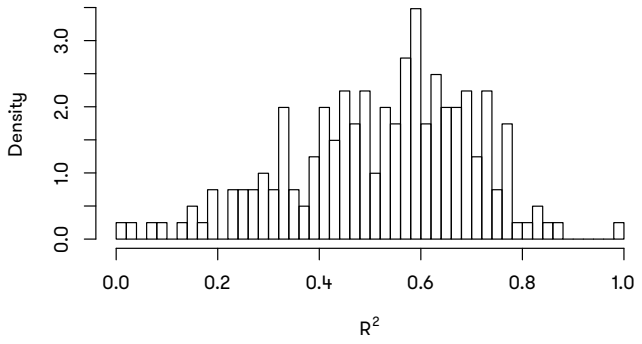


Figure 5.30: Histograms for R^2 computed from SCAP-T data.

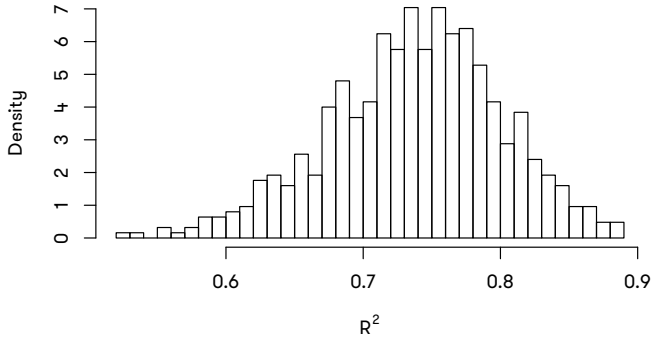


Figure 5.31: Histograms for R^2 computed from Zeisel et al. data.

Another characteristic of the SCAP-T data is the more varied sample size. We have plotted the normalized cell size factors computed from SCAP-T (Figure 5.32) and Zeisel et al. (Figure 5.33) data. It is obvious that the former has much wider range of cell size factors, which indicate that some of the cells in this data set might contain too many or too few reads coming from the biological genes, both of which will affect the accuracy of TASC.

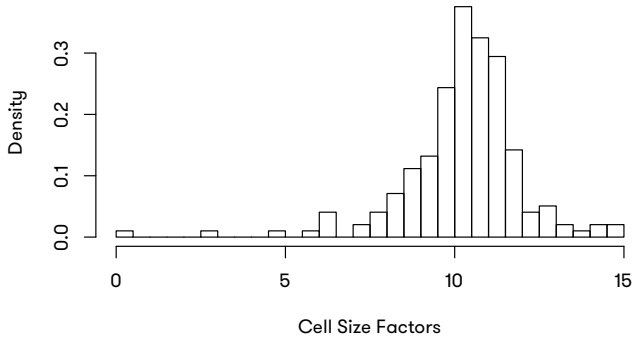


Figure 5.32: Histograms for normalized cell size factors computed from SCAP-T data.

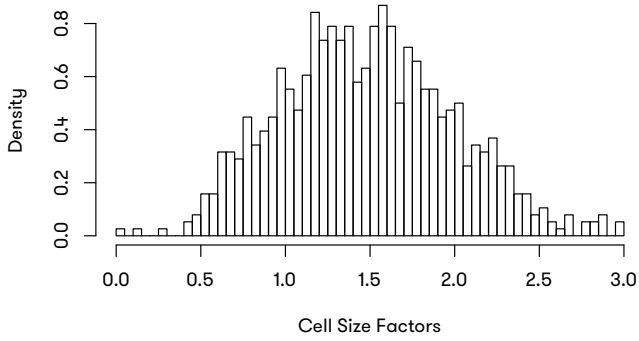


Figure 5.33: Histograms for normalized cell size factors computed from Zeisel et al. data.

To compare the methods with regards to their type I error rate under a real data scenario, we analyzed the SCAP-T data, which includes astrocytes and neurons that were processed on different days. This data set provides a perfect example to illustrate the impact of batch effect. To assess whether type I error is controlled under the null scenario, it is necessary to compare two groups of cells that are of the same type. To perform this assessment, we have derived a null comparison following these steps.

Step.*a* Estimate the technical parameters $(\alpha, \beta, \kappa, \tau)$ for 26 astrocytes and 198 neurons.

Step.*b* Among the 198 neurons, find 26 with the technical parameters closest in Euclidean dis-

tance to that of each astrocyte, and label these neurons as group 1. If multiple astrocytes share one closest neuron, then multiple neurons are selected for these astrocytes.

Step.*c* Label the unchosen 172 neurons as group 2.

Differential gene expression analyses have been performed on these two groups with all five methods (naïve SCRAN is not available due to the small sample size in group 1). The methods TASC, SCDE, MAST, DESeq2, SCRAN, and SCRAN.SP are then applied to these two groups, and the proportion of genes reported to be DE is reported in Table 5.8. Raw p-values are plotted using histograms (Figure 5.34). Negative logarithm of the raw p-values with base 10 are plotted with Q-Q plots (Figure 5.35).

We see that TASC has well controlled type I error rates at all assessed significance levels, whereas all other methods (SCDE, MAST, DESeq2, SCRAN, and SCRAN.SP) have severely inflated type I error rates, especially when the p-value threshold is reduced to 0.001 and 0.0001. For example, consider DESeq2, which, according to our simulations, has well-controlled type I error when there are no batch effects. At significance level of 0.001, DESeq2 has false positive rate of 1.7%, a 17-fold inflation, and at significance level of 0.0001, DESeq2 has false positive rate of 0.76%, corresponding to a 76-fold inflation. Even SCDE, which tends to be conservative when there are no batch effects, suffer from type I inflation in this real data scenario that contains a possible batch effect. The patterns are similar when we consider all genes in the evaluations.

Filter	Filter 1					Filter 2				
Significance Level	0.05	1.00E-02	1.00E-03	1.00E-04	1.00E-05	0.05	1.00E-02	1.00E-03	1.00E-04	1.00E-05
TASC	2.15E-02	2.33E-03	0	0	0	3.09E-02	5.78E-03	5.36E-04	8.93E-05	2.98E-05
SCDE	4.28E-02	1.65E-02	4.74E-03	1.46E-03	3.45E-04	9.42E-02	3.46E-02	8.72E-03	1.79E-03	1.79E-04
MAST	4.89E-02	1.12E-02	1.81E-03	7.75E-04	2.58E-04	7.42E-02	2.62E-02	1.21E-02	8.45E-03	6.25E-03
DESeq2	1.70E-01	8.44E-02	3.52E-02	1.69E-02	8.27E-03	1.25E-01	5.91E-02	2.32E-02	1.01E-02	4.79E-03
SCRAN.SP	2.50E-01	1.39E-01	6.63E-02	3.67E-02	2.11E-02	1.69E-01	9.13E-02	4.19E-02	2.21E-02	1.21E-02

Table 5.8: Proportion of DE genes identified by each method in SCAP-T data at varying significance levels. Filter 1 keeps the top 25% of genes in total read account across all the cells. Filter 2 keeps all the genes with non-zero counts in 5 cells or more. Nave SCRAN without the use of spike-ins is not included in this comparison, for the package fails to run due to there being “not enough cells in each cluster for specified ‘sizes’ ”.

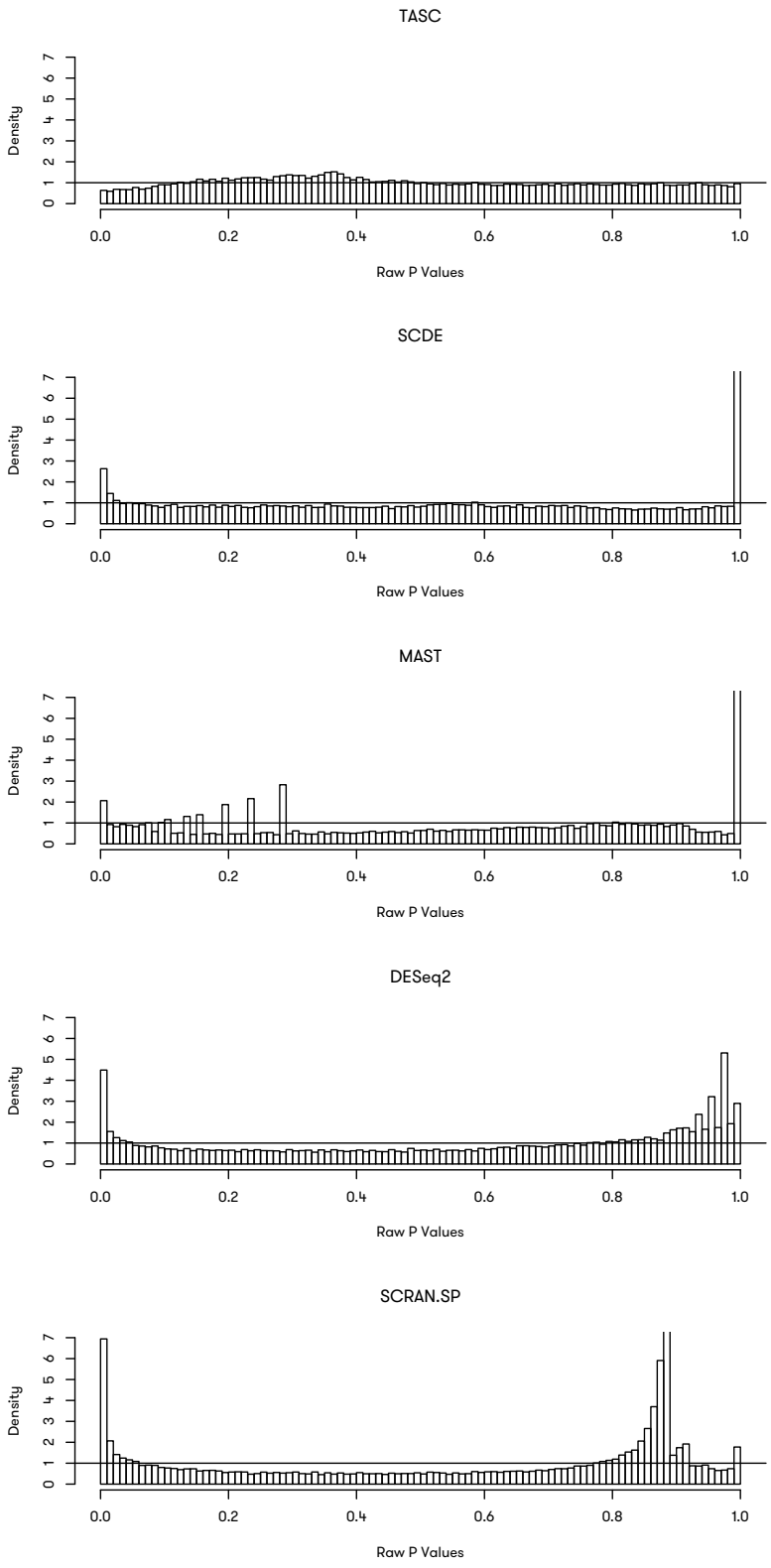


Figure 5.34: Histograms describing the distributions of raw p-values from various methods in the null comparison with SCAP-T data.

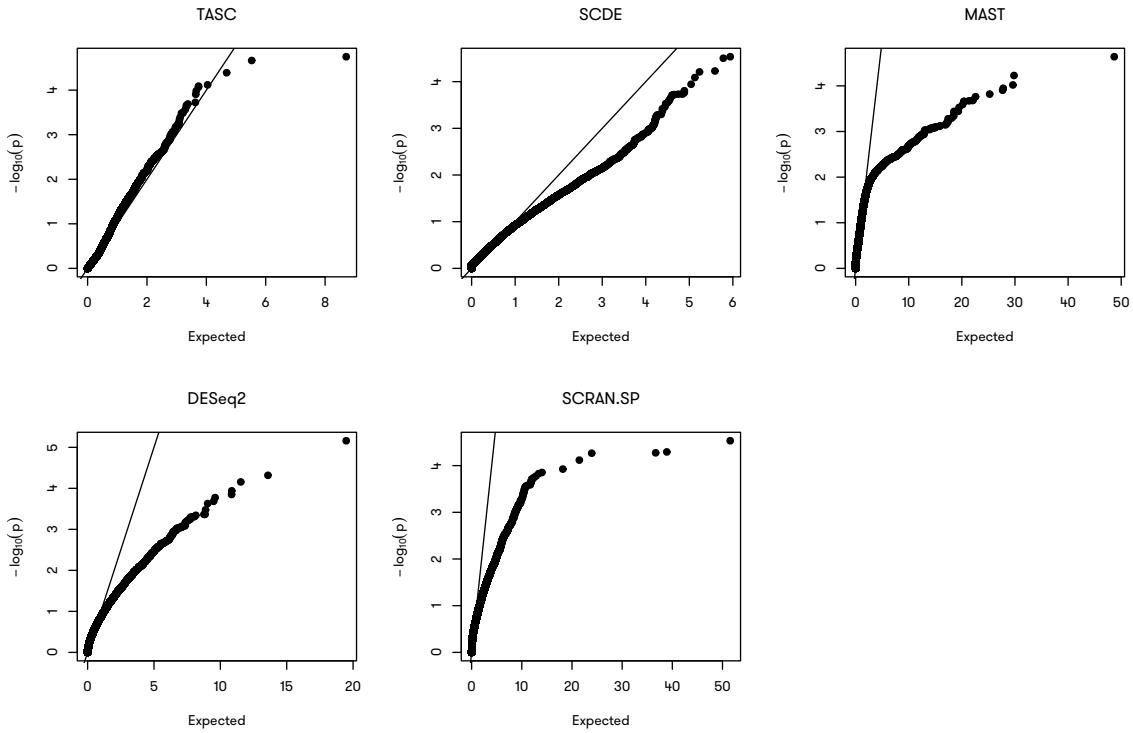


Figure 5.35: Q-Q plots describing the distributions of raw p-values from various methods in the null comparison with SCAP-T data.

5.4. Computational Details

TASC is implemented in an open-source program `https://github.com/scrna-seq/TASC`, with multithreading acceleration by openMP. For example, a data set of 104 cells and 6,405 genes takes 45MB of memory and 18.6 minutes using 20 cores (Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz) with Laplacian approximation using the binary we provided. Better performance can be achieved when using binaries compiled on the users hardware. We believe that TASC will provide a robust platform for researchers to leverage the power of scRNA-seq.

5.4.1. Laplace Approximation

In order to speed up the evaluation of integral, we have adopted Laplace's method to approximate the value and reduce the required computational resources. Briefly, the marginal likelihood of one cell in Equation 5.15 can be approximated with the

$$\log\left[\int_{\mu_{cg}} \Pr\left[Y_{cg}, \mu_{cg}\right] d\mu_{cg}\right] \approx \frac{1}{2}\left[\log(2\pi) - \log\left(h\left[\hat{\mu}_{cg}\right]\right)\right] \tag{5.24}$$

where $\hat{\mu}_{cg}$ is the maximizer of $\Pr\left[Y_{cg}, \mu_{cg}\right]$ over $\mu_{cg}$and $h\left[\mu_{cg}\right]$ is the second derivative of $\Pr\left[Y_{cg}, \mu_{cg}\right]$ over μ_{cg}.

In order to assess the performance of the Laplace's method, we have compared the $\hat{\beta}_1$, the estimated coefficient associated with the group indicator in the two group comparison settings in the Zeisel et al. data using Laplace's method (Laplace) and adaptive quadrature (Integration) in Figure 5.36. The estimates are highly correlated, indicating Laplace's method can give accurate estimates for the parameters of interests, under a wide range of sample sizes.

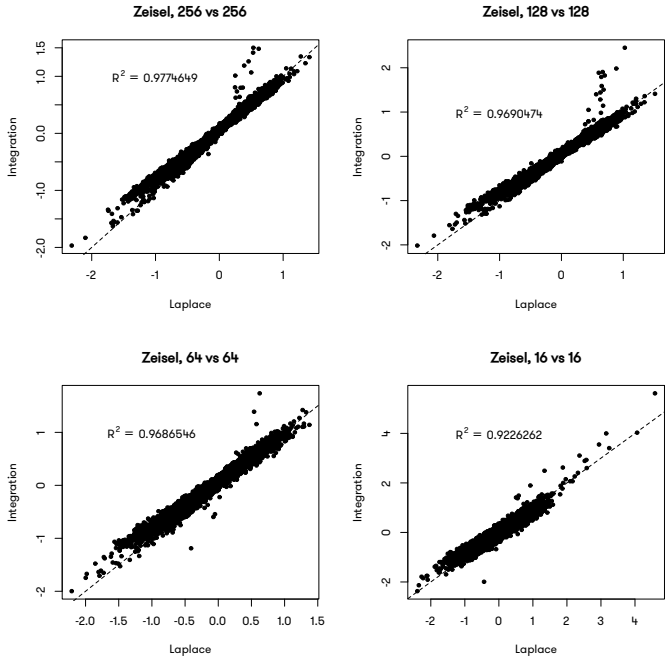


Figure 5.36: Comparison of Laplacian approximation and Adaptive Integration

Using Laplace's method can greatly reduce the CPU time required, as is show in Figure 5.37.

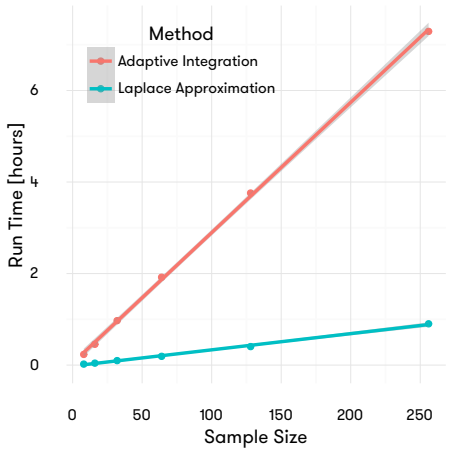


Figure 5.37: Comparison of run time of Laplacian approximation and Adaptive Integration, using 24 cores.

CHAPTER 6

MODELING TRANSCRIPTIONAL BURSTING WITH SCRNA-SEQ DATA

6.1. Motivation

As we have reviewed above, scRNA-seq is a promising technology for studying transcriptional bursting, due to its accurate profiling of transcriptome on a single-cell resolution, as well as its high throughput, the capability of monitoring tens of thousands of genes simultaneously. Previous studies (Kim and Marioni, 2013) attempted to infer the kinetic parameters by fitting a Beta-Poisson model with a Gibbs sampler. However, this study failed to address the intrinsic technical noise such as amplification bias and technical dropout. In addition, Kim and Marioni did not provide any testing procedures for comparing the parameters across experimental conditions. The TASC model can naturally adjust for technical biases present in scRNA-seq data. However, it is incapable of inferring bursting probabilities or testing for differential bursting. This has motivated us to develop TASC-B, an extension to TASC model incorporating additional parameters to characterize probabilities of genes being turned "on" and "off" in a homogeneous population of cells. Moreover, as a likelihood model, we have developed a series of likelihood-ratio tests to draw inference on the significance of differential bursting between groups.

6.2. Generative Model Incorporating Transcriptional Bursting

6.2.1. Extension to TASC model

According to the two-state model of transcriptional bursting, for certain genes, transcription randomly switches between "on" and "off" states. We propose that scRNA-seq data can be used to characterize transcriptional bursting provided that the following assumptions are satisfied.

- Homogeneity of cells sequenced, *i.e.*, all cells in the population of interest follow the same stochastic process of state-switching, with the same switching probabilities. This essentially requires that for a particular gene in consideration, the kinetic parameters of the two-state model are identical across all cells.

- Ergodicity: the population of cells have the same behavior averaged over time as averaged

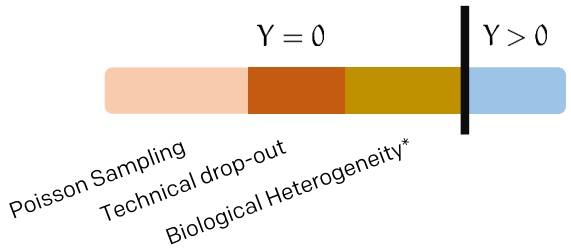


Figure 6.1: Illustration of sources of zeros in scRNA-seq data.

over the space of all the states of the entire population.

This will ensure that we get a sample of cells whose transcriptional status represents that of one individual cell at multiple random time points. Without bursting, this sample can be perfectly described with the TASC model 5.2. With bursting, however, excessive zeros will be observed in the recovered reads Y_{cg} due to some cells being in the "off" state. Intuitively, the zeros in Y_{cg} can be attributed to three distinctive sources, as illustrated in Figure 6.1.

The inflated zeros are primarily sourced from three contributing factors:

- Poisson sampling of sequencing. When the gene is constitutively on, the transcription follows a Poisson process. In some cells, the number of transcripts is 0 simply by stochasticity, especially when the mean expression of the gene is low. The probability of zeros decreases dramatically as the mean expression increases.

- Biological heterogeneity. In genes with significant bursting, excessive zeros that cannot be accounted for by the Poisson sampling alone can be observed in cells that are in the "off" state. Testing for the significant presence of this portion of zeros can provide evidence for the presence of transcriptional bursting.

- Technical drop-out. Due to the complexity of the scRNA-seq protocols, many steps can contribute to the loss of a particular transcript even when it is expressed in the cell sequenced. For example, even the most reverse transcriptase cannot capture 100% of the transcripts in one reaction. Losses due to PCR, sequencing and mapping can also cause zeros when

original cell did have the transcript expressed.

In TASC, we have accounted for the Poissonian zeros and the technical drop-out. To test for the presence of excessive zeros caused by transcriptional bursting, we can extend the TASC model to incorporate a parameter describing the probability of a cell being in the "on" state as follows. For purpose of simplicity, we lose the gene index g.

Step.*a* Let Z_c^B be the indicator representing the status of the bursting state in cell c. If $Z_c^B = 1$, cell c is "on", otherwise when $Z_c^B = 0$, cell c is "off".

Step.*b* $Z_c^B \sim Bernoulli(p^B)$, where p^B is a parameter of interest describing the overall propensity of a cell in this population to be in the "on" state.

Step.*c*

$$\mu_c = \begin{cases} 0, & \text{if } Z_c^B = 0 \\ \text{logNormal}(\theta_g, \sigma_g), & \text{if } Z_c^B = 1 \end{cases} \tag{6.1}$$

Step.*d* Follow identical steps from Step.*b* in subsection 5.2.2.

Following steps of algebra similar to that in subsection 5.2.2, the marginal likelihood of Y_{cg}, μ_{cg} can be written as,

$$\begin{aligned} & Pr\left[Y_{cg}, \mu_{cg}\right] \\ &= \sum_{Z_{cg}=0}^{1} \sum_{Z_{cg}^B=0}^{1} Pr\left[Y_{cg}, \mu_{cg}, Z_{cg}, Z_{cg}^B\right] \\ &= \sum_{Z_{cg}=0}^{1} \sum_{Z_{cg}^B=0}^{1} Pr\left[Y_{cg}|Z_{cg}, \mu_{cg}, Z_{cg}^B\right] Pr\left[Z_{cg}|\mu_{cg}, Z_{cg}^B\right] Pr\left[\mu_{cg}|Z_{cg}^B\right] Pr\left[Z_{cg}^B\right] \end{aligned} \tag{6.2}$$

When $Z_{cg}^B = 0$, $\mu_{cg} = 0$, therefore, $\pi_{cg} = expit\left[\kappa_c + \tau_c \log(\mu_{cg})\right] = 0$. Subsequently, Z_{cg} converges to 0. Therefore, $Pr\left[Z_{cg} = 1|\mu_{cg}, Z_{cg}^B = 0\right] = 0$. Therefore, only three components remain in

the above summation.

$$
\begin{aligned}
&\Pr\left[Y_{cg}, \mu_{cg}\right] \\
=&\Pr\left[Y_{cg}|Z_{cg}=0, \mu_{cg}, Z^B_{cg}=0\right]\Pr\left[Z_{cg}=0|\mu_{cg}, Z^B_{cg}=0\right]\Pr\left[\mu_{cg}|Z^B_{cg}=0\right]\Pr\left[Z^B_{cg}=0\right] \\
+&\Pr\left[Y_{cg}|Z_{cg}=0, \mu_{cg}, Z^B_{cg}=1\right]\Pr\left[Z_{cg}=0|\mu_{cg}, Z^B_{cg}=1\right]\Pr\left[\mu_{cg}|Z^B_{cg}=1\right]\Pr\left[Z^B_{cg}=1\right] \\
+&\Pr\left[Y_{cg}|Z_{cg}=1, \mu_{cg}, Z^B_{cg}=1\right]\Pr\left[Z_{cg}=1|\mu_{cg}, Z^B_{cg}=1\right]\Pr\left[\mu_{cg}|Z^B_{cg}=1\right]\Pr\left[Z^B_{cg}=1\right]
\end{aligned} \tag{6.3}
$$

When $Y_{cg} > 0, \mu_{cg} > 0$, the above three components can be further reduced to one, as the first two components are equal to 0, because $\Pr\left[Y_{cg} > 0|Z_{cg}=0, \mu_{cg}, Z^B_{cg}=0\right] = 0$. Therefore, in this case,

$$
\Pr\left[Y_{cg}, \mu_{cg}\right] = \frac{\left[e^{\alpha_c+\beta_c \log \mu_{cg}}\right]^{Y_{cg}} e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}}}{y_{cg}!}\pi_{cg} f_{LN}\left(\mu_{cg}|\theta_g, \sigma^2_g\right) p^B_g \tag{6.4}
$$

When $Y_{cg} > 0, \mu_{cg} = 0$, the above three components are all equal to 0. Because $\lambda_{cg} = \exp\left[\alpha_c + \beta_c \log \mu_{cg}\right] = \exp\left[-\infty\right] = 0$, and $\Pr\left[Y_{cg}|\lambda_{cg}\right] = f_{Poisson}(Y_{cg} > 0|\lambda_{cg} = 0) = 0$. Therefore, $\Pr\left[Y_{cg} > 0|\mu_c g = 0, Z_{cg}, Z^B_{cg}\right] = 0$, which means when $Y_{cg} > 0, \mu_{cg} = 0$,

$$
\Pr\left[Y_{cg}, \mu_{cg}\right] = 0 \tag{6.5}
$$

When $Y_{cg} = 0, \mu_{cg} = 0$, the last two components in Equation 6.3 are equal to 0. Because when $Z^B_c = 1$, $\mu_{cg} \sim LogNormal(\theta_g, \sigma^2_g)$, whose probability density function is equal to 0 when $\mu_{cg} = 0$. Therefore, when $Y_{cg} = 0, \mu_{cg} = 0$,

$$
\begin{aligned}
&\Pr\left[Y_{cg}, \mu_{cg}\right] \\
=&\Pr\left[Y_{cg}=0|\mu_{cg}=0, Z_{cg}=0, Z^B_{cg}=0\right] \\
&\Pr\left[Z_{cg}=0|\mu_{cg}=0, Z^B_{cg}=0\right]\Pr\left[\mu_{cg}=0|Z^B_{cg}=0\right]\Pr\left[Z^B_{cg}=0\right] \\
=&1 \times 1 \times 1 \times (1-p^B_g) \\
=&1-p^B_g
\end{aligned} \tag{6.6}
$$

When $Y_{cg} = 0, \mu_{cg} > 0$, the first component in Equation 6.3 is equal to 0, therefore, in this case,

$$\begin{aligned}
&\Pr\left[Y_{cg}, \mu_{cg}\right] \\
=&(1-\pi_{cg}) f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B + e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}} \pi_{cg} f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B \\
=&\left[(1-\pi_{cg}) + e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}} \pi_{cg}\right] f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B
\end{aligned} \tag{6.7}$$

Therefore, the marginal distribution of (Y_{cg}, μ_{cg}) can be written as,

$$\begin{aligned}
&\Pr\left[Y_{cg}, \mu_{cg}\right] \\
=&\begin{cases}
\frac{\left[e^{\alpha_c+\beta_c \log \mu_{cg}}\right]^{Y_{cg}} e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}}}{y_{cg}!} \pi_{cg} f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B, & \text{if } Y_{cg} > 0, \mu_{cg} > 0 \\
0, & \text{if } Y_{cg} > 0, \mu_{cg} = 0 \\
\left[(1-\pi_{cg}) + e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}} \pi_{cg}\right] f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B, & \text{if } Y_{cg} = 0, \mu_{cg} > 0 \\
1 - p_g^B, & \text{if } Y_{cg} = 0, \mu_{cg} = 0
\end{cases}
\end{aligned} \tag{6.8}$$

It's straightforward to compute the marginal distribution of Y_{cg} from this joint distribution in Equation 6.8.

$$\Pr\left[Y_{cg}\right] = \begin{cases}
p_g^B \int \left[\frac{\left[e^{\alpha_c+\beta_c \log \mu_{cg}}\right]^{Y_{cg}} e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}}}{y_{cg}!} \pi_{cg} f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B\right] d\mu_{cg}, & \text{if } Y_{cg} > 0 \\
1 - p_g^B + p_g^B \int \left[\left[(1-\pi_{cg}) + e^{-e^{\alpha_c+\beta_c \log \mu_{cg}}} \pi_{cg}\right] f_{LN}\left(\mu_{cg}|\theta_g, \sigma_g^2\right) p_g^B\right] d\mu_{cg}, & \text{if } Y_{cg} = 0
\end{cases} \tag{6.9}$$

6.2.2. Technical Parameters from ERCC

ERCC spike-ins are added after cell lysis, therefore, they do not exhibit the expression heterogeneity of a biological gene. The model for ERCC spike-ins is unchanged in this extension, and therefore, all technical parameters can still be estimated using methods developed for TASC, as described in subsection 5.2.1.

