STATISTICAL METHODS DEVELOPMENT FOR THE ANALYSIS OF SINGLE CELL RNA-seq Data

by

Xiuyu Ma

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2020

Date of final oral examination: 05/04/2020

The dissertation is approved by the following members of the Final Oral Committee:

Michael A. Newton, Professor, Statistics, Biostatistics and Medical Informatics
Christina Kendziorski, Professor, Biostatistics and Medical Informatics
Bret Larget, Professor, Statistics, Botany
Anru Zhang, Assistant Professor, Statistics
Qiongshi Lu, Assistant Professor, Biostatistics and Medical Informatics

Contents

List of Tables

List of Figures

Acknowledgments

My principal thanks are due to my advisor Prof. Michael A. Newton. The genesis of this dissertation owes much to Prof. Newton's idea. Intriguing as the problem appear, their solution entails numerous challenges,many of which are unseen in the literature. Throughout my years as doctoral student, Prof. Newton has lent invaluable assistance to my efforts in solving these problems. Further, his great tastes in research topics and deep thoughts in statistical methods benefit me a lot during my stay in Madison. He is also very supportive on students' decisions and generous to give advice. It is my honor and pleasure to work with Prof. Newton during my graduate study.

I would like to thank Prof. Christina Kendziorski, who is one of the committee members, for giving me a lot of useful suggestions for improving and better presenting the methodologies.

I would also express my thanks to other committee members, Prof. Bret Larget, Prof. Anru Zhang and Prof. Qiongshi Lu, for sharing valuable insights on this research.

A debt of gratitude is also owed to my parents for their ongoing support. I also thank my girlfriend Jiyuan Fang for her unwavering support.

Abstract

Single-cell analysis is a rapidly evolving approach to characterize genome-wide gene expression at the individual cell level. Overcoming unique variational structure underlying the data and studying cellular heterogeneity require statistical tools. In this dissertation, I develop and improve statistical methods focus on identifying genes with differential distributions across conditions. The first method uses a compositional structure which explicitly accounts for the cellular subtypes to characterize gene expression as a mixture over subtypes and quantify the distributional change between conditions. We also extend the distributional comparison to more than two conditions.
The second method accelerates the inference for patterns of how means are varied among multiple groups. It scales up the first method when more mixing components are considered.

The first method, called scDDboost, introduces an empirical Bayesian mixture approach and leverages cell-subtype structure revealed in cluster analysis in order to boost gene-level information on expression changes. Cell clustering informs gene-level analysis through a specially-constructed prior distribution over pairs of multinomial probability vectors; this prior meshes with available model-based tools that score patterns of differential expression over multiple subtypes. We derive an explicit formula for the posterior probability that a gene has the same distribution in two cellular conditions, allowing for a gene-specific mixture over subtypes in each condition. Advantage is gained by the compositional structure of the model, in which a host of gene-specific mixture components are allowed, but also in which the mixing proportions are constrained at the whole cell level. This structure leads to a novel form of information sharing through which the cell-clustering results support gene-level scoring of differential distribution. The result, according to our numerical

experiments, is improved sensitivity compared to several standard approaches for detecting distributional expression changes. The compositional model has great flexibility and we further extend it to more than two conditions.

The second method called EBSeq.v2 accelerates a widely used package EBSeq. The number of patterns for equivalent/differential means among groups grows fast with the number of groups. It introduces challenge for memory and computation. We provide a pruning algorithm to eliminates unlikely patterns that we can assess through preliminary checks over local Bayes factors. Further improvements are gained through a more efficient one-step EM for hyperparameters optimization and codes implementation in C++.

Chapter 1

Introduction

Statistical methods are widely used in genomics and molecular biology because they provide a reasoned approach to describing the complicated patterns of variation in experimental data. High-dimensional data are now routinely measured on cells from various biological systems, and such data expose limitations of existing statistical methodology. Comparative statistical tests, such as Student's t-test, when applied to each dimension of high-dimensional data tend to be underpowered, as they overlook collective properties of the system. Model-based statistical procedures provide one approach to improve operating characteristics, and this fact has guided work presented in this thesis. We are focused on the analysis of so-called RNA-Seq data. Such data aim to measure the abundance in biological samples of various molecular species of ribonucleic acid (RNA). These molecules are critical mediators of genetic information, and their measurement through modern sequencing technology has become a central to all sorts of biomedical investigations, from basic to translational to clinical studies. Single-cell RNA-seq(scRNA-seq) is a revolutionary tool to measure genome-scale transcripts at the individual cell level. The scRNA-seq data provides higher resolution that measures distribution of expression levels for each gene across a population of cells than the traditional profiling method, bulk RNA-seq, which measures average expression level for each gene. Such increased resolution demonstrates heterogeneity between cells and allows researchers to uncover new and potentially unexpected biological discoveries. With the development of technology, a wide variety of large-scale scRNA-seq approaches have been developed (Svensson et al. (2018)) to study

the biological systems more thoroughly. Because of the scale and the unique variation structure underlying the scRNA-seq data, statistical and computational challenges must be addressed to prevent inaccurate conclusions and to optimize novel discovery. One general problem is to compare transcripts between cells from different conditions and identifying those differences provides valuable insights into the complex biological systems, ranging from cancer genomics to diverse microbial communities (Hwang et al., 2018). This thesis focus on this general problem and presents three contributions that address some challenges in the single-cell data.

1.1 Differential Distribution Testing

From bulk RNA-seq data, interest lies in comparing average expression levels between different biological conditions. The differential expression test can be used to identify genes with certain properties. For scRNA-seq data, testing for difference remains important, but it is not sufficient to describe the difference only in terms of mean. scRNA-seq data presents different characteristics that requires a new definition for differential expression analysis. For example, due to the small number and low capture efficiency of RNA molecules in single cells (Saliba et al., 2014), many transcripts tend to be missed during the reverse transcription. As a result, we may observe that some transcripts are highly expressed in one cell but are missed in another cell. This phenomenon is defined as a "drop-out" event, which results a high prevalence of zero in the data. Also recent studies have shown that gene expression in a single cell is a stochastic process and that gene expression values in different cells are heterogeneous (Elowitz et al., 2002), which results in the gene-level multimodality. Those characteristics brought about more subtle changes for transcripts rather than change of average expression, such as differential proportion and differential modality, which are demonstrated to be biological meaningful.(Korthauer et al., 2016a, Wang et al., 2019).

Previous widely used methods for bulk data, such as `DESeq` and `EdgeR`, can be applied to scRNA-seq data but they do not consider the underlying characteristic. New statistical methods, such as `MAST` (Finak et al., 2015), `D3E` (Delmans and Hemberg, 2016) and `scDD` (Korthauer et al.,

2016a) are developed to account for those characteristics, but they are gene-specific tests that the information from other genes are not used when the tests are performed for one gene. Potentially, the sensitivity of testing for difference can be improved by allowing information sharing among genes.

The first contribution provides a test for distributional change of transcripts to capture those subtle differences between conditions. Specifically, we use an empirical Bayesian mixture approach to score genes for evidence of distributional changes, which we call it `scDDBoost`. We leverage the fact that cells populate distinct, identifiable subtypes determined by lineage history, epigenetic state, the activity of various transcriptional programs, or other distinguishing factors. Such subtype information may be injected into the transcripts profile for genome and can be inferred via other clustering methods, for example, `SC3` (Kiselev et al. (2017)), `CIDR` (Lin et al. (2017)) and `ZIFA` (Pierson and Yau (2015)). A gene's marginal distribution is modeled as mixture of cells with mixing structure determined by the subtype label. The mixture model is flexible and approximates a wide variety of distributions well, further, it accounts for other characteristics of single-cells data, e.g. high prevalence of zero counts and gene-level multimodality. The compositional model underlying `scDDBoost` provides a novel way of information sharing, as the celluar subtypes, which are inferred by clustering utilizing the expression data from whole genome, inform the analysis of gene-level expression. After identification of subtypes, our method combine two parts of computation, first is to quantify how the proportions of subtypes varies across conditions. The second is to quantify how the mean expression varies across subtypes, which faces a computational bottleneck. We address the issue in our next contribution. We show that `scDDBoost` increases power of testing under control of false discovery rate through simulation and empirical studies. More details can be found in chapter 2.

1.2 Scalability

The second contribution is to improve `EBSeq` (Leng et al., 2013), a method of inference for multi-group differential expression analysis. Specifically, given K groups of expression data, an empirical Bayes mixture model is applied to infer the patterns of how those means varies among groups for each gene. It has proven useful in many studies, including studies of development, viral transcription and cancer (Louro et al., 2020, Newhouse et al., 2017, Lee et al., 2020). `EBSeq` is most often used when $K = 2$, the multi-group feature is less frequently deployed, however it serves as a key component for `scDDBoost`, where in general $K \geq 3$ are expected, to characterize the difference between cellular subtypes that is used for analysis of distributional change. With the advance in large-scale sequencing technology, more cells are produced and likely to introduce big K. A computational bottleneck arises if we consider all possible patterns, which are equal in number to the Bell number of partitions of K objects (Gardner, 1978). For even moderate K, the memory and time costs of `EBSeq` become excessive. We first provide a pruning algorithm, where we identify those patterns that are negligible for the final inference through some preliminary check and remove them from the compute-intensive part of the code. Secondly, we apply a more efficient optimization scheme for the parameters and implement the codes in C++. We use numerical studies to demonstrate our implementation greatly improve the performance of `EBSeq`. More details can be found in Chapter 3.

1.3 Multiple Conditions Differential Distribution Testing

In order to get better understanding of some biological process, scientist increased the number of experiments and profiled transcripts over time to capture the underlying dynamics. These so-called time-course scRNA-seq is useful in understanding differences in differentiation maturity (Trapnell et al. (2014a), Qiu et al. (2017)). The time-course data after some pre-processing, for example, trajectory inference method is typically applied to organize the cells into a pseudotemporal ordering that is concordant with the development trajectory (Trapnell et al., 2014b), are then used to discover

genes that are associated with the lineages in the trajectory, or presenting difference between lineages, to illuminate the underlying biological processes.

The third contribution is to generalize the comparison for distributional change to more than two conditions. We provides a framework allowing testing for change of distributions along the process. Assuming we have data collected at T conditions(timepoints) labelled as $1, ..., T$ and the corresponding marginal distribution of a gene g are $f_g^1, ..., f_g^T$. Adopting the mixture modeling idea from `scDDboost`, marginal distribution f_g^i is determined by how cellular subtypes are mixed at condition i. We extend the framework of `scDDboost` to quantify changes of subtypes proportions over time and score the intensity of each pattern of distributional change. More details are provided in Chapter 4.

It is important to continue improving and developing statistical methods to study and test the biological signal especially as sequencing technologies continue to progress and experiments become more complex.

Chapter 2

A Compositional Model to Assess Expression Changes From Single-Cell RNA-Seq Data[1]

2.1 Background

The ability to measure genome-wide gene expression at single-cell resolution has accelerated the pace of biological discovery. Overcoming data analysis challenges caused by the scale and unique variation properties of single-cell data will surely fuel further advances in immunology (Papalexi and Satija (2017)), developmental biology (Marioni and Arendt (2017)), cancer (Navin (2015)), and other areas (Nawy (2013)). Computational tools and statistical methodologies created for data of lower-resolution (e.g., bulk RNA-seq) or lower dimension (e.g., flow cytometry) guide our response to the data-science demands of new measurement platforms, but they remain inadequate for efficient knowledge discovery in this rapidly advancing domain (Bacher and Kendziorski (2016)).

An important feature of single-cell studies that could be leveraged better statistically is the fact that cells populate distinct, identifiable subtypes determined by lineage history, epigenetic state, the

[1]This chapter is a reformated version of Technical Report 655795 at bioRxiv, written jointly with co-authors Drs. C. Kendziorski, K. Korthauer, and M.A. Newton. We are currently revising it for the Annals of Applied Statistics.

activity of various transcriptional programs, or other distinguishing factors. Extensive research on clustering cells has produced tools for identifying subtypes, including `SC3` (Kiselev et al. (2017)), `CIDR` (Lin et al. (2017)) and `ZIFA` (Pierson and Yau (2015)). We hypothesize that such subtype information may be usefully utilized in other inference procedures in order to improve their operating characteristics.

Assessing the magnitude and statistical significance of changes in gene expression associated with changes in cellular condition has been a central statistical problem in genomics. New tools specific to the single-cell RNA-seq data structure, including `MAST` (Finak et al. (2015)), `scDD` (Korthauer et al. (2016b)), and `D3E` (Delmans and Hemberg (2016)), have been deployed to address this problem. These tools respond to scRNA-seq characteristics, such as high prevalence of zero counts and gene-level multimodality, but they do not fully exploit cellular-subtype information. Our proposed method measures changes in a gene's marginal mixture distribution and acquires sensitivity to a variety of distributional effects by how it integrates gene-level data with estimated cellular subtypes. It is implemented in software in the R package `scDDboost` [2].

Through the compositional model underlying `scDDboost`, subtypes inferred by clustering inform the analysis of gene-level expression. The proposed methodology merges two lines of computation after cell clustering: one concerns patterns of differential expression among the cellular subtypes, and here we take advantage of the powerful `EBseq` method for detecting patterns in negative-binomially-distributed expression data (Leng et al. (2015)). The second concerns the counts of cells in various subtypes; for this we propose a Double-Dirichlet-Mixture distribution to model the pair of multinomial probability vectors for subtype counts in two experimental conditions. Further elements are developed, on the selection of the number of subtypes and on accounting for uncertainty in the cluster output, in order to provide an end-to-end solution to the differential distribution problem. We note that modularity in the necessary elements provides some methodological advantages. For example, improvements in clustering may be used in place of the default clustering without altering the form of downstream analysis. Also, by avoiding Markov chain Monte Carlo, `scDDboost` computations are relatively inexpensive for a Bayesian procedure.

[2] http://github.com/wiscstatman/scDDboost/

To set the context by way of example, Figure 2.1 highlights the ability of `scDDboost` to sense subtype composition changes and thus detect subtle gene expression changes between conditions. The three panels on the left compare expression from 91 human stem cells known to be in the G1 phase of the cell cycle, as well as from 76 such cells known to be in the G2/M phase (Leng et al. (2013)) in three genes (BIRC5, HMMR, and CKAP2), which we happen to know from prior studies have differential activity between G1 and G2/M (Li and Altieri (1999), Sohr and Engeland (2008), Dominguez et al. (2016)). Several standard statistical tools applied to the data behind Figure 2.1 do not find the observed differences in any of these genes to be statistically significant when controlling the false discovery rate (FDR) at 5%, but `scDDboost` does include these genes on its 5% FDR list. Considering prior studies, these subtle distributional changes are probably not false discoveries. The right panel in Figure 2.1 shows these three among many other genes also known to be involved in cell-cycle regulation but not identified by standard tools as altered between G1 and G2/M at the 5% FDR level. The color panel provides insight into why `scDDboost` has identified these genes. For this data set, six cellular subtypes were identified in the first step of `scDDboost` (colors red, blue, green, and orange are visible). These subtypes have changed in their proportions between G1 and G2/M; there is a lower proportion of red cells and a greater proportion of orange cells in G2/M, for example. These proportion shifts, which are inferred from genome-wide data, stabilize gene-specific statistics that measure changes between conditions in the mixture distribution of expression, and thereby increase power. We note that `scDDboost` agrees with other statistical tools on very strong differential-distribution signals (not shown), but it has the potential to increase power for subtle signals owing to its unique approach to leveraging cell subtype information.

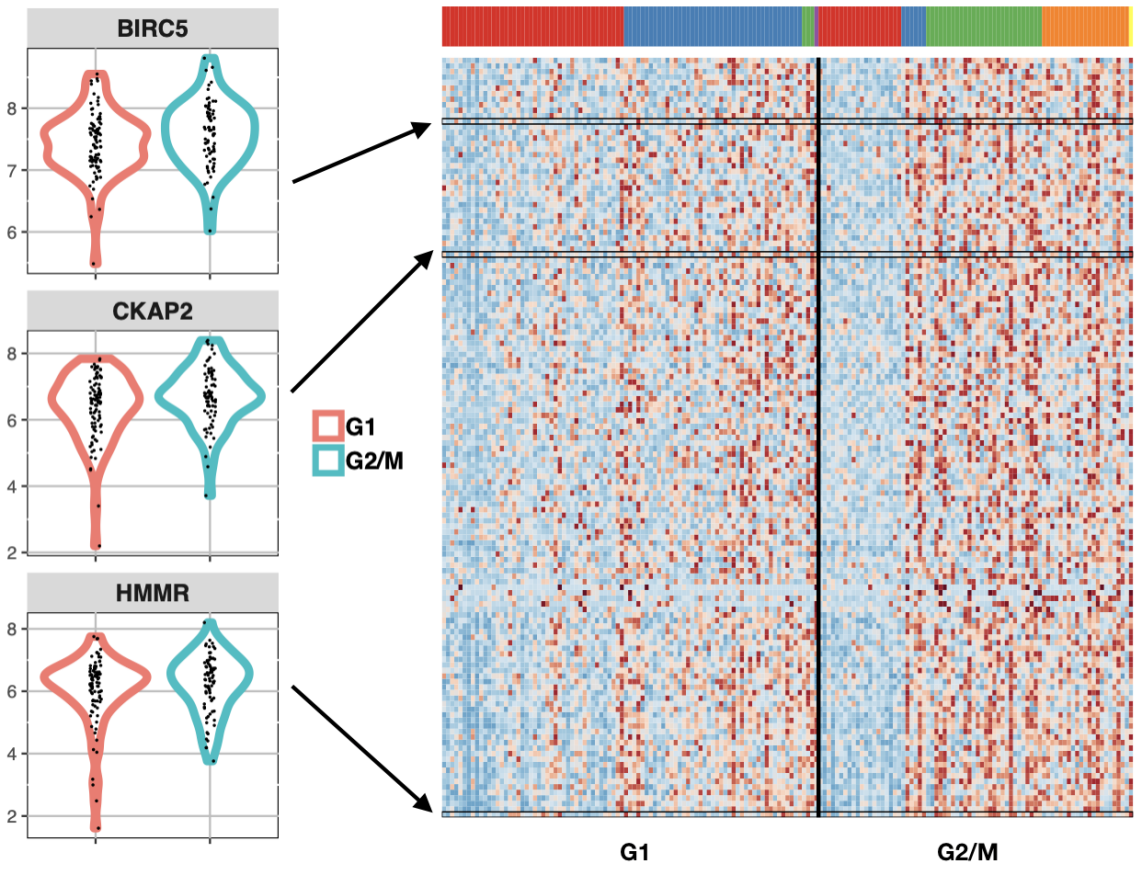


Figure 2.1: Genes involved in cell-cycle that are identified by scDDboost, but not standard approaches, as differentially distributed between cell-cycle phases G1 and G2/M in human embryonic stem cells. Density estimates on the left show expression data (log2 scale) of three genes identified by scDDboost at 5% FDR, but not similarly identified by MAST, scDD, and DESeq2. Prior studies have shown that the expression of BIRC5, CKAP2, and HMMR is dependent on the phase of cell-cycle, suggesting that these subtle shifts are not false positives. Heatmap (right) shows these three genes among 137 other cell-cycle genes (GO:0007049) identified exclusively by scDDboost, with expression from low (blue) to high (red). Cells (columns) are clustered by their genome-wide expression profiles into distinct cellular subtypes, as indicated by the color panel.

Numerical experiments on both synthetic and published scRNA-seq data bear out the incidental finding in Figure 2.1, that `scDDboost` has sensitivity for detecting subtle distribution changes. In these experiments we take advantage of `splatter` for generating synthetic data (Zappia et al. (2017)) as well as the compendium of scRNA-seq data available through `conquer` (Soneson and Robinson (2017)). Additional numerical experiments show a relationship between `scDDboost` findings and more mechanistic attempts to parameterize transcriptional activation (Delmans and

Hemberg (2016)). Finally, we establish first-order asymptotic results for the methodology.