6.2.3. Testing for Presence of Transcriptional Bursting

One advantage of a likelihood model is the theoretic and applied simplicity in testing for the significance of certain parameters using likelihood ratio tests. The incorporation of the parameter p_g^B allows us to directly test whether there is significant transcriptional bursting in a specific gene, with the following test (Test #1).

$$\begin{cases} H_0: & p_g^B = 1 \\ H_1: & p_g^B < 1 \end{cases} \tag{6.10}$$

And the naïve likelihood ratio test is to optimize the full model under the null hypothesis (*i.e.* the TASC model, $\hat{\mathcal{L}}_0$) and the alternative hypothesis ($0 < p_g^B < 1$, $\hat{\mathcal{L}}_1$). The likelihood ratio test can then be computed as

$$\hat{T} = -2\left[\log(\hat{\mathcal{L}}_1) - \log(\hat{\mathcal{L}}_0)\right] \tag{6.11}$$

Comparing $\hat{T}$ to a χ^2-distribution, with 1 degree of freedom, gives us the raw p-value for this test.

$$p = Pr\left[\chi_1^2 < \hat{T}\right] \tag{6.12}$$

Notice that the asymptotic distribution of $\hat{T}$ might not follow a χ^2-distribution, as p_g^G in the null hypothesis rests on the boundary of the parameter space. Volumes have been devoted to this specific topic in the statistical literature (Bartholomew, 1961; Kudo, 1963; Self and Liang, 1987). In our experience, the naïve implementation in this specific case does not severely affect its performance. However, further work needs to be done, incorporating methods such as Bartholomew's $\bar{\chi}^2$-tests (Kudo, 1963), or the modified χ^2-test from Edward Susko (Susko, 2013).

6.2.4. Testing for Differential Levels of Transcriptional Bursting

While it is important to interrogate the presence of transcriptional bursting in a homogeneous population of cells, most of the scRNA-seq experiments actually contain multiple groups or biological conditions. Significant insight on the regulatory mechanisms of transcription can be provided by testing the different levels of bursting between two groups, *i.e.*, testing the following hypothesis

(Test #2):

$$\begin{cases} H_0: & p_{g1}^B = p_{g2}^B = p_g^B \\ H_1: & p_{g1}^B \neq p_{g2}^B \end{cases} \tag{6.13}$$

A simple likelihood ratio test with the marginal likelihood optimized with one common p_g^B for the two conditions ($\hat{\mathcal{L}}_0$), or two distinctive probability parameters (p_{g1}^B and p_{g2}^B, $\hat{\mathcal{L}}_1$). The LRT statistic,

$$\hat{T} = -2\left[\log(\hat{\mathcal{L}}_1) - \log(\hat{\mathcal{L}}_0)\right] \tag{6.14}$$

Comparing $\hat{T}$ to a χ^2-distribution, with 1 degree of freedom, gives us the raw p-value for this test.

$$p = Pr\left[\chi_1^2 < \hat{T}\right] \tag{6.15}$$

6.2.5. Testing of Differential Expression With Adjustment for Transcriptional Bursting

With methods that simply model the true expression of the genes using Poisson distribution, testing for differential expression (DE) can be confounded by differential bursting (DB). In our method, it is possible to disentangle the former from the latter, testing for the differential levels of expression when the genes are in the "on" state only. A simple LRT can be derived to test for the following hypothesis (Test #3):

$$\begin{cases} H_0: & \theta_{g1} = \theta_{g2} = \theta_g \\ H_1: & \theta_{g1} \neq \theta_{g2} \end{cases} \tag{6.16}$$

The LRT statistic can be computed by optimizing the marginal likelihood with one common θ_g for the two conditions ($\hat{\mathcal{L}}_0$), or two distinctive probability parameters (θ_{g1} and θ_{g2}, $\hat{\mathcal{L}}_1$). The LRT statistic,

$$\hat{T} = -2\left[\log(\hat{\mathcal{L}}_1) - \log(\hat{\mathcal{L}}_0)\right] \tag{6.17}$$

Comparing $\hat{T}$ to a χ^2-distribution, with 1 degree of freedom, gives us the raw p-value for this test.

$$p = Pr\left[\chi_1^2 < \hat{T}\right] \tag{6.18}$$

6.2.6. Simultaneous Testing for Differential Levels of Expression and Bursting

In scenarios described in both subsection 6.2.4 and subsection 6.2.5, when testing for one parameter in p_g^B and θ_g, the other is allowed full degrees of freedom. As a great screening measure, one may want to test for the change in changes in either p_g^B or θ_g (Test #4). This can be achieved by testing:

$$\begin{cases} H_0: & \theta_{g1} = \theta_{g2} = \theta_g \text{ and } p_{g1}^B = p_{g2}^B = p_g^B \\ H_1: & \theta_{g1} \neq \theta_{g2} \text{ or } p_{g1}^B \neq p_{g2}^B \end{cases} \tag{6.19}$$

The LRT statistic can be computed by optimizing the marginal likelihood with one common θ_g and p_g^B for the two conditions ($\hat{\mathcal{L}}_0$), or two distinctive probability parameters (θ_{g1} and θ_{g2}, p_{g1}^B and p_{g2}^B, $\hat{\mathcal{L}}_1$). The LRT statistic,

$$\hat{T} = -2\left[\log(\hat{\mathcal{L}}_1) - \log(\hat{\mathcal{L}}_0)\right] \tag{6.20}$$

Comparing $\hat{T}$ to a χ^2-distribution, with 2 degree of freedom, gives us the raw p-value for this test.

$$p = Pr\left[\chi_2^2 < \hat{T}\right] \tag{6.21}$$

6.3. Evaluation of Performance and Comparison with Other Methods

6.3.1. Validation of Algorithms

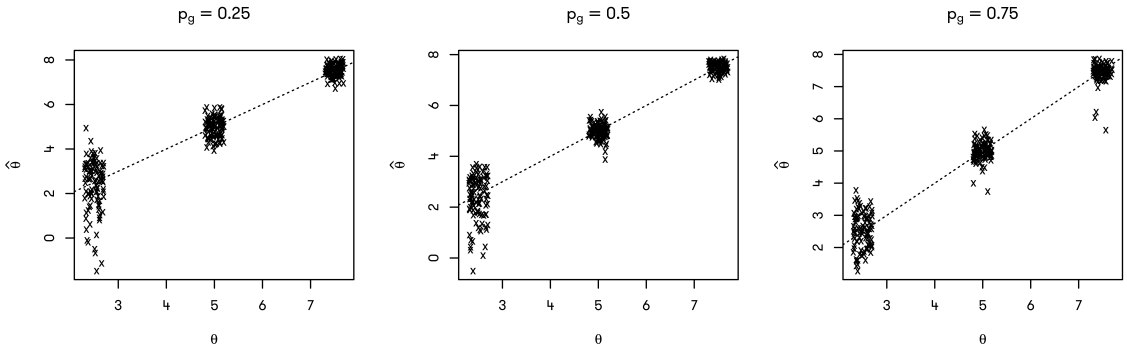
We have implemented the above algorithm using Cython and openMPI in Python (details in section 6.5). To validate our implementation, we have performed a series of simulation studies, where a range of values in p_g^B and θ_g are used to generate the read counts. The technical parameters are

fixed at,

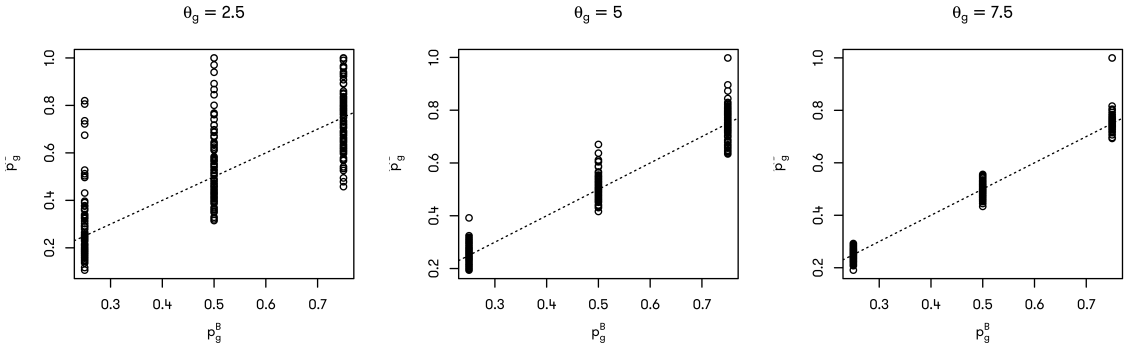
$$\begin{cases} \alpha = & 0.635308628209573 \\ \beta = & 1.01020809346668 \\ \kappa = & -4.39815121122278 \\ \tau = & 1.2364362618758 \\ \sigma = & 2.5 \end{cases} \tag{6.22}$$

All combinations of three different p_g^B $(0.25, 0.5, 0.75)$ and three different θ_g $(2.5, 5, 7.5)$ are used to generate the counts from the TASC-B model. 100 simulations are done for each combination of (θ_g, p_g^B). Each simulation contains the read counts of 600 cells. Two algorithms, original TASC, and TASC-B are used to fit the simulated data, in order to compare the performance of TASC-B against the more simplified TASC model. The bias and spread of the estimates are satisfactory considering the moderate sample size and the difficulties in accurately extrapolating the zero proportions from the non-zero counts, with details shown in Table 6.1. Some scenarios have proved to be more difficult than others for our algorithm. For example, when $p_g^B = 0.25$ and $\theta_g = 2.5$, *i.e.*, the gene is only turned on in a quarter of the cells, and the expression is relatively low, our algorithm displays the most severe bias, and highest estimation error, compared to other scenarios. We suspect it is due to the limited information there is to estimate the positive mean, as presumably only a handful of cells contain non-zero counts for these genes. With a poorly estimated θ_g, extrapolating p_g^B would be much more difficult, as is shown in the high variability in the estimate for p_g^B in Table 6.1. Interestingly, even when p_g^B is sufficiently large, if the mean expression of the gene is low enough ($\theta_g = 2.5$), the estimation error is still significant for both parameters. The spread of $\hat{p}_g^B$ is considerably tighter when $\theta_g = 5$ and $\theta_g = 7.5$ compared to $\theta_g = 2.5$ 6.2b. The $\hat{\theta}_g$ is estimated much more accurately as θ_g increases 6.2a.

Excluding p_g^B, as in TASC, estimates of θ_g can be confounded by p_g^B, as shown in Figure 6.3. TASC estimates of $\hat{\theta}_g$ decrease as p_g^B decreases, regardless of the true value of θ_g, while estimates of $\hat{\theta}_g$ by TASC-B are not confounded by p_g^B, centered around the true mean of θ_g in all simulated scenarios. Similar confounding is also observed for $\hat{\sigma}_g$ (Figure 6.4). TASC estimates of $\hat{\sigma}_g$ increases as p_g^B decreases, for it attributes the additional zeros produced by transcriptional bursting to σ_g. Under



(a) Scatter plots shown the estimated values of θ_g against its true values grouped by the true p_g^B



(b) Scatter plots shown the estimated values of p_g^B against its true values grouped by the true θ_g

Figure 6.2: Scatter plots illustrating the relationship between the estimated values and the true values of θ_g and p_g^B. The dotted line is the unit line with slope equal 1, and intercept equal to 0.

θ_g	p_g^B	$\overline{\hat{\theta}_g} - \theta_g$	$SD_{\hat{\theta}_g}$	$\overline{\hat{p}_g^B} - p_g^B$	$SD_{\hat{p}_g^B}$	$\overline{\hat{\sigma}_g} - \sigma$	$SD_{\hat{\sigma}_g}$
2.5	0.25	-1.315E-01	1.190E+00	2.993E-02	1.390E-01	-4.985E-03	5.047E-01
2.5	0.5	-1.553E-01	8.852E-01	4.593E-02	1.689E-01	6.097E-02	3.673E-01
2.5	0.75	4.424E-02	5.362E-01	-1.964E-03	1.374E-01	-3.404E-02	2.442E-01
5	0.25	2.025E-03	4.376E-01	7.418E-03	3.264E-02	-3.769E-02	2.601E-01
5	0.5	-2.528E-02	2.930E-01	9.754E-03	4.408E-02	1.016E-03	1.742E-01
5	0.75	-1.505E-02	2.881E-01	1.776E-03	6.326E-02	-5.249E-03	1.879E-01
7.5	0.25	2.906E-02	2.724E-01	2.572E-04	1.936E-02	-1.316E-02	1.776E-01
7.5	0.5	1.705E-02	1.789E-01	-7.927E-04	2.515E-02	-1.196E-02	1.294E-01
7.5	0.75	-6.214E-02	3.086E-01	8.398E-03	4.846E-02	2.750E-02	2.164E-01

Table 6.1: Bias and estimation error of $\hat{\theta}_g$, $\hat{p}_g^B$ and $\hat{\sigma}_g$ using TASC-B from all simulated scenarios. For each parameter, the bias is estimated by subtracting the true value of the parameter from the mean of all estimates from 100 simulations, and the estimation error is the standard deviation of the 100 estimates.

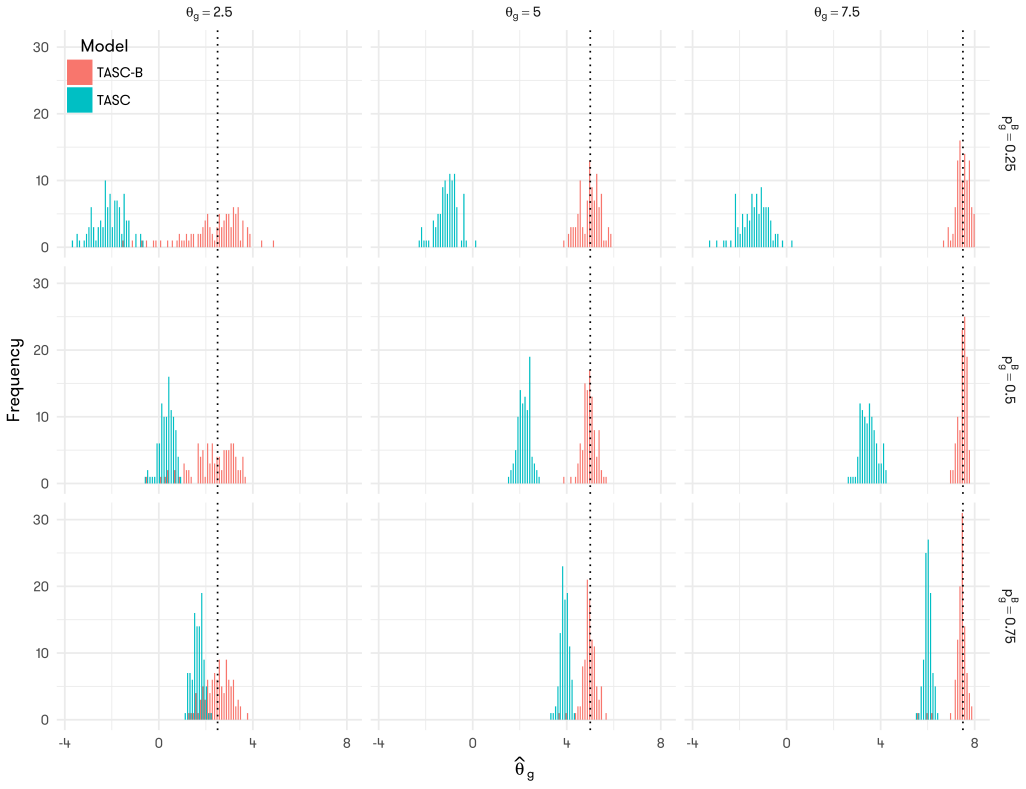


Figure 6.3: Histograms of estimated $\hat{\theta}_g$ using TASC and TASC-B. Different rows represent simulations with various p_g^G, and different columns represent simulations with various θ_g. In each panel, two histograms are plotted with distinct colors, representing the distribution of estimated $\hat{\theta}_g$ from TASC (blue) and TASC-B (red). The vertical dotted lines indicate the true values of θ_g.

the majority of the simulated scenarios, especially when p_g^B is relatively small, estimates of σ_g from TASC does not even overlap with the true value. The value of θ_g also confounds the estimation of σ_g for TASC. Larger θ_g increases the estimates of σ_g. In all simulated scenarios, estimates of $\hat{\sigma}_g$ by TASC-B are not confounded by p_g^B, as it correctly attributes the excessive zeros to bursting.

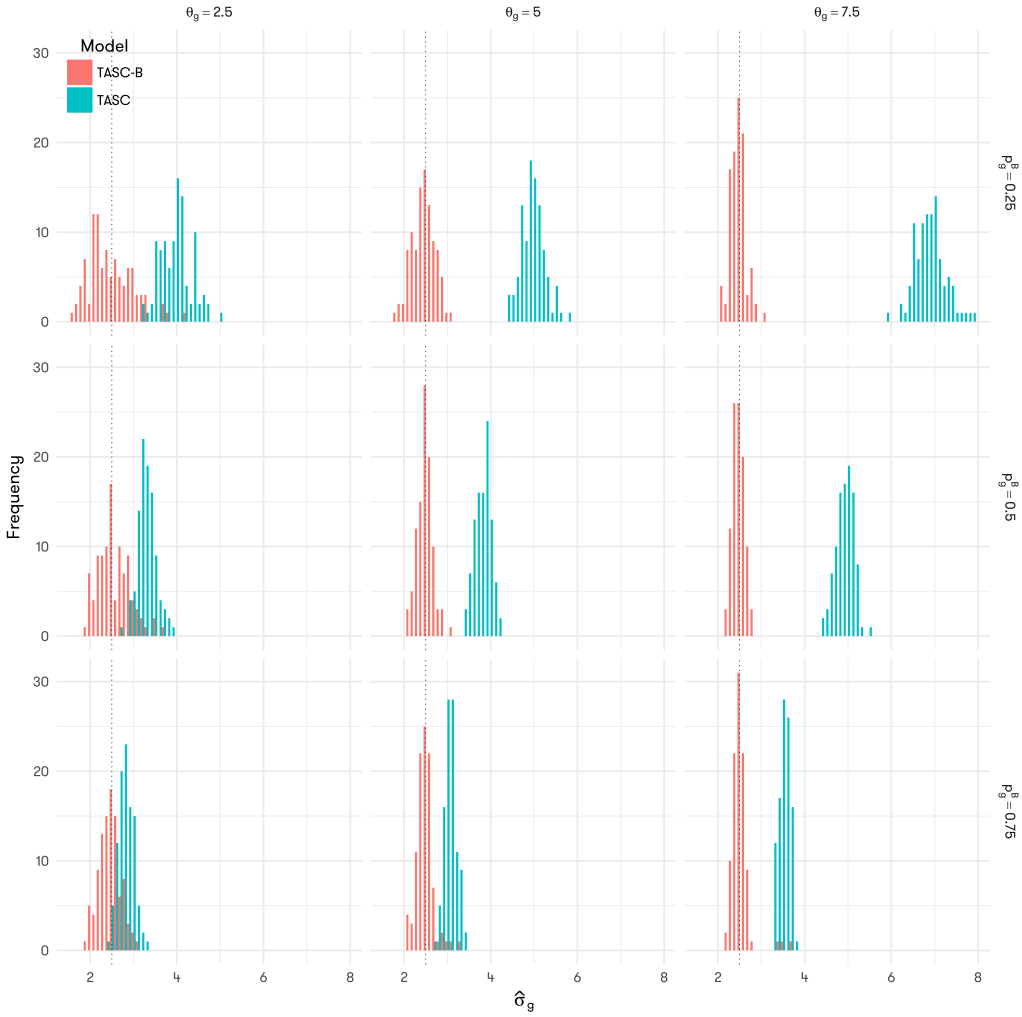


Figure 6.4: Histograms of estimated $\hat{\sigma}_g$ using TASC and TASC-B. Different rows represent simulations with various p_g^G, and different columns represent simulations with various σ_g. In each panel, two histograms are plotted with distinct colors, representing the distribution of estimated $\hat{\sigma}_g$ from TASC (blue) and TASC-B (red). The vertical dotted lines indicate the true values of σ_g.

6.3.2. Performance Under the Null Hypothesis

In order to investigate the propensity for false positives of our method, we have simulated 16 combinatory scenarios with $\theta_g \in \{2, 3, 4, 5\}$, and $p_g^B \in \{0.2, 0.4, 0.6, 0.8\}$ from our generative model, using

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.1022	0.0484	0.009	0.0044	0.001
0.2	3	0.0944	0.0456	0.0064	0.003	0.001
0.2	4	0.103	0.051	0.0094	0.0052	0.0012
0.2	5	0.0992	0.0486	0.0084	0.0036	0.0016
0.4	2	0.0978	0.0508	0.0108	0.0046	0.0014
0.4	3	0.101	0.0502	0.0092	0.0044	0.0002
0.4	4	0.0938	0.0486	0.0084	0.0036	0.001
0.4	5	0.0924	0.0458	0.008	0.0048	0.0006
0.6	2	0.095	0.049	0.0094	0.0042	0.0008
0.6	3	0.0936	0.0476	0.0092	0.0042	0.0006
0.6	4	0.1014	0.0522	0.0086	0.0026	0.0002
0.6	5	0.1	0.0534	0.0112	0.0058	0.0012
0.8	2	0.08	0.039	0.0096	0.0054	0.0012
0.8	3	0.09	0.0432	0.008	0.004	0.0002
0.8	4	0.0968	0.0444	0.0096	0.0048	0.0014
0.8	5	0.1052	0.0556	0.0122	0.0068	0.003

Table 6.2: Estimated false positive rates from the null simulation with TASC-B Test #2. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

technical parameters listed in Equation 6.22. Each scenario contains the counts of 5000 artificial genes in 600 cells (300 cells in each condition). The parameters are identical across biological conditions, therefore, for a method with well-controlled type I error, the p-values should be distributed uniformly on $(0, 1)$. Test #2 (subsection 6.2.4) is implemented with Python and Cython. Raw p-values are extracted from the 5000 genes for each scenario and $-\log_{10}(p)$ is compared with the expected percentile drawn from a uniform distribution. From Figure 6.5, under all scenarios simulated, TASC-B has well-controlled type I error, with the distribution of the raw p-values closely resembling a uniformly distributed random variable. To quantitatively illustrate the behavior of TASC-B under the null conditions, we estimated the type I error rate by computing the fraction of genes that are smaller than a specified significance level ($\alpha \in (0.1, 0.05, 0.01, 0.005, 0.001)$) among all 5000 genes in a given scenario. The numbers are summarized in Table 6.2. Under all the scenarios simulated, estimated type I error from TASC-B dovetails perfectly with the nominal significance level, indicating well-controlled false positive rates under all scenarios. While Figure 6.5 and Table 6.2 focus on the lower range of the p-values, Figure 6.6 shows that p-values from TASC-B are uniformly distributed in the mid and upper regions as well. Overall, TASC-B Test #2 has well-controlled type I error rate, and is well-behaved under the null.

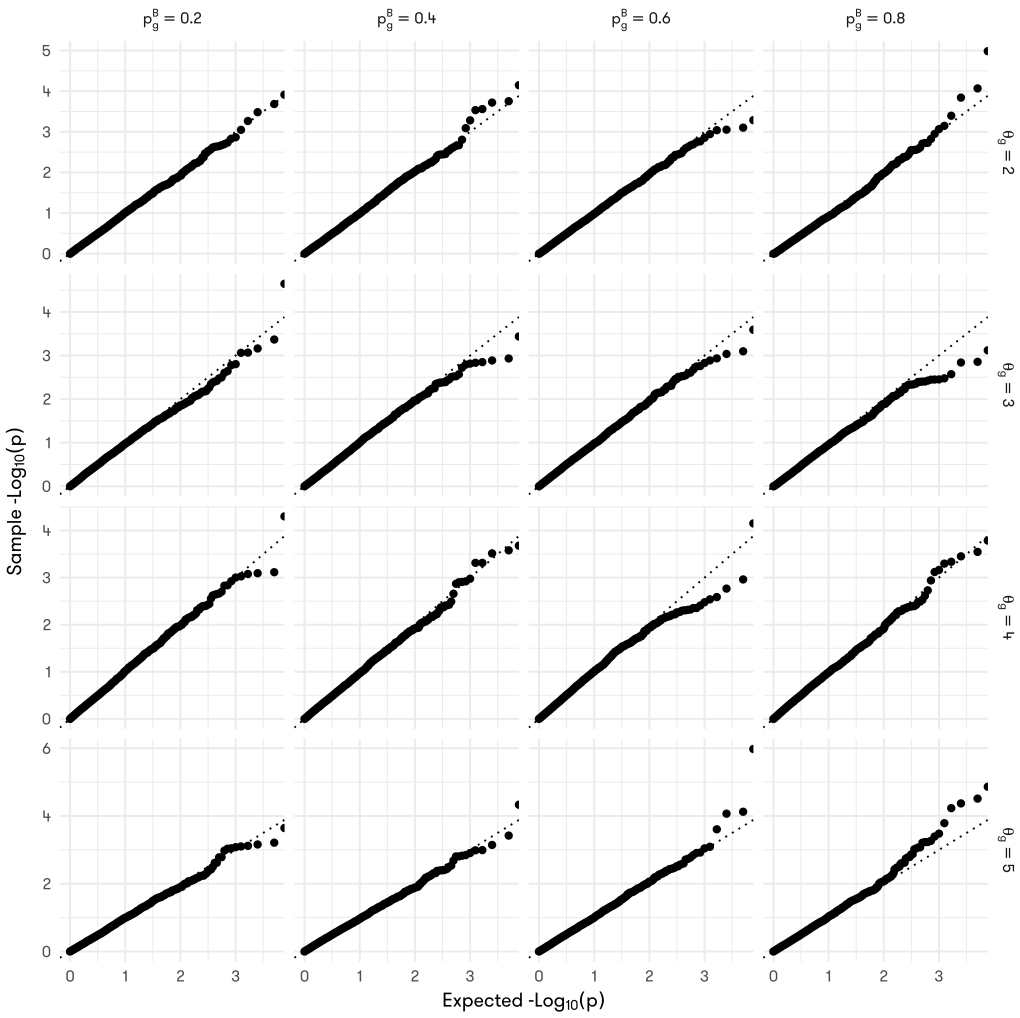


Figure 6.5: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from Test #2 (subsection 6.2.4) performed with TASC-B model, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

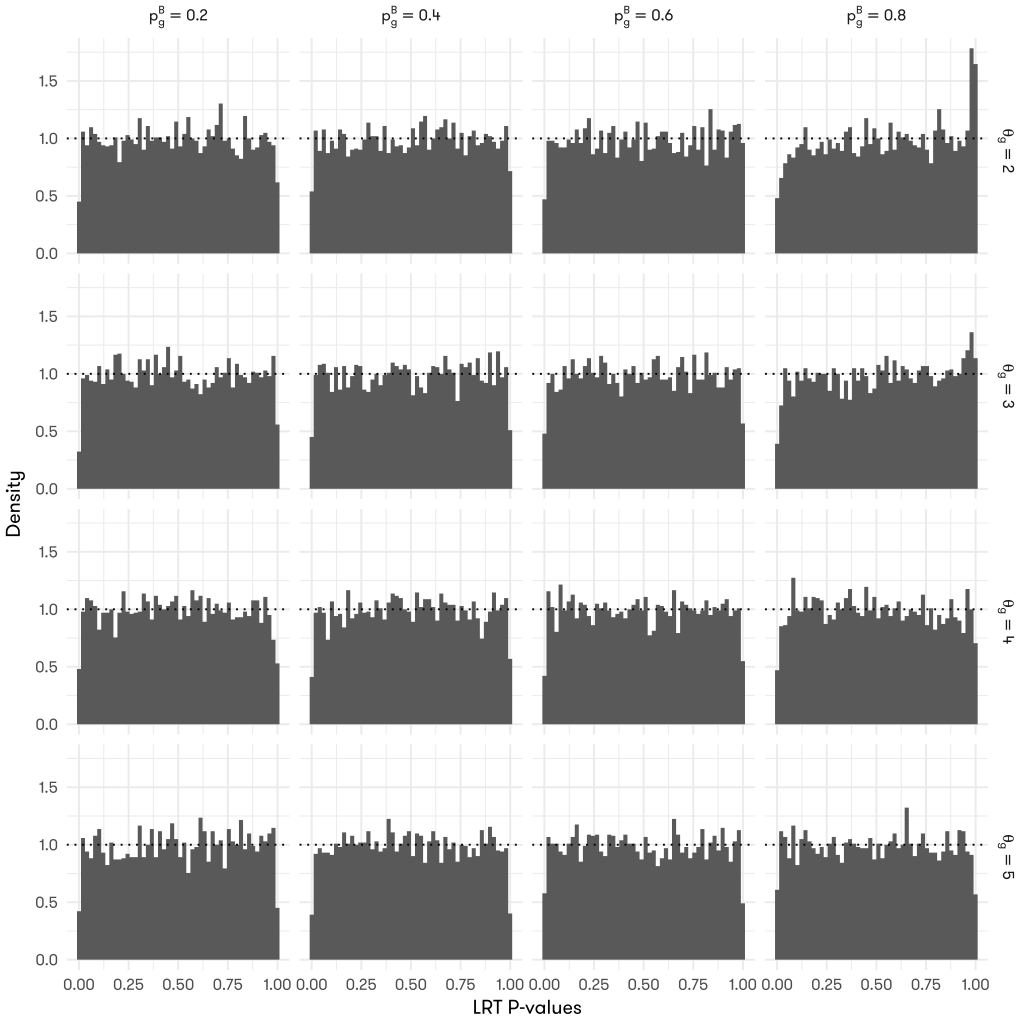


Figure 6.6: Histograms of raw p-values extracted from TASC-B Test #2 results (subsection 6.2.4). With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.1098	0.058	0.0126	0.0058	0.001
0.2	3	0.1068	0.0496	0.012	0.007	0.0008
0.2	4	0.1088	0.0558	0.0114	0.0056	0.0008
0.2	5	0.1044	0.056	0.01	0.0052	0.0006
0.4	2	0.1054	0.0562	0.0086	0.0052	0.0018
0.4	3	0.103	0.0514	0.0098	0.0052	0.0006
0.4	4	0.1082	0.0558	0.0122	0.0062	0.001
0.4	5	0.1038	0.0502	0.008	0.0044	0.0004
0.6	2	0.0982	0.0522	0.0098	0.005	0.0014
0.6	3	0.1012	0.0502	0.0088	0.0034	0.0004
0.6	4	0.0958	0.048	0.0098	0.005	0.0012
0.6	5	0.1018	0.0508	0.0112	0.0054	0.0008
0.8	2	0.0936	0.0462	0.008	0.0044	0.0014
0.8	3	0.0942	0.0492	0.0082	0.005	0.0008
0.8	4	0.1012	0.0532	0.0074	0.003	0.0008
0.8	5	0.1134	0.0532	0.0114	0.0054	0.0012

Table 6.3: Estimated false positive rates from the null simulation with TASC-B Test #3 subsection 6.2.5. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

Using the simulated data described at the beginning of subsection 6.3.2, we also tested the performance of TASC-B Test #3 under the null. From Figure 6.7, under all scenarios simulated, Test #3 has well-controlled type I error, with the distribution of raw p-values closely resembling a uniformly distributed random variable. To quantitatively illustrate the behavior of Test #3 under the null conditions, we estimated the type I error rate by computing the fraction of genes that are smaller than a specified significance level ($\alpha \in (0.1, 0.05, 0.01, 0.005, 0.001)$) among all 5000 genes in a given scenario. The numbers are summarized in Table 6.3. Under all the scenarios simulated, estimated type I error from TASC-B dovetails perfectly with the nominal significance level, indicating well-controlled false positive rates under all scenarios. In addition, Figure 6.8 shows that p-values from Test #3 are uniformly distributed in the mid and upper regions as well. Overall, TASC-B Test #3 has well-controlled type I error rate, and is well-behaved under the null.

The same simulated null dataset (subsection 6.3.2) is also used to benchmark Test #4 (subsection 6.2.6) under the null. From Figure 6.9, under all scenarios simulated, Test #4 has well-controlled type I error, with the distribution of raw p-values closely resembling a uniformly distributed random variable. To quantitatively illustrate the behavior of Test #3 under the null conditions, we estimated the type I error rate by computing the fraction of genes that are smaller than a specified

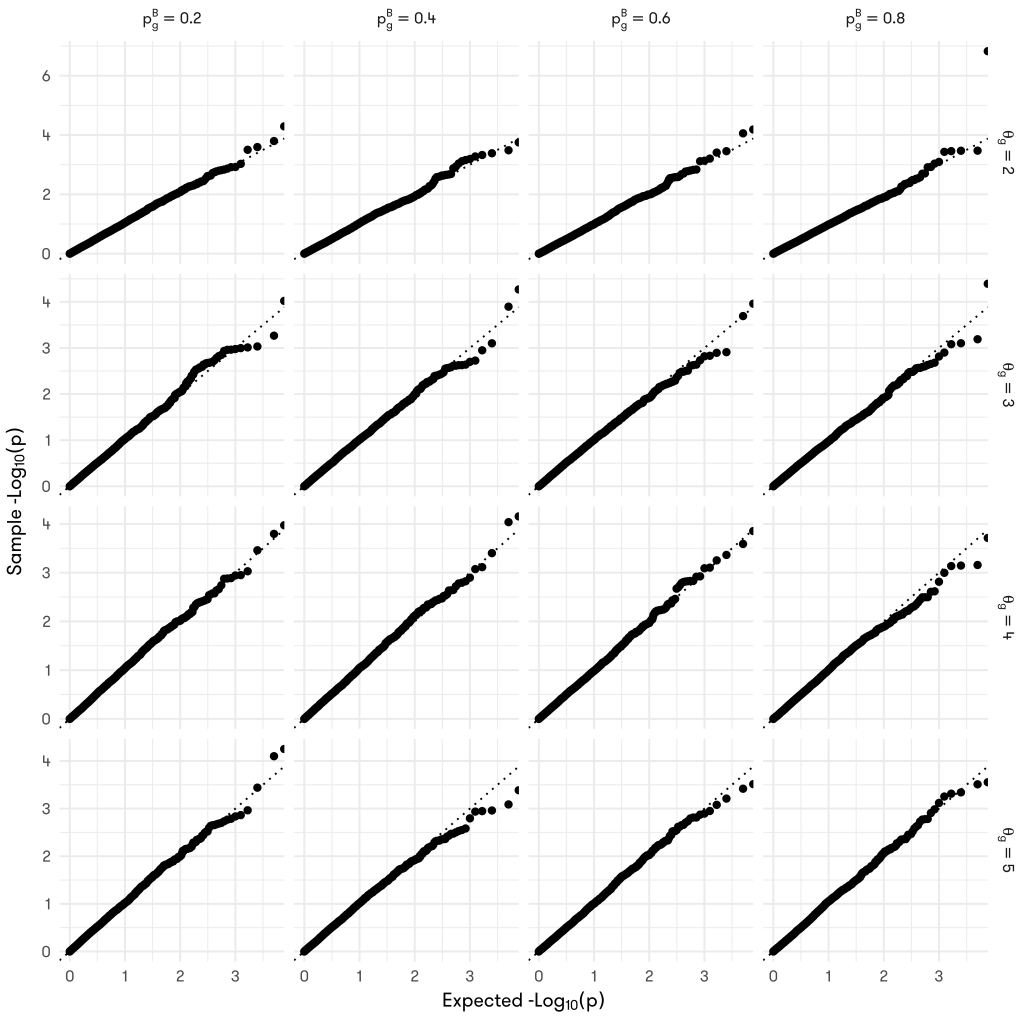


Figure 6.7: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from Test #3 (subsection 6.2.5) performed with TASC-B model, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

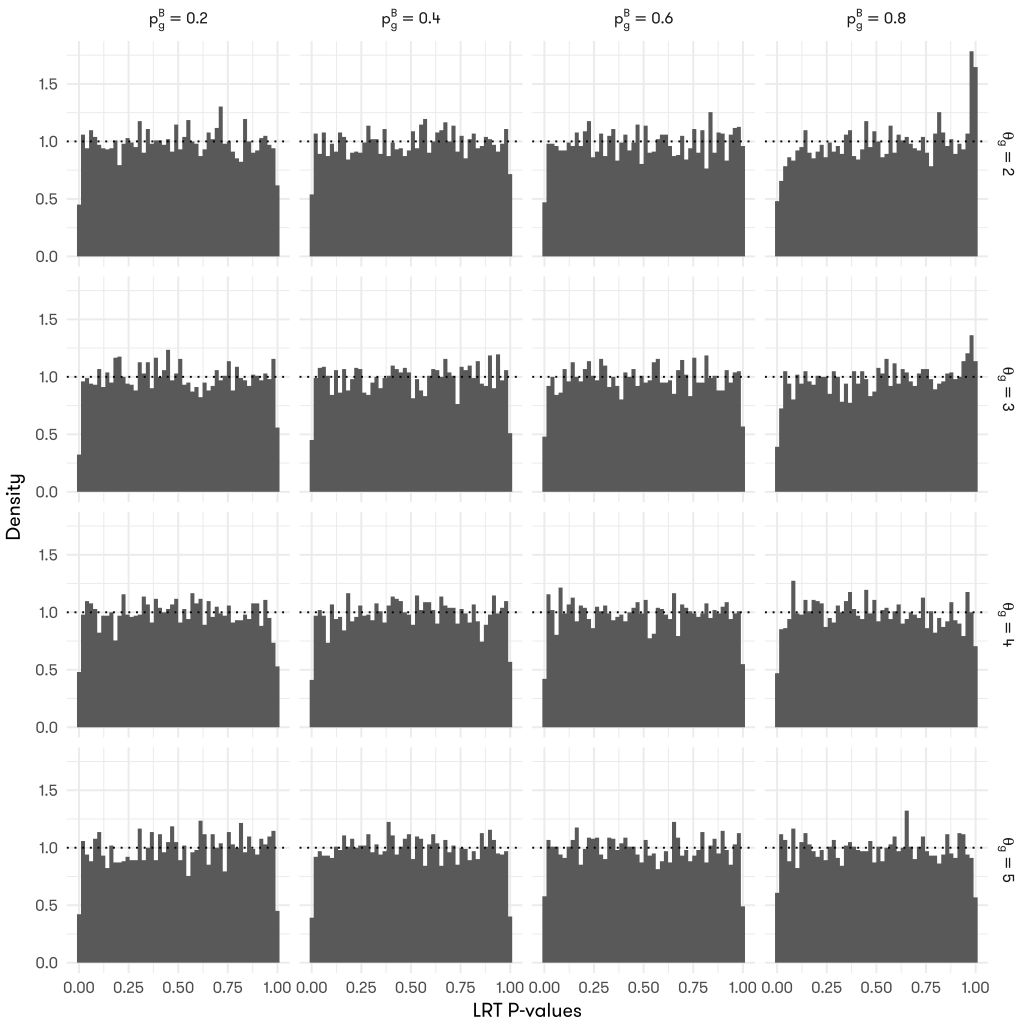


Figure 6.8: Histograms of raw p-values extracted from TASC-B Test #3 results (subsection 6.2.4). With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

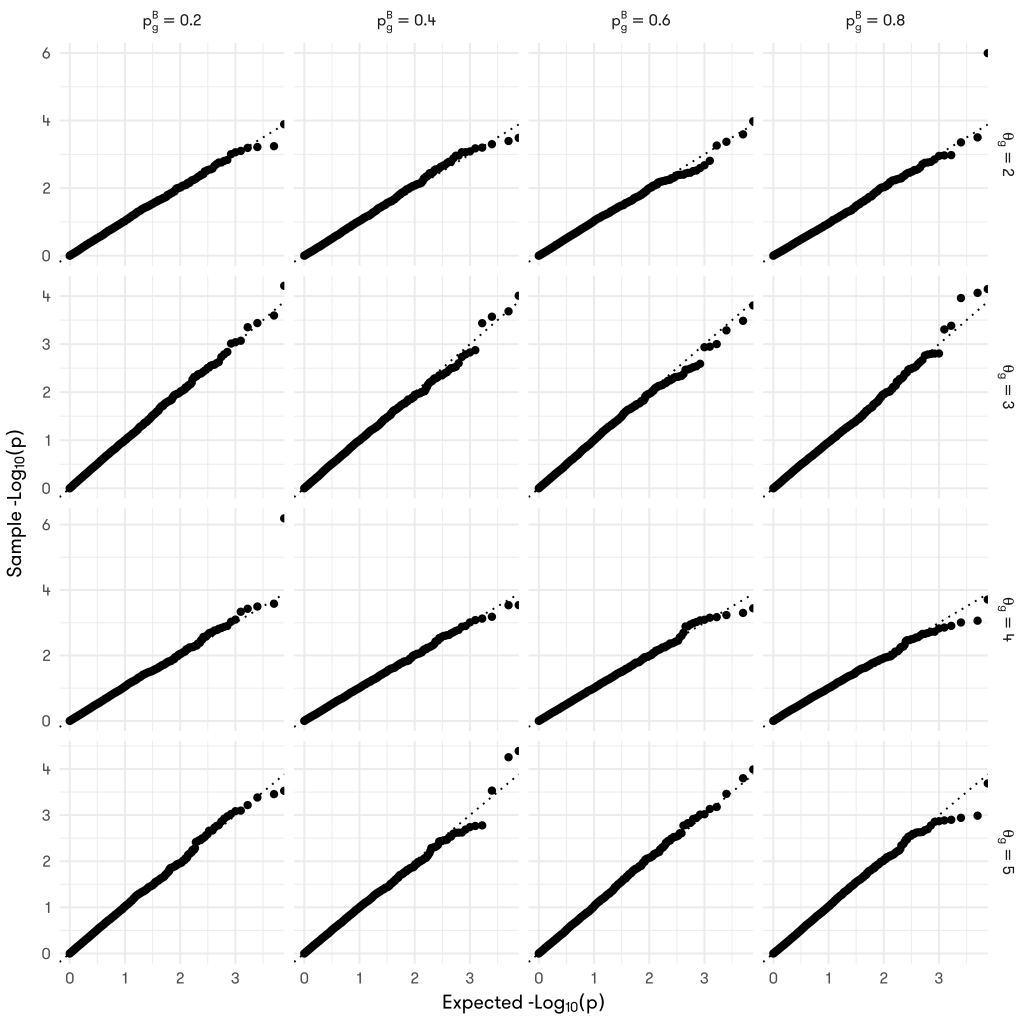


Figure 6.9: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from Test #4 (subsection 6.2.5) performed with TASC-B model, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

significance level ($\alpha \in (0.1, 0.05, 0.01, 0.005, 0.001)$) among all 5000 genes in a given scenario. The numbers are summarized in Table 6.4. Under all the scenarios simulated, estimated type I error from TASC-B dovetails perfectly with the nominal significance level, indicating well-controlled false positive rates under all scenarios. In addition, Figure 6.10 shows that p-values from Test #4 are uniformly distributed in the mid and upper regions as well. Overall, TASC-B Test #4 has well-controlled type I error rate, and is well-behaved under the null.

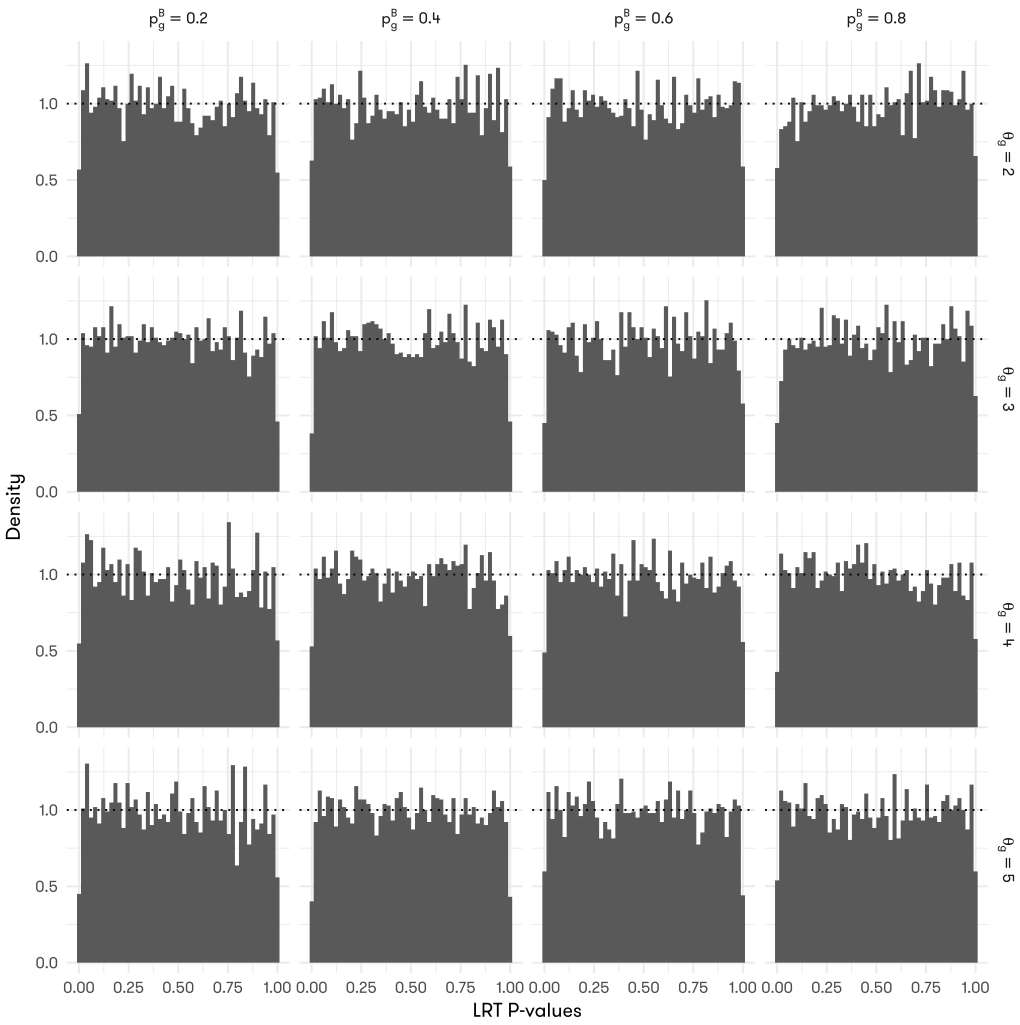


Figure 6.10: Histograms of raw p-values extracted from TASC-B Test #4 results (subsection 6.2.4). With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.1068	0.058	0.0114	0.0052	0.0014
0.2	3	0.0998	0.0498	0.0102	0.0058	0.0014
0.2	4	0.1104	0.0586	0.0112	0.0052	0.0014
0.2	5	0.1048	0.0546	0.0092	0.0054	0.0014
0.4	2	0.1078	0.0542	0.0126	0.0066	0.0016
0.4	3	0.0992	0.046	0.0078	0.004	0.0008
0.4	4	0.1022	0.051	0.0106	0.0048	0.0012
0.4	5	0.1004	0.049	0.008	0.0046	0.0006
0.6	2	0.1048	0.0506	0.01	0.0038	0.0008
0.6	3	0.1006	0.0504	0.0092	0.0036	0.0008
0.6	4	0.101	0.0508	0.0098	0.0048	0.0016
0.6	5	0.104	0.0532	0.0122	0.0056	0.0014
0.8	2	0.0904	0.045	0.0114	0.0048	0.0006
0.8	3	0.0896	0.0424	0.009	0.0046	0.001
0.8	4	0.0992	0.0498	0.0074	0.0044	0.0006
0.8	5	0.1042	0.0548	0.0106	0.005	0.0002

Table 6.4: Estimated false positive rates from the null simulation with TASC-B Test #4 subsection 6.2.6. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

6.3.3. Existing Methods Perform Unfavorably Compared to TASC-B Under the Null

SCRAN with DESeq2 As we have reviewed, DESeq2 (Love, Huber, and Anders, 2014) uses an empirical Bayes approach for estimating gene-specific variance, by fitting a curve with the naïve estimates of mean and variance of all the genes, and shrinking the variance estimates to the curve with a Bayesian prior. When used with single-cell data, this approach cannot take into consideration the cell-to-cell variations of technical noises. In order to compensate for this, SCRAN (Lun, Bach, and Marioni, 2016) is designed to normalize the read counts adjusting for specific noises that only occur in scRNA-seq. We have combined these two methods as a benchmark for popular analysis pipeline.

SCRAN with DESeq2 displays severe type I inflation under the null (Figure 6.11, Figure 6.12, Table 6.5). In Figure 6.11, SCRAN with DESeq2 produces smaller p-values than expected from a uniform distribution under the null. The severity of deviation increases as θ_g or p_g^B increases, with significant inflation in the scenario with $p_g^B = 0.8$ and $\theta_g = 5$. Similar trend can be spotted in the table of estimated false positive rates (Table 6.5) as well. With $p_g^B = 0.2$ and $\theta_g = 2$, estimated false positive rate is 0.07 at significance level $\alpha = 0.1$, which is conservative. With $\theta_g = 2$ and $p_g^B = 0.2$, the estimated false positive rate becomes 0.171 at $\alpha = 0.1$, and with $\theta_g = 2$ and $p_g^B = 0.8$, the

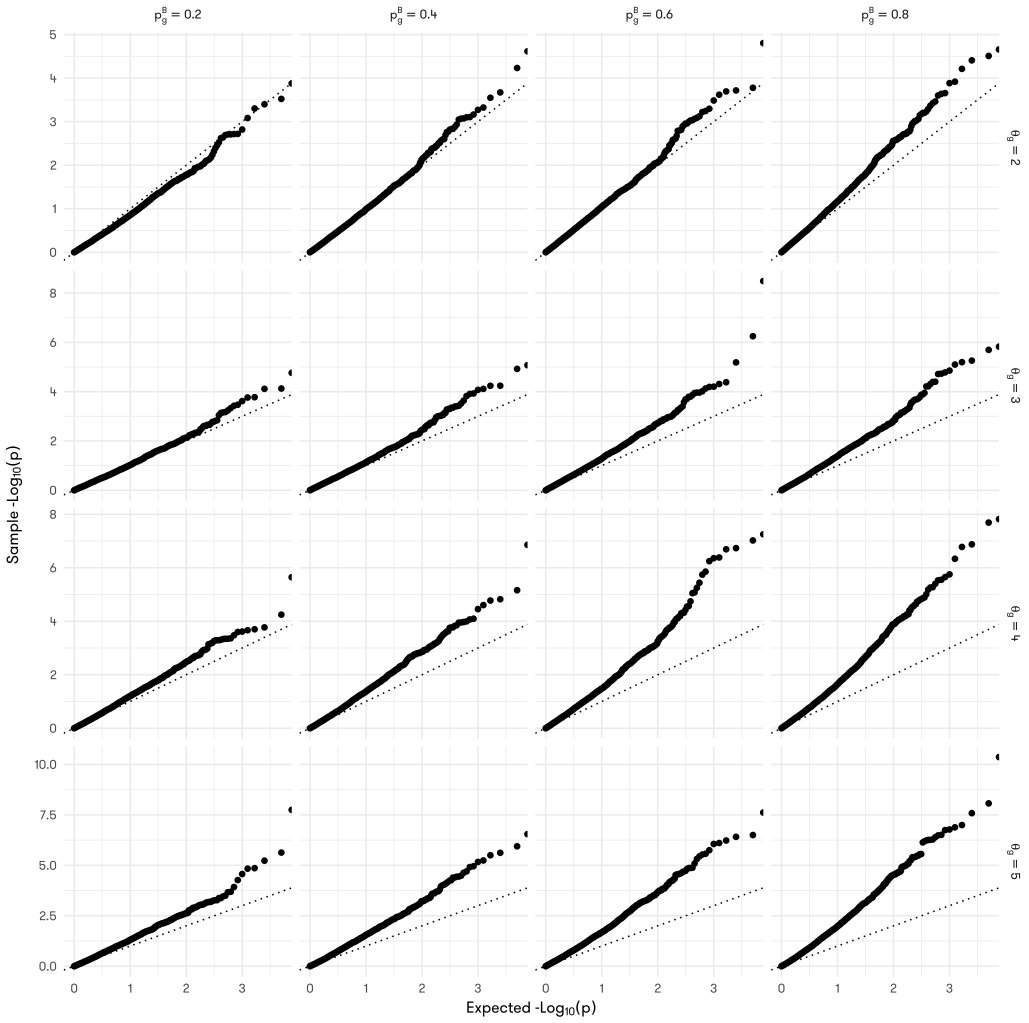


Figure 6.11: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from SCRAN coupled with DESeq2, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

estimated FPR is 0.1376, both of which are seriously anti-conservative. This trend happens in all significance levels we have tested.

Type I inflation is observed in the histograms of the raw p-values as well (Figure 6.12). As θ_g and p_g^B increase, the density concentrates on the lower bound of $(0, 1)$, indicating more p-values smaller than the expected value under the null.

MAST is a method for detecting DE genes, with considerations for inflated zeros in scRNA-seq

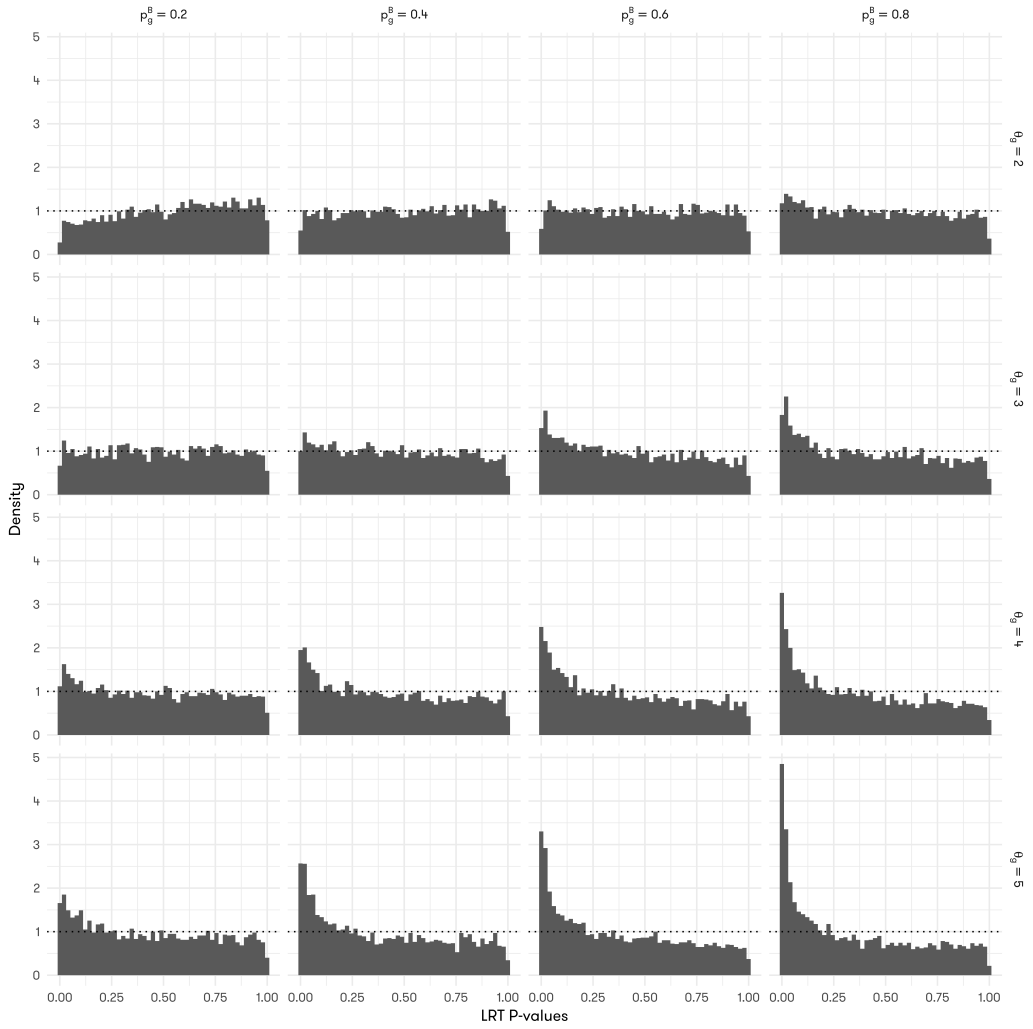


Figure 6.12: Histograms of raw p-values extracted from SCRAN coupled with DESeq2. With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.07	0.0358	0.0054	0.0034	0.001
0.2	3	0.1056	0.0572	0.0132	0.0072	0.0028
0.2	4	0.1452	0.0834	0.0218	0.0136	0.0042
0.2	5	0.171	0.1002	0.0336	0.0184	0.0062
0.4	2	0.0976	0.0496	0.0112	0.0072	0.0024
0.4	3	0.1304	0.0726	0.0204	0.0112	0.0054
0.4	4	0.1828	0.1122	0.0394	0.0254	0.0078
0.4	5	0.219	0.1408	0.0516	0.0338	0.0126
0.6	2	0.109	0.0572	0.012	0.0074	0.0028
0.6	3	0.1636	0.0974	0.031	0.0204	0.0058
0.6	4	0.207	0.1318	0.05	0.0374	0.0138
0.6	5	0.2402	0.1652	0.0668	0.049	0.0246
0.8	2	0.1376	0.0782	0.0236	0.0148	0.0044
0.8	3	0.1822	0.1146	0.0372	0.0222	0.0088
0.8	4	0.2306	0.1554	0.0656	0.0452	0.0228
0.8	5	0.2852	0.2096	0.098	0.0732	0.0378

Table 6.5: Estimated false positive rates from the null simulation with SCRAN coupled with DE-Seq2. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

data. It tests for the differences in proportions of zero between different groups using Bayesian regularized logistic regression (discrete), tests for difference in mean positive expression using generalized linear regression with a conditional normal distribution (continuous), and combine the χ^2-statistics from the above two tests to test for changes in either (hurdle). It does not consider the technical noises intrinsic in the dataset, but rather rely on *a priori* normalization methods like SCRAN. This not only renders MAST susceptible for mistakes from normalization, but also limits the power of this method due to this two-step approach. This is suggested in the null simulation where all three tests of MAST is significantly conservative compared to TASC-B (Figure 6.13, Figure 6.15, Figure 6.17,Figure 6.14,Figure 6.16,Figure 6.18).