2.2 Modeling

2.2.1 Data structure, sampling model, and parameters

In modeling scRNA-seq data, we imagine that each cell c falls into one of $K > 1$ classes, which we think of as subtypes or subpopulations of cells. For notation, $z_c = k$ means that cell c is of subtype k, with the vector $z = (z_c)$ recording the states of all sampled cells. Knowledge of this class structure prior to measurement is not required, as it will be inferred as necessary from available genomic data. We expect that cells arise from multiple experimental conditions, such as by treatment-control status or some other factors measured at the cell level, but we present our development for the special case of two conditions. Notationally, $y = (y_c)$ records the experimental condition, say $y_c = 1$ or $y_c = 2$. Let's say condition j measures $n_j = \sum_c 1[y_c = j]$ cells, and in total we have $n = n_1 + n_2$ cells in the analysis. The examples in Section 3 involve hundreds to thousands of cells. Further let

$$t_k^j = t_k^j(y, z) = \sum_c 1[y_c = j, z_c = k] \tag{2.1}$$

denote the number of cells of subtype k in condition j and $X_{g,c}$ denote the normalized expression of gene g in cell c. This is one entry in a typically large genes-by-cells data matrix X. Thus, the data structure entails an expression matrix X, a treatment label vector y, and a vector z of latent subtype labels.

We treat subtype counts in the two conditions, $t^1 = (t_1^1, t_2^1, \cdots, t_K^1)$ and $t^2 = (t_1^2, t_2^2, \cdots, t_K^2)$, as independent multinomial vectors, reflecting the experimental design. Explicitly,

$$t^1|y \sim \text{Multinomial}_K(n_1, \phi) \quad \text{and} \quad t^2|y \sim \text{Multinomial}_K(n_2, \psi) \tag{2.2}$$

for probability vectors $\phi = (\phi_1, \phi_2, \cdots, \phi_K)$ and $\psi = (\psi_1, \psi_2, \cdots, \psi_K)$ that characterize the populations of cells from which the n observed cells are sampled. This follows from the more basic sampling model: $P(z_c = k|y_c = 1) = \phi_k$ and $P(z_c = k|y_c = 2) = \psi_k$.

Our working hypothesis, referred to as the *compositional model*, is that any differences in the distribution of expression $X_{g,c}$ between $y_c = 1$ and $y_c = 2$ (i.e., any condition effects) are attributable to differences between the conditions in the underlying composition of cell types; i.e., owing to $\phi \neq \psi$. We suppose that cells of any given subtype k will present data according to a distribution reflecting technical and biological variation specific to that class of cells, regardless of the condition y_c of the cell. Some care is needed in this, as an overly broad cell subtype (e.g., *epithelial cells*) could have further subtypes that show differential response to some treatment, for example, and so cellular condition (treatment) would then affect the distribution of expression data within the subtype, which is contrary to our working hypothesis. Were that the case, we could have refined the subtype definition to allow a greater number of population classes K in order to mitigate the problem of within-subtype heterogeneity. A risk in this approach is that K could approach n, as if every cell were its own subtype. We find, however, that data sets often encountered do not display this theoretical phenomenon when considering a broad class of within-subtype expression distributions. We revisit the issue in Section 5, but for now, we proceed assuming that cellular condition affects the composition of subtypes but not the distribution of expression within a subtype.

Within the compositional model, let $f_{g,k}$ denote the sampling distribution of expression measurement $X_{g,c}$ assuming that cell c is from subtype k. Then for the two cellular conditions, and at some expression level x, the marginal distributions over subtypes are finite mixtures:

$$f_g^1(x) = \sum_{k=1}^{K} \phi_k f_{g,k}(x) \quad \text{and} \quad f_g^2(x) = \sum_{k=1}^{K} \psi_k f_{g,k}(x).$$

In other words, $X_{g,c}|[y_c = j] \sim f_g^j$ and $X_{g,c}|[z_c = k, y_c = j] \sim f_{g,k}$.

We say that gene g is *differentially distributed*, denoted DD_g and indicated by $f_g^1 \neq f_g^2$, if $f_g^1(x) \neq f_g^2(x)$ for some x, and otherwise it is equivalently distributed (ED_g). Motivated by findings from bulk RNA-seq data analysis, we further set each $f_{g,k}$ to have a a negative-binomial form, with mean $\mu_{g,k}$ and shape parameter σ_g, as in (Leng et al. (2013), Anders and Huber (2010), Love et al. (2014a) and Chen et al. (2018)). This choice is effective in our numerical experiments though it is not critical to the modeling formulation. The use of mixtures per gene has proven useful in related

model-based approaches (e.g., Finak et al. (2015); McDavid et al. (2014); Huang et al. (2018)).

We seek methodology to prioritize genes for evidence of DD_g. Interestingly, even if we have evidence for condition effects on the subtype frequencies, it does not follow that a given gene will have $f_g^1 \neq f_g^2$; that depends on whether or not the subtypes show the right pattern of *differential expression* at g, to use the standard terminology from bulk RNA-seq. For example, if two subtypes have different frequencies between the two conditions ($\phi_1 \neq \psi_1$ and $\phi_2 \neq \psi_2$) but the same aggregate frequency ($\phi_1 + \phi_2 = \psi_1 + \psi_2$), and also if $\mu_{g,1} = \mu_{g,2}$ then, other things being equal, $f_g^1 = f_g^2$ even though $\phi \neq \psi$. The fact is so central that we emphasize:

Key issue: A gene that does not distinguish two subtypes will also not distinguish the cellular conditions if those subtypes appear in the same aggregate frequency in the two conditions, regardless of changes in the individual subtype frequencies.

We formalize this issue in order that our methodology has the necessary functionality. To do so, first consider the parameter space $\Theta = \{\theta = (\phi, \psi, \mu, \sigma)\}$, where $\phi = (\phi_1, \phi_2, \cdots, \phi_K)$ and $\psi = (\psi_1, \psi_2, \cdots, \psi_K)$ are as before, where $\mu = \{\mu_{g,k}\}$ holds all the subtype-and-gene-specific expected values, and where $\sigma = \{\sigma_g\}$ holds all the gene-specific negative-binomial shape parameters. Critical to our construction are special subsets of Θ corresponding to partitions of the K cell subtypes. A single partition, π, is a set of mutually exclusive and exhaustive blocks, b, where each block is a subset of $\{1, 2, \cdots, K\}$, and we write $\pi = \{b\}$. Of course, the set Π containing all partitions π of $\{1, 2, \cdots, K\}$ has cardinality that grows rapidly with K. We carry along an example involving $K = 7$ cell types, and one three-block partition taken from the set of 877 possible partitions of $\{1, 2, \cdots, 7\}$ (Figure 3.2). For any partition $\pi = \{b\}$, consider aggregate subtype frequencies

$$\Phi_b = \sum_{k \in b} \phi_k \quad \text{and} \quad \Psi_b = \sum_{k \in b} \psi_k,$$

and extend the notation, allowing vectors $\Phi_\pi = \{\Phi_b : b \in \pi\}$ and similarly for Ψ_π. Recall the partial ordering of partitions based on refinement, and note that as long as π is not the most refined partition (every cell type is in its own block), then the mapping from (ϕ, ψ) to (Φ_π, Ψ_π) is many-

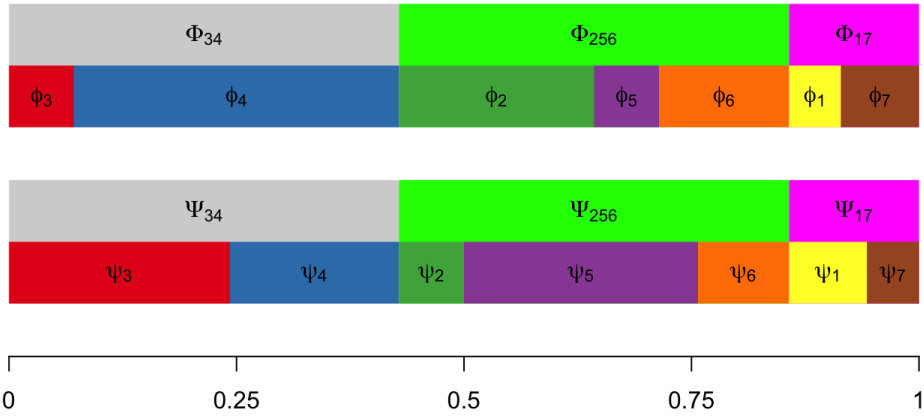


Figure 2.2: Proportions of $K = 7$ cellular subtypes in two different conditions. Aggregated proportions of subtypes 3 and 4, subtypes 2, 5, and 6, and subtypes 1, and 7 remain same across conditions, while individual subtype frequencies change. Depending on the changes in average expression among subtypes, these frequency changes may or may not induce changes between two conditions in the marginal distribution of some gene's expression.

to-one. Further, define sets

$$A_\pi = \{\theta \in \Theta : \ \Phi_b = \Psi_b \, \forall b \in \pi\}. \tag{2.3}$$

and

$$M_{g,\pi} = \{\theta \in \Theta : \ \mu_{g,k} = \mu_{g,k'} \iff k, k' \in b, b \in \pi\}. \tag{2.4}$$

Under A_π there are constraints on cell subtype frequencies; under $M_{g,\pi}$ there is equivalence in the gene-level distribution of expression between certain subtypes. These sets are precisely the structures needed to address differential distribution DD_g (and it complement, equivalent distribution, ED_g) at a given gene g, since:

Theorem 2.2.1. *Let $C_{g,\pi} = A_\pi \cap M_{g,\pi}$. For partitions $\pi_1 \neq \pi_2$, $C_{g,\pi_1} \cap C_{g,\pi_2} = \emptyset$. Further, at*

any gene g, equivalent distribution is

$$\mathrm{ED}_g = \bigcup_{\pi \in \Pi} C_{g,\pi}.$$

With additional probability structure on the parameter space, we immediately obtain from Theorem 1 a formula for local false discovery rates:

$$1 - P(\mathrm{DD}_g | X, y) = P(\mathrm{ED}_g | X, y) = \sum_{\pi \in \Pi} P\left(A_\pi \cap M_{g,\pi} | X, y\right). \tag{2.5}$$

Local false discovery rates are important empirical Bayesian statistics in large-scale testing (Efron (2007); Muralidharan (2010); Newton et al. (2004)). For example, the conditional false discovery rate of a list of genes is the arithmetic mean of the associated local false discovery rates. The partition representation guides the construction of a prior distribution (Section 2.3) and a model-based method (Section 2.2) for scoring differential distribution. Setting the stage, Figure 2.3 shows the dependency structure of the proposed compositional model and the partition-reliant prior specification.

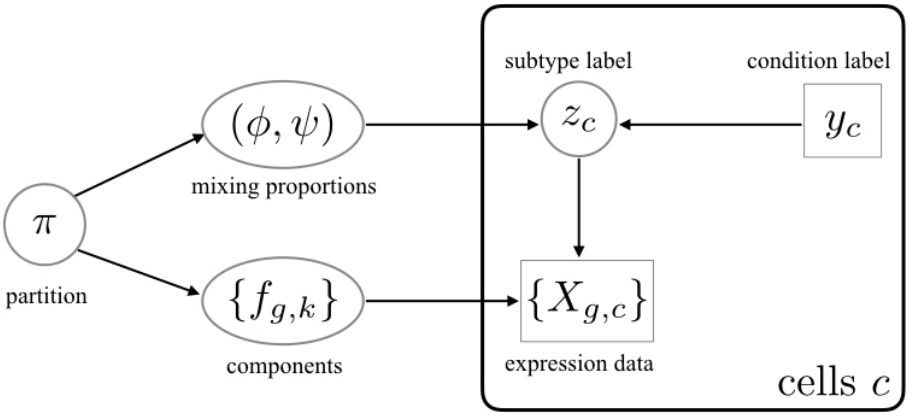


Figure 2.3: Directed acyclic graph structure of the compositional model and partition-reliant prior. The plate on the right side indicates i.i.d. copies over cells c, conditionally on mixing proportions and mixing components. Observed data are indicated in rectangles/squares, and unobserved variables are in circles/ovals.

Key to computing the gene-specific local false discovery rate $P(\text{ED}_g|X, y)$ is evaluating probabilities $P(A_\pi \cap M_{g,\pi}|X, y)$. The dependence structure (Figure 2.3) implies a useful reduction of this quantity, at least conditionally upon subtype labels $z = (z_c)$. For each subtype partition π and gene g,

Theorem 2.2.2. $P(A_\pi \cap M_{g,\pi}|X, y, z) = P(A_\pi|y, z)\, P(M_{g,\pi}|X, z)$.

In what follows, we develop the modeling and computational elements necessary to efficiently evaluate inference summaries (2.5) taking advantage of Theorems 1 and 2. Roughly, the methodological idea is that subtype labels z have relatively low uncertainty, and may be estimated from genome-wide clustering of cells in the absence of condition information y (up to an arbitrary label permutation). The modest uncertainty in z we handle through a computationally efficient randomized clustering scheme. Theorem 2 indicates that our computational task then separates into two parts given z. On one hand, cell subtype frequencies combine with condition labels to give $P(A_\pi|y, z)$. Then gene-level data locally drive the posterior probabilities $P(M_{g,\pi}|X, z)$ that measure differential expression between subtypes. Essentially, the model provides a specific form of information sharing between genes that leverages the compositional structure of single-cell data in order to sharpen our assessments of between-condition expression changes.

2.2.2 Method structure and clustering

To infer subtypes, we leverage the extensive research on how to cluster cells using scRNA-seq data: for example, SC3 (Kiselev et al. (2017)), CIDR (Lin et al. (2017)), and ZIFA (Pierson and Yau (2015)). We propose distance-based clustering on the full set of profiles in a way that is blind to the condition label vector y, in order to have as many cells as possible to inform the subtype structure. We investigated several clustering schemes in numerical experiments and allow flexibility in this choice within the SCDDBOOST software. Associating clusters with subtype labels $\widehat{z}_c$ estimates the actual subtypes z_c, and prepares us to use Theorems 1 and 2 in order to compute separate posterior probabilities $P(A_\pi|y, \widehat{z})$ and $P(M_{g,\pi}|X, \widehat{z})$ that are necessary for scoring differential distribution. The first probability concerns patterns of cell counts over subtypes in the two conditions, and has

a convenient closed form within the double-Dirichlet model (Section 2.3). The second probability concerns patterns of changes in expected expression levels among subtypes, and this is also conveniently computed for negative-binomial counts using `EBSeq` (Leng et al. (2013)). Algorithm 1 summarizes how these elements combine to get the posterior probability of differential distribution per gene, conditional on an estimate of the subtype labels.

Algorithm 1 scDDBoost-core

Input:

GENES by CELLS expression data matrix $X = (X_{g,c})$

cell condition labels $y = (y_c)$

cell subtype labels (estimated) $\widehat{z} = (\widehat{z}_c)$

Output: posterior probabilities of differential distribution from estimated subtypes

1: **procedure** scDDBoost-core($X, y, \widehat{z}$)
2: number of cell subtypes $K = \text{length}(\text{unique}(\widehat{\text{z}}))$
3: subtype differential expression: $\forall g, \pi$ compute $P(M_{g,\pi}|X, \widehat{z})$ using `EBSeq`
4: cell frequency changes: $\forall \pi$ compute $P(A_\pi|y, \widehat{z})$ using Double Dirichlet model
5: posterior probability: $\forall g,\ P(\text{ED}_g|X, y, \widehat{z}) \leftarrow \sum_\pi P(M_{g,\pi}|X, \widehat{z})\ P(A_\pi|y, \widehat{z})$
6: **return** $\forall g,\ P(\text{DD}_g|X, y, \widehat{z}) = 1 - P(\text{ED}_g|X, y, \widehat{z})$
7: **end procedure**

We invoke $K-$medoids (Kaufman and Rousseeuw (1987)) as the default clustering method in `scDDboost`, and customize the cell-cell distance by integrating two measures. The first assembles gene-level information by cluster-based-similarity partitioning (Strehl and Ghosh (2003)). Separately at each gene, modal clustering (Dahl (2009a) and Appendix A.6) partitions the cells, and then we define dissimilarity between cells as the Manhattan distance between gene-specific partition labels. A second measure defines dissimilarity by one minus the Pearson correlation between cells, which is computationally inexpensive, less sensitive to outliers than Euclidean distance, and effective at detecting cellular clusters in scRNA-seq (Kim et al. (2018)). The default clustering in `scDDboost` combines these two measures by weighted average, with $w_C = \frac{\sigma_P}{\sigma_C + \sigma_P}$ and $w_P = 1 - w_C$, where $w_C, \sigma_C, w_P, \sigma_P$ are the weights and standard deviations of cluster-based distance and Pearson-correlation distance, respectively. The software allows other distances; in any case the final distance matrix is denoted $D = (d_{i,j})$.

Any clustering method entails classification errors, and so $\widehat{z}_c \neq z_c$ for some cells. To mitigate the effects of this uncertainty, `scDDboost` averages output probabilities from SCDDBOOST-CORE over randomized clusterings $\widehat{z}^*$. These are not uniformly random, but rather are generated by applying $K-$medoids to a randomized distance matrix $D^* = (d_{i,j}/w_{i,j})$, where $w_{i,j}$ are non-negative weights $w_{i,j} = (e_i + e_j)$, and where (e_i) are independent and identically Gamma distributed deviates with shape $\widehat{a}/2$ and rate $\widehat{a}$, and where $\widehat{a}$ is estimated from D. (Thus $w_{i,j}$ is Gamma$(\widehat{a}, \widehat{a})$ and has unit mean.) The distribution of clusterings induced by this simple computational scheme approximates a Bayesian posterior analysis, as we argue in the Appendix, where we also present pseudo-code for the resulting `scDDboost` Algorithm 5. Averaging over results from randomized clusterings gives additional stability to the posterior probability statistics (Appendix Figure A10).

Computations become more intensive the larger is the number K of cell subtypes. Version 1.0 of `scDDboost` is restricted to $K \leq 9$; we consider further computational strategies in Section 5. Inferentially, taking K to be too large may inflate the false positive rate (Appendix Figure A11). The approach taken in `scDDboost` is to set K using the validity score (Ray and Turi (2000)), which measures changes in within-cluster sum of squares as we increase K. Our implementation, in Appendix A.8, shows good operating characteristics in simulation.

2.2.3 $P(A_\pi|y, z)$

We introduce the Double Dirichlet Mixture (DDM), which is the partition-reliant prior $p(\phi, \psi)$ indicated in Figure 2.3, in order to derive an explicit formula for $P(A_\pi|y, z)$. We lose no generality here by defining $A_\pi = \{(\phi, \psi) : \Phi_b = \Psi_b \ \forall b \in \pi\}$, rather than as a subset of the full parameter space as in (2.3). Each A_π is closed and convex subset of the product space holding all possible pairs of length-K probability vectors.

We propose a spike-slab-style mixture prior with the following form:

$$p(\phi, \psi) = \sum_{\pi \in \Pi} \omega_\pi \, p_\pi(\phi, \psi). \tag{2.6}$$

Each mixture component $p_\pi(\phi, \psi)$ has support A_π; the mixing proportions ω_π are positive constants

summing to one. To specify component p_π, notice that on A_π there is a 1-1 correspondence between pairs (ϕ, ψ) and parameter states:

$$\left\{(\widetilde{\phi}_b, \widetilde{\psi}_b, \Phi_b),\ \forall b \in \pi\right\}, \tag{2.7}$$

where

$$\widetilde{\phi}_b = \frac{\phi_b}{\Phi_b}, \quad \widetilde{\psi}_b = \frac{\psi_b}{\Psi_b}, \quad \text{and} \quad \Phi_b = \sum_{k\in b} \phi_k = \sum_{k\in b} \psi_k = \Psi_b.$$

For example, $\widetilde{\phi}_b$ is a vector of conditional probabilities for each subtype given that a cell from the first condition is one of the subtypes in b.