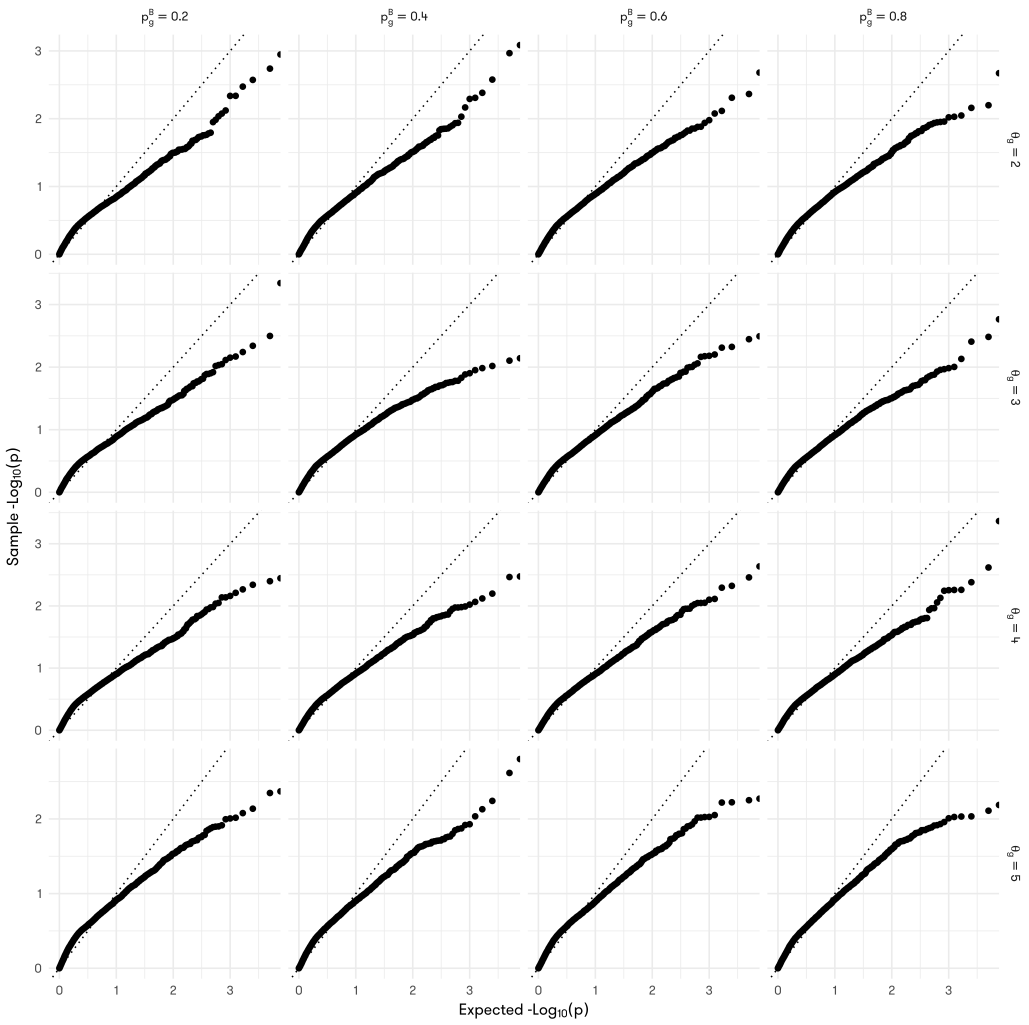


Figure 6.13: Q-Q plots of $-\log_{10}(\mathrm{p})$ comparing the p-values extracted from MAST Continuous Test, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and $\mathrm{p}_g^{\mathrm{B}}$ used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns $\mathrm{p}_g^{\mathrm{B}}$.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.0568	0.0208	0.0018	0.0012	0
0.2	3	0.0684	0.0198	0.002	0.0006	0.0002
0.2	4	0.0718	0.0212	0.002	0.0006	0
0.2	5	0.0728	0.0228	0.0012	0.0004	0
0.4	2	0.0728	0.0222	0.0016	0.001	0.0002
0.4	3	0.0726	0.0248	0.0006	0	0
0.4	4	0.0718	0.0256	0.0012	0.0004	0
0.4	5	0.0688	0.0244	0.001	0.0004	0
0.6	2	0.0668	0.0224	0.001	0.0006	0
0.6	3	0.0756	0.0256	0.0022	0.0008	0
0.6	4	0.0722	0.0252	0.0022	0.0006	0
0.6	5	0.0708	0.0238	0.0018	0	0
0.8	2	0.0716	0.0214	0.0012	0.0002	0
0.8	3	0.0742	0.0276	0.001	0.0006	0
0.8	4	0.0708	0.0236	0.0018	0.0006	0.0002
0.8	5	0.0766	0.0282	0.0012	0	0

Table 6.6: Estimated false positive rates from the null simulation with MAST Continuous Test. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

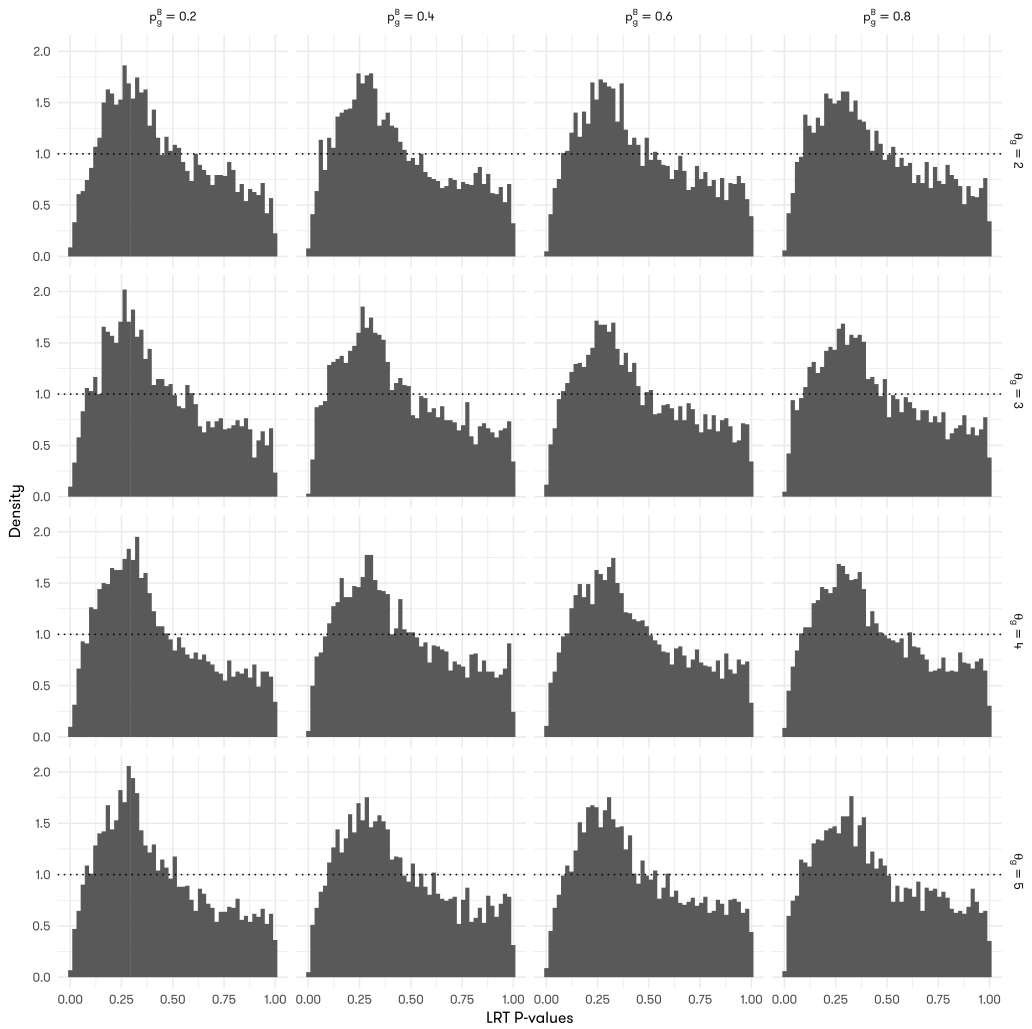


Figure 6.14: Histograms of raw p-values extracted from MAST Continuous Test. With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

The continuous test of MAST consistently over-estimates the p-values under the null in our simulations (Figure 6.13), with p-value density more concentrated around 0.25 (Figure 6.14). This happens consistently across all simulated scenarios, suggesting a systemic under-reporting of significant. With $\alpha = 0.1$, in all simulated scenarios, the estimated false positive rates range from 0.0568 to 0.0756, at least a quarter lower than the significance level.

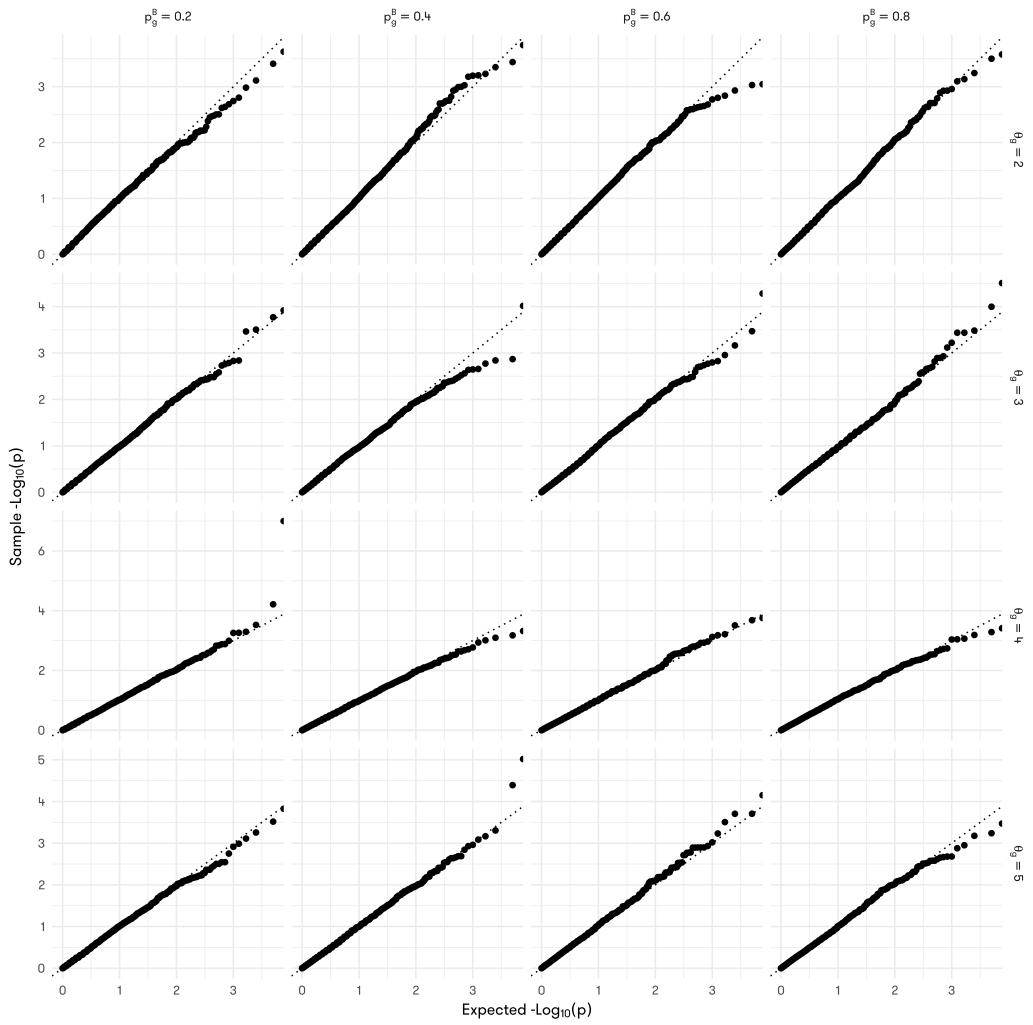


Figure 6.15: Q-Q plots of $-\log_{10}(\mathrm{p})$ comparing the p-values extracted from MAST Discrete Test, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and $\mathrm{p}_g^{\mathrm{B}}$ used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns $\mathrm{p}_g^{\mathrm{B}}$.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.0992	0.0488	0.0072	0.003	0.0006
0.2	3	0.0974	0.047	0.0108	0.0052	0.0008
0.2	4	0.1058	0.0546	0.0104	0.0058	0.0012
0.2	5	0.1014	0.0492	0.0098	0.0032	0.0008
0.4	2	0.1024	0.0546	0.0124	0.0074	0.0016
0.4	3	0.0938	0.0472	0.0084	0.0034	0.0002
0.4	4	0.0974	0.0482	0.0094	0.004	0.0008
0.4	5	0.0988	0.0506	0.0094	0.005	0.001
0.6	2	0.1042	0.0526	0.0116	0.0048	0.0004
0.6	3	0.1032	0.0524	0.0104	0.0054	0.0006
0.6	4	0.1004	0.0572	0.0104	0.0066	0.0012
0.6	5	0.1078	0.0572	0.0132	0.0058	0.0012
0.8	2	0.1018	0.0454	0.0108	0.0054	0.001
0.8	3	0.0906	0.048	0.0092	0.0048	0.0014
0.8	4	0.1126	0.0544	0.0106	0.0052	0.0012
0.8	5	0.1012	0.0516	0.011	0.005	0.0006

Table 6.7: Estimated false positive rates from the null simulation with MAST Discrete Test. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

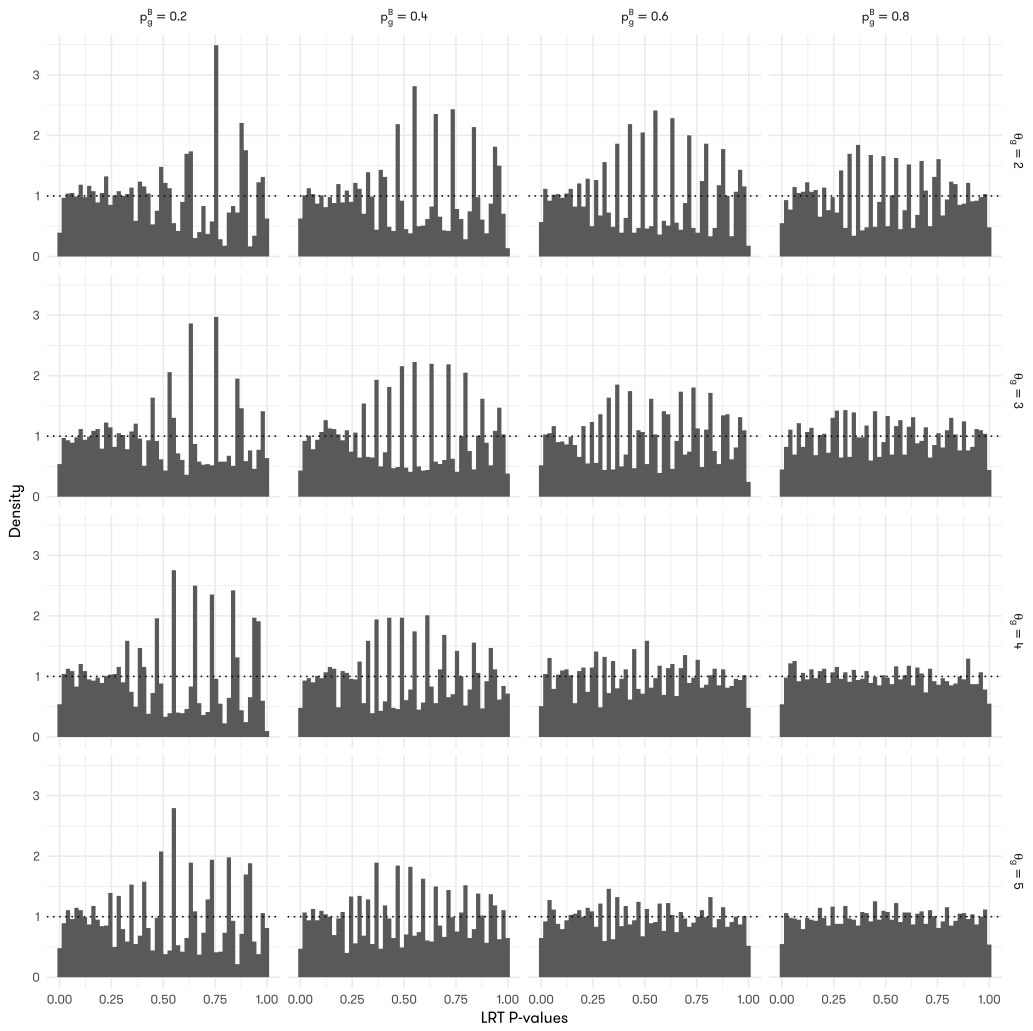


Figure 6.16: Histograms of raw p-values extracted from MAST Discrete Test. With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

This tendency to be conservative does not extend to the discrete test. P-values from this test are consistent with a uniform distribution after $-\log_{10}$ transformation (Figure 6.15). A closer look at the histograms (Figure 6.16), however, reveals that the distribution of these p-values are not completely uniform on $(0, 1)$, but rather concentrated on certain values, forming spikes on the histograms. We suspect this is due to the Bayesian shrinkage used for regularization. For the discrete test, p-values follow the nominal significance level quite closely, at least in levels we have tested (Table 6.7).

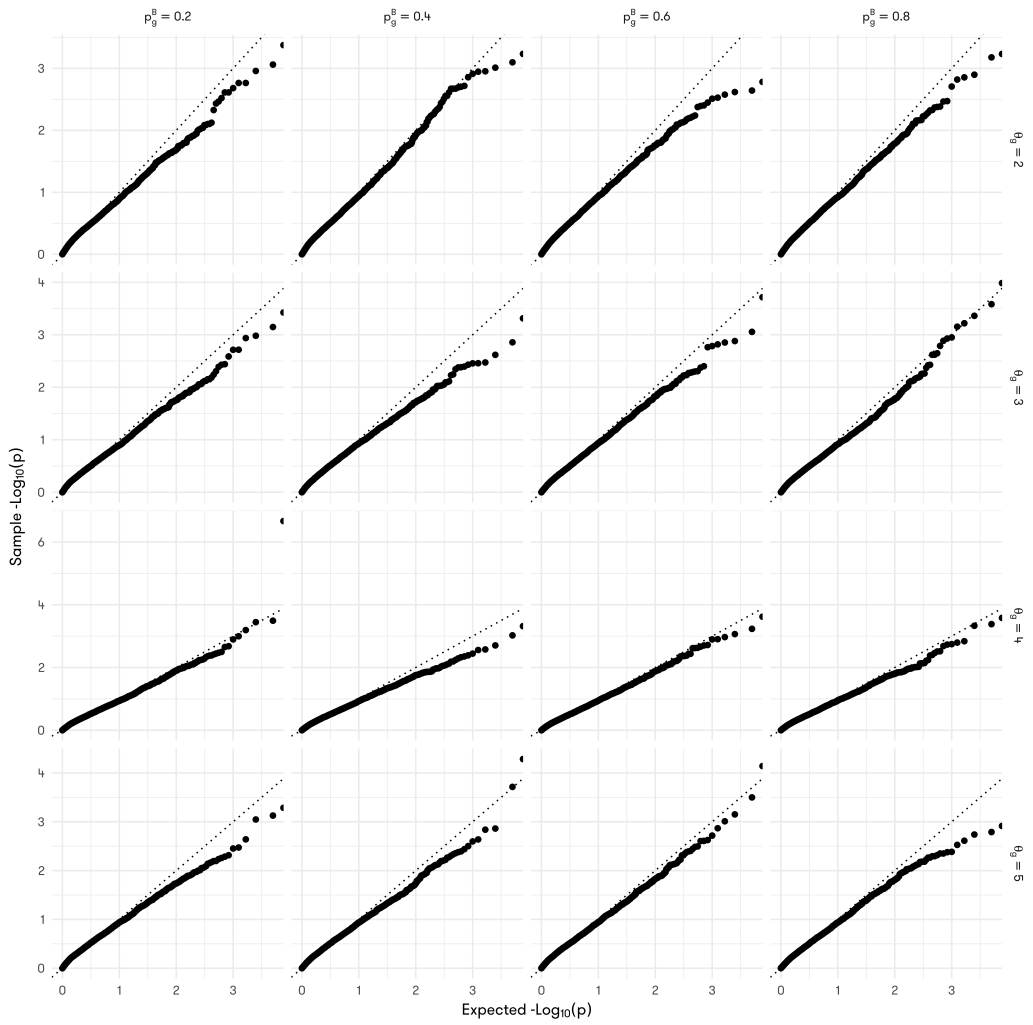


Figure 6.17: Q-Q plots of $-\log_{10}(\mathrm{p})$ comparing the p-values extracted from MAST Hurdle Test, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and $\mathrm{p}_g^{\mathrm{B}}$ used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns $\mathrm{p}_g^{\mathrm{B}}$.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.0768	0.033	0.0042	0.0024	0.0004
0.2	3	0.0756	0.0356	0.0046	0.0022	0.0004
0.2	4	0.0828	0.0418	0.0074	0.0032	0.0008
0.2	5	0.08	0.036	0.0042	0.0014	0.0006
0.4	2	0.0836	0.0414	0.008	0.0046	0.0006
0.4	3	0.0792	0.033	0.0044	0.0022	0.0002
0.4	4	0.083	0.0346	0.0038	0.0018	0.0004
0.4	5	0.083	0.0342	0.0064	0.0026	0.0004
0.6	2	0.0816	0.0358	0.0054	0.002	0
0.6	3	0.0824	0.0388	0.0054	0.0022	0.0004
0.6	4	0.0882	0.039	0.0072	0.0038	0.0006
0.6	5	0.087	0.039	0.0066	0.0036	0.0008
0.8	2	0.0782	0.0374	0.0058	0.0026	0.0004
0.8	3	0.0756	0.0322	0.0062	0.003	0.001
0.8	4	0.0878	0.037	0.0048	0.0026	0.0006
0.8	5	0.0846	0.0398	0.0064	0.002	0

Table 6.8: Estimated false positive rates from the null simulation with MAST Hurdle Test. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

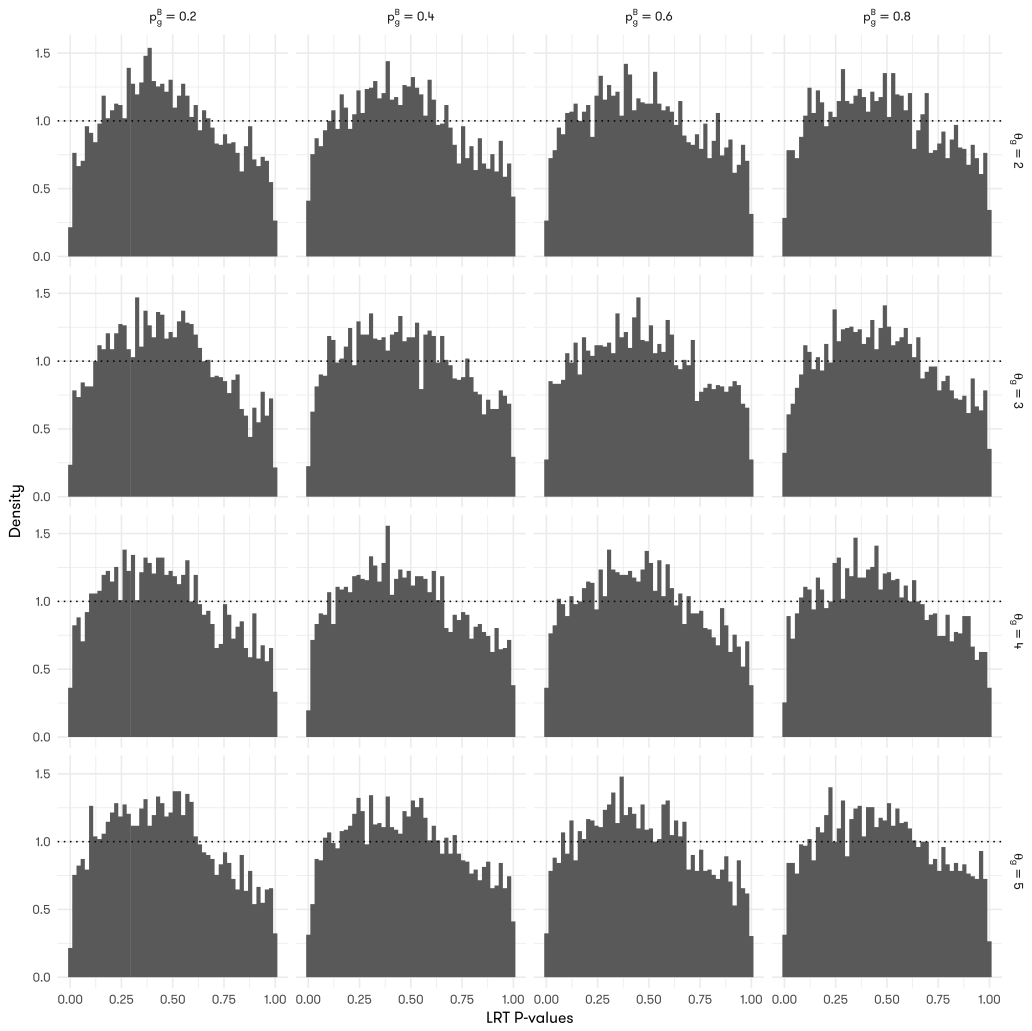


Figure 6.18: Histograms of raw p-values extracted from MAST Hurdle Test. With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

Combining the discrete and continuous tests, MAST can generate one unified p-value for testing whether there is any change in either the proportions of zero or the positive mean expression between groups. Since this hurdle test simply combines the value and degrees of freedom of the two χ^2 statistics, it inherits the tendency to be conservative from the continuous test (Figure 6.17), but smooths out the spikes from the discrete test (Figure 6.18). The end result is a still conservative test with slightly improved empirical FPR (Table 6.8).

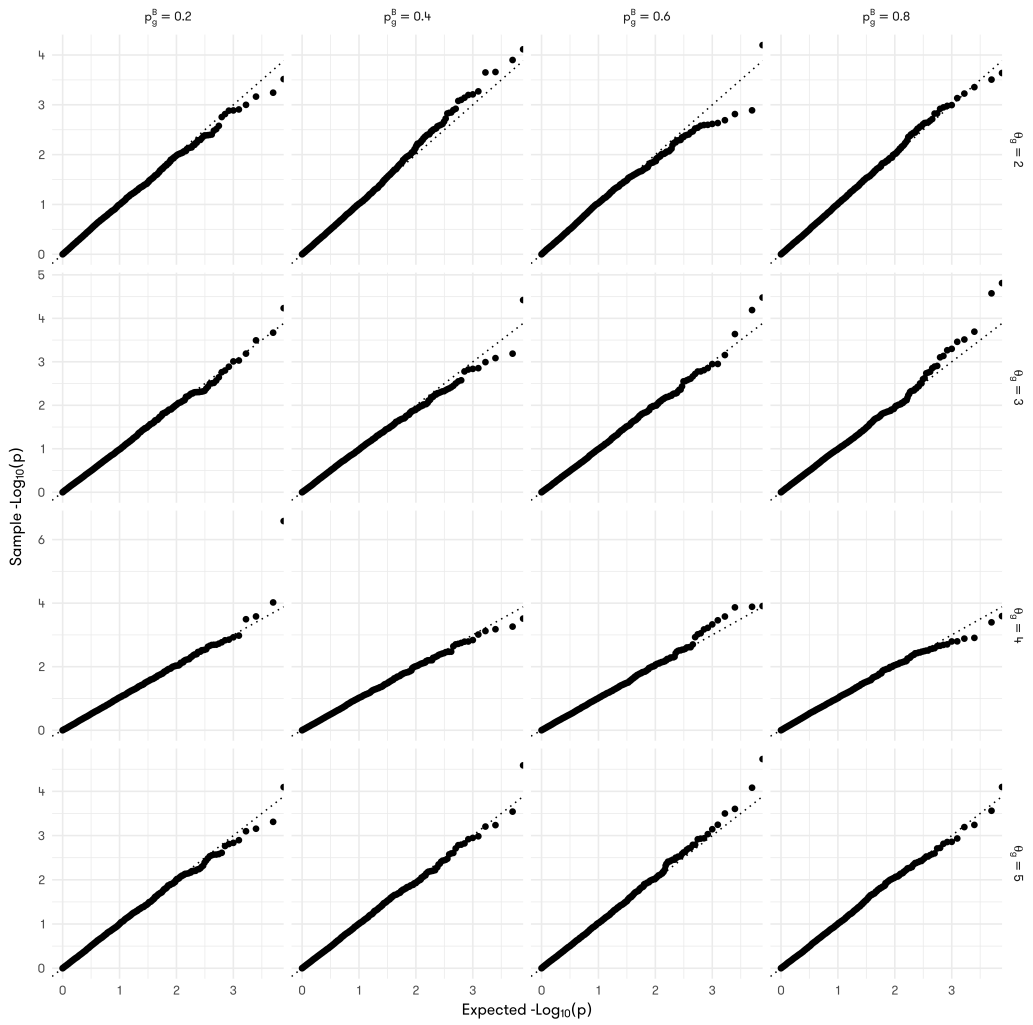


Figure 6.19: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from the original TASC package, and the expected quantile drawn from a uniform distribution on $(0, 1)$. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

		Significance Level				
p_g^B	θ_g	0.1	0.05	0.01	0.005	0.001
0.2	2	0.0994	0.0472	0.0096	0.0038	0.0006
0.2	3	0.0958	0.0498	0.0104	0.0048	0.0012
0.2	4	0.1074	0.053	0.0118	0.0056	0.0008
0.2	5	0.1004	0.049	0.0102	0.0036	0.0008
0.4	2	0.1036	0.0526	0.0126	0.0076	0.002
0.4	3	0.0964	0.048	0.0076	0.0038	0.0006
0.4	4	0.1044	0.0494	0.0106	0.0046	0.001
0.4	5	0.102	0.052	0.0084	0.0042	0.0008
0.6	2	0.1104	0.0538	0.008	0.0036	0.0002
0.6	3	0.0978	0.0504	0.0096	0.0042	0.0008
0.6	4	0.1024	0.0506	0.0116	0.0048	0.002
0.6	5	0.1064	0.054	0.0112	0.0068	0.0014
0.8	2	0.1028	0.0526	0.0104	0.006	0.001
0.8	3	0.098	0.0474	0.0092	0.0056	0.0018
0.8	4	0.1034	0.0518	0.0112	0.0054	0.0004
0.8	5	0.1004	0.0518	0.0122	0.005	0.0008

Table 6.9: Estimated false positive rates from the null simulation with the original TASC package. For each simulated scenario, the fraction of genes with raw p-value smaller than or equal to a specific significance level among all 5000 genes is computed. The estimated FPR using five different significance levels $(0.1, 0.05, 0.01, 0.005, 0.001)$ are listed.

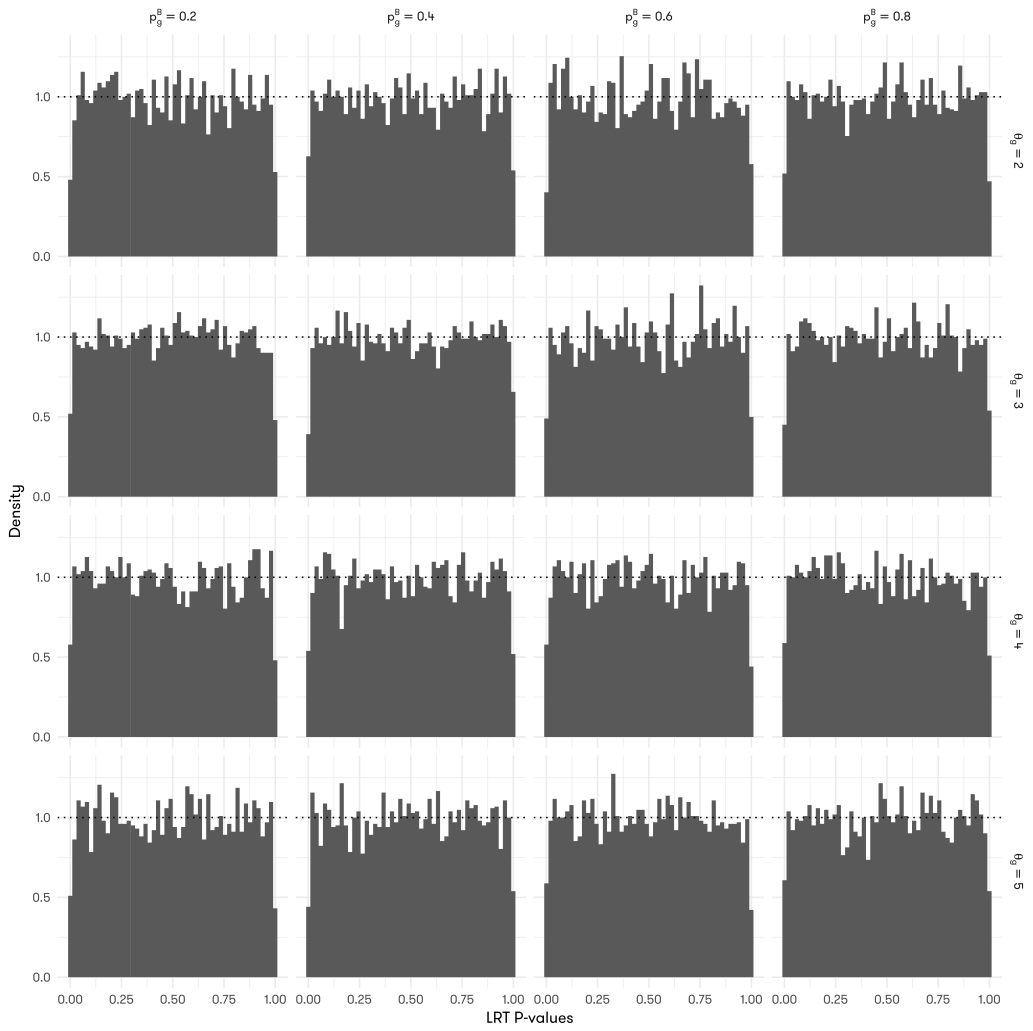


Figure 6.20: Histograms of raw p-values extracted from the original TASC package. With 5000 genes in 50 bins, each bin is expected to have a density of $50/5000 = 1\%$, which is indicated with the dotted line. Methods with well-controlled type I error should generate a histogram that evenly distributed between $(0, 1)$. θ_g and p_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

The original TASC does not consider p_g^B. Fortunately this does not cause any inflation in any simulated scenarios, even when significant proportions of cells are biologically zero *i.e.*, $p_g^B = 0.2$ (Figure 6.19). The overall distribution of the p-values from TASC resembles that of the $\mathrm{uniform}(0, 1)$ (Figure 6.20). Upon closer look at the numbers, we have found slight evidence of anti-conservativeness in certain scenarios. For example, when $p_g^B = 0.2$ and $\theta_g = 4$ (Table 6.9), empirical FPR is 7.4% higher than the nominal significance level ($\alpha = 0.1$). This could easily due to the estimation er-

ror of the simulation. Overall, the original TASC package will not experience type I inflation in the presence of significant bursting, as long as there is no difference in bursting levels between groups.

6.3.4. Performance Under the Alternative Hypothesis

To assess the sensitivity of the likelihood ratio tests implemented in TASC-B, data simulated under a variety of alternative hypotheses are fitted with TASC-B and other methods for comparison. All combinatorial scenarios with $\Delta p_g^B = p_{g1}^B - p_{g0}^B \in \{0, 0.1, 0.2, 0.3, 0.4\}$ and $\theta_g = \theta_{g1} - \theta_{g0} \in \{-3, -2, -1, 0, 1, 2, 3\}$, except for the combinations $(\Delta p_g^B, \Delta\theta_g) \in \{(0, -3), (0, -2), (0, -1)\}$, as these conditions are identical to $(0, 3), (0, 2), (0, 1)$ respectively. Other parameters are identical to the nulls simulation (subsection 6.3.2). Two groups of 300 cells each are simulated, compared using three LRTs (TASC-B Test2 #2, 3 and 4).

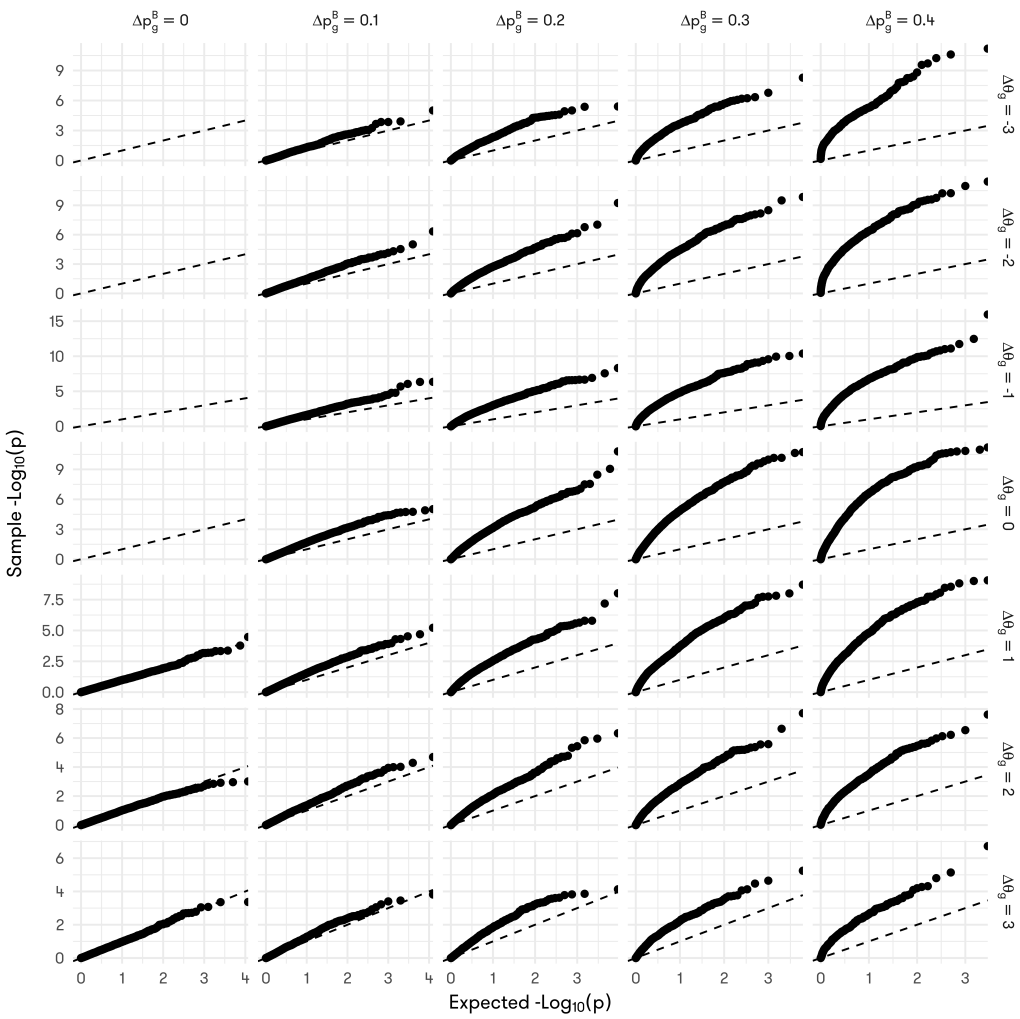


Figure 6.21: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from TASC-B Test #2, and the expected quantile drawn from a uniform distribution on $(0, 1)$ under a series of alternative hypotheses. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

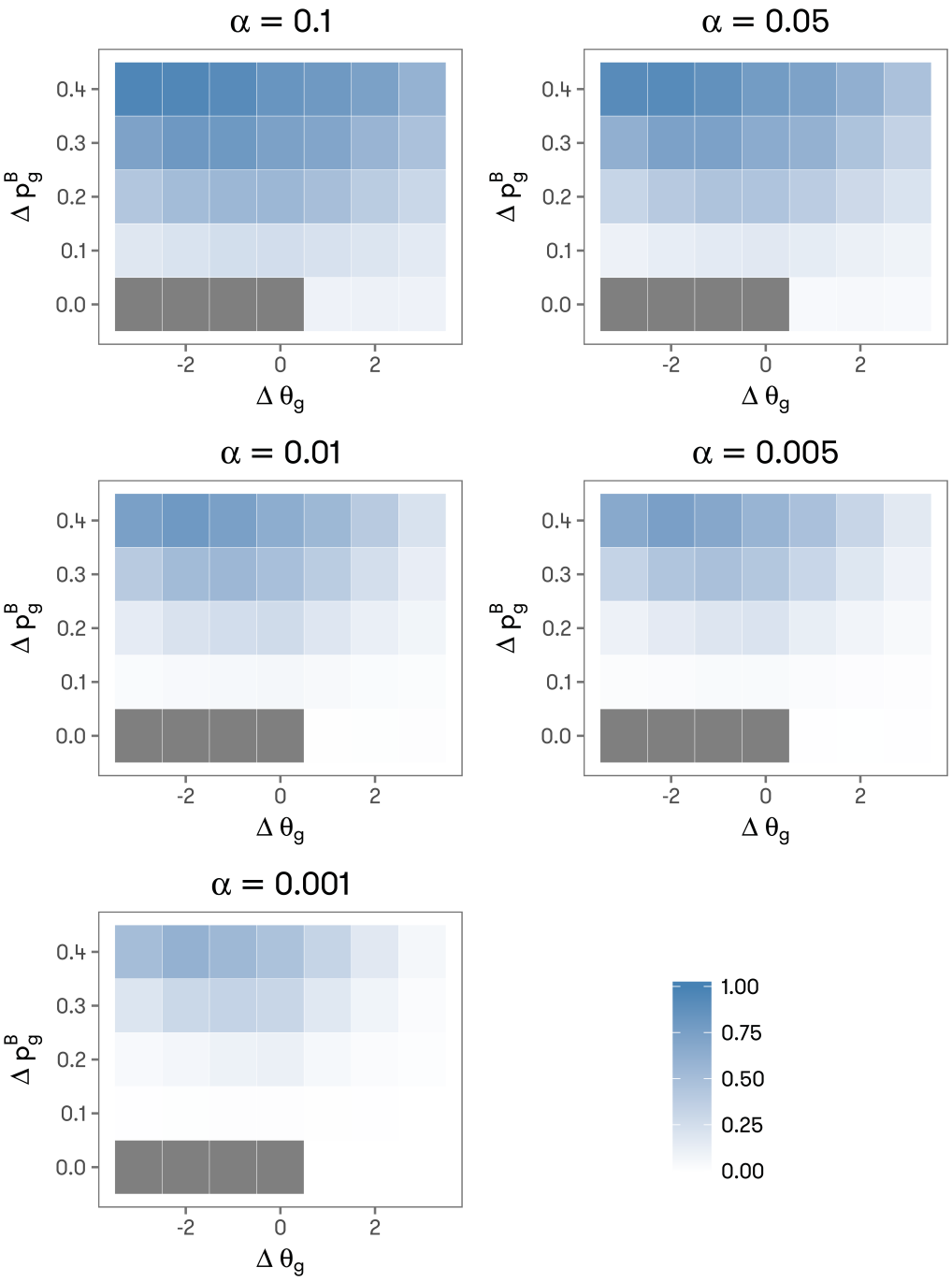


Figure 6.22: Heat maps illustrating the estimated power of the TASC-B Test #2 under various simulated scenarios. Empirical power is represented by the fraction of the genes with LRT p-values smaller than the specified significance levels ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$) among all genes simulated. Darker color represents higher power. Scenarios that are omitted from the simulation are filled with grey. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the left and bottom side of the graph, with rows differing in θ_g and columns p_g^B.

TASC-B Test #2 is testing for differences in bursting probability p_g^B between conditions. This test in our simulation successfully picks up the differences in p_g^B, and the power is highly dependent on the difference of bursting levels between the two groups (Figure 6.21). As Δp_g^B increases, the Q-Q plot deviates more severely from the unit line, suggesting that p-values are increasingly smaller than expected from a random uniform distribution. The empirical power at different significance levels, when plotted against the difference in p_g^B and θ_g on a heat map, corroborates the above claim (Figure 6.22). Interestingly, Test #2 has the most power when $\Delta\theta_g$ is not extreme. For example, when $\Delta p_g^B - 0.3$, significant power loss is observed when $\Delta\theta_g = 3$ or $\Delta\theta_g = -3$, with the former causing more power loss than the latter. Overall, the power curve of Test #2 is a function not only of the effect size Δp_g^B, but is also influenced by the positive mean difference $\Delta\theta_g$. This is primarily caused by the difficulties of attributing zeros to bursting, when $\Delta\theta_g$ can also explain the difference in proportions of zero, *i.e.*, causing it to fluctuate in the same direction.

Moreover, TASC-B Test #2 behaves quite well when $\Delta\theta_g \neq 0$ while $\Delta p_g^B = 0$. In this scenario, no type I inflation is observed, indicating resistance to confounding by $\Delta\theta_g$ with this test.

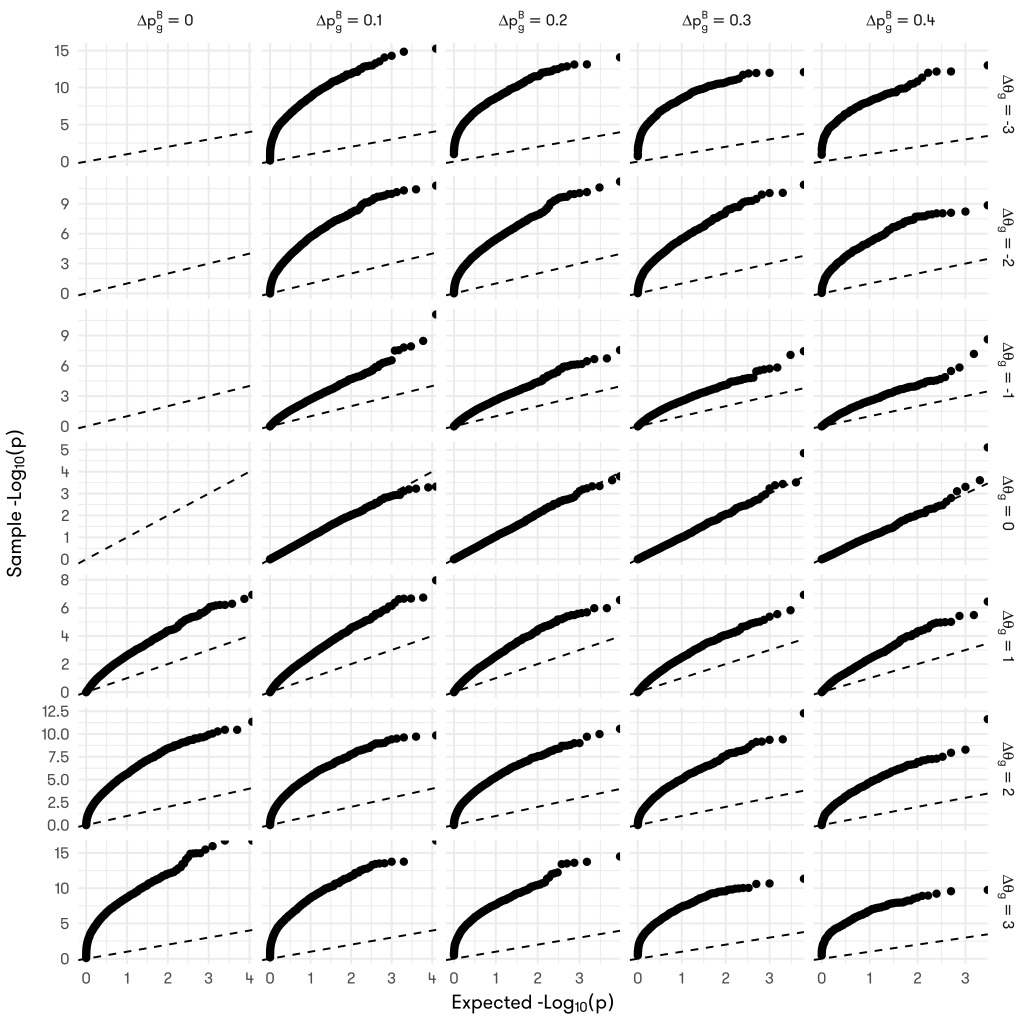


Figure 6.23: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from TASC-B Test #3, and the expected quantile drawn from a uniform distribution on $(0, 1)$ under a series of alternative hypotheses. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

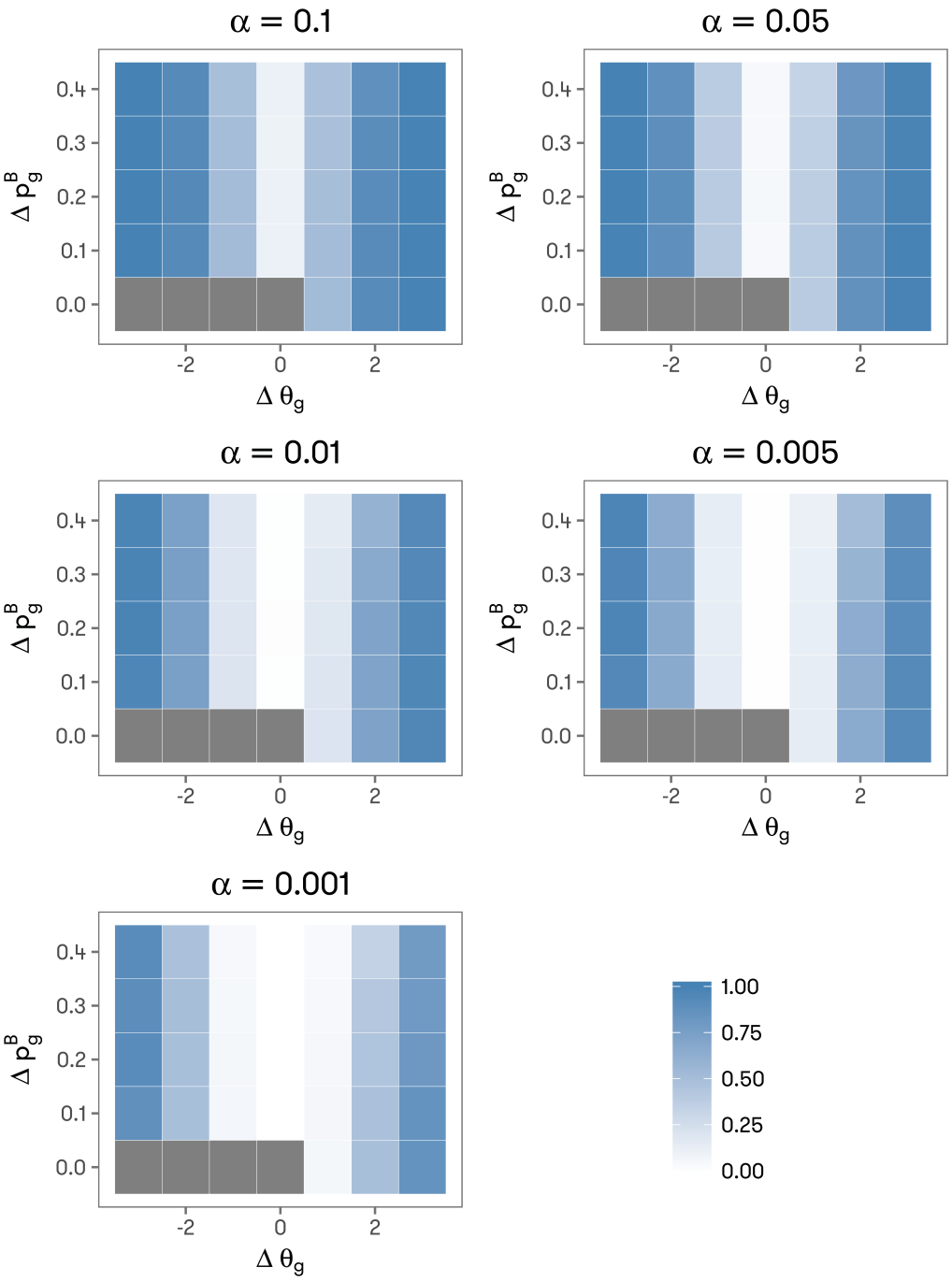


Figure 6.24: Heat maps illustrating the estimated power of the TASC-B Test #3 under various simulated scenarios. Empirical power is represented by the fraction of the genes with LRT p-values smaller than the specified significance levels ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$) among all genes simulated. Darker color represents higher power. Scenarios that are omitted from the simulation are filled with grey. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the left and bottom side of the graph, with rows differing in θ_g and columns p_g^B.

TASC-B Test #3 tests for the difference in the biological mean after taking into consideration possible differences in the bursting probabilities between groups. This test behaves very well under our simulation, a modest change of 2 on the log scale can be detected with sufficient power (Figure 6.23). The power of Test #3 is primarily influenced by the effect size (Figure 6.24). Levels of Δp_g^B does not affect the empirical power at any significance level. It is an overall powerful method at modest effect size. For example, at $\alpha = 0.05$, a difference of $|\Delta\theta_g| = 2$ can be detected with almost 100% power.

Similar to Test #2, Test #3 is not confounded by changes in Δp_g^B. When $\Delta\theta_g = 0$ and $\Delta p_g^B \neq 0$, no type I inflation is observed.

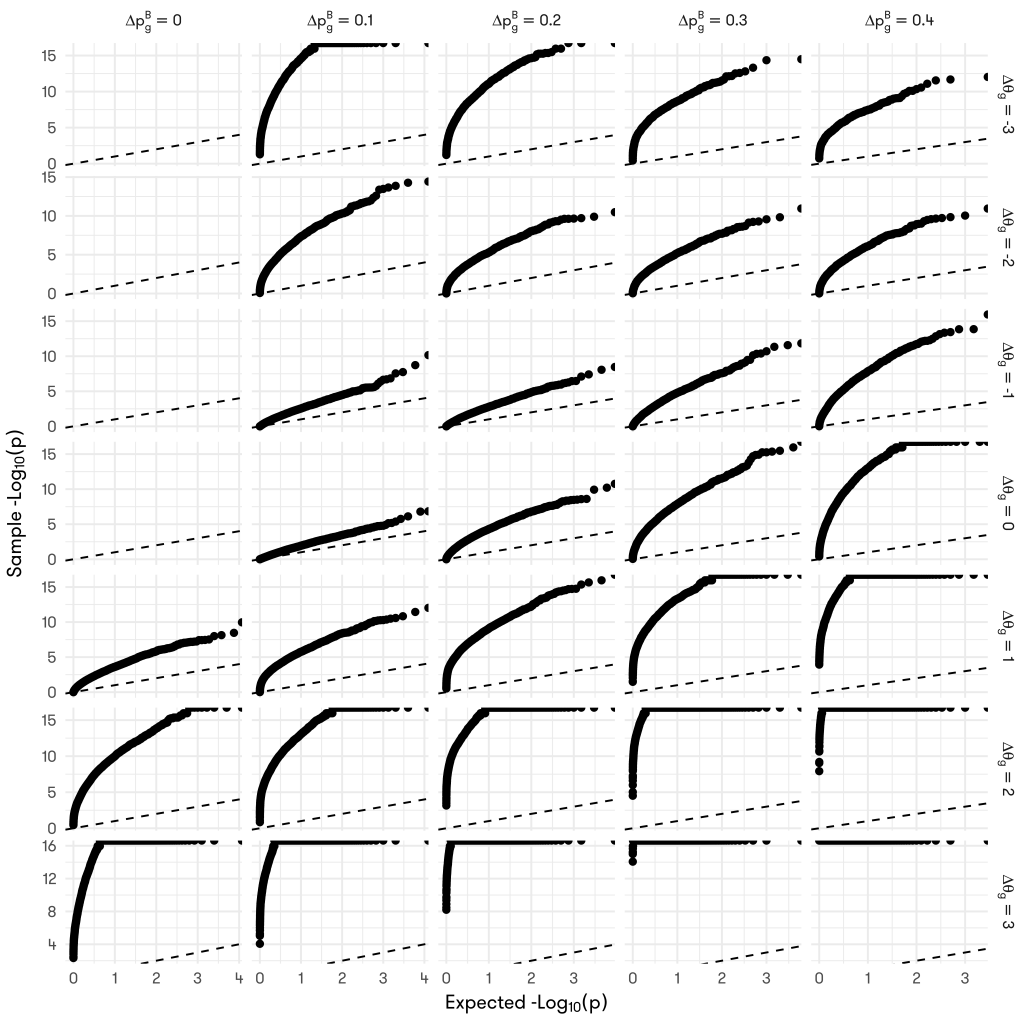


Figure 6.25: Q-Q plots of $-\log_{10}(p)$ comparing the p-values extracted from TASC-B Test #4, and the expected quantile drawn from a uniform distribution on $(0, 1)$ under a series of alternative hypotheses. Dotted line indicates the unit line, with intercept equal to 0, and slope equal to 1. Methods with well-controlled type I error should generate a line overlapping the unit line. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

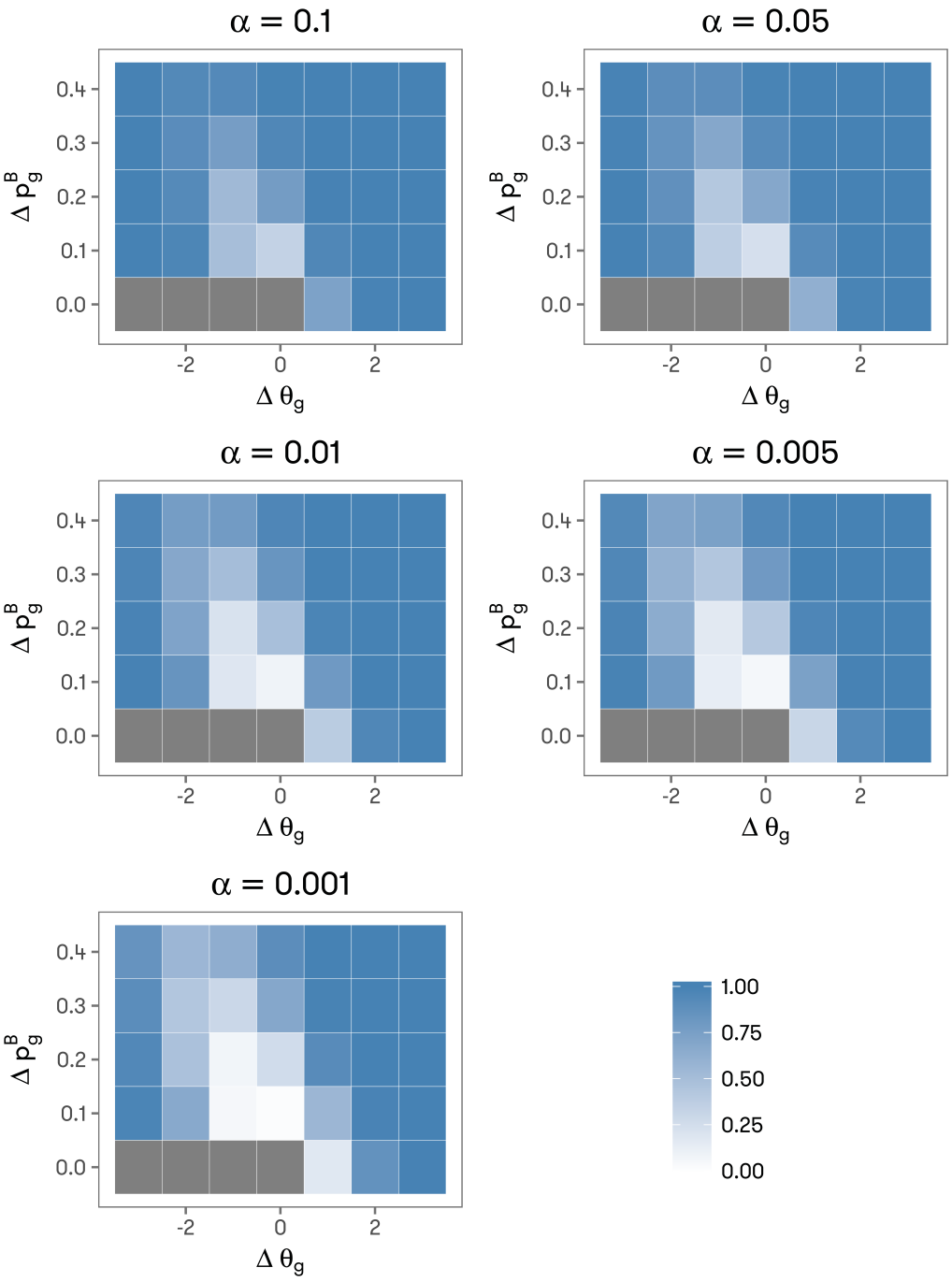


Figure 6.26: Heat maps illustrating the estimated power of the TASC-B Test #4 under various simulated scenarios. Empirical power is represented by the fraction of genes with LRT p-values smaller than the specified significance levels ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$) among all genes simulated. Darker color represents higher power. Scenarios that are omitted from the simulation are filled with grey. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the left and bottom side of the graph, with rows differing in θ_g and columns p_g^B.