We introduce hyperparameters $\alpha_k^1, \alpha_k^2 > 0$ for each subtype k, and set $\beta_b = \sum_{k\in b} \left(\alpha_k^1 + \alpha_k^2\right)$ for any possible block b. Extending notation, let α_b^j be the vector of α_k^j for $k \in b$, β_π be the vector of β_b for $b \in \pi$, ϕ_b and ψ_b be vectors of ϕ_k and ψ_k, respectively, for $k \in b$, and Φ_π and Ψ_π be the vectors of Φ_b and Ψ_b for $b \in \pi$. The proposed double-Dirichlet component p_π is determined in the transformed scale by assuming $\Psi_\pi = \Phi_\pi$ and further:

$$\begin{aligned} \Phi_\pi &\sim \text{Dirichet}_{N(\pi)}[\beta_\pi] \\ \widetilde{\phi}_b &\sim \text{Dirichlet}_{N(b)}[\alpha_b^1] \qquad \forall b \in \pi \\ \widetilde{\psi}_b &\sim \text{Dirichlet}_{N(b)}[\alpha_b^2] \qquad \forall b \in \pi \end{aligned} \tag{2.8}$$

where $N(\pi)$ is the number of blocks in π and $N(b)$ is the number of subtypes in b, and where all random vectors in (2.8) are mutually independent. Mixing over π as in (2.6), we write $(\phi, \psi) \sim \text{DDM}\left[\omega = (\omega_\pi), \alpha^1 = (\alpha_k^1), \alpha^2 = (\alpha_k^2)\right]$.

We record some properties of the component distributions p_π:

Property 1: In $p_\pi(\phi, \psi)$, ψ and ϕ are dependent, unless π is the null partition in which all subtypes constitute a single block.

Property 2: With $k \in b$, marginal means are:

$$E_\pi(\phi_k) = \frac{\alpha_k^1}{\sum_{k' \in b} \alpha_{k'}^1} \frac{\beta_b}{\sum_{b' \in \pi} \beta_{b'}} \quad \text{and} \quad E_\pi(\psi_k) = \frac{\alpha_k^2}{\sum_{k' \in b} \alpha_{k'}^2} \frac{\beta_b}{\sum_{b' \in \pi} \beta_{b'}}.$$

Recall from (2.1) the vectors t^1 and t^2 holding counts of cells in each subtype in each condition, computed from y and z. Relative to a block $b \in \pi$, let $t_b^j = \sum_{k \in b} t_k^j$, for cell conditions $j = 1, 2$, and, let t_π^j be the vector of these counts over $b \in \pi$. The following properties refer to marginal distributions in which (ϕ, ψ) have been integrated out of the joint distribution involving (2.2) and the component p_π.

Property 3: t^1 and t^2 are conditionally independent given y, t_π^1 and t_π^2.

Property 4: For $j = 1, 2$,

$$p_\pi(t^j | t_\pi^j, y) = \prod_{b \in \pi} \left\{ \left[\frac{\Gamma(t_b^j + 1)}{\prod_{k \in b} \Gamma(t_k^j + 1)} \right] \left[\frac{\Gamma(\sum_{k \in b} \alpha_k^j)}{\prod_{k \in b} \Gamma(\alpha_k^j)} \right] \left[\frac{\prod_{k \in b} \Gamma(\alpha_k^j + t_k^j)}{\Gamma(t_b^j + \sum_{k \in b} \alpha_k^j))} \right] \right\}$$

Property 5:

$$p_\pi(t_\pi^1, t_\pi^2 | y) = \left[\frac{\Gamma(n_1 + 1)\Gamma(n_2 + 1)}{\prod_{b \in \pi} \Gamma(t_b^1 + 1)\Gamma(t_b^2 + 1)} \right] \left[\frac{\Gamma(\sum_{b \in \pi} \beta_b)}{\prod_{b \in \pi} \Gamma(\beta_b)} \right] \left[\frac{\prod_{b \in \pi} \Gamma(\beta_b + t_b^1 + t_b^2)}{\Gamma(n_1 + n_2 + \sum_{b \in \pi} \beta_b)} \right].$$

Let's look at some special cases to dissect this result.

Case 1. If π has a single block equal to the entire set of cell types $\{1, 2, \cdots, K\}$, then $t_b^j = n_j$ for both $j = 1, 2$, and Property 5 reduces, correctly, to $p_\pi(t_\pi^1, t_\pi^2 | y) = 1$. Further,

$$p_\pi(t^j | t_\pi^j, y) = \left[\frac{\Gamma(n_j + 1)}{\Gamma(n_j + \sum_{k=1}^K \alpha_k^j)} \right] \left[\frac{\Gamma(\sum_{k=1}^K \alpha_k^j)}{\prod_{k=1}^K \Gamma(\alpha_k^j)} \right] \left[\prod_{k=1}^K \frac{\Gamma(\alpha_k^j + t_k^j)}{\Gamma(t_k^j + 1)} \right]$$

which is the well-known Dirichlet-multinomial predictive distribution for counts t^j (Wagner and Taudes (1986)). E.g, taking $\alpha_k^j = 1$ for all types k we get the uniform distribution

$$p_\pi(t^j | t_\pi^j, y) = \frac{\Gamma(n_j + 1)\Gamma(K)}{\Gamma(n_j + K)}.$$

Case 2. At the opposite extreme, π has one block b for each class k, so $\phi = \psi$. Then

$p_\pi(t^j|t^j_\pi, y) = 1$, and further, writing $b = k$,

$$p_\pi(t^1_\pi, t^2_\pi|y) = \left[\frac{\Gamma(n_1+1)\Gamma(n_2+1)}{\prod_{k=1}^K \Gamma(t^1_k+1)\Gamma(t^2_k+1)}\right]\left[\frac{\Gamma(\sum_{k=1}^K \beta_k)}{\prod_{k=1}^K \Gamma(\beta_k)}\right]\left[\frac{\prod_{k=1}^K \Gamma(\beta_k + t^1_k + t^2_k)}{\Gamma(n_1+n_2+\beta_k)}\right].$$

which corresponds to Dirichlet-multinomial predictive distribution for counts $t^1 + t^2$ since t^1 and t^2 are identical distributed given (ϕ, ψ) in this case. These properties are useful in establishing:

Theorem 2.2.3. *DDM is conjugate to multinomial sampling of t^1 and t^2:*

$$(\phi, \psi)|y, z \sim \text{DDM}\left[\omega^{\text{post}} = (\omega^{\text{post}}_\pi), \alpha^1 + t^1, \alpha^2 + t^2\right]$$

where

$$\omega^{\text{post}}_\pi \propto p_\pi(t^1|t^1_\pi, y)\, p_\pi(t^2|t^2_\pi, y)\, p_\pi(t^1_\pi, t^2_\pi|y)\, \omega_\pi. \tag{2.9}$$

The target probability $P(A_\pi|y, z)$ is an integral of the posterior distribution in Theorem 3. To evaluate it, we need to contend with the fact that sets $\{A_\pi : \pi \in \Pi\}$ are not disjoint. Relevant overlaps have to do with partition refinement. Recall that a partition π^r is a refinement of a partition π^c if for any $b \in \pi^c$ there exists $s \subset \pi^r$ such that $\underset{b' \in s}{\cup} b' = b$. We say π^c coarsens π^r when π^r refines π^c. Any partition both refines and coarsens itself, as a trivial case. Generally, refinements increase the number of blocks. If subtype frequency vectors (ϕ, ψ) satisfy the constraints in A_{π^r} then they also satisfy the constraints of any π^c that coarsens π^r: i.e., $A_{\pi^r} \subset A_{\pi^c}$. Refinements reduce the dimension of allowable parameter states. For the double-Dirichlet component distributions P_π, we find:

Property 6: For two partitions $\widetilde{\pi}$ and π, $P_{\widetilde{\pi}}(A_\pi|y, z) = 1[\widetilde{\pi} \text{ refines } \pi\,]$.

This supports the main finding of this section:

$$P(A_\pi|y, z) = \sum_{\widetilde{\pi} \in \Pi} \omega^{\text{post}}_{\widetilde{\pi}}\, 1[\widetilde{\pi} \text{ refines } \pi]. \tag{2.10}$$

2.2.4 $P(M_{g,\pi}|X, z)$

We leverage well-established modeling techniques for transcript analysis, including (Leng et al. (2013), Kendziorski et al. (2003b), and Jensen et al. (2009)), which characterize equivalent or differential expression in terms of shared or independently drawn mean effects. Let $X_{g,b}$ denote the subvector of expression values at gene g over cells c with $z_c = k$ for which subtype k is part of block b of partition π. Conditioning on subtype labels $z = (z_c)$, we assume that under $M_{g,\pi}$:

1. *between blocks:* subvectors $\{X_{g,b} : b \in \pi\}$ are mutually independent,
2. *within blocks:* for cells mapping to block b, observations $X_{g,c}$ are i.i.d.
3. *mean effects:* for each block b, there is a univariate mean, $\mu_{g,b}$, shared by cells mapping to that block. *a priori* these means are i.i.d. between blocks.

These assumptions imply a useful factorization marginally to latent means,

$$P(X_g|M_{g,\pi}, z) = \prod_{b\in\pi} f(X_{g,b}), \tag{2.11}$$

where f is a customized density kernel. In our case we use `EBseq` from (Leng et al. (2013)): the sampling distribution of $X_{g,c}$ is negative binomial, and f becomes a particular compound multivariate negative binomial formed from integrating uncertainty in the block-specific means (see Appendix A.5). Through its gene-level mixing model, `EBseq` also gives estimates of $\{P(M_{g,\pi}|z)\}$: the proportions of genes governed by any of the different patterns π of equivalent/differential expression among subtypes. With these estimates and (2.11) we compute by Bayes's rule:

$$P(M_{g,\pi}|X, z) \propto P(M_{g,\pi}|z) \prod_{b\in\pi} f(X_{g,b}).$$

The proportionality is resolved by calculating over all partitions π.

2.3 Numerical experiments

2.3.1 Synthetic data

We used `splatter` (v. 1.2.0) to generate synthetic scRNA-seq data for which the DD status of genes is known (Zappia et al. (2017)), thereby allowing us to measure operating characteristics of `scDDboost`. Our hypothetical two-condition comparison involved 17421 genes, 10% of which exhibited actual shifts in distribution between two conditions. We entertained 12 different parameter settings encoding these distributional shifts, varying the number of subtypes K, the subtype frequency profiles (ϕ, ψ), as well as the `splatter`-specific parameters θ and γ controlling location and scale characteristics of expression levels. These settings cover a range of scenarios we might expect to see in practice. Two replicate data sets were simulated under each parameter setting. Further details are in Appendix A.10.

Figures 2.4 and 2.5 summarize the true positive rate and false discovery rate of `scDDboost` compared to three other methodologies: `MAST` (v. 1.4.0), `scDD` (v. 1.2.0), and `DESeq2` (v. 1.18.1). `scDDboost` exhibits very good operating characteristics in this study, as it controls the FDR in all cases while also delivering a relatively high rate of true positives in all cases.

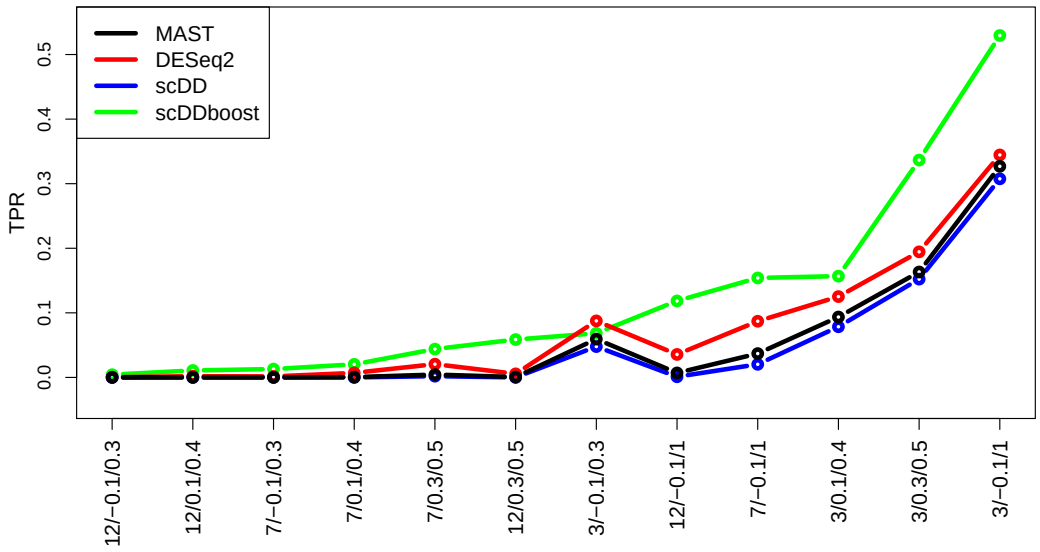


Figure 2.4: True positive rate (vertical) of four DD detection methods in 12 synthetic-data settings (horizontal). Settings are labeled for $K/\theta/\gamma$ and ranked by scDDboost values. Each method is targeting a 5% false discovery rate (FDR). The plot shows average rates over replicate simulated data in each setting.

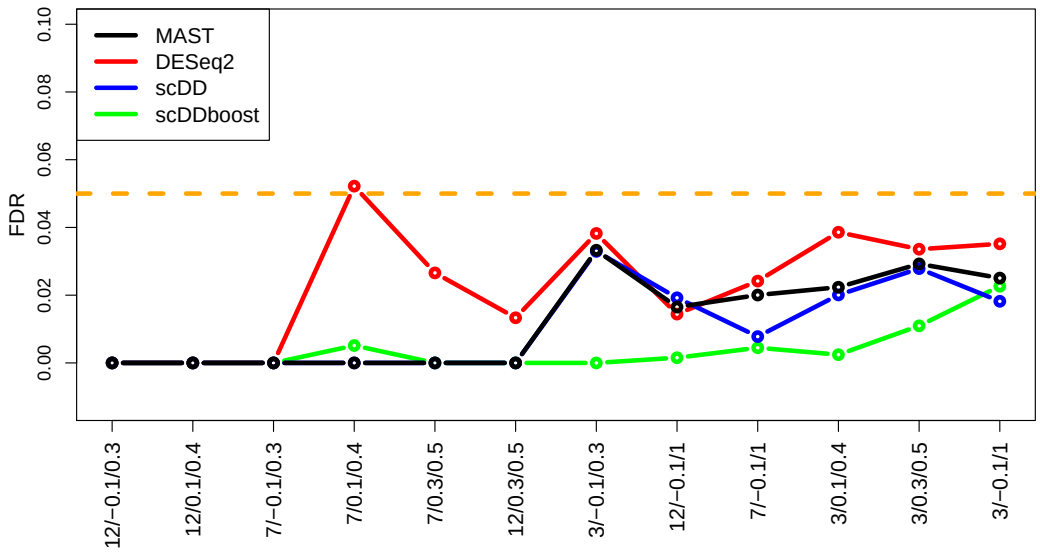


Figure 2.5: False discovery rate (vertical) of methods in settings (horizontal, same order) from Figure 2.4

2.3.2 Empirical study

We applied `scDDboost` to a collection of previously published data sets that are recorded at `conquer` (Soneson and Robinson (2017)). Though not knowing the truly DD genes, we can examine how `scDDboost` output compares to output from several standard methods. We selected 12 data sets from `conquer` representing different species and experimental settings and involving hundreds to thousands of cells. Appendix Table A7 provides details. Figure 2.6 compares methods in terms of the size of the reported list of DD genes at the 5% FDR target level. We see a consistently high yield of `scDDboost` among the evaluated methods. For reference, one of these data sets (GSE64016) happens to be the data behind Figure 2.1, where we know from other information that some of the uniquely identified genes are likely not to be false positives.

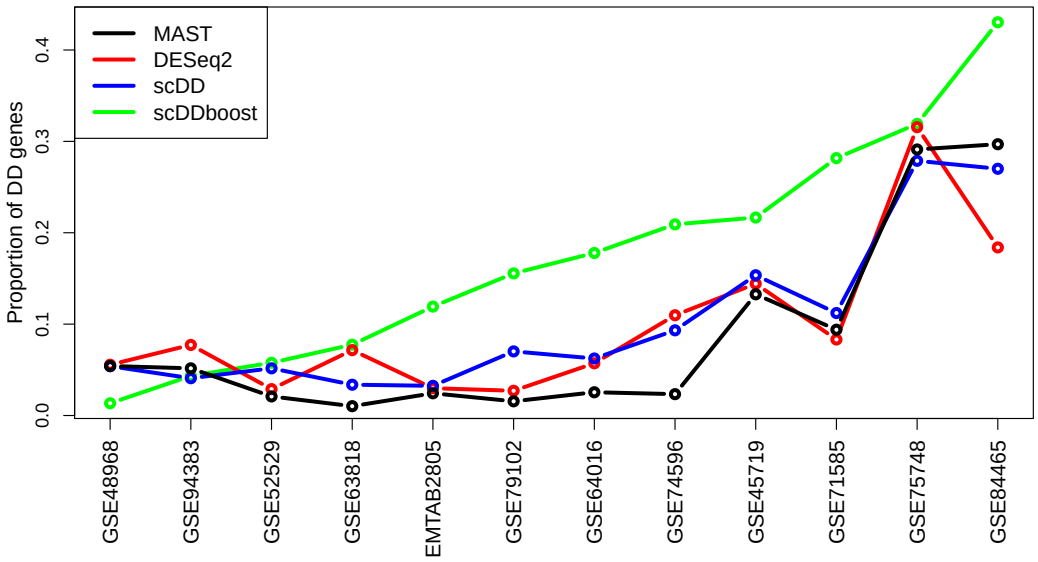


Figure 2.6: Proportion of DD genes at 5% FDR threshold with respect to total number of genes identified by each method. Ranked by scDDboost list size

To check that the increased discovery rate of `scDDboost` is not associated with an increased rate of false calls, we applied it to a series of random splits of single-condition data sets (Appendix Table A8). Figure 2.7 confirms a very low call rate in cases where no changes in distribution are expected.

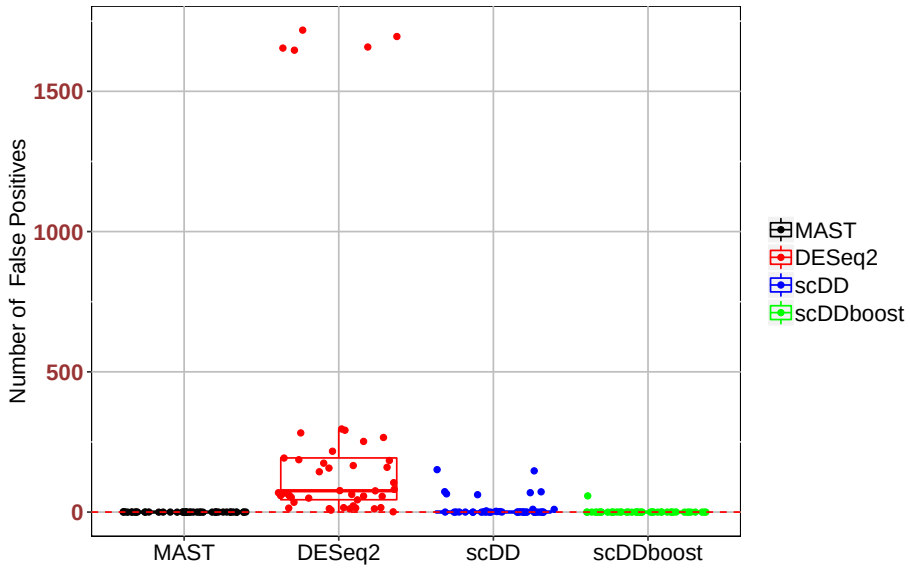


Figure 2.7: False positive counts at 5% FDR threshold by several methods on 5 random splits of 9 single-condition data sets from Appendix Table A8

We conjecture that `scDDboost` gains power through its novel approach to borrowing strength across genes; i.e., that the genomic data are providing information about cell subtypes and mixing proportions, leaving gene-level data to guide gene-specific mixture components. One way to drill into this idea is to consider how many genes have similar expression characteristics to a given gene. By virtue of the `EBseq` analysis inside `scDDboost`, we may assign each gene to a set of related genes that all have the same highest-probability pattern of equality/inequality of means across the subtypes. Say $\widehat{\pi}_g = \text{argmax}_\pi P(M_{g,\pi}|\widehat{z}, X)$. In Figure 2.8, we show that compared to DD genes commonly identified by multiple methods (blue), the set sizes for genes uniquely identified by `scDDboost` (red) tend to be larger. Essentially, the proposed methodology boosts weak DD evidence when a gene's pattern of differential expression among cell subtypes matches a large number of other genes.

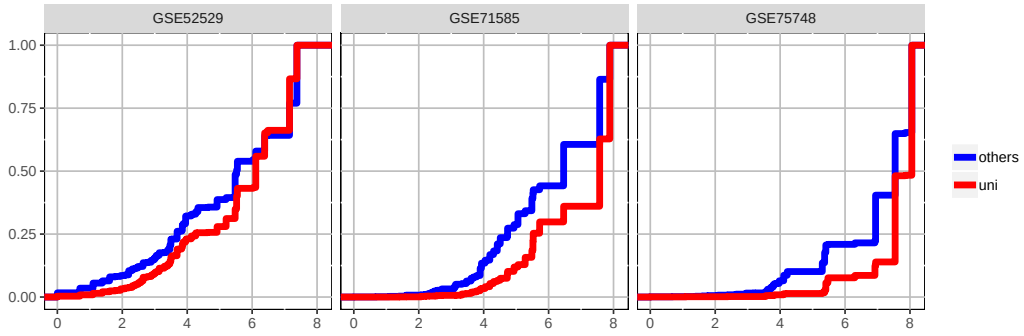


Figure 2.8: Genes are grouped by their pattern of differential expression across subtypes as inferred by the EBseq computation within scDDboost for three example datasets. Cumulative distribution functions of the log-scale size statistic for all genes identified by scDDboost are plotted; red is the subset uniquely identified by scDDboost; blue are those also identified by the comparison methods (MAST, scDD, or DESeq2). Sets of similarly-patterned genes tend to be larger (horizontal axis, log size) for genes uniquely identified by scDDboost (red) compared to other DD genes (blue), at 5% FDR.

2.3.3 Bursting

Transcriptional bursting is a fundamental property of genes, wherein transcription is either negligible or attains a certain probability of activation (Raj and van Oudenaarden (2008)). D3E (Delmans and Hemberg (2016)) is a computationally intensive method for DE gene analysis rooted in modeling the bursting process. It considers transcripts as in the stationary distribution from an experimentally validated stochastic process of single-cell gene expression (Peccoud and Ycart (1995)). Three mechanistic parameters (rate of promoter activation, rate of promoter inactivation, and the conditional rate of transcription given an active promoter) characterize the model, which allow distributional changes between conditions without changing the mean expression level. For genes identified as DD by `scDDboost` in dataset GSE71585, either uniquely or in common with comparison methods, Figure 2.9 shows changes of these bursting parameters. Interestingly, genes uniquely identified by `scDDboost` are associated with more significant changes between estimated bursting parameters compared to commonly identified genes. This finding and similar findings on other data sets (not shown) provide some evidence that `scDDboost` is able to detect biologically meaningful changes in the expression distribution.

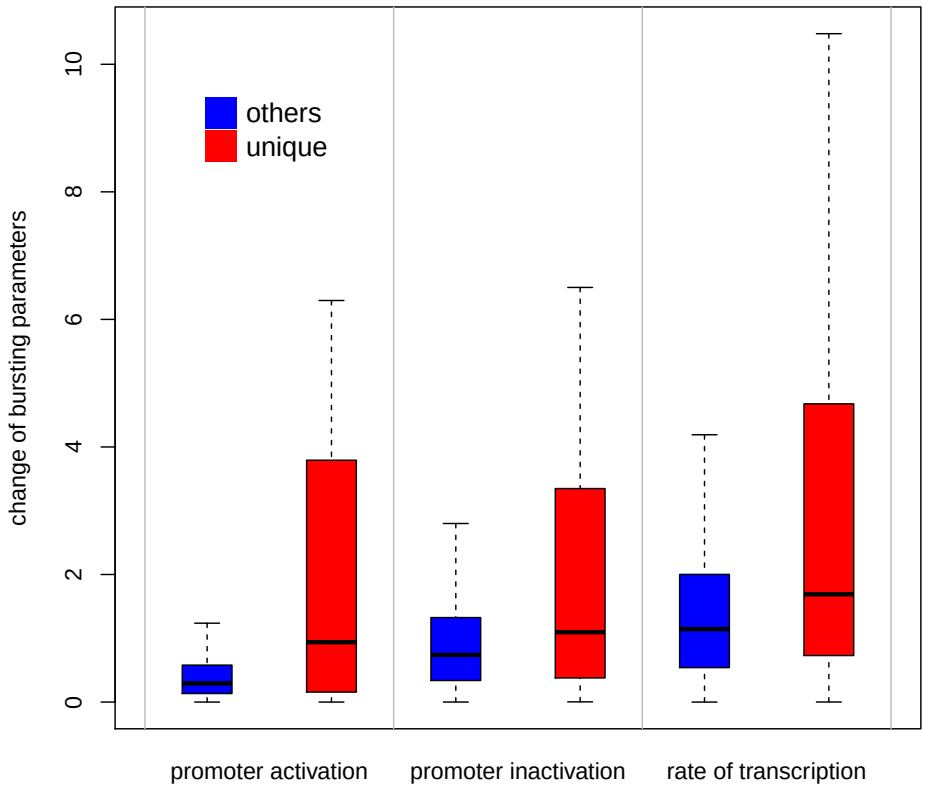


Figure 2.9: Absolute values of log fold changes of bursting parameters tend to be larger for 1758 genes uniquely identified by scDDboost (red) compare to other 2983 genes (blue) at 5% FDR

2.3.4 Time Complexity

Run time complexity of `scDDboost` is dominated by the cost of clustering cells and of running `EBSeq` to measure differences between subtypes. Recall the notation that n for number of cells, G for number of genes and K for number of subtypes. Our distance-based clustering of n cells measuring G genes requires on the order of $G \times n^2$ operations (see Appendix A.6). Further, `EBSeq` uses summed counts within each subtype for each gene to compute its density kernel, and there are Bell(K) differential patterns to compute, where Bell counts the partitions of K. Our implementation

`scDDboost` efficiently deals with large n under moderate K. We have imposed the computational limit $K \leq 9$ in `scDDboost` (v. 1.0). In a typical case involving 20000 genes and 200 cells, using 50 of randomized distances, `scDDboost` is relatively efficient for $K \leq 6$ requiring less than 15 CPU minutes on, for example, a quad-core 2.2 GHz Intel Core i7 with 16 Gb of RAM. The same data might require 20 to 40 CPU hours when $K = 9$. In Section 5 we mention some opportunities to improve this speed.

2.4 Asymptotics of the Double Dirichlet Mixture

Summary statistics $P(A_\pi|y, z)$, from Section 2.3, are amenable to a first-order asymptotic analysis that provides further insight into DDM model behavior. The fact that support sets A_π for component distributions $p_\pi(\phi, \psi)$ are not disjoint becomes an important issue. Consider distinct partitions π_1 and π_2 of subtypes $\{1, 2, \cdots, K\}$, and recall that $N(\pi)$ counts the number of blocks in partition π. In case π_2 refines π_1, then $N(\pi_1) < N(\pi_2)$, and we also know that $A_{\pi_2} \subset A_{\pi_1}$, since refinement imposes additional constraints on the pair (ϕ, ψ) of probability vectors. If the data-generating state $(\phi, \psi) \in A_{\pi_2}$, one might ask how posterior probability mass tends to be allocated among the other mixture components whose support sets also contain this state. The question is addressed by the following:

Theorem 2.4.1. *Let π_1 and π_2 denote two partitions for which $N(\pi_1) < N(\pi_2)$ and $A_{\pi_1} \cap A_{\pi_2}$ is non-empty. Let $(\phi, \psi) \in A_{\pi_1} \cap A_{\pi_2}$ denote the data generating state for subtype labels $z_1, z_2, \cdots, z_n$ given i.i.d. Bernoulli condition labels $y_1, y_2, \cdots, y_n$, and recall the posterior mixing proportions ω_π^{post} from equation (2.9) with hyper-parameters $\alpha_i^j \geq 1$ for $i = 1, \cdots, K$, $j = 1, 2$. Then*

$$\frac{\omega_{\pi_1}^{\text{post}}}{\omega_{\pi_2}^{\text{post}}} \longrightarrow_{a.s.} 0 \qquad \text{as } n \longrightarrow \infty.$$

Essentially, mixing mass is transferred to components associated with the most refined partition consistent with a given parameter state. To be precise, let $H(\phi, \psi) = \{\pi : (\phi, \psi) \in A_\pi\}$ record all the partitions associated with one state. Typically, there is a most refined partition, $\pi^* = \pi^*(\phi, \psi)$,

such that

$$A_{\pi^*} = \bigcap_{\pi \in H(\phi,\psi)} A_\pi. \tag{2.12}$$

This always happens when $K \leq 3$. In Appendix A.12 we characterize the exceptional set of states where (2.12) does not hold. Notably, if (2.12) does hold for state (ϕ, ψ), then for any $\pi \in H(\phi, \psi)$, using Theorem 4 and (2.10), we have

$$P(A_\pi | y_1, \cdots, y_n; z_1, \cdots z_n) \longrightarrow_{a.s.} 1 \quad \text{as } n \longrightarrow \infty.$$

This provides conditions under which we expect good performance for large numbers of cells.

2.5 Compositional model for more than two conditions

Many scRNA data are measured through more than 2 conditions. The ability to perform tests for multiple conditions has potential to better analyze the data. We present a framework allowing inference for various patterns of how those distributions are varied under multiple conditions. Using the compositional model, the pattern of differential/equivalent distributions can be fully determined by the pattern of means and pattern of aggregated proportions. We extended the double Dirichlet prior in section 2.2 to infer patterns of aggregated proportions under more than two conditions.

2.5.1 Method

Assuming we have data from conditions $1, ..., T$, denoted as $X^1, ..., X^T$. Recall f_g^i is marginal density of gene g at condition i. A pattern of distributional changes can be represented as

$$D_{g,\Delta} = \{f_g^i = f_g^j, \forall i, j \in b, \forall b \in \Delta\}$$

Where Δ is a partition for conditions. We use Δ to distinguish with π, which is the partition for subtypes. Any two conditions from the same block have identical distribution and any two

conditions from different blocks have differential distribution. Assume there is K subtypes over the T conditions. Recall $f_{g,i}$ is the sampling distribution for expression value at gene g from subtype i. Let ϕ_i^j be proportion of subtype i at condition j. We have the marginal density of a gene in one condition

$$f_g^j = \sum_{i=1}^{K} \phi_i^j f_{g,i}$$

As the mixing components are common to conditions, whether $f_g^i = f_g^j$ is fully attributed to how the mixing proportions are changed. Recalling that there is no distributional change if the aggregated proportions remain the same on block of subtypes sharing the same distribution. Let $\phi^j = (\phi_i^j)$ be the vector for proportions in condition j and $\Phi_b^j = \sum_{i \in b} \phi_i^j$ be the aggregated proportions over block b in condition j. Consider patterns of aggregated proportions

$$A_{\pi,\Delta} = \{\forall b_1 \in \Delta, \forall j_1, j_2 \in b_1, \forall b_2 \in \pi, \Phi_{b_2}^{j_1} = \Phi_{b_2}^{j_2}\} \tag{2.13}$$

Recalling that patterns of means are

$$M_{g,\pi} = \{\theta \in \Theta : \ \mu_{g,k} = \mu_{g,k'} \iff k, k' \in b, b \in \pi\} \tag{2.14}$$

Further, we denote the most coarse partition and the most refined partition for conditions as $\Delta_{\mathrm{C}} = \{1, ..., T\}, \Delta_{\mathrm{R}} = \{\{1\}, ..., \{T\}\}$. Similarly we have π_{C} for the most coarse partition of subtypes. We have a theorem

Theorem 2.5.1. *Let $\Omega = \{(\mu, \phi^1, ..., \phi^T), \mu \in \mathbb{R}^K, \phi^j \in \mathbb{R}^K\}$ be the whole parameter space, then the patterns of distributional change can be characterized as*

$$D_{g,\Delta} = \begin{cases} \bigcup\limits_{\pi \neq \pi_C} M_{g,\pi} \cap A_{\pi,\Delta} & \text{if } \Delta \neq \Delta_C, \Delta_R \\ \bigcup\limits_{\pi} M_{g,\pi} \cap A_{\pi,\Delta} & \text{if } \Delta = \Delta_C \\ \Omega \setminus \bigcup\limits_{\Delta' \neq \Delta} D_{g,\Delta'} & \text{if } \Delta = \Delta_R \end{cases}$$

We propose a spike-slab-style mixture for $(\phi^1, ..., \phi^T)$, i.e.

$$p(\phi^1, ..., \phi^T) = \sum_{\pi}\sum_{\Delta}\omega_{\pi,\Delta}p_{\pi,\Delta}(\phi^1, ..., \phi^T)$$

To specify $p_{\pi,\Delta}$, similar in section 2.2, there is a 1-1 correspondence between $(\phi^1, ..., \phi^T)$ and parameter states:

$$\{(\widetilde{\phi}^j_{b_1}, \Phi^{b_2}_{b_1}, j \in b_2), b_1 \in \pi, b_2 \in \Delta\}$$

where

$$\widetilde{\phi}^j_{b_1} = \frac{\phi^j_{b_1}}{\Phi^{b_2}_{b_1}}, \text{ for } i \in b_2, \quad \text{and} \quad \Phi^{b_2}_{b_1} = \sum_{i \in b}\phi^j_i \text{ for } j \in b_2$$

For example, $\widetilde{\phi}^j_{b_1}$ is a vector of conditional probabilities for each subtype given that a cell from the condition j is one of the subtypes in b_1.

We introduce hyperparameters $\alpha^1_i, ..., \alpha^T_i > 0$ for any subtype i, and set $\beta^{b_2}_{b_1} = \sum_{j \in b_2}\sum_{i \in b_1}\alpha^j_i$ for any subtype block b_1 and condition block b_2. Extending notation, Let $\alpha^j_{b_1}$ be the vector of $\alpha^j_i, i \in b_1$, $\beta^{b_2}_{\pi}$ be the vector of $\beta^{b_2}_{b_1}, b_1 \in \pi$, and $\Phi^{b_2}_{\pi}$ be the vector of $\Phi^{b_2}_{b_1}, b_1 \in \pi$. Then $p_{\pi,\Delta}$ is determined in the transformation scale with

$$\Phi^{b_2}_{\pi} \sim \text{Dirichlet}_{N(\pi)}[\beta^{b_2}_{\pi}]$$
$$\widetilde{\phi}^j_{b_1} \sim \text{Dirichlet}_{N(b_1)}[\alpha^j_{b_1}] \quad \forall b_1 \in \pi$$

2.6 Concluding remarks

We have presented `scDDboost`, a tool for detecting differentially distributed genes from scRNA-seq data, where transcripts are modeled as a mixture of cellular subtypes. The methodology links established model-based techniques with novel empirical Bayesian modeling and computational

elements to provide a powerful detection method showing comparatively good operating characteristics in simulation, empirical, and asymptotic studies.

In the software and numerical experiments we made specific choices, such as to use mixtures of negative binomial components per gene, and to use K-medoids clustering on particular cell-cell distances. These choices have evident advantages, but the model structure and theory developed in Section 2 carry through for other cases. Future experiments could study other formulations within the same schema; for example there may be cell-cell distances that better capture the intrinsic dimensionality of expression programs, including, perhaps distances based on diffusions (Haghverdi et al. (2015)) or the longest-leg path distance (Little et al. (2017)). Future experiments could also further assess operating characteristics when the number of cells is very large and the number of reads is relatively small, as may arise with unique molecular identifiers (Chen et al. (2018)). Further, assuming a compositional structure to drive model-based computations may not be restrictive, since it allows great flexibility in the form of each gene/condition-specific expression distribution (as coded, they are finite mixtures of negative binomials).

`EBSeq` currently presents a computational bottleneck for `scDDboost`, since it searches all partitions of K and encodes a hyper-parameter estimation algorithm that scales poorly with K. Several approximations present themselves that may redress the problem, since, in the mixture model context, only patterns π corresponding to relatively probable expression-change patterns over subtypes have a big impact on the final posterior inference. Even resolving this bottleneck there are advantages to having K small compared to n. Numerical experiments (see Appendix) show increased false discoveries when K is over-estimated. But accurate estimation with large K would not be expected to provide much improved power, since that depends on accurate estimation of subtypes and their frequencies which relies on K being relatively small compared to n.

Chapter 3

Improved EBSeq: [1]

3.1 Background

Since their introduction over twenty years ago, technologies to measure genome-wide gene expression have revolutionized science and medicine. The resulting data have had a major impact on statistical sciences as well, by introducing challenges arising from ”small n, large p” datasets. One of the central statistical challenges has been the differential expression problem: namely, how do we identify genes whose expression levels vary significantly across biological conditions? Dozens of methods have been developed toward this end and a few have endured. For RNA-Seq data, the empirical Bayesian hierarchical modeling approach, encoded in R package EBSeq has a number of advantages owing to how it captures variation characteristics of genes and isoforms and how it scores differential expression over two or more conditions (Leng et al., 2013, Leng and Kendziorski, 2019). It has proven useful in hundreds of studies including studies of development (Louro et al., 2020, Sanders and Cartwright, 2015, Yoon et al., 2017, Sabbagh et al., 2018), viral transcription (Newhouse et al., 2017, O‘Grady et al., 2017, Baños-Lara et al., 2018, Zhang et al., 2017), and cancer (Lee et al., 2020, Song et al., 2018, Son et al., 2017, Yang et al., 2016). The most common use of EBSeq is to score differential expression between two biological conditions. The package’s multi-condition feature is less frequently deployed; but recently it has been recognized that EB-

[1] This chapter is a reformated version of a manuscript written jointly with co-authors Drs. C. Kendziorski and M.A. Newton, and prepared for the Journal of Statistical Software

Seq multi-condition scores are uniquely suited to characterizing multiple cellular subtypes from the growing body of single-cell RNA-seq data (Ma et al., 2019). The work reported here is motivated by the need to improve computational performance of the multi-condition calculations within the original system, say EBSeq.v1, which at the time of writing is version 1.26.0 at Bioconductor(Huber et al., 2015) In addition to some basic code improvements, we deploy in EBSeq.v2 a new algorithmic approach to determining multi-condition differential expression scores.