Test #4 tests for changes in either p_g^B or θ_g between the two groups. It does not simply combine the two χ^2 statistics and their degrees of freedom like MAST, but computes a separate LRT by relaxing the constraints on the two parameters compared to the null hypothesis. It is highly sensitive to any changes in the vector (p_g^B, θ_g). Due to the stricter constraint on the null hypothesis, it is usually more powerful than Test #2 or #3, even when only 1 parameters is different between groups (Figure 6.25). For example, when $\Delta\theta_g = 0$, Test #4 is overall more powerful than Test #2, because it can achieve a higher level of confidence that at least one parameter has changed.

Except for a few difficult scenarios with only slight changes in p_g^B and θ_g, Test # can achieve almost 100% power at low significance level such as $\alpha = 0.05$ (Figure 6.26). It will be a great tool for screening any changes in either p_g^B or θ_g prior to running Tests #2 or #3 for more detailed testing.

6.3.5. Comparison to Other Methods Under Alternative Hypotheses

The same simulated groups are also compared with existing methods such as SCRAN coupled with DESeq2, MAST and the original TASC package. Similar graphs to those in subsection 6.3.4 are produced for all methods. For the interest of concision, two summary graphs are shown here, depicting the empirical power at significance levels $\alpha = 0.05$ for all methods under all simulated scenarios.

SCRAN with DESeq2

SCRAN coupled with DESeq2 suffers from severe type I inflation in our null simulations. It tends to be anti-conservative, especially when p_g^B and θ_g gets larger. This trend continues in situations where only one parameter is unchanged ($\Delta p_g^B = 0$ or $\Delta\theta_g = 0$). For example, when $\Delta\theta_g = 0$ and $\Delta p_g^B = 0.4$, SCRAN with DESeq2 is reporting 40% of the genes are differentially expressed, while in reality none of them are. This tendency to be anti-conservative may cause serious trouble with genes that are constitutively expressed, which account for the majority of the transcriptome in certain cell types, as we will see in Figure 6.28.

MAST The three tests of MAST (discrete, continuous and hurdle) vaguely overlap with TASC-B Tests #2, #3 and #4. However, their tests do not naturally translate to biologically interpretable concepts. When $\Delta p_g^B = 0$, the MAST discrete test, which simply compares the proportion of zeros, without considerations of technical dropout or Poisson sampling, is severely confounded by the

change in mean expression θ_g, while their continuous tests are notably underpowered in the same situations. In our previous studies, we have shown that under the null, MAST continuous tests are overly conservative in all simulated scenarios. This has affected the power of this test in this new simulation. In all scenarios simulated, TASC-B Test #3 overpowers the MAST continuous tests by a significant percentage. In scenarios where $\Delta p_g^B = 0.4$ and $\Delta\theta_g = 3$, MAST continuous almost has no power, while the power of TASC-B is almost 100% under these circumstances.

TASC The original TASC model, due to its lack of consideration for true biological zeros, suffer in some of the simulated scenarios. For example, when Δp_g^B and $\Delta\theta_g$ influences the mean expression in opposite directions (*e.g.*, $\Delta p_g^B = 0.4$, $\Delta\theta_g = -3$), thus canceling each other out, TASC is not immune to this type of confounding when testing for the difference in mean, while TASC-B can easily achieve a power of 100% in these cases. Moreover, this confounding can also cause TASC to report false positive results when $\Delta\theta_g = 0$ but $\Delta p_g^B \neq 0$.

6.4. Application to Real World Dataset

We have explored the performance of our TASC-B model, especially Test #2, as our package is one of the first methods capable of detecting the differences in levels of bursting probability to our knowledge. We have looked at all the level-1 classes from the Zeisel et al. dataset using Test #1 and #2 of TASC-B.

6.4.1. Test #1 on Zeisel et al. Data

Surprisingly, the majority of genes we have investigated are constitutively expressed (Figure 6.28), with some level-1 classes (e.g., microglia and endothelial mural) containing more genes with significant bursting than others. Interestingly, distribution is bimodal, with the mode closer to zero representing genes that are turned off in a significant portion of cells. The other mode around $\text{logit}\left[p_g^B\right] = 15$ represent those genes that are constitutively "on", *i.e.*, $p_g^B \approx 1$.

Looking closer, we have plotted the distribution of log read counts from the most significant genes that display the pattern of transcriptional bursting from all seven level-1 classes (only two are shown in Figure 6.29 and Figure 6.30). Notice that the distributions of the log read counts from the significantly bursting genes show severe zero inflation with an additional mode distant from 0. This visual confirmation gives us confidence that the model is picking up actual bursty genes rather than ran-

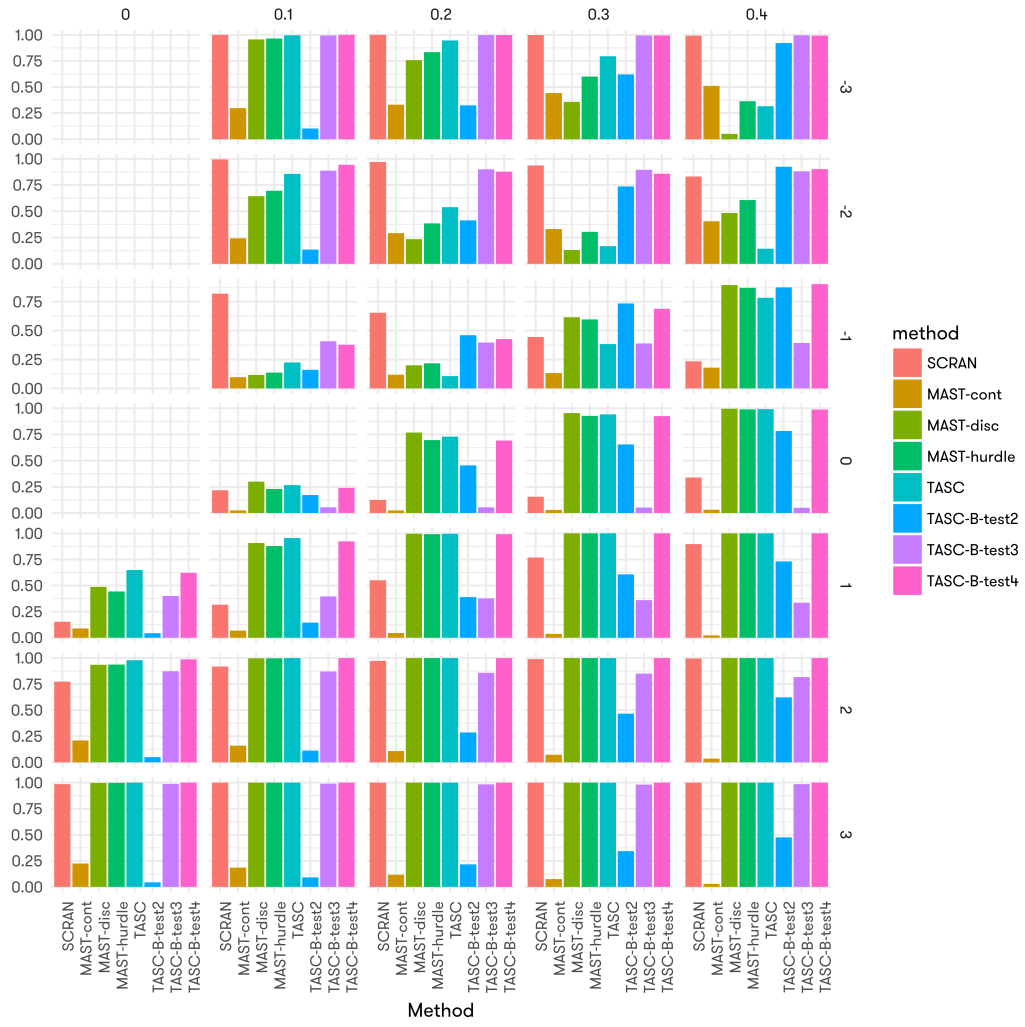


Figure 6.27: Bar plots illustrating the estimated power of the TASC-B tests (Tests #2, 3 and 4), and existing methods (SCRAN coupled with DESeq2, MAST and the original TASC package) under various simulated scenarios. Empirical power is represented by the fraction of genes with p-values smaller than the specified significance level ($\alpha = 0.05$) among all genes simulated. $\Delta\theta_g$ and Δp_g^B used to simulate the scenario is labeled on the right and top side of the graph, with rows differing in θ_g and columns p_g^B.

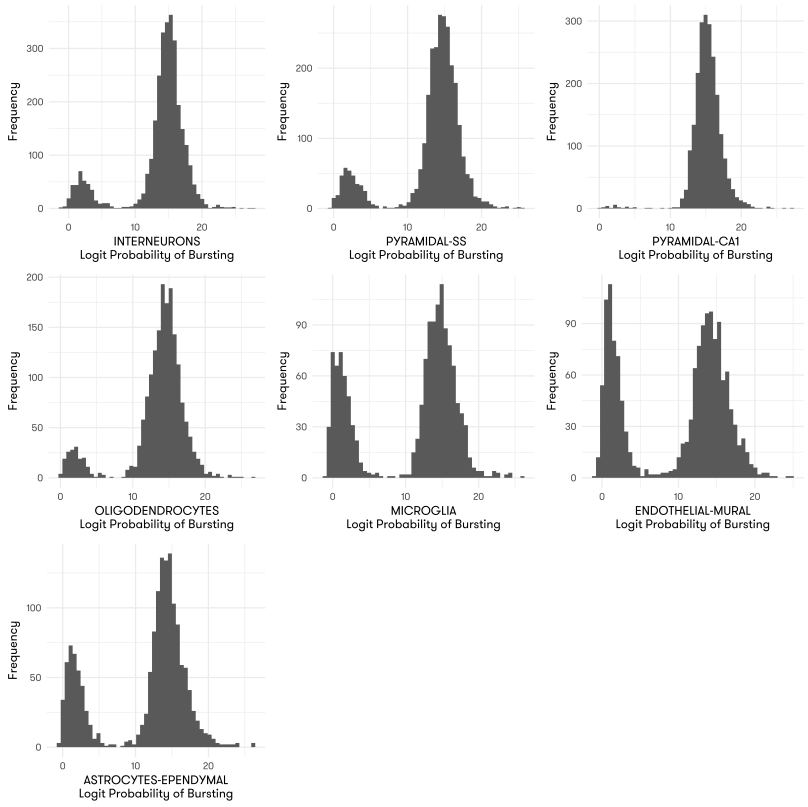


Figure 6.28: Histograms illustrating the distribution of logit-transformed probability of bursting in all seven level-1 classes from the Zeisel et al. data.

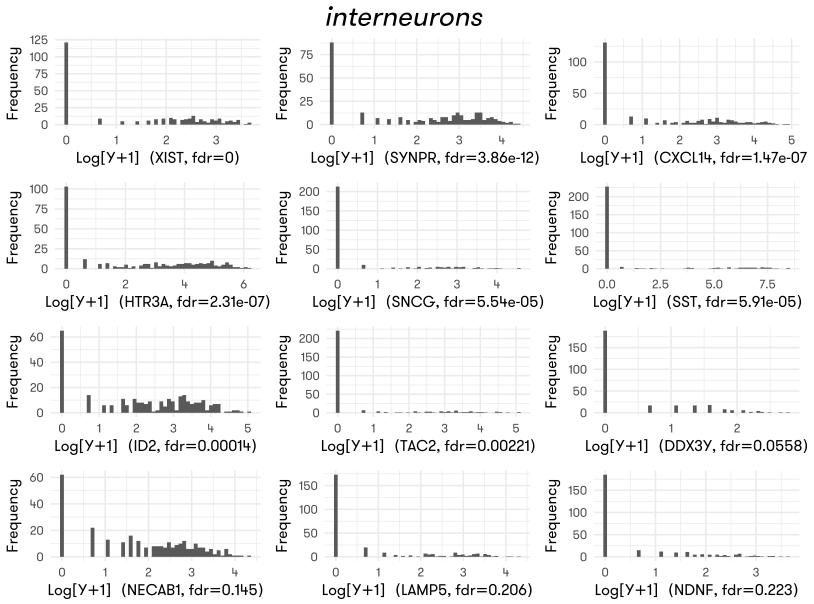


Figure 6.29: Histograms illustrating the distribution of log read counts ($\log(Y_{cg}+1)$) of the most significantly bursty genes in interneurons of the Zeisel et al. data.

dom noise in the signal. Judging from the FDR-adjusted p-values, our method is quite conservative in calling significantly bursting genes. For example, the gene LAMP5 displays an obvious bimodal pattern in interneurons, however, the false discovery rate is only 0.206. Test #1 is ideal for stringent screening of the whole genome for bursting genes.

Interestingly, one gene in particular, "Xist" has been discovered repeatedly by TASC-B to be consistently bursty in all the level-1 classes. This fact has never been reported before, and since Xist is a major effector in the process of X-chromosome inactivation, suggesting that transcriptional bursting might be related to these downstream epigenetic effects.

6.4.2. Test #2 on Zeisel et al. Data

We have also compared the difference in levels of bursting in the selected genes from any two of the level-1 classes in the Zeisel et al. data, using Tests #2, #3 and #4. 21 comparisons have been made, but only one is shown here (Figure 6.31) as an example.

We are able to visually inspect that the distributions of log read counts from the two groups under comparison (interneurons vs endothelial mural) are distinctively different, confirming that the method is picking up genes that are indeed with differential levels of bursting probability. Take the

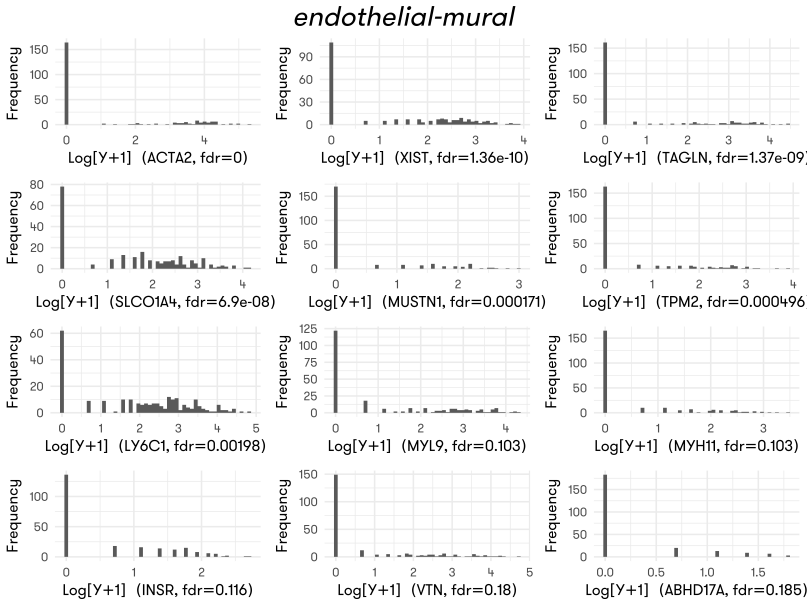


Figure 6.30: Histograms illustrating the distribution of log read counts ($\log(Y_{cg}+1)$) of the most significantly bursty genes in endothelial mural of the Zeisel et al. data.

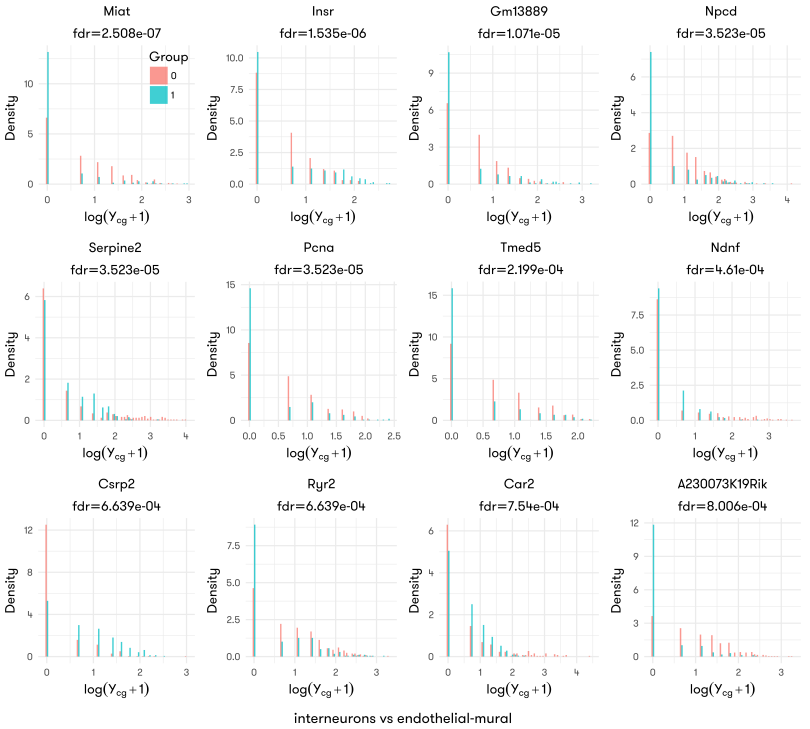


Figure 6.31: Histograms illustrating the distribution of log read counts ($\log(Y_{cg}+1)$) of the most significantly differentially bursting genes called by Test #2 in endothelial mural compared to interneurons of the Zeisel et al. data.

gene “Miat” for example, in interneurons, the pattern of expression (red) closely resembles a constitutively expressed gene. The majority of interneurons express this gene, and the inflated zeros can be attributed to technical dropouts as well as Poisson sampling. The majority of endothelial mural cells do not express Miat. For those that do, the expression is on average similar to that in interneurons, which suggests a scenario of modulation by transcriptional bursting. By turning off the expression of this gene in some cells, while leaving the rest in the “on” state, the tissue is able to lower the average expression (including the zeros) of Miat in the tissue, without changing the expression level in each single cell. The situation with the gene “Insr” is different. The overall proportion of zeros is somewhat similar between the endothelial mural cells and the interneurons. However, in interneurons, Insr is in a state of lowly constitutive expression. In endothelial murals, the mean positive expression of Insr' is increased compared to that in interneurons. The reduction of zeros from Poisson sampling is compensated by an increase of zeros from Insr being turned off in some of the endothelial cells. Although the proportion of zeros is unchanged between groups, there is evidence that changes in transcriptional bursting happens due to the different polarity in the expression pattern.

Overall, Test #2 has discovered evidence of DB in all comparisons from the Zeisel *et al.* data, suggesting the pervasive presence of regulation by transcriptional bursting.

Group 0	Group 1	Total Genes	False Discovery Rate						
			$FDR \leqslant 0.1$	$FDR \leqslant 0.05$	$FDR \leqslant 0.01$	$FDR \leqslant 0.005$	$FDR \leqslant 0.001$	$FDR \leqslant 5 \times 10^{-4}$	$FDR \leqslant 10^{-4}$
endothelial mural	astrocytes ependymal	883	0.0079275	0.0033975	0.002265	0.0011325	0.0011325	0.0011325	0.0011325
interneurons	astrocytes ependymal	551	0.2413793	0.1470054	0.0508167	0.0344828	0.0163339	0.0036298	0
interneurons	endothelial mural	518	0.3648649	0.2625483	0.0926641	0.0405405	0.0250965	0.015444	0.011583
interneurons	microglia	434	0.3686636	0.2511521	0.0668203	0.0253456	0.0046083	0.0023041	0.0023041
interneurons	oligodendrocytes	748	0.0508021	0.0240642	0.013369	0.0106952	0.0066845	0.0053476	0.0026738
interneurons	pyramidal CA1	1617	0.030303	0.0290662	0.0241187	0.021645	0.0166976	0.0136054	0.0111317
interneurons	pyramidal SS	1978	0.0085945	0.0060667	0.0050556	0.0035389	0.0030334	0.0020222	0.0010111
microglia	astrocytes ependymal	774	0.0077519	0.003876	0.001292	0.001292	0.001292	0.001292	0.001292
microglia	endothelial mural	871	0	0	0	0	0	0	0
oligodendrocytes	astrocytes ependymal	893	0.0145577	0.012318	0.0033595	0	0	0	0
oligodendrocytes	endothelial mural	830	0.0722892	0.0349398	0.0120482	0.0120482	0.0084337	0.0084337	0.0036145
oligodendrocytes	microglia	721	0.0915395	0.0208044	0.0041609	0.0041609	0.001387	0.001387	0.001387
pyramidal CA1	astrocytes ependymal	420	0.5952381	0.4880952	0.2928571	0.2166667	0.1238095	0.102381	0.047619
pyramidal CA1	endothelial mural	399	0.6466165	0.5964912	0.4235589	0.3283208	0.1804511	0.1378446	0.0802005
pyramidal CA1	microglia	323	0.6873065	0.5634675	0.3529412	0.2198142	0.1021672	0.0588235	0.0433437
pyramidal CA1	oligodendrocytes	569	0.2741652	0.2372583	0.1775044	0.1652021	0.1300527	0.1142355	0.0755712
pyramidal SS	astrocytes ependymal	569	0.1054482	0.0421793	0.0070299	0.0052724	0.0052724	0.0035149	0
pyramidal SS	endothelial mural	542	0.1476015	0.0811808	0.0276753	0.0166052	0.0110701	0.0110701	0.001845
pyramidal SS	microglia	458	0.1572052	0.0895197	0.0131004	0.0065502	0	0	0
pyramidal SS	oligodendrocytes	782	0.0191816	0.0153453	0.0076726	0.0051151	0.0012788	0.0012788	0.0012788
pyramidal SS	pyramidal CA1	1659	0.0620856	0.0433996	0.0247137	0.0174804	0.013261	0.0120555	0.0084388

Table 6.10: Fraction of genes with FDR adjusted p-values smaller than or equal to the given false discovery threshold from Test #2 in all comparisons. For each comparison, the names of the cell types compared, the total number of genes in this comparison, as well as seven different false discovery thresholds ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$) are included.

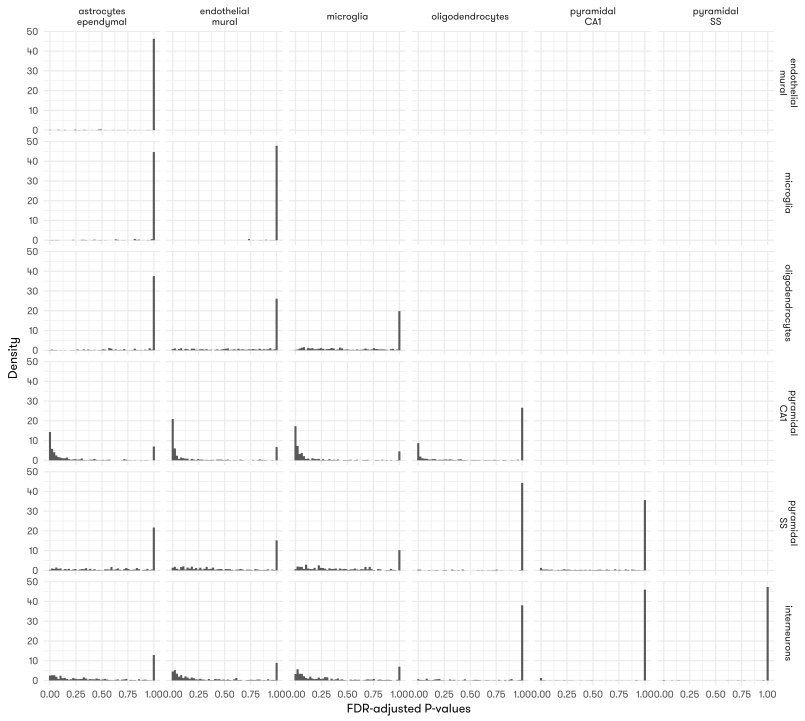


Figure 6.32: Histograms illustrating the distribution of the FDR-adjusted p-values from Test #2 comparing cell types from the Zeisel et al. data. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel.

Overall in Zeisel et al. data, DB varies depending on the groups being compared (Figure 6.32, Table 6.10). In some comparisons, significant portion of genes display DB behaviors, for example, the majority of genes tested that are co-expressed in pyramidal CA1 and cells from one of the following tissues: microglia, astrocytes ependymal, or endothelial mural, show distinctive patterns of bursting (Figure 6.32). However, when comparing pyramidal CA1 to pyramidal SS, interneurons or oligodendrocytes, no significant DB is observed. Bursting patterns essentially delineates the seven level-1 classes into two categories:

- pyramidal CA1, pyramidal SS, interneurons(neuron-like cells)
- astrocytes ependymal, endothelial mural, microglia (non-neuron-like cells)

The above classes essentially recapitulate the original classification, Figure 1C from Zeisel et al., 2015, in which the two groups form the two major clusters. Genes from oligodendrocytes display moderate amount of DB when compared to both types. In the original paper, it is reported as a

member of non-neuron-like cell clusters, although this could be an artifact of the clustering technique used in the paper. In our opinion, oligodendrocytes should be a transitional type between the neuron-like and non-neuron-like cells, and should belong to a separate category, judging from the bursting probability data alone.

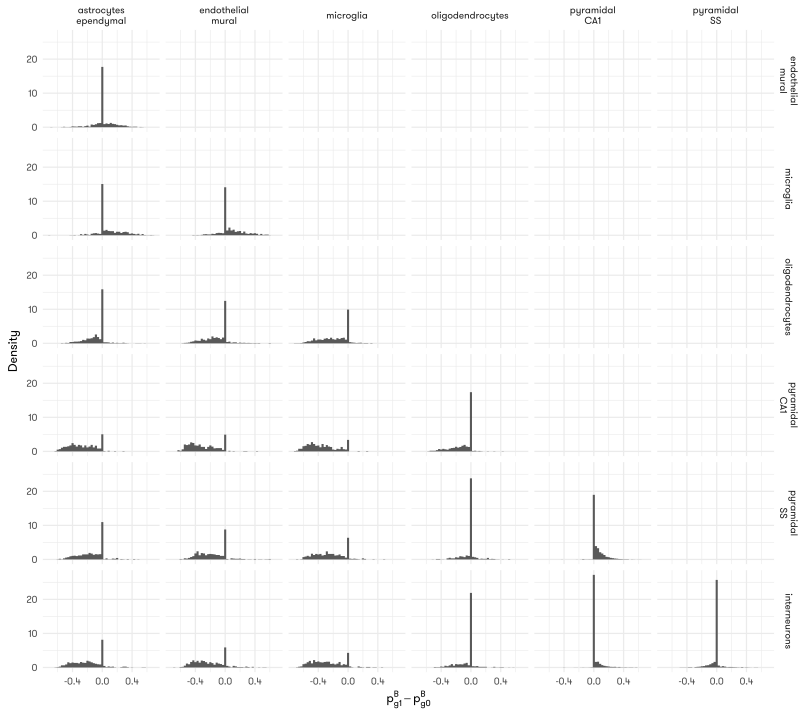


Figure 6.33: Histograms illustrating the distribution of the difference in level of transcriptional bursting ($\Delta p_g^B = p_{g1}^B - p_{g0}^B$) in any two cell types from the Zeisel et al. data. The difference is computed with fitted values from Test #2. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel. The dotted lines indicate zero.

Generally, the genes are more constitutively "on" in neuron-like cells, *i.e.*, less likely to be bursty compared to non-neuron-like cells (Figure 6.33). The difference in bursting probabilities can be as drastic as over 40%.

This suggests that transcriptional bursting, especially bursting probabilities, might be more indicative of the underlying cellular functions than we previously thought. Further investigations might be needed to see if bursting probabilities can be used to cluster cellular functions *ad hoc* in novel tissues.

6.4.3. Test #3 on Zeisel et al. Data

Test #3 aims to detect genes whose expression is changed between the two groups compared, after accounting for technical noise as well as bursting discrepancies. This is equivalent to testing the difference in the mean positive expression of a gene provided that it is in the "on" state. From the histograms of read counts of the most significant genes called by Test #3 (Figure 6.34), one can observe an obvious shift of the histograms. This visual inspection confirms the validity of our results.