With samples from K biological conditions, EBSeq.v1 calculates posterior probability scores for various patterns of differential expression among these conditions. Typically the null pattern, in which expected expression is the same across all groups, receives the highest score, on the average over genes; the software can consider many patterns. A computational bottleneck arises if we ask the code to consider all possible patterns, which are equal in number to the Bell number, B_K, of partitions of K objects (Gardner, 1978). For even moderate K, the memory and time costs of EBSeq.v1 become excessive. Section 2 describes an alternative pruning/clustering algorithm which leverages the finding that many of the differential expression patterns will have small mixing rates; and we can know this without fitting the full model. By identifying patterns that are probably inconsequential to the final inferences, we remove them from other compute-intensive parts of the code and improve the overall operating characteristics, as demonstrated in a battery of benchmark tests in Section 3.3.

EBSeq.v2 improves the performance of EBSeq.v1. In addition to improved handling of group partitions indicated above and discussed in Section 2, the core code is converted to C++ and adopts open-source, peer-reviewed, and fast libraries Eigen and Boost for internals (Guennebaud et al., 2010, Boost, 2015). We also modify the EM algorithm by changing how the hyperparameters are recomputed in each cycle. We substitute the Nelder-Mead optimization (optim from package stats) with a single gradient step within each EM update. The overall effect of these changes is to dramatically improve the performance of EBSeq in the multi-group setting.

3.2 The statistical problem

For each inference unit in the system, we have real-valued measurements $X = \{X_i\}$ for a sample index $i = 1, 2, \cdots, n$, as well as discrete sample labels, say y_i, taking values in a label set $\{1, 2, \cdots, K\}$. The labels refer to different sampling groups, or conditions, that underly the measurements, and we are especially interested in the case when K exceeds 2. In applications of interest we expect K to be small compared to n, perhaps in the tens when n is in the hundreds to thousands. Statistical inference is focused on evaluating hypotheses about the unknown mean values $\mu_k = E(X_i|y_i = k)$. For example, in the case of $K = 4$ groups, we have $B_4 = 15$ different patterns of equality and inequality among the group means, a few of which are:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$
$$H_1 : \mu_1 \neq \mu_2 = \mu_3 = \mu_4$$
$$H_2 : \mu_1 \neq \mu_2 \neq \mu_3 = \mu_4, \mu_1 \neq \mu_3.$$

The set of partitions of $\{1, 2, \cdots, K\}$ is in one-one correspondence with the set of such hypotheses regarding equalities and inequalities among the means. More specifically, for a partition $\pi = \{b\}$ composed of mutually disjoint blocks $b \subset \{1, 2, \cdots, K\}$, we say that mean vector $\mu = (\mu_k)$ satisfies pattern π if $\mu_j = \mu_k$ whenever $j, k \in b$ for some $b \in \pi$ and also if $\mu_j \neq \mu_k$ whenever j and k are in different blocks. Thus, for example, H_0 above corresponds to $\pi = \{\{1, 2, 3, 4\}\}$, and H_1 corresponds to $\pi = \{\{1\}, \{2, 3, 4\}\}$. In the remainder, we use notation M_π to denote the the constraint (i.e., hypothesis) on the mean vector associated with partition π:

$$M_\pi = \left\{\mu \in \mathbb{R}^K : \mu_j = \mu_k \iff j, k \in b,\, b \in \pi\right\}.$$

Notice that any mean vector μ is an element of exactly one of these subsets, and, considering all partitions,

$$\mathbb{R}^K = \bigcup_\pi M_\pi.$$

The use of partition/pattern structures for expected values appears in many statistical problems. They are a central feature of Dirichlet-process mixture computations (MacEachern, 1998), various Bayesian clustering algorithms (Quintana and Iglesias, 2003, Heller and Ghahramani, 2005), as well as empirical Bayesian methodologies for genomic applications, such as the EBSeq tool introduced earlier, and a similar tool for microarray-based data, EBarrays (Kendziorski et al., 2003a).

In empirical Bayesian applications, there are very many inference units (genes), and for each sample i there is a measurement on every unit. The methodology entails both discrete and continuous mixing over the parameter space in order to deliver posterior probability scores for each unit: $p_{\text{local},\pi} = P(M_\pi|X, y)$. We use the subscript 'local' to remind us that the the probability is local to the particular unit (e.g., gene), in the same way that the local false discovery rate is local to each testing unit in large-scale inference (Efron, 2005). The discrete mixing further involves probabilities estimated for the whole system, $p_{\text{global},\pi}$. By the use of maximum likelihood estimation, the fitted probabilities satisfy:

$$p_{\text{global},\pi} = \text{mean}_{\text{units}}\left(p_{\text{local},\pi}\right)$$

The conditional independence assumptions behind EBSeq induce a product-partition form on $p_{\text{local},\pi}$: with $X_b = \{X_i : y_i = k, \text{and } k \in b\}$ denoting all n_b measurements on a given unit (gene) from samples i whose condition status y_i maps them to a block b of partition π,

$$p_{\text{local},\pi} \propto p_{\text{global},\pi} \prod_{b\in\pi} f(X_b).$$

The proportionality is resolved by summing over all B_K partitions, which presents a computational challenge even for moderate K. In EBarrays, the joint predictive mass $f(x)$ takes either a compound Gamma form or a log-normal form; our focus is to improve EBSeq, which brings empirical Bayes methodology to RNA-Seq data. In EBSeq, $f(x)$ takes a special form as a Beta mixture of Negative Binomial (NB) mass functions. These distributional choices respond to properties and variance

characteristics of RNA-Seq data, and also lead to a convenient closed-form predictive mass function:

$$f(x) = \Big[\prod_{i=1}^{m}\binom{x_i+\gamma-1}{x_i}\Big]\frac{\text{Beta}(\alpha+m\gamma, \beta+\Sigma_{i=1}^{m}x_i)}{\text{Beta}(\alpha,\beta)} \tag{3.1}$$

EBSeq.v1 sets hyperparameters α, β, and γ as well as mixing rates $p_{\text{global},\pi}$ using both local data on the unit and global data on from all units. See Appendix A.13 for additional details.

3.3 Pruning

EBSeq.v1 fits a mixture over all B_K partitions. Even if this were computationally feasible for moderately large K, we expect in applications that very many, perhaps most, of these partitions would contribute a negligible amount to the fitted model. We propose a pruning algorithm that uses filtering statistics to identify partitions that are likely to carry most of the mixture mass without the need to fit the mixture over all partitions. The algorithm works by selecting probable partitions at each unit and taking their union as our global pool of partitions possibly of much smaller size than B_K.

Consider first a single inference unit (gene) and the group sample means $\widehat{\mu} = (\widehat{\mu}_1, ..., \widehat{\mu}_K)$:

$$\widehat{\mu}_k = \frac{\sum_{i=1}^{n} X_i 1[y_i = k]}{\sum_{i=1}^{n} 1[y_i = k]}$$

and let $r = (r_1, ..., r_K)$ represent the rank of $\widehat{\mu}$, i.e. r_k is the position of $\widehat{\mu}_k$ in the permutation rearranging $\widehat{\mu}$ into ascending order. We use an overlap concept that was also used in (Dahl, 2009b) to trim partitions in a modal clustering application. Two sets E_1 and E_2 of finite and integer elements overlap if E_1 contains a number between the smallest and largest numbers of E_2, or vice versa. For example, $E_1 = \{1, 3\}$ overlaps with $E_2 = \{2\}$, but $E_1 = \{1, 2\}$ and $E_2 = \{3\}$ do not overlap. Relative to a partition $\pi = \{b\}$, consider sets of indices $A_r(b) = \{r_k, k \in b\}$. Compatibility: A partition π is compatible with an empirical rank r if either π contains only one block or if for any two different blocks $b_1, b_2 \in \pi$, $A_r(b_1), A_r(b_2)$ do not overlap. For example with $K = 4$, and the empirical ordering $\widehat{\mu}_1 < \widehat{\mu}_2 < \widehat{\mu}_3 < \widehat{\mu}_4$ corresponds to rank $r = (1, 2, 3, 4)$.

The partitions compatible with r, and their corresponding hypotheses, are

$$\begin{aligned}
\{\{1,2,3,4\}\} &\quad \mu_1 = \mu_2 = \mu_3 = \mu_4 \\
\{\{1\},\{2,3,4\}\} &\quad \mu_1 \neq \mu_2 = \mu_3 = \mu_4 \\
\{\{1,2\},\{3,4\}\} &\quad \mu_1 = \mu_2 \neq \mu_3 = \mu_4 \\
\{\{1,2,3\},\{4\}\} &\quad \mu_1 = \mu_2 = \mu_3 \neq \mu_4 \\
\{\{1,2\},\{3\},\{4\}\} &\quad \mu_1 = \mu_2 \neq \mu_3 \neq \mu_4 \\
\{\{1\},\{2,3\},\{4\}\} &\quad \mu_1 \neq \mu_2 = \mu_3 \neq \mu_4, \mu_1 \neq \mu_4 \\
\{\{1\},\{2\},\{3,4\}\} &\quad \mu_1 \neq \mu_2 \neq \mu_3 = \mu_4, \mu_1 \neq \mu_3 \\
\{\{1\},\{2\},\{3\},\{4\}\} &\quad \text{All } \mu\text{s are distinct,}
\end{aligned}$$

while, for example, $\mu_1 = \mu_3 \neq \mu_2 = \mu_4$, i.e., $\{\{1,3\},\{2,4\}\}$, is not compatible with r. In fact, for the set $\mathcal{C}_g$ of partitions π that are compatible with the empirical ranks at unit g, we find:

Lemma 3.3.1. *The cardinality of $\mathcal{C}_g$ is $|\mathcal{C}_g| = 2^{K-1}$.*

Compatible partitions vary from unit to unit, and over the full system there may still be close to B_K different partitions that are compatible for at least one unit. But compatibility is a useful first property to consider as we filter the total number of partitions to a manageable number.

Sampling theory offers one argument in support of considering compatible partitions when the total number of samples n is large compared to the number of groups K. Then, deviations $\widehat{\mu} - \mu$ tend to be relatively small. For the pattern M_π corresponding to μ, π is not compatible with the empirical ranks only if for some pair of groups j and k, we have $\mu_j < \mu_k$ and also $\widehat{\mu}_j > \widehat{\mu}_k$. By the law of large numbers, this event is increasingly improbable as n increases.

We investigate compatibility empirically by running EBSeq.v1 in three example data sets where K is sufficiently small that computations are feasible: GSE45719 (Deng et al., 2014), GSE57872 (Patel et al., 2014), GSE74596 (Engel et al., 2016a). For each data set, Table 1 reports

$$p_{\text{compatible}} = \text{mean}_g \left(\sum_{\pi \in \mathcal{C}_g} p_{\text{local},\pi} \right),$$

which measures the posterior probability mass of compatible partitions in a fitted model. Compatible partitions cover most of the mixture mass in these cases. Table 1 also reports $N_{95\%}$, which measures the concentration of local posterior probability mass over the full set of partitions. Specifically, at each unit g we consider the most probable partitions and we count how few of these are required to capture at least 95% of the local probability mass, we then average over units to get $N_{95\%}$. We also keep track of $N_{c,95\%}$, which measures the average number of compatible partitions within the set capturing at least 95% local mass. Notice that $N_{c,95\%}$ and $N_{95\%}$ are much smaller than $|\mathcal{C}_g| = 2^{K-1}$, but we we seek still further pruning to have a manageable number of partitions when considering all units at once.

Data Set	N_{samples}	N_{units}	K	$p_{\text{compatible}}$	B_K	$N_{95\%}$	$N_{c,95\%}$
GSE45719	110	27083	4	95%	15	4.9	3.56
GSE74596	114	23337	6	82%	203	13.11	6.57
GSE52529	143	22112	7	85%	877	40.5	21.2

Table 1: Empirical properties of partitions in several data sets. $N_{\text{samples}}, N_{\text{units}}$ are numbers of samples and units

A key observation is that some pairs of groups present strong information that we can use to filter partitions. For example, at typical levels of variation seen in RNA-Seq data, observing $\widehat{\mu}_j = 10, \widehat{\mu}_k = 1000$ at two groups would suggest $\mu_j \neq \mu_k$, and we may not lose much by dropping partitions π asserting the opposite: $\mu_j = \mu_k$. Such information is measured by a Bayes factor comparing differential mean with equivalent mean, namely

$$D_{j,k} = \frac{P(X_{\{j,k\}}|\mu_j \neq \mu_k, y)}{P(X_{\{j,k\}}|\mu_j = \mu_k, y)} = \frac{f(X_j)f(X_k)}{f(X_{\{j,k\}})}$$

where $X_{\{j,k\}}$ represents vector of all expression values at a given unit for samples from groups j and k and f is the predictive density function from (3.1). If the two-group Bayes factor is sufficiently extreme (small or large), we are guided about partitions that may be dropped or included before doing a a full mixture computation.

In restricting to compatible partitions $\mathcal{C}_g$, the two-group Bayes factors are relevant to pairs of

groups that are adjacent after ranking by empirical mean. Let o_i be the antirank representing the group label having ith smallest sample mean. Then each $\pi \in \mathcal{C}_g$ is set by filling either $\neq$ or $=$ into the slots of equation (3.2),

$$\mu_{o_1} \_ \mu_{o_2} \_ \cdots \_ \mu_{o_K}. \tag{3.2}$$

These $K-1$ slots may be assessed using two-group Bayes factors. To identify a unit specific set of pruned partitions $\mathcal{S}_g$, we consider three assignment states for each slot in (3.2): (equivalent, differential, and uncertain) based on the Bayes factor and a user-specified threshold $t_1 > 0$.

$$\begin{aligned} \log(D_{o_i,o_{i+1}}) > t_1 &\Longrightarrow \mu_{o_i} \neq \mu_{o_{i+1}} \\ \log(D_{o_i,o_{i+1}}) < -t_1 &\Longrightarrow \mu_{o_i} = \mu_{o_{i+1}} \\ -t_1 < \log(D_{o_i,o_{i+1}}) < t_1 &\Longrightarrow \text{uncertain} \end{aligned}$$

Any partitions consistent with the filled-in (3.2) constitute a restricted (i.e. pruned) set of partitions $\mathcal{S}_g \subset \mathcal{C}_g$. The user-specified threshold t_1 gauges the size of $\mathcal{S}_g$, with larger t_1 being more inclusive and smaller values being more restrictive. We say $N_{\text{UC},g}$ is the number of uncertain positions at unit g, and we take $\mathcal{S} = \cup_g \mathcal{S}_g$ to be the selected set of partitions to be used in the full model-fitting computation. We say $N_{\text{selected}} = |\mathcal{S}|$. To prevent some corner case, we also provide users options to upperbound the number of uncertainty positions at each unit, details can be found in the appendix A.15

Algorithm 2 provides pseudo-code for the pruning algorithm of one unit.

To investigate the pruning algorithm we ran it on two example data sets that are too large for EBSeq.v1: Hrvatin (Hrvatin et al., 2018), Retina (Shekhar et al., 2016), and we set the threshold $t_1 = 1$. Table 2 shows that units $N_{\text{UC},g}$ is quite small for this threshold setting.

We also consider the pruning method in the smaller examples from Table 1. Here we can calculate the overall mixture mass (from the full model and EBSeq.v1) that is associated with the

Data Set	N_{samples}	N_{units}	K	mean $N_{\text{UC},g}$	max $N_{\text{UC},g}$	N_{selected}	B_K (million)
Hrvatin	2164	17228	12	0.016	4	354	4.2
Retina	15551	22156	14	0.008	4	527	190.9

Table 2: Properties of two-group Bayes factor filtering in two example data sets

selected set of partitions: $p_{\mathcal{S}} = \sum_{\pi \in \mathcal{S}} p_{\text{global},\pi}$. Table 3 confirms that the pruning algorithm is finding dominant mixture components.

Data Set	N_{selected}	B_K	$p_{\mathcal{S}}$
GSE45719	15	15	1
GSE74596	126	203	0.99
GSE52529	227	877	0.88

Table 3: Empirical properties of selected partitions in three example data sets.

There is no theoretical guarantee that pruning will retain the true, data-generating partitions. However, We take the union of local selected partitions to construct global pool, which make it less likely to miss a true signal. And we have theorem 3.3.1 partially supporting consistency of our selection rule. Namely it tells if the Bayes factor of the two adjacent groups is favorable for differential means. Then the partition maximizing the $p_{\text{local},\pi}$ will also have differential means for this two groups. The regularity conditions for theorem 3.3.1 essentially requiring $\widehat{\mu}_{o_{i+1}} - \widehat{\mu}_{o_i} \geq O(\frac{1}{min(n_{o_i}, n_{o_{i+1}})})$, which is easily satisfied given moderate size of samples.

Theorem 3.3.1. *If some regularity conditions are satisfied (see appendix). Then if $D_{o_i,o_{i+1}} > 1$, the partition $\pi^* = argmax\ p_{local,\pi}$ must have adjacent group o_i and o_{i+1} to be classified in different groups*

Algorithm 2 PRUNING FOR ONE UNIT

Input:

expression data at one gene X_g

subtype label $y = (y_i)$

threshold t_1 for Bayes factor

threshold t_2 to filter small mean values

threshold K^* for number of uncertainty positions

hyperparameters (α, β)

Output: $\mathcal{S}_g$: pruned partitions for gene g

procedure PRUNING($X_g, y, t_1, t_2, K^*, \alpha, \beta$)
  group sample means: $\widehat{\mu} \leftarrow X_g, y$
  empirical ordering of $\widehat{\mu}$: $o \leftarrow \widehat{\mu}$
  groups sizes: n_o
  initialize state between adjacent groups: $ST, ST_j \leftarrow$ uncertain
  uncertain positions: $UC \leftarrow \{\}$
  for o_j in o_1 to o_{K-1} **do**
    if max($\widehat{\mu}_{o_j}, \widehat{\mu}_{o_{j+1}}$) $< t_2$ **then**
      $ST_j \leftarrow =$
    else
      $D_{o_j,o_{j+1}} \leftarrow$ calculate two-group Bayes factor from($\widehat{\mu}_{o_j}, \widehat{\mu}_{o_{j+1}}, n_{o_j}, n_{o_{j+1}}, \alpha, \beta$)
      if $\log(D_{o_j,o_{j+1}}) > t_1$ **then**
        $ST_j \leftarrow \neq$
      else if $\log(D_{o_j,o_{j+1}}) < -t_1$ **then**
        $ST_j \leftarrow =$
      else
        $UC \leftarrow UC \cup \{o_j\}$
      end if
    end if
  end for
  $N_{UC} \leftarrow \min(K^*, |UC|)$
  $UC^* \leftarrow$ positions with smallest N_{UC} absolute value of log Bayes factors in UC.
  $\mathcal{S}_g \leftarrow$ use ST and UC^* to generate
end procedure

3.4 Crowding issue

In contrast to pruning, we also want to prevent the scenario that too few patterns are selected. This could happen when there is a long chain of equal states, which we called crowding phenomena.

$$\mu_{o_j} = \mu_{o_{j+1}} = \mu_{o_{j+2}} ... = \mu_{o_k}$$

Even though the Bayes factors for adjacent groups can be significant favorable for equal states, but the difference between head and tail of the chain may be big. In the pruning part, we do not compare groups when they are not adjacent. This may induce contradiction. For example, $D_{o_j,o_{j+1}}, D_{o_{j+1},o_{j+2}}$ are small and favorable for equivalent state, we have $\mu_{o_j} = \mu_{o_{j+1}} = \mu_{o_{j+2}}$. However $D_{o_j,o_{j+2}}$ may support uncertain or differential states. To further check the difference between μ_{o_j} and $\mu_{o_{j+2}}$, we provide an "equal-handle" algorithm to further investigate such equality chains and break them down to sub partitions when necessary. The algorithm adopts the idea for agglomerative hierarchical clustering. We iteratively check the state of adjacent groups with the smallest Bayes factor along the chain. If the state is equal, we combine them to form a new group. Otherwise we break the chain. Figure 3.1 demonstrate the algorithm on an equality chain of length 5. Algorithm 3 gives the equal-handle pseudo-code.

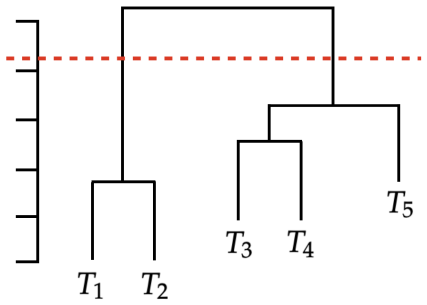


Figure 3.1: Illustration of equal-handle algorithm, $o_1, ..., o_5$ are groups on the equal chain. we sequentially merge two groups having the smallest Bayes factor(strongest evidence for having equal mean) to build a dendrogram. Red line is the threshold where we choose to break down the chain. There are two clusters, groups within the same cluster having same mean. The state between clusters become uncertain. Thus we have two patterns now 1) $o_1 = o_2 = o_3 = o_4 = o_5$ and 2) $o_1 = o_2 \neq o_3 = o_4 = o_5$

Algorithm 3 EQUALHANDLE FOR ONE UNIT

Input:

expression data at one gene X_g

subtype label $y = (y_i)$

positions for chain of equal states $\widetilde{o} = (o_j, ...o_k)$

vector of Bayes factors for adjacent elements along the chain $D_{\widetilde{o}} = (D_{o_j,o_{j+1}}, ...D_{o_{k-1},o_k})$

threshold t_1 for Bayes factor

Output: updated $\mathcal{S}_g$

procedure EQUALHANDLE(X_g, y, D_e, e, t_1)
    while $\min |\log(D_e)| > t_1$ **do**
        pick j^* such that $D_{o_{j^*},o_{j^*+1}} = \min(D_{\widetilde{o}})$
        update $\widetilde{o}$ by merging group o_{j^*} and o_{j^*+1}
        update $D_{\widetilde{o}}$ by recalculating Bayes factor between the merged group and its adjacent groups.

    end while
    if $|\widetilde{o}| > 1$ **then**
        mark state between groups that are not merged as uncertain
    else
        return
    end if
end procedure

3.5 Full algorithm

EBSeq.v2 works as follows: the global pool of partitions is determined by pruning and equalhandle, then EM algorithm is applied to estimate the parameters. We improve the optimization within the EM iterations. EBSeq.v1 does a full optimization within each M-step to update the hyper parameters (α, β) and the mixing rates $p_{\text{global},\pi}$. There is a closed formula for updated mixing rates but hyperparameter optimization by optim is more compute intensive. EBSeq.v2 uses one gradient step toward the optimal solution within each EM iteration. Numerical experiments in Section 3 demonstrate the two methods produce comparable results. Further, EBSeq.v1 is implemented primarily in R, which, like any other interpreted language, is relatively slow for intensive computing. We deploy core components of EBSeq.v2 in C++ for efficiency.

In some applications of EBSeq, we may be able to further prune partitions by $p_{\text{global},\pi}$. For example, in single cell study, many papers argue that the transcripts have a lower dimensional structure (Lopez et al., 2018). The co-expression characteristics of genes is a potential explanation and a consequence is that the true partitions will present themselves in quite a few genes and thus $p_{\text{global},\pi}$ can not be too small. Thus, we provide an option for users to prune more aggressively, which is achieved by iteratively throwing out partitions with small $p_{\text{global},\pi}$ after each run of EM. The complete framework is presented as Algorithm 4.

3.6 Results

We compare package versions EBSeq.v2 and EBSeq.v1 on benchmark data sets where the number of groups K is small. We go on to assess the performance of EBSeq.v2 in both synthetic and empirical data sets where K is larger.

3.6.1 Benchmarks with small K

We check EBSeq.v2 on the benchmark data sets built into EBSeq.v1. Since $K \leq 3$ in these cases, we use no pruning in deploying EBSeq.v2, and so this tests speed and the effect of hyper-parameter

Algorithm 4 EBSEQ.V2

Input:

GENES by SAMPLES expression data $\{X_{g,i}\}$

sample subtype labels $y = (y_i)$

threshold t_1 for Bayes factor

threshold t_2 to filter small mean values

threshold t_3 for further trimming

threshold K^* for number of uncertainty positions

minimum length L_e of equality chain for doing equal-Handle

initial values for hyper parameters $\alpha^{(0)}, \beta^{(0)}$

Output: final selected partitions $\mathcal{S}$, matrix of $\{p_{\text{local},\pi}\}$

procedure EBSEQ2($X_{g,c}, y, t_1, t_2, t_3, K^*, L_e$)
  for gene in Genome **do**
    $S_g \leftarrow$ run pruning algorithm
    if there is equality chain with length $> L_e$ **then**
      updated $S_g \leftarrow$ run equal-Handle algorithm
    end if
  end for
  $\mathcal{S} \leftarrow \cup_g \mathcal{S}_g$
  $p^{(0)}_{\text{global},\pi} \leftarrow \frac{1}{|\mathcal{S}|}$,uniform initialization
  $t = 0$, number of iterations.
  repeat
    if $t > 0$ **then**
      if $p^{(t)}_{\text{global},\pi} < t_3$ **then**
        $p^{(t)}_{\text{global},\pi} \leftarrow 0$
      end if
    end if
    E-step: $p^{(t+1)}_{\text{local},\pi} \leftarrow p^{(t)}_{\text{global},\pi} P(X|M_{g,\pi}, y, \alpha^{(t)}, \beta^{(t)})$
    M-step: $\alpha^{(t+1)}, \beta^{(t+1)}$ updated by one step gradient ascent.
    M-step: $p^{(t+1)}_{\text{global},\pi} \leftarrow \text{mean}(p^{(t+1)}_{\text{local},\pi})$
    $t = t + 1$
  until convergence criterion is met
end procedure

optimization differences between the algorithms. The proposed method expects normalized data: if input raw data, we provide a median normalization (Love et al., 2014b). Basically, function MedianNorm calculates the size factors that are further used for normalization and are passed in the EBTest function. EBSeq.v2 inherits all the functions from version 1.

The first example contains 1000 genes with 2 groups and 5 samples per group. We found that the correlation between the estimations of posterior of DE under two versions is 0.995.

We also consider another example, where we have 500 genes with 3 groups and 2 samples per group. At each gene, we consider the patterns π with maximum $p_{\text{local},\pi}$ under the two algorithms and use adjusted rand index (ARI) to measure their similarity. The average ARI over genes is 0.961, which also confirms the compatibility between the two versions of EBSeq. We increase the number of simulated genes to 10000 and compare the running time of two algorithms under $K = 2, 3$. Even at small K, EBSeq.v2 is a lot faster than EBSeq.v1 (Table 4)

EBSeq version	K	Average Run Time (minutes)
v1	2	3.3
v2	2	0.05
v1	3	45
v2	3	0.05

Table 4: Average run time comparison, 20 samples per group, 10000 genes. EM iteration for EBSeq.v1 is set to 5

3.6.2 Synthetic data, larger K

We use the Chinese restaurant process (CRP) over both genes and samples to simulate synthetic expression data with plausible variation characteristics. Each setting has 20000 units and 200 samples per group; details are in Appendix A.16.

We evaluate the performance of EBSeq.v2 with four metrics: coverage, extra patterns, scores and time. Let A be the set of patterns selected by EBSeq.v2, and B be the true underlying patterns. Coverage is $|A \cap B|/|B|$, extra patterns is $|A \setminus B|$. Coverage measures how well we can capture the underlying patterns. Extra patterns measures the efficiency, namely how many extra patterns

we are not able to filter out. For each gene, there is an underlying pattern π_g and an estimated pattern π'_g maximizing the posterior. We also consider adjusted Rand index (ARI) between π_g and π'_g to measure the accuracy of our estimation. Time is the CPU time of running EBSeq.v2. The computation was done using a single core 2.4GHz Intel Xeon E5645 with 126 Gb of RAM.

Figure 3.2 shows the results for 15 and 20 groups (1.38 billion and 51.7 trillion patterns). Our pruning algorithm covers almost every true patterns (Figure 3.2a). The ARI scores are close to 1 with small standard deviation (Figure 3.2c,3.2d), which shows our algorithm can accurately identify the true patterns. The number of extra patterns and time are acceptable and a great improvement to make the problem doable (Figure 3.2b,3.2e).

3.6.3 Empirical study

We apply EBSeq.v2 to three unique-molecule-index (UMI)-based scRNA-seq data sets. The first one is a mouse visual cortex dataset in which $47,209$ cells were classified into main cell types and subtypes through extensive analysis (Hrvatin et al., 2018). We preprocessed the data and consider a subset of $n = 2164$ cells and $K = 12$ cell types. Figure 3.3 showed log mean expression of each cell types at the genes favorable for a specific DE pattern, We can clearly see the differences between the four blocks and similarity within each block.

The second dataset contains cells from $K = 10$ purified populations derived from peripheral blood freely available from 10X Genomics, where 1,000 cells were sampled randomly from each cell type and combined to form a $n = 10,000$-cell data set. We carried out UMAP visualizations of the cells considering whole genome versus considering genes favorable for specific patterns(Figure 3.4). In the UMAP plot, cells are clustered together if we use those genes favorable for equivalent expression across all cell types. On the other hand, if we use genes favorable for other DE pattern, the projection reflected the difference of means between blocks.

The third dataset contained $n = 27,499$ mouse retinal bipolar cells, we used cluster annotation from $K = 14$ cell types from the author (Shekhar et al., 2016). We randomly selected 2 genes "Anp32a" and "BC030499" favorable for 2 patterns and show the cumulative distribution functions across the blocks (Figure 3.5). In the cdf plot, we observe distributional differences between cells

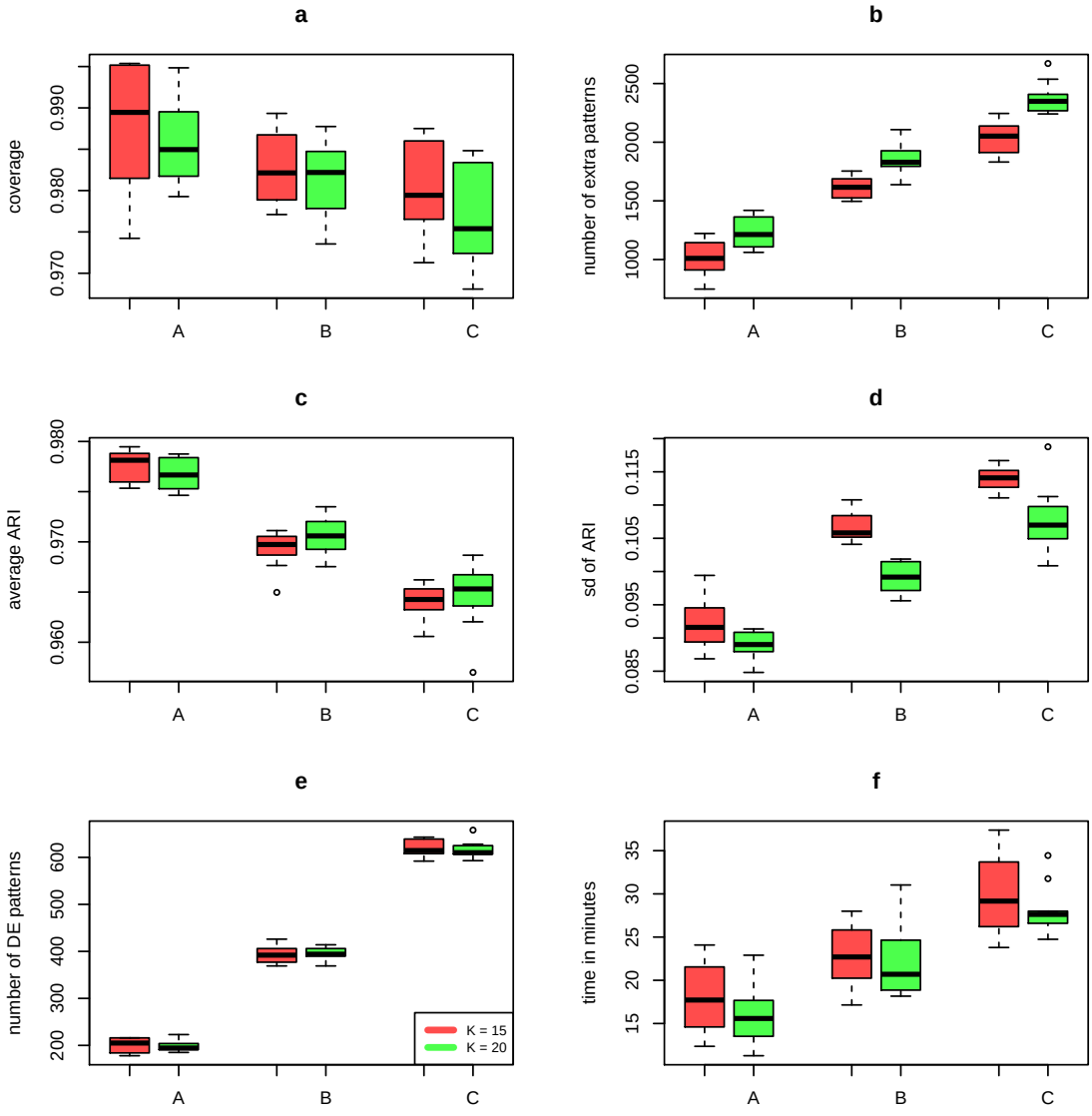


Figure 3.2: simulation setting: 200 samples each group, 20000 genes in total. We choose number of groups K to be either 15 or 20. Blocks of genes are generated from Chinese restaurant process, genes within the same block will have same DE patterns. x-axis label "A", "B", "C" represent 3 parameters setting (α_0) of the Chinese restaurant process governing total number of patterns underneath (here roughly 200, 400 and 600 patterns for each case). Under each choice of (K, α_0), we simulated 10 datasets. Here are the boxplots of the 10 datasets. Fig a presents the coverage percentage, Fig b presents the extra patterns we selected but does not belong to the set of true underlying DE patterns. Fig c presents the average ARI(adjusted rand index) between the MAP pattern and true pattern. Fig d presents the standard deviation of ARI. Fig e presents the number of underlying DE patterns across the genome. Fig f presents the computation time (minutes)

presenting differential means.

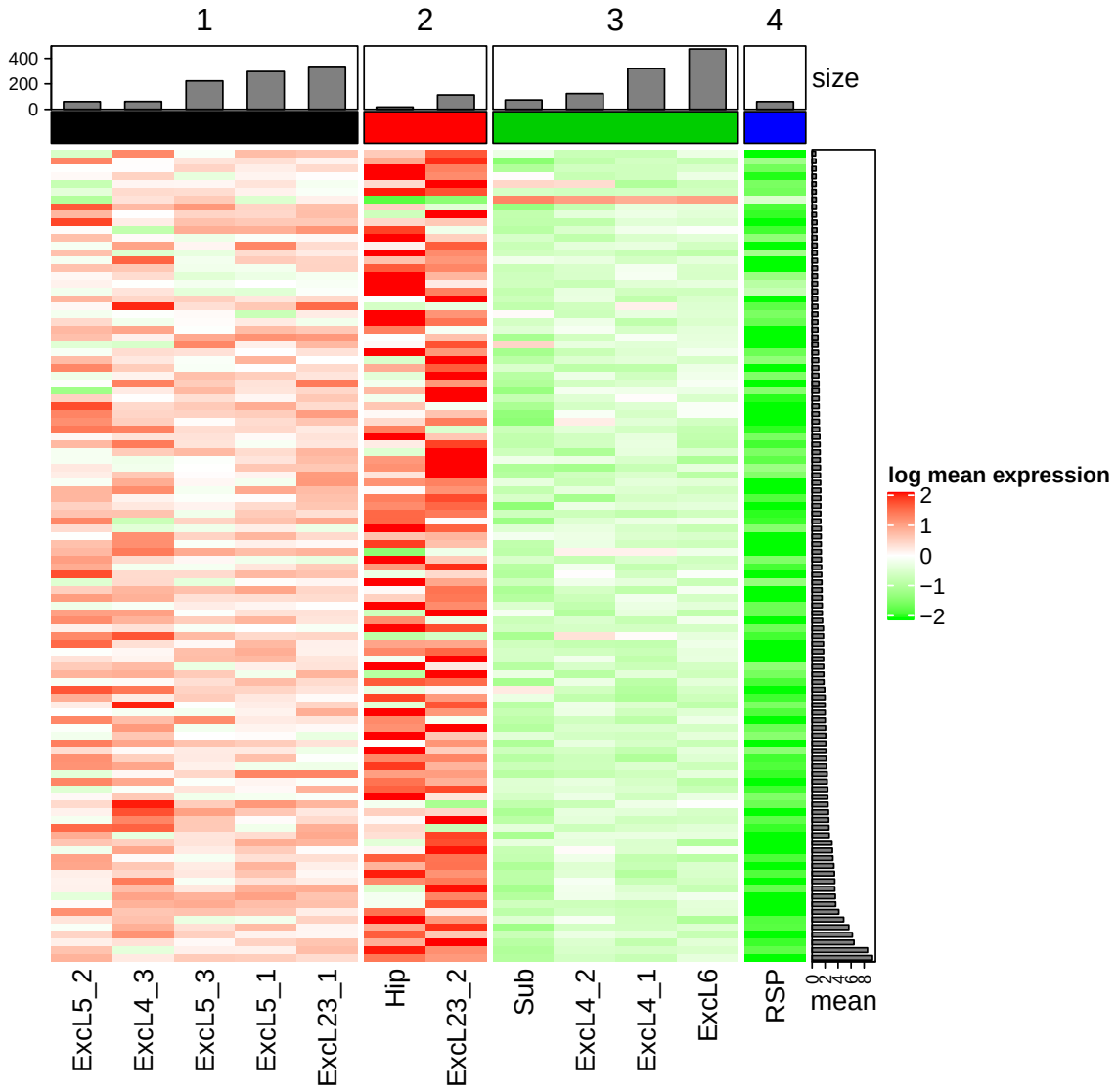


Figure 3.3: Heat-map of the mean expression on log scale across groups at those genes with one MAP pattern, genes are filtered by mean expression bigger than 0.5. The blocks are groups shared same mean. The top bar plot shows the number of cells each group ordered within each block, the right barplot shows the ordered marginal mean across all cells. Data are mouse cortex cells from (Hrvatin et al., 2018)

To further check the overall performance of EBSeq.v2, we consider another metric of consistency. Namely, we have empirical fold change of means over any two groups. We also have our

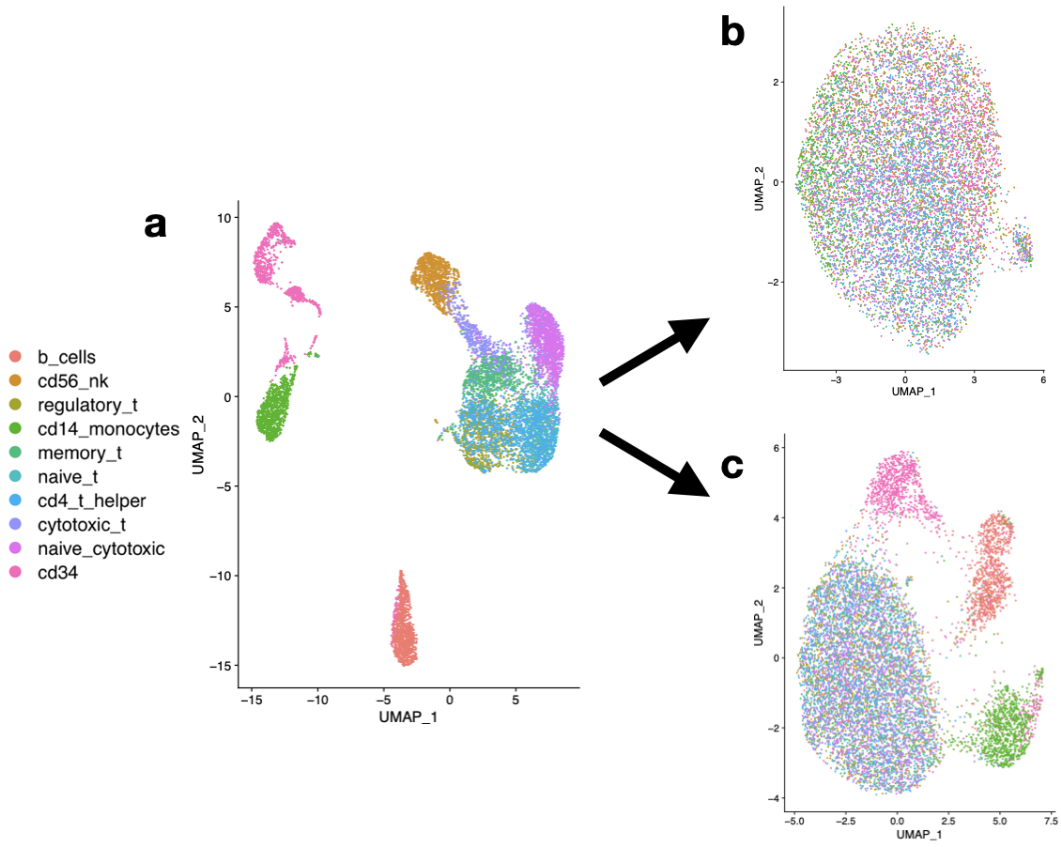


Figure 3.4: Umap of PBMC data, left considering whole genome. Top right considering those genes identified to have all equal means across cell types, bottom right considering those genes identified to have maximum a specific posterior pattern

posterior estimates of groups i and j having differential means as $P(\mu_i \neq \mu_j|X) = \sum_{\pi \in \Pi_{i,j}} p_{\text{local},\pi}$ where $\Pi_{i,j}$ is the collection of partitions that i and j belongs to different blocks. We average the empirical fold change and the posterior estimates over whole genome. From Figure 3.6, the upper triangle(averaged log fold change) is quite similar to the lower triangle(averaged posterior estimates).