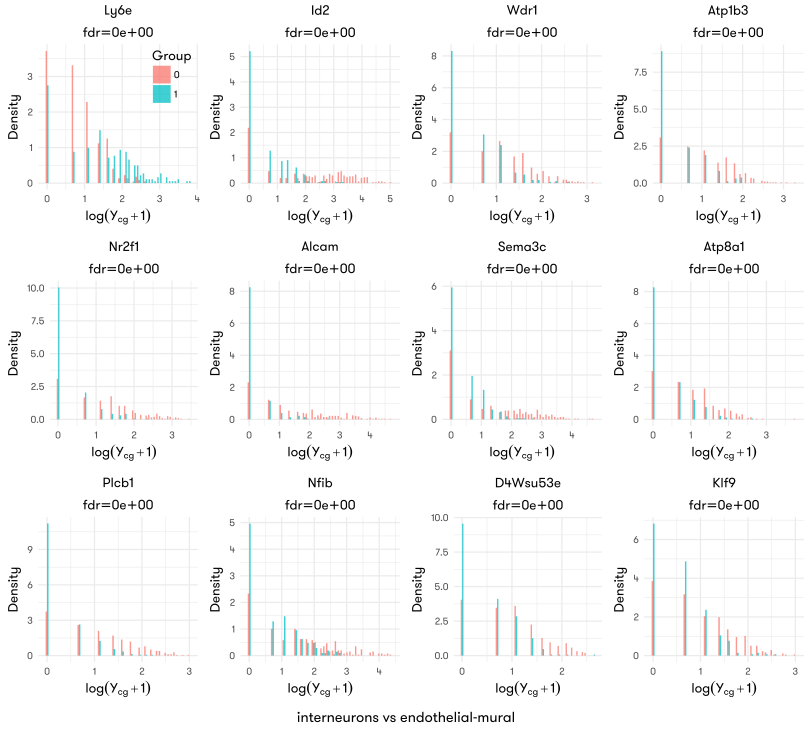


Figure 6.34: Histograms illustrating the distribution of log read counts ($\log(Y_{cg}+1)$) of the most significantly DB genes called by Test #3 in endothelial mural compared to interneurons of the Zeisel et al. data.

Interestingly, the number of significant genes detected from Test #3 shows a slightly different patterns than Test #2 (Figure 6.35, Table 6.11). The neuron-like cells (pyramidal CA1, pyramidal SS and interneurons) still cluster together by containing only a moderate number of genes that are significantly DE. Within-group difference in the non-neuron-like cells including astrocytes ependymal, endothelial mural and microglia is also mild, consistent with the patterns from bursting probabilities. Oligodendrocytes in the case of DE, cluster closer to the non-neuron-like group, with a majority of

Group 0	Group 1	Total Genes	Significance Levels						
			$FDR \leqslant 0.1$	$FDR \leqslant 0.05$	$FDR \leqslant 0.01$	$FDR \leqslant 0.005$	$FDR \leqslant 0.001$	$FDR \leqslant 5 \times 10^{-4}$	$FDR \leqslant 10^{-4}$
endothelial mural	astrocytes ependymal	883	0.2287656	0.1857305	0.1325028	0.1109853	0.0781427	0.0668177	0.0532276
interneurons	astrocytes ependymal	551	0.753176	0.7168784	0.6333938	0.5862069	0.4954628	0.4500907	0.4047187
interneurons	endothelial mural	518	0.7393822	0.6911197	0.5888031	0.5637066	0.4826255	0.4710425	0.4054054
interneurons	microglia	434	0.5069124	0.4308756	0.2695853	0.2165899	0.1129032	0.09447	0.0645161
interneurons	oligodendrocytes	748	0.855615	0.8355615	0.7847594	0.7540107	0.709893	0.6951872	0.6631016
interneurons	pyramidal CA1	1617	0.5862709	0.5213358	0.4223871	0.3778602	0.3129252	0.2869511	0.249227
interneurons	pyramidal SS	1978	0.4140546	0.3498483	0.2512639	0.2143579	0.1587462	0.1466127	0.1162791
microglia	astrocytes ependymal	774	0.1356589	0.0930233	0.0516796	0.0439276	0.0219638	0.0193798	0.0116279
microglia	endothelial mural	871	0.0861079	0.053961	0.0332951	0.0298507	0.0241102	0.0206659	0.0126292
oligodendrocytes	astrocytes ependymal	893	0.4389698	0.387458	0.2900336	0.256439	0.1926092	0.1769317	0.1399776
oligodendrocytes	endothelial mural	830	0.4289157	0.3698795	0.3012048	0.2674699	0.1975904	0.1807229	0.139759
oligodendrocytes	microglia	721	0.2024965	0.149792	0.0818308	0.0665742	0.0332871	0.0305132	0.0263523
pyramidal CA1	astrocytes ependymal	420	0.647619	0.6095238	0.4857143	0.4309524	0.3452381	0.3261905	0.252381
pyramidal CA1	endothelial mural	399	0.679198	0.6140351	0.4862155	0.4411028	0.3684211	0.3483709	0.273183
pyramidal CA1	microglia	323	0.3498452	0.2693498	0.1145511	0.1114551	0.0650155	0.0619195	0.0526316
pyramidal CA1	oligodendrocytes	569	0.8594025	0.8383128	0.7609842	0.7504394	0.7135325	0.7065026	0.6766257
pyramidal SS	astrocytes ependymal	569	0.741652	0.6889279	0.5922671	0.5448155	0.4674868	0.4411248	0.370826
pyramidal SS	endothelial mural	542	0.7546125	0.697417	0.5867159	0.5516605	0.4612546	0.4188192	0.3708487
pyramidal SS	microglia	458	0.5021834	0.3842795	0.231441	0.1899563	0.1157205	0.080786	0.0611354
pyramidal SS	oligodendrocytes	782	0.8414322	0.8043478	0.7608696	0.7276215	0.6905371	0.6713555	0.6342711
pyramidal SS	pyramidal CA1	1659	0.3990356	0.3254973	0.2405063	0.2169982	0.1639542	0.1488849	0.1127185

Table 6.11: Fraction of genes with FDR adjusted p-values smaller than or equal to the given false discovery threshold from Test #3 in all comparisons. For each comparison, the names of the cell types compared, the total number of genes in this comparison, as well as seven different false discovery thresholds ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$) are included.

the genes DE when compared to the neuron-like cell types. This is actually consistent with the classification from the original paper (Zeisel et al., 2015).

Another exception is genes in microglia. While displaying severe DB, they do not tend to be DE when compared to neuron-like cells such as pyramidal CA1, pyramidal SS, oligodendrocytes or interneurons. Genes in endothelial mural, however, show significance difference in both bursting and expression patterns from the neuron-like cells.

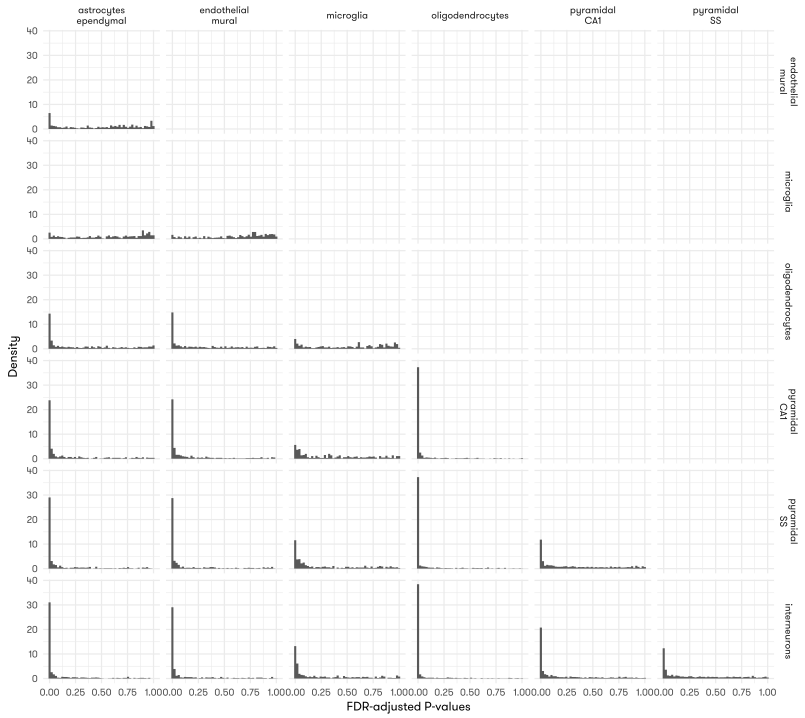


Figure 6.35: Histograms illustrating the distribution of the FDR-adjusted p-values from Test #3 comparing cell types from the Zeisel et al. data. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel.

Within neuron-like cells, the mean difference between two cells types is close to 0 (Figure 6.36). Similar situation is observed within the non-neuron-like cells, albeit with a slightly larger spread (Figure 6.36). When comparing across groups, however, gene expression is overall lower in non-neuron-like cells, such as astrocytes ependymal, endothelial mural, and microglia, compared to neuron-like cells such as pyramidal CA1, pyramidal SS and interneurons (Figure 6.36). Again consistent with the original report, the expression pattern of oligodendrocytes are closer to non-

neuron-like cells.

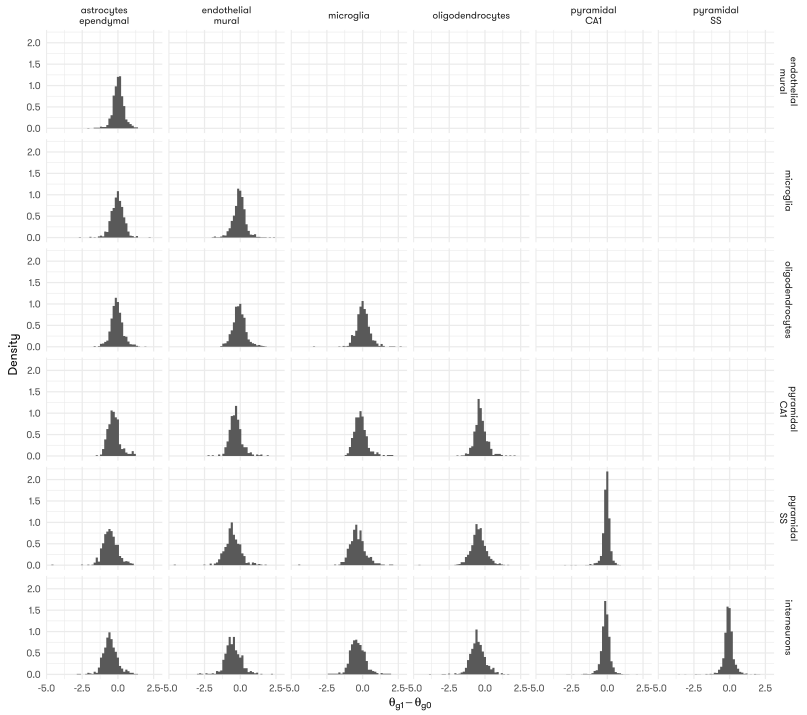


Figure 6.36: Histograms illustrating the distribution of the difference in positive mean expression ($\Delta\theta_g = \theta_{g1} - \theta_{g0}$) in any two cell types from the Zeisel et al. data. The difference is computed with fitted values from Test #3. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel. The dotted lines indicate zero.

In general, there is a slight decoupling of DE and DB, despite their close relationship with each other. Further investigation is needed on whether independent mechanisms are responsible for this decoupling, and if there is any biological relevance behind this phenomenon.

6.4.4. Test #4 on Zeisel et al. Data

Co-testing of significance of DB and DE (Test #4) is primarily intended as a pre-screening procedure for downstream testing for DB or DE separately with Tests #2 and #3. As testing procedures are computationally expensive.

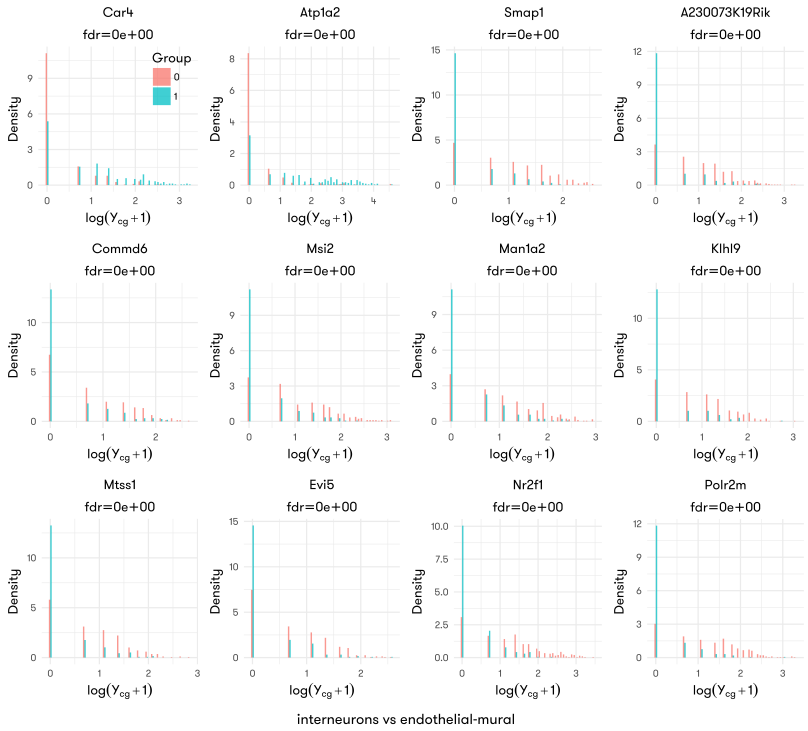


Figure 6.37: Histograms illustrating the distribution of log read counts ($\log(Y_{cg}+1)$) of the most significantly DB genes called by Test #4 in endothelial mural compared to interneurons of the Zeisel et al. data.

A visual inspection of the distribution of read counts from genes discovered by Test #4 shows a great separation of histograms, indicating high confidence in either DB or DE (Figure 6.37).

Group 0	Group 1	Total Genes	Significance Levels						
			$FDR \leqslant 0.1$	$FDR \leqslant 0.05$	$FDR \leqslant 0.01$	$FDR \leqslant 0.005$	$FDR \leqslant 0.001$	$FDR \leqslant 5 \times 10^{-4}$	$FDR \leqslant 10^{-4}$
endothelial mural	astrocytes ependymal	883	0.3431484	0.3001133	0.2242356	0.209513	0.1574179	0.1506229	0.1347678
interneurons	astrocytes ependymal	551	0.9019964	0.8820327	0.8439201	0.8275862	0.8058076	0.7858439	0.7604356
interneurons	endothelial mural	518	0.9092664	0.8899614	0.8474903	0.8397683	0.8069498	0.7972973	0.7779923
interneurons	microglia	434	0.8663594	0.843318	0.7857143	0.7695853	0.7281106	0.7096774	0.6520737
interneurons	oligodendrocytes	748	0.8716578	0.8596257	0.815508	0.8074866	0.7780749	0.7673797	0.7406417
interneurons	pyramidal CA1	1617	0.5528757	0.4990724	0.4032158	0.3692022	0.3129252	0.2999382	0.2603587
interneurons	pyramidal SS	1978	0.4287159	0.3822042	0.2836198	0.2548028	0.2022245	0.1850354	0.1562184
microglia	astrocytes ependymal	774	0.2777778	0.2416021	0.1705426	0.1485788	0.0956072	0.0891473	0.0723514
microglia	endothelial mural	871	0.1492537	0.1262916	0.0884041	0.0826636	0.0700344	0.0688863	0.0551091
oligodendrocytes	astrocytes ependymal	893	0.6170213	0.568869	0.4893617	0.4680851	0.3896976	0.3706607	0.3236282
oligodendrocytes	endothelial mural	830	0.6759036	0.6313253	0.5337349	0.5120482	0.453012	0.4277108	0.3831325
oligodendrocytes	microglia	721	0.5242718	0.4715673	0.3744799	0.3522885	0.2760055	0.2565881	0.2260749
pyramidal CA1	astrocytes ependymal	420	0.9357143	0.9261905	0.8738095	0.8666667	0.8404762	0.8309524	0.8
pyramidal CA1	endothelial mural	399	0.9423559	0.9398496	0.9273183	0.914787	0.8947368	0.8847118	0.8621554
pyramidal CA1	microglia	323	0.9287926	0.9164087	0.8544892	0.8328173	0.8018576	0.7863777	0.7399381
pyramidal CA1	oligodendrocytes	569	0.9033392	0.884007	0.8594025	0.8506151	0.8347979	0.8260105	0.8049209
pyramidal SS	astrocytes ependymal	569	0.8857645	0.8646749	0.8224956	0.8031634	0.7662566	0.7504394	0.7170475
pyramidal SS	endothelial mural	542	0.8837638	0.8634686	0.8413284	0.8265683	0.7915129	0.7712177	0.7380074
pyramidal SS	microglia	458	0.8318777	0.8187773	0.7489083	0.7074236	0.650655	0.6179039	0.5786026
pyramidal SS	oligodendrocytes	782	0.8478261	0.8337596	0.7941176	0.7749361	0.7404092	0.7365729	0.7007673
pyramidal SS	pyramidal CA1	1659	0.4454491	0.3881857	0.2899337	0.2597951	0.2091621	0.1886679	0.1597348

Table 6.12: Fraction of genes with FDR adjusted p-values smaller than or equal to the given false discovery threshold from Test #4 in all comparisons. For each comparison, the names of the cell types compared, the total number of genes in this comparison, as well as seven different false discovery thresholds ($\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$) are included.

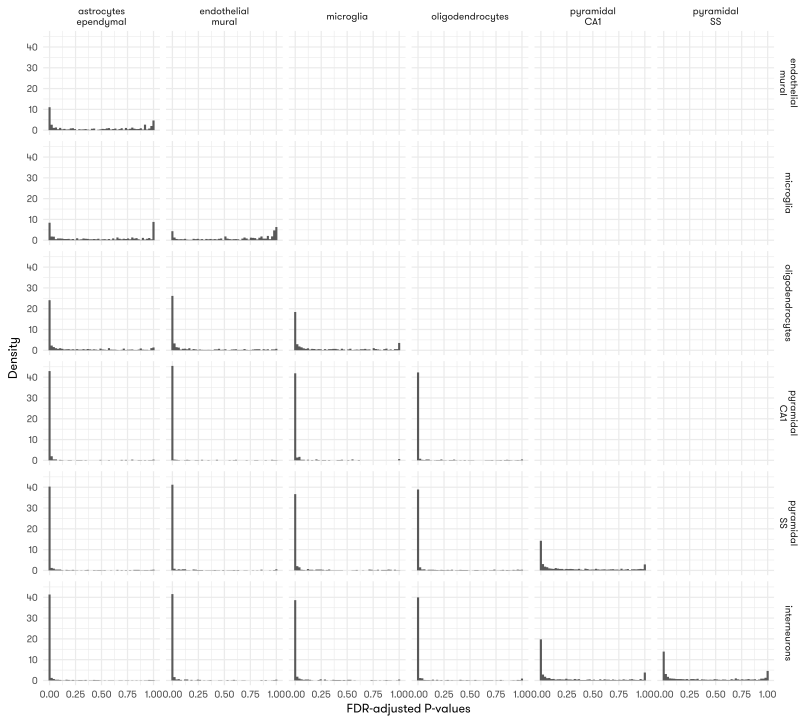


Figure 6.38: Histograms illustrating the distribution of the FDR-adjusted p-values from Test #4 comparing cell types from the Zeisel et al. data. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel.

Overall, the patterns resemble that of Tests #2 and #3, with clear clusters of two groups (neuron-like and non-neuron-like). Oligodendrocytes cluster closer with the non-neuron-like group just like in Test #3 (Figure 6.38), which is consistent with the fact that Test #4 is essentially a combination of the Tests #2 and #3. In addition, the proportion of genes significant for Test #4 is higher than Test #2 or #3 alone (Table 6.12).

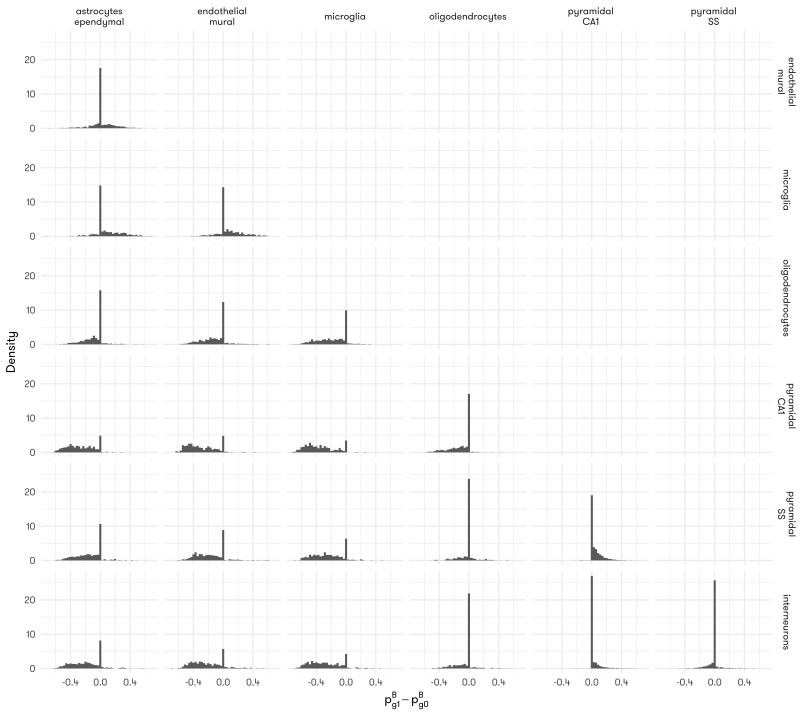


Figure 6.39: Histograms illustrating the distribution of the difference in level of transcriptional bursting ($\Delta p_g^B = p_{g1}^B - p_{g0}^B$) in any two cell types from the Zeisel et al. data. The difference is computed with fitted values from Test #4. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel. The dotted lines indicate zero.

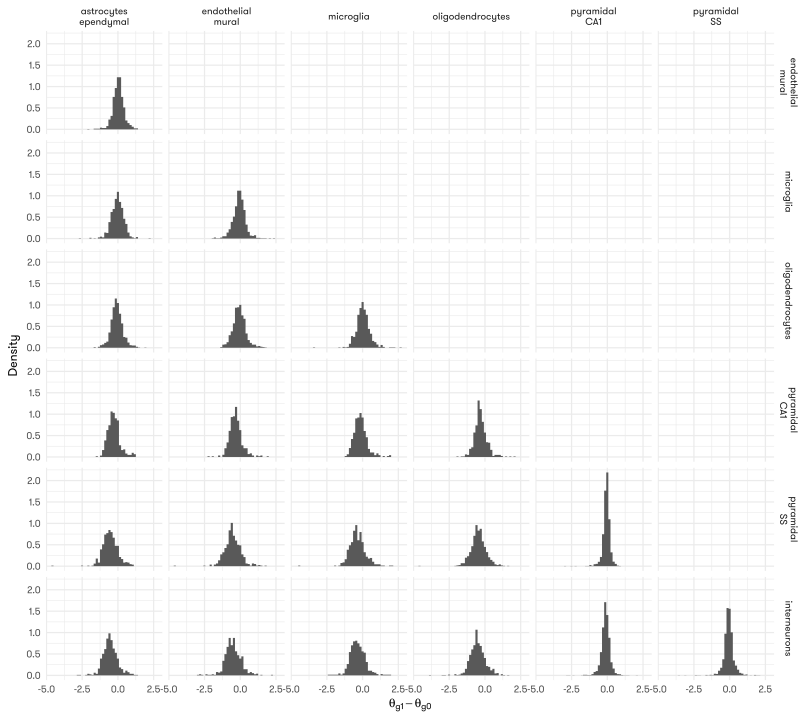


Figure 6.40: Histograms illustrating the distribution of the difference in positive mean expression ($\Delta\theta_g = \theta_{g1} - \theta_{g0}$) in any two cell types from the Zeisel et al. data. The difference is computed with fitted values from Test #4. Names of the two groups compared are labeled to the left (group 0) and top (group 1) of the panel. The dotted lines indicate zero.

Histograms of Δp_g^B and $\Delta\theta_g$ are also very close to the results from previous tests (Figure 6.39 and Figure 6.40).

In general, Test #4 is extremely sensitive in detecting the difference in the vector (p_g^B, θ_g). It will be a potentially useful test for pre-screening for genes that are different in at least on parameter between groups.

6.5. Computational Details

TASC-B is implemented in Python 2.7.12 and Cython 0.25.2. Other dependencies include numpy, scipy, mpi4py. Optimization is performed with the L-BFGS-B algorithm in scipy. Random restarts are implemented to avoid local optima. Taking advantage of the openMPI interface mpi4py, TASC-B can utilize hundreds of cores for speeding up the computation. The source code can be found on

github repo `https://github.com/scrna-seq/TASC-B`.

CHAPTER 7

DISCUSSION AND CONCLUDING REMARKS

RNA-seq has empowered a new generation of molecular biologists in their study of transcriptional regulation, by providing better tools for analysis of differentially expressed genes, transcripts. However, due to the intrinsic noise in the data collection process, it is of utmost importance that one properly adjusts for these technical biases. This dissertation has proposed several frameworks for analyzing RNA-seq data, from bulk to single-cell, accounting for the technical variations, which has greatly improved the statistical performance of the testing procedures.

7.1. MetaDiff

A major application of bulk RNA-seq is to detect differential isoform expression across experimental conditions. In this case, it is vital to account for the estimation error, due to the fact that isoform expression levels are estimated rather than observed, that they are estimated with various precision across samples, and that covariates and confounding factors may play a role to influence gene expression. To do so, we have proposed a flexible regression framework, utilizing the well-established random-effects meta-regression approach. Through computer simulations and the analysis of a real RNA-Seq dataset on human heart failure, we demonstrated that the proposed method can improve the power of isoform differential analysis while controlling for false positives due to the effect of covariates or confounding variables. The meta-regression approach we used is computationally efficient and widely available in existing statistical software packages. We have provided a tool and instructions on how to use meta-regression for isoform differential expression analysis with RNA-Seq data.

We have compared the performance of our method and other commonly used methods for differential expression analysis, including Cuffdiff, DESeq, DESeq2, EdgeR, and EBSeq. Both Cuffdiff and EBSeq take into account the estimation uncertainty for isoform expression levels, although EBSeq does not explicitly model the degree of uncertainty. In our simulated data, when no covariate and confounder influenced isoform expression (Scenario I), Cuffdiff had lower power than our method when m = 8 or 16, but better power when sample size was small (m = 4). In contrast, EBSeq had conservative FDR among non-DE transcripts and correspondingly lower power for detecting true

DE transcripts. When a covariate or confounder was present, these two methods showed either lower statistical power (in Scenario II) or inflated FDR (in Scenario III) as they are unable to adjust for covariates.

In our simulations, although EdgeR could control FDR in the presence of a confounder, it was conservative in all three scenarios. It also had lower power to detect true DE transcripts compared to our method and Cuffdiff in Scenario I. In contrast, DESeq showed inflated FDR for non-DE transcripts regardless the presence of confounder, especially when sample size was small. When a covariate or a confounder was present, DESeq had little power to detect DE transcripts that were correlated with the covariate. DESeq2 showed better performance than DESeq, however, its overall performance was not as satisfactory as BcLR and Student's *t*-test. We note that EdgeR, DESeq, and DESeq2 cannot take into account the uncertainty in isoform expression estimation, which may lead to biased testing results.

The uncertainty in an isoform FPKM estimate can be quantified as a standard error, which can be calculated in Cufflinks (Trapnell et al., 2010). In differential expression analysis, the FPKM value is usually log-transformed, and we approximate the variance of log(FPKM) using the delta method, which is also used in Cuffdiff 1.0 (Trapnell et al., 2010). However, this approximation can be poor and lead to false positive results when the variance of FPKM value is large compared to its magnitude. Hence we filtered out transcripts with large CVs in meta-regression. To avoid using delta method to approximate the variance of log(FPKM), one could use MMSEQ estimated isoform expression because MMSEQ directly gives the variance estimate of log(FPKM).

We note that meta-regression only requires estimates of isoform expression and the corresponding estimation uncertainty, but it is not tied to any particular estimation method.

we explained the equivalence between a random-effects model that accounts for estimation uncertainty in differential expression analysis and the random-effects metaregression. Meta-regression has been well studied in statistics and epidemiology literatures (Berkey et al., 1995; Greenwood et al., 1999; Higgins and Thompson, 2004) and is easy to implement using standard software. Both RNA-Seq analysis and metaanalysis face the same problem of small sample size. The BcLR test uses a correction factor to modify the standard LR test for small sample sizes. Huizenga, Visser, and Dolan, 2011b compared several testing procedures for meta-regression and showed that the BcLR test and t-test are the two best options. In our simulation study, we found that the BcLR test outperformed t-test with less conservative FDR and more power to detect DE transcripts.

7.2. sQTL

In sQTL analysis using RNA-Seq data, it is important to account for exon-inclusion level estimation uncertainty within a sample, directly model variation in the precision of exon-inclusion level estimates between samples, and allow for non-uniform read distribution.

We evaluated three statistical methods, including random effects meta regression, beta regression, and generalized linear mixed model, for the analysis of sQTLs. In contrast to GLiMMPS, which uses junction reads only to quantify exon inclusion levels, we used PennSeq (Hu et al., 2013), a statistical method that utilizes all available reads and allows non-uniform read distribution. Using both simulated and real RNA-Seq datasets, we demonstrated that all three methods outperformed GLiMMPS, and identified sQTLs at low false discovery rates but higher power.

The main reason for power improvement over GLiMMPS is due to the efficient use of additional information in exon-inclusion level estimation. Closer examination of the simulated data showed that the exon-inclusion levels using junction reads only were less well estimated as compared to PennSeq, which uses all available reads including those from flanking constitutive exons. Another reason is that GLiMMPS cannot model paired-end data structure, but PennSeq can effectively utilize paired-end read information in its modeling. In paired-end RNA-Seq data with tight distribution of insert size, reads mapped to flanking constitutive exons can provide useful information about the exon inclusion level. By using the generalized linear mixed model with estimates obtained from PennSeq, we confirmed that the power loss of GLiMMPS was due to the use of less accurate estimate of exon-inclusion levels.

We also examined the impact of non-uniformity on the performance of different methods. Not surprisingly, the power of all methods decreased for exon trios that demonstrate severe non-uniformity. Among the four methods we evaluated, PSGLMM and PSBeta had slightly inflated FDRs. The FDRs of both PSMeta and GLiMMPS were under control, but PSMeta had greater power. Overall, PSMeta appeared to be the most reliable yet powerful method for sQTL analysis. This further corroborates the importance of modeling non-uniform read distribution in exon-inclusion level estimation.