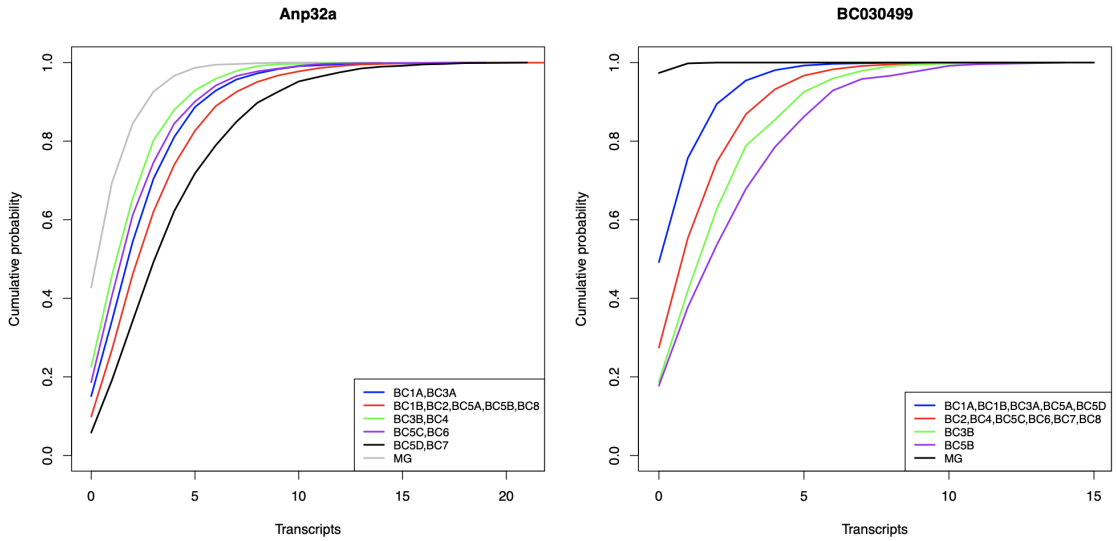


Figure 3.5: Cumulative distribution of transcripts at two genes "Anp32a" and "BC030499". Cells shared same mean are pooled. Using data from RETINA (Shekhar et al., 2016), bipolar cells from mouse

3.7 Summary and discussion

We have presented algorithms to accelerate and scale up EBSeq, a tool to score genes according to likely pattern of differential expression they display over two or more biological conditions. The pruning algorithm reduces the size of the mixture and serves as core for the acceleration when the number of conditions is moderately large. We worked out a theoretical result that partially supports the consistency of the pruning. The equal-handle algorithm deals with a corner case and keeps the model sensitive to the difference between groups. Through simulated and empirical examples we demonstrate the efficiency and accuracy of the new EBSeq.v2 tool.

Technologies and tools for RNA-seq data are still developing, and large-scale datasets composed of complex cell types are expected. Understanding patterns of mean expression among a lot of cell types can be an essential unit for many analysis, e.g. identifying genes with distributional changes across multiple conditions. Currently, negative binomial model approximates gene expression data well. One possible extension is to incorporate other densities (e.g. Gamma and log normal distribution) into EBSeq.v2 as they may better fit some data.

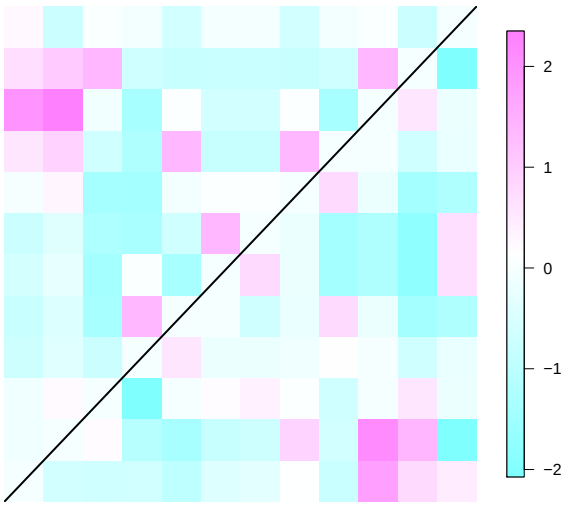


Figure 3.6: Heatmap of estimated log fold change v.s. posterior probabilities that two groups are DE using data (Hrvatin et al., 2018). Both the log fold change and posterior estimated are normalized by deducting corresponding mean and dividing by corresponding standard deviation. Given a gene, there are two matrices, one for log fold change and one for posterior probability of DE over all possible pair of groups. We average those matrices over all genes. All values are normalized by demean and divided by standard deviation. Upper triangle is for averaged log fold change and lower triangle is for averaged posterior of DE. We observe consistency between those two heatmaps, which demonstrates large differences are corresponding to high probability of DE while small differencs are corresponding to low probability of DE.

Computational details

The results in this paper were obtained using R 3.5.1 with the packages Rcpp 0.12.11, RcppEigen 0.3.2.9.0, BH 1.69.0-1. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Appendices

A.1 Proof of Theorem 2.2.1 in Chapter 2

If $\theta \in \bigcup_{\pi \in \Pi} [A_\pi \cap M_{g,\pi}]$, then there exists a partition π for which $\theta \in A_\pi$ and $\theta \in M_{g,\pi}$. By construction

$$f_g^1(x) = \sum_{k=1}^{K} \phi_k f_{g,k}(x) = \sum_{b \in \pi} \sum_{k \in b} \phi_k f_{g,k}(x) = \sum_{b \in \pi} \Phi_b f_{g,k^*(b)}(x),$$

where $k^*(b)$ indexes any component in b, since all components in that block have the same component distribution owing to constraint $M_{g,\pi}$. Continuing, using the constraint $\theta \in A_\pi$,

$$f_g^1(x) = \sum_{b \in \pi} \Psi_b f_{g,k^*(b)}(x) = f_g^2(x) \qquad \forall x.$$

That is, $\theta \in \text{ED}_g$.

If $\theta \in \text{ED}_g$, then $f_g^1(x) = f_g^2(x)$ for all x. Noting that both are mixtures over the same set of components $\{f_{g,k}\}$, let $\{h_{g,l} : l = 1, 2, \cdots, L\}$ be the set of distinct components over this set, and so

$$f_g^1(x) = \sum_{k=1}^{k} \phi_k f_{g,k}(x) = \sum_{l=1}^{L} c_{g,l}(\phi) h_{g,l}(x) = \sum_{l=1}^{L} c_{g,l}(\psi) h_{g,l}(x) = f_g^2(x)$$

where

$$c_{g,l}(\phi) = \sum_{k=1}^{K} \phi_k 1[f_{g,k} = h_{g,l}] \qquad c_{g,l}(\psi) = \sum_{k=1}^{K} \psi_k 1[f_{g,k} = h_{g,l}]. \tag{3}$$

Finite mixtures of distinct negative binomial components are identifiable (Proposition 5 from Yakowitz and Spragins (1968)), and so the equality of f_g^1 and f_g^2 implies $c_{g,l}(\phi) = c_{g,l}(\psi)$ for all $l = 1, 2, \cdots, L$. Identifying the partition blocks $b_l = \{k : f_{g,k} = h_{g,l}\}$, and the partition $\widetilde{\pi} = \{b_l\}$, we find $\theta \in A_{\widetilde{\pi}} \cap M_{g,\widetilde{\pi}}$. The accumulated probabilities in (3) correspond to $\Phi_{\widetilde{\pi}}$ and $\Psi_{\widetilde{\pi}}$, which are equal on $A_{\widetilde{\pi}}$.

A.2 Randomizing distances for approximate posterior inference

One way to frame the subtype problem is to suppose that subtype labels $z = (z_i)$ satisfy $z = f(\Delta)$, where $\Delta = (\delta_{i,j})$ is a $n \times n$ matrix holding *true*, unobservable distances, such as $\delta_{i,j}$ between cells i and j, and that f is some assignment function, like the one induced by the $K-$medoids algorithm. Then posterior uncertainty in z would follow directly from posterior uncertainty in Δ. On one hand, we could proceed via formal Bayesian analysis, say under a simple conjugate prior in which $1/\delta_{i,j} \sim \text{Gamma}(a_0, d_0)$, for hyperparameters a_0 and d_0, and in which the observed distance $d_{i,j}|\delta_{i,j} \sim \text{Gamma}(a_1, a_1/\delta_{i,j})$. This would assure that $\delta_{i,j}$ is the expectation of $d_{i,j}$, with shape parameter a_1 affecting variation of measured distances about their expected values. Not accounting for any constraints imposed by both D and Δ being distance matrices, we would have the posterior distribution $1/\delta_{i,j}|D \sim \text{Gamma}(a_0 + a_1, d_0 + a_1 d_{i,j})$. For any threshold $c > 0$, we would find

$$P\left(\delta_{i,j} \leq c|D\right) = P\left(U \geq \frac{d_0 + a_1 d_{i,j}}{c(a_0 + a_1)}\right) \tag{4}$$

where $U \sim \text{Gamma}(a_0 + a_1, a_0 + a_1)$

Alternatively, we could form randomized distances $d^*_{i,j} = d_{i,j}/w_{i,j}$ where $w_{i,j}$ is the analyst-supplied random weight distributed as $\text{Gamma}(\widehat{a}, \widehat{a})$ as in Section 2.2. Notice that

$$P(d^*_{i,j} \leq c|D) = P(w_{i,j} > d_{i,j}/c|D)$$

which is also an upper tail probability for a unit-mean Gamma deviate with shape and rate equal to $\widehat{a}$. Comparing to (4), by setting $\widehat{a}$ to equal $a_0 + a_1$, and if a_0 and d_0 are relatively small, we find

$$P(d^*_{i,j} \leq c|D) \approx P(\delta_{i,j} \leq c|D).$$

In other words, the randomized distance procedure is providing approximate posterior draws of the underlying distance matrix. In spite of limitations of this procedure for full Bayesian inference, it provides an elementary scheme to account for uncertainty in subtype allocations. Numerical experiments in Appendix make comparisons to a full, Dirichlet-process-based, posterior analysis.

Pseudo-code of scDDboost

Algorithm 5 SCDDBOOST

Input:

GENES by CELLS expression data matrix $X = (X_{g,c})$

cell condition labels $y = (y_c)$

number of cell subtypes K

number of randomized clusterings n_r

Output: posterior probabilities of differential distribution

procedure SCDDBOOST(X, y, K, n_r)
  distance matrix: $D = \mathrm{dist(X)} \leftarrow$ pairwise distances between cells (columns of X)
  hyper-parameters $(a_0, a_1, d_0) \leftarrow$ hyper(D). Set $\widehat{a} = a_0 + a_1$.
  repeat
    Gamma noise vector: e, with components $\sim$ Gamma($\widehat{a}/2, \widehat{a}$)
    randomized distance matrix: $D^* \leftarrow D/(e\mathbf{1}^T + \mathbf{1}e^T)$
    $\widehat{z}^* \leftarrow K-$medoids($D^*$)
    $P^* \leftarrow$ SCDDBOOST-CORE($X, y, \widehat{z}^*$)
  until n_r randomized distance matrices
  return $\forall$genes g, $P(\mathrm{DD}_g|X, y) = \frac{1}{n_r}\sum_{D^*} P^*_g$
end procedure

A.3 Empirical datasets

Data set	Conditions	# cells	Organism	Ref
GSE94383	0 min unstim vs 75min stim	186,145	human	Lane et al. (2017)
GSE48968-GPL13112	BMDC (2h LPS stimulation) vs 6h LPS	96,96	mouse	Shalek et al. (2014)
GSE52529	T0 vs T72	69,74	human	Trapnell et al. (2014c)
GSE74596	NKT1 vs NTK2	46,68	mouse	Engel et al. (2016b)
EMTAB2805	G1 vs G2M	95,96	mouse	Buettner et al. (2015)
GSE71585-GPL13112	Gad2tdTpositive vs Cux2tdTnegative	80,140	mouse	Tasic et al. (2016)
GSE64016	G1 vs G2	91,76	human	Leng et al. (2015)
GSE79102	patient1 vs patient2	51, 89	human	Kiselev et al. (2017)
GSE45719	16-cell stage blastomere vs mid blastocyst cell	50, 60	mouse	Deng et al. (2014)
GSE63818	Primordial Germ Cells, develop- mental stage: 7 week gestation vs Somatic Cells, developmental stage: 7 week gestation	40,26	mouse	Guo et al. (2015)
GSE75748	DEC vs EC	64, 64	human	Chu et al. (2016)
GSE84465	neoplastic cells vs non-neoplastic cells	1000, 1000	human	Darmanis et al. (2017)

Appendix Table A5: Data sets used for the empirical study of scDDboost

Data set	Condition	# cells
GSE63818null	7 week gestation	40
GSE75748null	DEC	64
GSE94383null	T0	186
GSE48968-GPL13112null	BMDC (2h LPS stimulation)	96
GSE74596null	NKT1	46
EMTAB2805null	G1	96
GSE71585-GPL13112null	Gad2tdTpositive	80
GSE64016null	G1	91
GSE79102null	patient1	51

Appendix Table A6: Single-condition data sets used in the random-splitting experiment.

A.4 Proof of Theorem 2.2.2 in Chapter 2

Proof. [Proof of Theorem 2] Recall $\theta = (\phi, \psi, \mu, \sigma)$. Through the graphical structure of our model (Figure 3), given n_1 and n_2 numbers of cells within each condition, we note that z^1, z^2 are multinomial draws, also given ϕ and ψ. Also given z, $X_{g,c}$ is sampled through NB(μ_{g,z_c}, σ_g), and only depends on (μ, σ). Thus $P(X, y, z|\theta) = P(y, z|\phi, \psi)P(X|z, \mu, \sigma)$, and also (μ, σ) and (ϕ, ψ) are independent a priori given π. By Bayes's rule (and always conditioning on π),

$$
\begin{aligned}
P(\theta|X, y, z) &\propto P(X, y, z|\theta)P(\theta) \\
P(X, y, z|\theta)P(\theta) &= P(y, z|\phi, \psi)P(X|z, \mu, \sigma)P(\mu, \sigma|z)P(\phi, \psi) \\
P(\phi, \psi|y, z) &\propto P(y, z|\phi, \psi)P(\phi, \psi) \\
P(\mu, \sigma|X, z) &\propto P(X|z, \mu, \sigma)P(\mu, \sigma|z) \\
\text{Thus } P(\theta|X, y, z) &\propto P(\phi, \psi|y, z)P(\mu, \sigma|X, z)
\end{aligned}
$$

It thus follows by integration over the parameter space that $P\left(A_\pi \cap M_{g,\pi}|X, y, z\right) = P\left(A_\pi|y, z\right)\ P\left(M_{g,\pi}|X, z\right)$.

A.5 EBSeq

Here we recall some key elements from Leng et al. (2013) on the model behind `EBSeq`, which we adapt to get $P(M_{g,\pi}|X, z)$. Suppose we have K subtypes, let $X_g^I = X_{g,1}^I, \cdots, X_{g,S_1}^I$ denote transcripts at gene g from subtype $I, I = 1, \cdots, K$. The `EBSeq` model assumes that counts within subtype I are distributed as Negative Binomial: $X_{g,s}^I|r_{g,s}, q_g^I \sim NB(r_{g,s}, q_g^I)$. Due to sample-specific size factor in the raw counts, r is made sample-specific. However, we are dealing with normalized counts rather than raw counts in `EBSeq`, we instead make r shared at gene level across all samples, i.e. $X_{g,s}^I|\sigma_g, q_g^I \sim NB(\sigma_g, q_g^I)$

$$
P(X_{g,s}^I|\sigma_g, q_g^I) = \binom{X_{g,s} + \sigma_g - 1}{X_{g,s}} (1 - q_g^I)^{X_{g,s}^I} (q_g^I)^{\sigma_g}
$$

and $\mu^I_{g,s} = \sigma_g(1 - q^I_g)/q^I_g$; For ease in later deriving the density kernel f, we use q rather than μ to parameterize the NB.

Following Leng et al. (2013), we assume a prior distribution on q^I_g : $q^I_g|\alpha, \beta^g \sim Beta(\alpha, \beta^g)$. The hyperparameter α is shared by the whole genome and β^g is gene-specific. We force the size factor to be 1 for all cells and use the same procedure as `EBSeq` to estimate the shape parameter σ_g. Namely, we have

1. gene-level sample mean $m_g = \frac{1}{n}\sum_{s=1}^{n} X_{g,s}$, where $n = n_1 + n_2$ is the total number of cells

2. average of sample variances over subtypes $v_g = \frac{1}{K}\sum_{I=1}^{K} v^I_g$.

3. v^I_g is the unadjusted sample variance for subtype I, i.e. $v^I_g = \frac{1}{n^I}\sum_{s,z_s=I}(X_{g,s} - m^I_g)^2$ where m^I_g is the sample mean within subtype I and n^I is the number of cells within subtype I.