We only focused on exon-skipping events, but the framework we presented here can be easily generalized to examine other types of alternative splicing, including alternative 5' splice site, alternative 3' splice site, and mutually exclusive exons. In our analysis, we estimated the exon-inclusion

levels first and then identified sQTLs using regression based methods. This two-stage approach might be less powerful than identifying sQTLs using a one-stage approach, which avoids estimating exon-inclusion levels directly. We are currently pursuing extensions in this direction. Another possible direction of future research is to consider the overall splicing pattern of a gene instead of considering one exon-trio at a time. We anticipate that this approach will lead to more meaningful results.

7.3. TASC

Single-cell RNA-seq technology has enabled the exploration of between-cell heterogeneity in single-cell resolution. However, due to the limitations of current technology, scRNA-seq data are often noisy. Failure to account for technical noise can lead to biased downstream analyses and misleading results. To take full advantage of scRNA-seq, it is crucial to account for technical noise so as to better quantify biological variation. Here we have described a statistical framework, TASC, that accurately estimates cell-specific technical biases, adjusts for them in differential expression analysis, and consequently produces results that are more robust to batch effects that exhibit as systematic differences between cells.

TASC utilizes information in spike-ins to account for technical noise in a cell-specific manner. Compared to the traditional bulk RNA sequencing, in scRNA-seq the reverse transcription and preamplification steps can lead to pervasive dropout events and amplification bias. While amplification bias can be alleviated by the use of UMIs, dropout events are harder to control. To reliably estimate cell-specific dropout parameters under the paucity of reliable spike-ins at low concentrations, we have developed an empirical Bayes procedure that borrows information across cells. The accuracy of this empirical Bayes procedure has been examined in simulations based on real scRNA-seq data. Our evaluations show that TASC is always slightly conservative. However, we are willing to accept this slight conservativeness, since the data used in our evaluations are generated under an ideal null distribution, and we believe it is more meaningful to examine each methods performance in noisier data where strong deviations from the null can be observed. This is the motivation for our analysis of the data set from SCAP-T involving the comparisons of two groups of neurons with batch effects. Our results show that TASC achieves accurate type I error control under this noisy setting, whereas other methods have substantially inflated type I errors.

An important feature of TASC is the ability to adjust for covariates such as cell size and cell cycle. If

the goal is to find genes that differ in concentration between two cell types, then one should adjust for cell size. If one doesnt adjust for cell size, then most genes would be significant, since the expression of most genes scale with cell size, and thus, the genes that are markers for real pathway differences between cell-types would be hard to detect. Ultimately, whether to adjust for cell size is a decision for the user, and our goal through TASC is to provide the flexibility.

The hierarchical mixture model underlying TASC allows for flexible modeling of the true biological variation of gene expression across cells, and thus can be adapted to tackle many interesting biological questions. For example, ranking the estimated values of σ_g^2 allows us to identify biologically variable genes. The posterior expectation of μ_{cg} also gives us the inferred true expression value given the observed read counts. To illustrate the importance of accurate adjustment for cell-specific technical noise, we have benchmarked TASC against existing methods for differential expression analysis.

TASC incorporates the estimated technical parameters, which reflect cell-to-cell differences that may lead to batch effects, into a hierarchical mixture model to estimate the biological variance of a gene and to detect DE genes. The EM algorithm implemented in TASC offers a flexible and efficient approach to adjust for additional covariates to further eliminate confounding originated from cell size and cell cycle differences. In our evaluations, TASC appears to be robust in the detection of DE genes when batch effects are present.

TASC is implemented in an open-source program (`https://github.com/scrna-seq/TASC`), with multithreading acceleration by openMP. For example, a data set of 104 cells and 6,405 genes takes 45MB of memory and 18.6 minutes using 20 cores (Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz) with Laplacian approximation using the binary we provided. Better performance can be achieved when using binaries compiled on the users hardware. We believe that TASC will provide a robust platform for researchers to leverage the power of scRNA-seq.

7.4. TASC-B

The original implementation of TASC assumes that the true expression levels of a gene follow a logNormal distribution in cells. We recognize that logNormal does not entirely reflect the true distribution as transcriptional bursting could lead to zeros in the expression. A more realistic distribution is zero-inflated logNormal, which accounts for true zeros in gene expression. This has motivated us to develop TASC-B, an extension from TASC, incorporating an additional parameter p_g^B to describe

the probability of a gene being in the "on" state in the cell population of interest.

We have shown with simulation studies that the maximum likelihood estimators are consistent, showing little bias at moderately sized samples ($\mathrm{n} = 300$). In typical scRNA-seq experiments, one batch usually contains up to hundreds, if not thousands of cells, using the Fluidigm system. Experiments performed with the new droplet-based scRNA-seq technologies, can further increase the sample size to one or two magnitudes larger. This suggests that in actual experimental conditions, the bias of our estimators is negligible, across a wide range of values for both parameters. We note that difficulties arise when the expression of the gene is relatively low, or the gene is turned off in a large fraction of the cells. But this mainly affects the estimation uncertainty without causing serious bias.

Single-cell RNA-seq data can also be used to compare bursting probabilities across distinct populations of cells. We have devised additional LRTs specifically for testing the differences in bursting probabilities (TASC-B Test #2), levels of constitutive expression (TASC-B Test #3), or both (TASC-B Test #4). Due to the proper adjustment for the technical noise, as well as careful avoidance of local maxima in our optimization procedures, all of the above tests have well-controlled type I error rates. We also note that currently no method exists to test for the exact changes we aim to test with TASC-B. The three tests proposed by MAST may sound similar to our component-wise testing schemes, however, the former fails to account for technical noises and their tests are not naturally relatable to biological concepts. Tests proposed by TASC-B on the other hand, have clear biological relevance. When testing for changes in gene expression, it is important to avoid confounding by transcriptional bursting and cell-specific technical dropout. We have shown that without explicitly modeling these variations, tests can produce false positive results, or be greatly reduced in power.

Due to technological limitations, studies on transcriptional bursting rarely cover the entire transcriptome, which makes scRNA-seq the only available method capable of screening for transcriptionally bursty genes in a high-throughput fashion. We have shown using a likelihood-ratio test (TASC-B Test #1) from the Zeisel et al. data that different cell types have varied proportions of genes expressed in a bursty manner. Most genes in neuron-like cells, such as pyramidal CA1, pyramidal SS and interneurons, exhibit patterns of constitutive expression. More genes in non-neuron-like cells, such as microglia, astrocytes ependymal and endothelial mural, are turned off in a significant proportion of cells, demonstrating a pattern of bursty transcription. One gene in particular "Xist" is turned off in a majority of cells in most of the tissues we have looked at. Further investi-

gation is needed to explain the potential correlation between the bursty expression and its role in X-chromosome inactivation.

Screening for differentially bursting genes has produced a generous list of genes that demonstrate significantly distinctive patterns of bursting and expression between cell types from the Zeisel *et al.* data. These genes have produced a great starting point for functional annotations and gene ontology enrichment studies.

7.5. Concluding Remarks

In this dissertation, we have proposed MetaDiff, a flexible regression framework for isoform differential expression analysis that can take into account isoform expression estimation uncertainty and variation across biological replicates, and allow for covariate adjustment. Our method can effectively control for false positives due to confounding and increase the power to detect true DE transcripts.

We have also evaluated three statistical methods for the analysis of sQTLs in RNA-Seq. As shown by both simulations and the analysis of real data, the most robust method is PSMeta, a random effects meta regression based approach. An appealing feature of PSMeta is that it can be easily implemented using existing software packages. Results from this study will be instructive for researchers in selecting the appropriate statistical methods for sQTL analysis.

We have proposed a new statistical framework TASC, that allows a more robust utilization of spike-ins to account for cell-specific technical noise. To obtain reliable estimates of cell-specific dropout parameters, we have developed an empirical Bayes procedure that borrows information across cells. We have demonstrated an application of this general framework by a likelihood-based test for differential expression. TASC can flexibly and efficiently adjust for cell-specific covariates, such as cell cycle stage or cell size, which may confound differential expression analysis. We believe that TASC will provide a robust platform for researchers to leverage the power of scRNA-seq.

We have finally extended the TASC model to TASC-B, incorporating additional parameters to characterize the bursting probabilities in a cell population. We have developed MLEs to infer the values of the bursting parameters. In addition, likelihood ratio tests have been developed to test for differences in bursting parameters between groups. TASC-B is advantageous for it is immune to the confounding of transcriptional bursting when testing for changes in mean expression. We have discovered Xist as a potential gene that is generally bursting in all neuronal cells we have investigated.

TASC-B can be a valuable tool in studying transcriptional bursting using scRNA-seq data.

APPENDIX

SOFTWARE

The software mentioned in this dissertation can be found at:

- MetaDiff: `https://github.com/jiach/MetaDiff`
- TASC: `https://github.com/scrna-seq/TASC`
- TASC: `https://github.com/scrna-seq/TASC-B`

BIBLIOGRAPHY

Anders, S and Huber, W (2010). Differential expression analysis for sequence count data. *Genome biology* 11.10, R106.

Anders, S, Reyes, A, and Huber, W (2012). Detecting differential usage of exons from RNA-seq data. *Genome research* 22.10, 2008–2017.

Andrews, S et al. (2010). *FastQC: a quality control tool for high throughput sequence data*.

Ansari, AM, Ahmed, AK, Matsangos, AE, Lay, F, Born, LJ, Marti, G, Harmon, JW, and Sun, Z (2016). Cellular GFP toxicity and immunogenicity: potential confounders in in vivo cell tracking experiments. *Stem Cell Reviews and Reports* 12.5, 553–559.

Aschoff, M, Hotz-Wagenblatt, A, Glatting, K-H, Fischer, M, Eils, R, and König, R (2013). SplicingCompass: differential splicing detection using RNA-Seq data. *Bioinformatics*, btt101.

Bacher, R and Kendziorski, C (2016). Design and computational analysis of single-cell RNA-sequencing experiments. *Genome biology* 17.1, 63.

Baker, SC, Bauer, SR, Beyer, RP, Brenton, JD, Bromley, B, Burrill, J, Causton, H, Conley, MP, Elespuru, R, Fero, M, et al. (2005). The external RNA controls consortium: a progress report. *Nature methods* 2.10, 731–734.

Bartholomew, D (1961). A test of homogeneity of means under restricted alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, 239–281.

Bengtsson, M, Ståhlberg, A, Rorsman, P, and Kubista, M (2005). Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome research* 15.10, 1388–1392.

Berkey, CS, Hoaglin, DC, Mosteller, F, and Colditz, GA (1995). A random-effects regression model for meta-analysis. *Statistics in medicine* 14.4, 395–411.

Blake, WJ, Balázsi, G, Kohanski, MA, Isaacs, FJ, Murphy, KF, Kuang, Y, Cantor, CR, Walt, DR, and Collins, JJ (2006). Phenotypic consequences of promoter-mediated transcriptional noise. *Molecular cell* 24.6, 853–865.

Bloom, J, Khan, Z, Kruglyak, L, Singh, M, and Caudy, A (2009). Measuring differential gene expression by short read sequencing: quantitative comparison to 2-channel gene expression microarrays. *BMC Genomics* 10.1, 221.

Browne, WJ, Subramanian, SV, Jones, K, and Goldstein, H (2005). Variance partitioning in multilevel logistic models that exhibit overdispersion. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 168.3, 599–613.

Buettner, F, Natarajan, KN, Casale, FP, Proserpio, V, Scialdone, A, Theis, FJ, Teichmann, SA, Marioni, JC, and Stegle, O (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology* 33.2, 155–160.

Cao, A and Galanello, R (2010). Beta-thalassemia. *Genet Med* 12.2, 61–76. ISSN: 1098-3600.

Carlin, BP and Chib, S (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 473–484.

Cartegni, L and Krainer, AR (2002). Disruption of an SF2/ASF-dependent exonic splicing enhancer in SMN2 causes spinal muscular atrophy in the absence of SMN1. *Nature genetics* 30.4, 377–384.

Cho, S and Dreyfuss, G (2010). A degron created by SMN2 exon 7 skipping is a principal contributor to spinal muscular atrophy severity. *Genes & Development* 24.5, 438–442.

Chong, S, Chen, C, Ge, H, and Xie, XS (2014). Mechanism of transcriptional bursting in bacteria. *Cell* 158.2, 314–326.

Chubb, JR, Trcek, T, Shenoy, SM, and Singer, RH (2006). Transcriptional pulsing of a developmental gene. *Current biology* 16.10, 1018–1025.

Conesa, A, Madrigal, P, Tarazona, S, Gomez-Cabrero, D, Cervera, A, McPherson, A, Szcześniak, MW, Gaffney, DJ, Elo, LL, Zhang, X, et al. (2016). A survey of best practices for RNA-seq data analysis. *Genome biology* 17.1, 13.

Crick, F (1979). Split genes and RNA splicing. *Science* 204.4390, 264–271.

Ding, B, Zheng, L, Zhu, Y, Li, N, Jia, H, Ai, R, Wildberg, A, and Wang, W (2015). Normalization and noise reduction for single cell RNA-seq experiments. *Bioinformatics*, btv122.

Eberwine, J, Sul, J-Y, Bartfai, T, and Kim, J (2014). The promise of single-cell sequencing. *Nature methods* 11.1, 25–27.

Emmert-Buck, MR, Bonner, RF, Smith, PD, Chuaqui, RF, et al. (1996). Laser capture microdissection. *Science* 274.5289, 998.

Femino, AM, Fay, FS, Fogarty, K, and Singer, RH (1998). Visualization of single RNA transcripts in situ. *Science* 280.5363, 585–590.

Ferrari, S and Cribari-Neto, F (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics* 31.7, 799–815.

Fiering, S, Northrop, JP, Nolan, GP, Mattila, PS, Crabtree, GR, and Herzenberg, LA (1990). Single cell assay of a transcription factor reveals a threshold in transcription activated by signals emanating from the T-cell antigen receptor. *Genes & Development* 4.10, 1823–1834.

Finak, G, McDavid, A, Yajima, M, Deng, J, Gersuk, V, Shalek, AK, Slichter, CK, Miller, HW, McElrath, MJ, Prlic, M, et al. (2015). MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome biology* 16.1, 278.

Fukaya, T, Lim, B, and Levine, M (2016). Enhancer control of transcriptional bursting. *Cell* 166.2, 358–368.

Gole, J, Gore, A, Richards, A, Chiu, Y-J, Fung, H-L, Bushman, D, Chiang, H-I, Chun, J, Lo, Y-H, and Zhang, K (2013). Massively parallel polymerase cloning and genome sequencing of single cells using nanoliter microwells. *Nature biotechnology* 31.12, 1126–1132.

Greenwood, CM, Midgley, JP, Matthew, AG, and Logan, AG (1999). Statistical issues in a metaregression analysis of randomized trials: impact on the dietary sodium intake and blood pressure relationship. *Biometrics* 55.2, 630–636.

Griebel, T, Zacher, B, Ribeca, P, Raineri, E, Lacroix, V, Guigó, R, and Sammeth, M (2012). Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic acids research* 40.20, 10073–10083.

Grün, D and Oudenaarden, A van (2015). Design and analysis of single-cell sequencing experiments. *Cell* 163.4, 799–810.

Hahn, CN and Scott, HS (2012). Spliceosome mutations in hematopoietic malignancies. *Nature genetics* 44.1, 9–10.

Ham, RG (1965). Clonal growth of mammalian cells in a chemically defined, synthetic medium. *Proceedings of the National Academy of Sciences* 53.2, 288–293.

Han, Y, Gao, S, Muegge, K, Zhang, W, and Zhou, B (2015). Advanced applications of RNA sequencing and challenges. *Bioinformatics and biology insights* 9.Suppl 1, 29.

Hicks, SC, Teng, M, and Irizarry, RA (2015). On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data. *bioRxiv*, 025528.

Higgins, J and Thompson, SG (2004). Controlling the risk of spurious findings from meta-regression. *Statistics in medicine* 23.11, 1663–1682.

Hu, Y, Liu, Y, Mao, X, Jia, C, Ferguson, JF, Xue, C, Reilly, MP, Li, H, and Li, M (2013). PennSeq: accurate isoform-specific gene expression quantification in RNA-Seq by modeling non-uniform read distribution. *Nucleic acids research*, gkt1304.

Hu, Y, Liu, Y, Mao, X, Jia, C, Ferguson, JF, Xue, C, Reilly, MP, Li, H, and Li, M (2014). PennSeq: accurate isoform-specific gene expression quantification in RNA-Seq by modeling non-uniform read distribution. *Nucleic acids research* 42.3, e20–e20.

Huizenga, HM, Visser, I, and Dolan, CV (2011a). Testing overall and moderator effects in random effects meta-regression. *British Journal of Mathematical and Statistical Psychology* 64.1, 1–19.

Huizenga, HM, Visser, I, and Dolan, CV (2011b). Testing overall and moderator effects in random effects meta-regression. *British Journal of Mathematical and Statistical Psychology* 64.1, 1–19.

Imielinski, M, Berger, AH, Hammerman, PS, Hernandez, B, Pugh, TJ, Hodis, E, Cho, J, Suh, J, Capelletti, M, Sivachenko, A, et al. (2012). Mapping the hallmarks of lung adenocarcinoma with massively parallel sequencing. *Cell* 150.6, 1107–1120.

Islam, S, Zeisel, A, Joost, S, La Manno, G, Zajac, P, Kasper, M, Lönnerberg, P, and Linnarsson, S (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature methods* 11.2, 163–166.

Jia, C, Guan, W, Yang, A, Xiao, R, Tang, W, Moravec, C, Margulies, K, Cappola, T, Li, C, and Mingyaos, L (2015). MetaDiff: Differential Isoform Expression Analysis using Random-Effects Meta-Regression. *BMC Bioinformatics*.

Jia, C, Kelly, D, Kim, J, Li, M, and Zhang, N (2017). Accounting for technical noise in single-cell RNA sequencing analysis. *bioRxiv*, 116939.

Kashima, T and Manley, JL (2003). A negative element in SMN2 exon 7 inhibits splicing in spinal muscular atrophy. *Nature genetics* 34.4, 460–463.

Katayama, S, Töhönen, V, Linnarsson, S, and Kere, J (2013). SAMstrt: statistical test for differential expression in single-cell transcriptome with spike-in normalization. *Bioinformatics* 29.22, 2943–2945.

Katz, Y, Wang, ET, Airoldi, EM, and Burge, CB (2010). Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nature methods* 7.12, 1009–1015.

Kharchenko, PV, Silberstein, L, and Scadden, DT (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods* 11.7, 740–742.

Kim, HJ, Kim, NC, Wang, Y-D, Scarborough, EA, Moore, J, Diaz, Z, MacLea, KS, Freibaum, B, Li, S, Molliex, A, et al. (2013). Mutations in prion-like domains in hnRNPA2B1 and hnRNPA1 cause multisystem proteinopathy and ALS. *Nature* 495.7442, 467–473.

Kim, JK and Marioni, JC (2013). Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome biology* 14.1, R7.

Kim, JK, Kolodziejczyk, AA, Illicic, T, Teichmann, SA, and Marioni, JC (2015). Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature communications* 6.

Kim, M-S, Pinto, SM, Getnet, D, Nirujogi, RS, Manda, SS, Chaerkady, R, Madugundu, AK, Kelkar, DS, Isserlin, R, Jain, S, et al. (2014). A draft map of the human proteome. *Nature* 509.7502, 575–581.

Kolodziejczyk, AA, Kim, JK, Svensson, V, Marioni, JC, and Teichmann, SA (2015). The technology and biology of single-cell RNA sequencing. *Molecular cell* 58.4, 610–620.

Korthauer, KD, Chu, L-F, Newton, MA, Li, Y, Thomson, J, Stewart, R, and Kendziorski, C (2016). A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biology* 17.1, 222.

Kudo, A (1963). A multivariate analogue of the one-sided test. *Biometrika* 50.3/4, 403–418.

Lander, ES (2011). Initial impact of the sequencing of the human genome. *Nature* 470.7333, 187–197.

Landry, ZC, Giovanonni, SJ, Quake, SR, and Blainey, PC (2013). Optofluidic cell selection from complex microbial communities for single-genome analysis. *Methods in enzymology* 531.

Lappalainen, T, Sammeth, M, Friedländer, MR, ACt Hoen, P, Monlong, J, Rivas, MA, Gonzàlez-Porta, M, Kurbatova, N, Griebel, T, Ferreira, PG, et al. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* 501.7468, 506–511.

Leng, N, Dawson, JA, Thomson, JA, Ruotti, V, Rissman, AI, Smits, BMG, Haag, JD, Gould, MN, Stewart, RM, and Kendziorski, C (2013). EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics* 29.8, 1035–1043.

Li, B and Dewey, CN (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics* 12.1, 323.

Li, J and Tibshirani, R (2013). Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. *Statistical methods in medical research* 22.5, 519–536.

Love, M, Huber, W, and Anders, S (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15.12, 550.

Lun, AT, Bach, K, and Marioni, JC (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology* 17.1, 75.

Macosko, EZ, Basu, A, Satija, R, Nemesh, J, Shekhar, K, Goldman, M, Tirosh, I, Bialas, AR, Kamitaki, N, Martersteck, EM, et al. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 161.5, 1202–1214.

Marioni, J, Mason, C, Mane, S, Stephens, M, and Gilad, Y (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res* 18.9, 1509–1517.

Martin, KC and Ephrussi, A (2009). mRNA localization: gene expression in the spatial dimension. *Cell* 136.4, 719–730.

McCarthy, DJ, Chen, Y, and Smyth, GK (2012). Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic acids research*, gks042.

Montgomery, SB, Sammeth, M, Gutierrez-Arcelus, M, Lach, RP, Ingle, C, Nisbett, J, Guigo, R, and Dermitzakis, ET (2010). Transcriptome genetics using second generation sequencing in a Caucasian population. *Nature* 464.7289, 773–777.

Navin, N, Kendall, J, Troge, J, Andrews, P, Rodgers, L, McIndoo, J, Cook, K, Stepansky, A, Levy, D, Esposito, D, et al. (2011). Tumour evolution inferred by single-cell sequencing. *Nature* 472.7341, 90–94.

Niu, L, Huang, W, Umbach, DM, and Li, L (2014). IUTA: a tool for effectively detecting differential isoform usage from RNA-Seq data. *BMC genomics* 15.1, 862.

Padovan-Merhar, O, Nair, GP, Biaesch, AG, Mayer, A, Scarfone, S, Foley, SW, Wu, AR, Churchman, LS, Singh, A, and Raj, A (2015). Single mammalian cells compensate for differences in cellular volume and DNA copy number through independent global transcriptional mechanisms. *Molecular cell* 58.2, 339–352.

Pickrell, JK, Marioni, JC, Pai, AA, Degner, JF, Engelhardt, BE, Nkadori, E, Veyrieras, J-B, Stephens, M, Gilad, Y, and Pritchard, JK (2010). Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature* 464.7289, 768–772.

Pierson, E and Yau, C (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology* 16.1, 241.

Raj, A, Peskin, CS, Tranchina, D, Vargas, DY, and Tyagi, S (2006). Stochastic mRNA synthesis in mammalian cells. *PLoS Biol* 4.10, e309.

Robinson, MD and Smyth, GK (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics* 9.2, 321–332.

Sandberg, R (2014). Entering the era of single-cell transcriptomics in biology and medicine. *Nature methods* 11.1, 22.

Santangelo, PJ, Lifland, AW, Curt, P, Sasaki, Y, Bassell, GJ, Lindquist, ME, and Crowe, JE (2009). Single molecule–sensitive probes for imaging RNA in live cells. *Nature methods* 6.5, 347–349.

Self, SG and Liang, K-Y (1987). Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82.398, 605–610.

Shalek, AK, Satija, R, Adiconis, X, Gertner, RS, Gaublomme, JT, Raychowdhury, R, Schwartz, S, Yosef, N, Malboeuf, C, Lu, D, et al. (2013). Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature* 498.7453, 236–240.

Shalek, AK, Satija, R, Shuga, J, Trombetta, JJ, Gennert, D, Lu, D, Chen, P, Gertner, RS, Gaublomme, JT, Yosef, N, et al. (2014). Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* 510.7505, 363–369.

Shen, S, Park, JW, Huang, J, Dittmar, KA, Lu, Z-x, Zhou, Q, Carstens, RP, and Xing, Y (2012). MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic acids research*, gkr1291.

Shen, S, Park, JW, Lu, Z-x, Lin, L, Henry, MD, Wu, YN, Zhou, Q, and Xing, Y (2014). rMATS: Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proceedings of the National Academy of Sciences* 111.51, E5593–E5601.

Shi, Y and Jiang, H (2013). rSeqDiff: Detecting differential isoform expression from RNA-Seq data using hierarchical likelihood ratio test. *PloS one* 8.11, e79448.

Stegle, O, Teichmann, SA, and Marioni, JC (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics* 16.3, 133–145.

Sugarbaker, DJ, Richards, WG, Gordon, GJ, Dong, L, De Rienzo, A, Maulik, G, Glickman, JN, Chirieac, LR, Hartman, M-L, Taillon, BE, et al. (2008). Transcriptome sequencing of malignant pleural mesothelioma tumors. *Proceedings of the National Academy of Sciences* 105.9, 3521–3526.

Susko, E (2013). Likelihood ratio tests with boundary constraints using data-dependent degrees of freedom. *Biometrika* 100.4.

Suter, DM, Molina, N, Gatfield, D, Schneider, K, Schibler, U, and Naef, F (2011). Mammalian genes are transcribed with widely different bursting kinetics. *Science* 332.6028, 472–474.

Tang, F, Barbacioru, C, Wang, Y, Nordman, E, Lee, C, Xu, N, Wang, X, Bodeau, J, Tuch, BB, Siddiqui, A, et al. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods* 6.5, 377–382.

Tirosh, I, Izar, B, Prakadan, SM, Wadsworth, MH, Treacy, D, Trombetta, JJ, Rotem, A, Rodman, C, Lian, C, Murphy, G, et al. (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* 352.6282, 189–196.

Torres, TT, Metta, M, Ottenwälder, B, and Schlötterer, C (2008). Gene expression profiling by massively parallel sequencing. *Genome research* 18.1, 172–177.

Trapnell, C, Williams, BA, Pertea, G, Mortazavi, A, Kwan, G, Baren, MJ van, Salzberg, SL, Wold, BJ, and Pachter, L (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* 28.5, 511–515.

Trapnell, C, Roberts, A, Goff, L, Pertea, G, Kim, D, Kelley, DR, Pimentel, H, Salzberg, SL, Rinn, JL, and Pachter, L (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols* 7.3, 562–578.

Treisman, R, Orkinl, SH, and Maniatis, T (1983). Specific transcription and RNA splicing defects in five cloned B-thalassaemia genes. *Nature* 302, 14.

Treisman, R, Proudfoot, NJ, Shander, M, and Maniatis, T (1982). A single-base change at a splice site in a β 0-thalassemic gene causes abnormal RNA splicing. *Cell* 29.3, 903–911.

Turro, E, Astle, WJ, and Tavaré, S (2014). Flexible analysis of RNA-seq data using mixed effects models. *Bioinformatics* 30.2, 180–188.

Tyagi, S (2009). Imaging intracellular RNA distribution and dynamics in living cells. *natuRe methods* 6.5, 331–338.

Vallejos, CA, Marioni, JC, and Richardson, S (2015). BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS Comput Biol* 11.6, e1004333.

Wang, L, Feng, Z, Wang, X, Wang, X, and Zhang, X (2010). DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics* 26.1, 136–138.

Wang, W, Qin, Z, Feng, Z, Wang, X, and Zhang, X (2013). Identifying differentially spliced genes from two groups of RNA-seq samples. *Gene* 518.1. Proceedings of the 23rd International Conference on Genome Informatics (GIW 2012) Proceedings of the 23rd International Conference on Genome Informatics (GIW 2012), 164 –170. ISSN: 0378-1119.

Weber, AP, Weber, KL, Carr, K, Wilkerson, C, and Ohlrogge, JB (2007). Sampling the Arabidopsis transcriptome with massively parallel pyrosequencing. *Plant physiology* 144.1, 32–42.

White, AK, VanInsberghe, M, Petriv, I, Hamidi, M, Sikorski, D, Marra, MA, Piret, J, Aparicio, S, and Hansen, CL (2011). High-throughput microfluidic single-cell RT-qPCR. *Proceedings of the National Academy of Sciences* 108.34, 13999–14004.

Wu, H, Wang, C, and Wu, Z (2013). A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. *Biostatistics* 14.2, 232–243.

Zeisel, A, Muñoz-Manchado, AB, Codeluppi, S, Lönnerberg, P, La Manno, G, Juréus, A, Marques, S, Munguba, H, He, L, Betsholtz, C, et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347.6226, 1138–1142.

Zenklusen, D, Larson, DR, and Singer, RH (2008). Single-RNA counting reveals alternative modes of gene expression in yeast. *Nature structural & molecular biology* 15.12, 1263–1271.

Zhao, K, Lu, Z-x, Park, JW, Zhou, Q, and Xing, Y (2013). GLiMMPS: robust statistical model for regulatory variation of alternative splicing using RNA-seq data. *Genome biology* 14.7, R74.

Zhou, Y-H, Xia, K, and Wright, FA (2011). A powerful and flexible approach to the analysis of RNA sequence count data. *Bioinformatics* 27.19, 2672–2678.

Zong, C, Lu, S, Chapman, AR, and Xie, XS (2012). Genome-wide detection of single-nucleotide and copy-number variations of a single human cell. *Science* 338.6114, 1622–1626.