We estimate the pooled over-dispersion rate by $o_g = \frac{v_g}{m_g}$ and obtain $\sigma_g = m_g\frac{o_g}{1-o_g}$ from the first moment of NB. Our aim is to quantify the expression pattern among K groups:

$$M_{g,\pi} = \{\theta \in \Theta : \ \mu_{g,k} = \mu_{g,k'} \iff k, k' \in b, b \in \pi\}.$$

For example, if $K = 3$, there are 5 expression patterns, which may be written equivalently in terms of parameters q:

$$P1 : q^1_g = q^2_g = q^3_g$$
$$P2 : q^1_g = q^2_g \neq q^3_g$$
$$P3 : q^1_g \neq q^2_g = q^3_g$$
$$P4 : q^1_g = q^3_g \neq q^2_g$$
$$P5 : q^1_g \neq q^2_g \neq q^3_g \text{ and } q^1_g \neq q^3_g$$

In a pattern where two groups I and J share the same q_g the counts from these groups are essentially pooled: i.e. $X_g^{I,J}|\sigma_g, q_g \sim NB(\sigma_g, q_g)$, $q_g|\alpha, \beta^g \sim$ Beta(α, β^g). The prior predictive function is $f(X_g^{I,J}) = \int_0^1 P(X_g^{I,J}|r_g, q_g) * P(q_g|\alpha, \beta^g) dq_g = \left[\prod_{s=1}^{S} \binom{X_{g,s}+\sigma_g-1}{X_{g,s}}\right] \frac{Beta(\alpha+\Sigma_{s=1}^S \sigma_g, \beta^g+\Sigma_{s=1}^S X_{g,s})}{Beta(\alpha,\beta^g)}$. Consequently, the prior predictive function for $P1, \cdots, P5$ takes a convenient form if we further treat the distinct q's as independently drawn from the common Beta mixing distribution:

$$
\begin{aligned}
h_1^g(X_g) &= f(X_g^{1,2,3}) \\
h_2^g(X_g) &= f(X_g^{1,2}) f(X_g^3) \\
h_3^g(X_g) &= f(X_g^1) f(X_g^{2,3}) \\
h_4^g(X_g) &= f(X_g^{1,3}) f(X_g^2) \\
h_5^g(X_g) &= f(X_g^1) f(X_g^2) f(X_g^3)
\end{aligned}
$$

where $h_k^g(X_g) = P(X_g|M_{g,\pi_k}, z)$ for the associated pattern π_k. Then the marginal distribution of count vector X_g is $\sum_{k=1}^{5} p_k h_k^g(X_g)$, where the mixing mass $p_k = P(M_{g,\pi}|z)$ is shared by all genes. Then, the posterior probability of an expression pattern k is obtained by:

$$
\frac{p_k h_k^g(X_g)}{\sum_{l=1}^{5} p_l h_l^g(X_g)}.
$$

In the optimization for determining the hyperparameters (α, β^g, p), we use EM for the mixing proportions and we use in each cycle a single gradient ascent step for α and β^g, in contrast to a full root-finding step used by `EBSeq`.

A.6 modalClust

In this section, we review and extend Dahl's modal clustering procedure (Dahl (2009a)). This extension is part of the default cell clustering method of `scDDboost`. It operates on data from one

gene at a time, and extends to Poisson-distributed observations the modal-clust procedure.

Product Partition Model (PPM): Let $X = (X_1, X_2, ..., X_n)$ be a vector of data (say at one gene). Given a partition $\pi = \{S_1, \cdots, S_q\}$, where S_i are disjoint subsets of $\{1, 2, \cdots, n\}$ and $\bigcup_{i=1}^{q} S_i = \{1, 2, \cdots, n\}$, a PPM for X entails

$$p(X|\pi) = \prod_{i=1}^{q} f(X_{S_i})$$

where X_{S_i} is the vector of observations corresponding to the items of component S_i. The component likelihood $f(X_S)$ is defined for any non-empty component S and can take many forms. The partition π, which clusters cells, is the parameter we are interested in. Other parameters that may have been involved in the model are integrated out. (Note the partition here has no relation to the partition of subtypes, as, e.g. in Figure 3.)

When the prior distribution for a partition π also takes a product form then so does the posterior. We aim to compute the MAP partition (maximizing the posterior $p(\pi|X) \propto p(X|\pi)p(\pi)$) to be used as an initial estimated clustering. Dahl (2009a) demonstrated that by some choice of f and prior of π, we can reduce the time complexity of finding the MAP partition to $O(n^2)$. The crucial condition for f is the **non-overlapping** condition: if X_{S_1} and X_{S_2} are overlapped in the sense that $\min\{X_{S_2}\} < \max\{X_{S_1}\} < \max\{X_{S_2}\}$ or $\min\{X_{S_1}\} < \max\{X_{S_2}\} < \max\{X_{S_1}\}$, let $X_{S_1^*}$ and $X_{S_2^*}$ be the sets of swapping one pair of those overlapped terms and keep the other unchanged. Then $f(X_{S_1})f(X_{S_2}) \leq f(X_{S_1^*})f(X_{S_2^*})$. Here we confirm the non-overlapping condition for Poisson-Gamma observations.

Under the non-overlapping condition of density kernel f, the MAP partition π satisfies that for any two blocks $b_1, b_2 \in \pi$, either $\max_{i \in b_1}(X_i) \leq \min_{j \in b_2}(X_j)$ or $\min_{i \in b_1}(X_i) \geq \max_{j \in b_2}(X_j)$. Thus we reduce the solution space and reduce the time complexity. In the Poisson-Gamma model we have:

$$\begin{aligned} X_i|\pi, \lambda &\sim Poisson(X_i|\lambda_1 \mathbf{I}\{i \in S_1\} + \cdots + \lambda_q \mathbf{I}\{i \in S_q\}) \\ \pi &\sim p(\pi) \\ \lambda_j &\sim Gamma(\alpha_0, \beta_0) \end{aligned}$$

where $p(\pi) \propto \prod_{i=1}^{q} \eta_0 \Gamma(|S_i|)$. Integrate out λ, $f(X_S)$ is obtained as:

$$f(X_S) = \frac{\beta^\alpha}{(|S|+\beta)^{\sum_{i\in S} X_i+\alpha}} \frac{\Gamma(\sum_{i\in S} X_i+\alpha)}{\Gamma(\alpha)} \frac{1}{\prod_{i\in S} X_i}$$

To apply modal-clustering on Poisson-Gamma model, we need to show the kernel $f(X_S)$ satisfies the non-overlapping condition.

Proof. if X_{S_1} and X_{S_2} are overlapping, without loss of generality, we assume $\min\{X_{S_2}\} < \max\{X_{S_1}\} < \max\{X_{S_2}\}$, and we swap $\max\{X_{S_1}\}$ with $\min\{X_{S_2}\}$ and keep the rest unchanged or we could also swap $\max\{X_{S_1}\}$ with $\max\{X_{S_2}\}$. We denote the new set formed by swap of $\max\{X_{S_1}\}$ with $\min\{X_{S_2}\}$ as S_1^* and S_2^* and swap of $\max\{X_{S_1}\}$ with $\max\{X_{S_2}\}$ as S_1^{**}, S_2^{**} accordingly.

Then we need to show at least one of the following happens

$$f(X_{S_1^*}) f(X_{S_2^*}) \geq f(X_{S_1}) f(X_{S_2}) \tag{5}$$

$$f(X_{S_1^{**}}) f(X_{S_2^{**}}) \geq f(X_{S_1}) f(X_{S_2}) \tag{6}$$

Let $a = \max\{X_{S_1}\}$, $b = \min\{X_{S_2}\}$ and $c = \max\{X_{S_2}\}$. $h_1 = \sum_{i\in S_1} X_i - a$ and $h_2 = \sum_{i\in S_2} X_i - b$, n_1 and n_2 are the number of elements in S_1 and S_2. Then

$$f(X_{S_1^*}) f(X_{S_2^*}) \geq f(X_{S_1}) f(X_{S_2})$$

$$\Longleftrightarrow$$

$$\frac{\Gamma(h_1+a+\alpha)}{(n_1+\beta)^{h_1+a+\alpha}} \frac{\Gamma(h_2+b+\alpha)}{(n_2+\beta)^{h_2+b+\alpha}} \leq \frac{\Gamma(h_2+a+\alpha)}{(n_2+\beta)^{h_2+a+\alpha}} \frac{\Gamma(h_1+b+\alpha)}{(n_2+\beta)^{h_1+b+\alpha}}$$

$$\Longleftrightarrow$$

$$\frac{\Gamma(h_1+a+\alpha)}{\Gamma(h_1+b+\alpha)} \frac{\Gamma(h_2+b+\alpha)}{\Gamma(h_2+a+\alpha)} \leq \left(\frac{n_1+\beta}{n_2+\beta}\right)^{a-b}$$

The left hand side of the above formula is $\text{LHS}_1 = \frac{(h_1+b+\alpha)\ldots(h_1+a-1+\alpha)}{(h_2+b+\alpha)\ldots(h_2+a-1+\alpha)}$ by the property of

Gamma function and that X_i are integers.

Similarly,

$$f(X_{S_1^{**}})f(X_{S_2^{**}}) \geq f(X_{S_1})f(X_{S_2})$$

$$\Longleftrightarrow$$

$$\frac{\Gamma(h_2+c+\alpha)}{\Gamma(h_2+a+\alpha)}\frac{\Gamma(h_1+a+\alpha)}{\Gamma(h_1+c+\alpha)} \leq (\frac{n_2+\beta}{n_1+\beta})^{c-a}$$

The left hand side of above formula is $\text{LHS}_2 = \frac{(h_2+a+\alpha)...(h_2+c-1+\alpha)}{(h_1+a+\alpha)...(h_1+c-1+\alpha)}$.

If $h_1 \leq h_2$, then $\text{LHS}_1 \leq (\frac{h_1+a-1+\alpha}{h_2+a-1+\alpha})^{a-b}$ and $\text{LHS}_2 \leq (\frac{h_2+c-1+\alpha}{h_1+c-1+\alpha})^{a-b}$.

So if $\frac{h_1+a-1+\alpha}{h_2+a-1+\alpha} \leq \frac{n_1+\beta}{n_2+\beta}$ then (12) holds, if $\frac{h_2+c-1+\alpha}{h_1+c-1+\alpha} \leq \frac{n_1+\beta}{n_2+\beta}$ then (13) holds.

We multiply those two inequalities, and find that $\frac{h_1+a-1+\alpha}{h_2+a-1+\alpha} * \frac{h_2+c-1+\alpha}{h_1+c-1+\alpha} = \frac{h_1+a-1+\alpha}{h_1+c-1+\alpha} * \frac{h_2+c-1+\alpha}{h_2+a-1+\alpha} \leq 1$ as $c > a$ and $h_1 \leq h_2$. But $\frac{n_1+\beta}{n_2+\beta} * \frac{n_1+\beta}{n_2+\beta} = 1$. At least one equality holds, consequently at least one of (12) and (13) holds.

The proof for case $h_1 > h_2$ follows similarly.

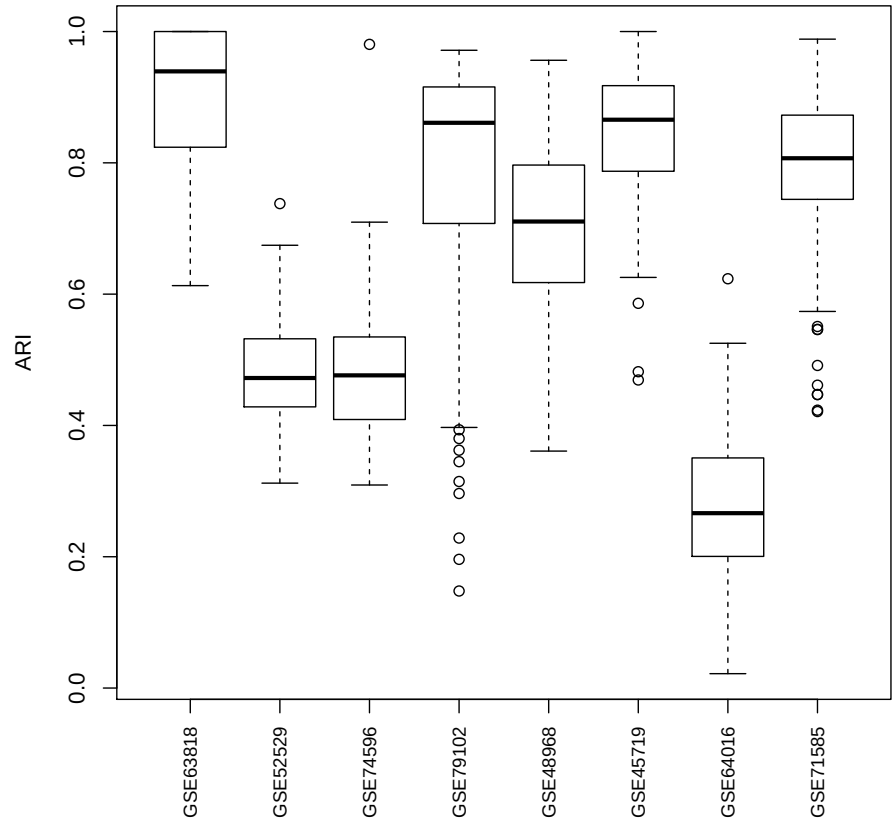
A.7 Randomized $K-$means

In this section, we consider parameters for the distribution of random weights and some properties the induced distribution over cell clusterings. Referring to the appendix in the main paper, to find the value of a_0, a_1 and d_0, we have the marginal likelihood of $d_{i,j}$

$$P(d_{i,j}|a_0, a_1, d_0) = \frac{\Gamma(a_0+a_1)}{\Gamma(a_0)\Gamma(a_1)} \frac{d_0^{a_0} d_{i,j}^{a_1-1} a_1^{a_1}}{(d_0 + a_1 * d_{i,j})^{a_0+a_1}}$$

We estimate d_0 by treating $d_{i,j} \approx \Delta_{i,j}$ and based on the mean-variance ratio ($\frac{\text{E}(1/\Delta_{i,j})}{\text{Var}(1/\Delta_{i,j})} = d_0$), d_0 can be approximately estimated by moments of $1/d_{i,j}$. Then we obtain a_0, a_1 from maximizing marginal density of $d_{i,j}$. The MLE estimators are obtained through `nlminb` function in `R`. One issue that arises is that the default value for tolerance rate of stopping is 1e-10, which yields large value of $a_1 + a_0$ and results in non-randomness of our weighting matrix. To avoid this issue, we set

tolerance rate as 1e-3 to obtain moderate deviation from D (Appendix Figure A7).



Appendix Figure A7: Adjusted RAND index of clusterings generated by randomizing distances. We investigate the variation of clustering given by random weighting through 8 datasets and each dataset we are using 100 random distances.

We plot the ARI (adjusted RAND index) between randomly generated clusterings to the clustering from the original distances across eight datasets. The boxplots indicate that random weighting is inducing substantial variation in the distribution of cell partitions.

We also check validity of random weighting by comparing it to Dirichlet-process-based clustering (Jara et al. (2011)) on a simulated dataset. We simulate one-dimensional data X from a mixture of 5 normal distributions with different means and same variance ($\mu = (-6, -2, 0, 2, 10), \sigma = 1$). We compare clustering results between random weighting and Bayesian clustering using the Dirich-

let process prior (using `DPpackage`) in terms of posterior probabilities that two elements belong to the same class given the whole data. We also compare accuracy of the two procedures by looking at the ARI comparing to true class label (Appendix Figure A8). We find that random weighting scheme closely matches the distributional features of the Dirichlet-process computation, and, in this case, tends to put more mass close to the data-generating partition.

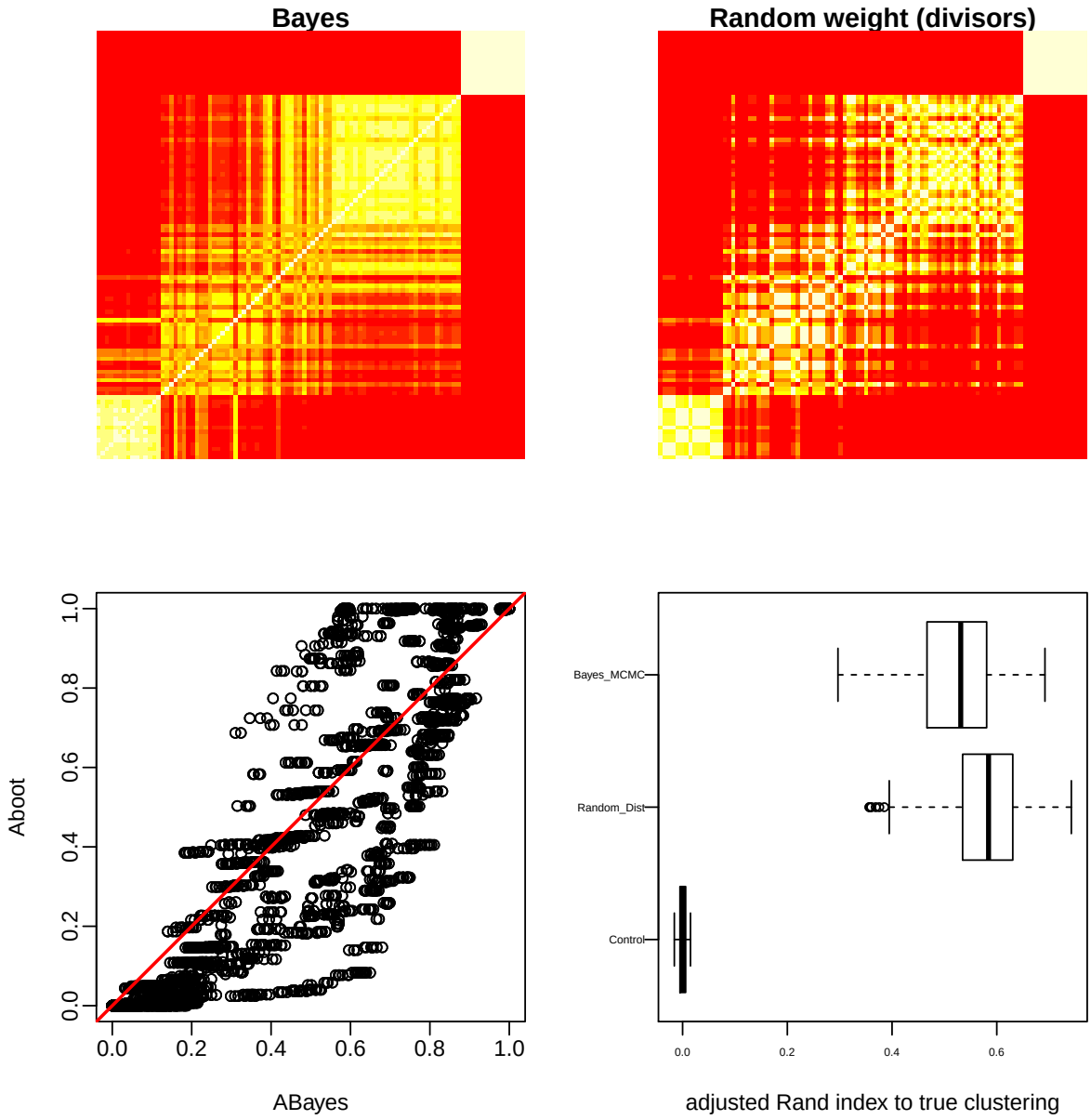
A.8 Selecting K

In this section, we give the criterion to select the number of subtypes K. We implement a procedure inspired by `validity`, as defined in Ray and Turi (2000). We consider a modified `validity`= $\frac{\textbf{intra}}{\textbf{inter}}$, where **intra** $= \frac{1}{N}\sum_{i=1}^{K}\sum_{x\in C_i}||x - z_i||^2$, **inter** $=$ mean$(||z_i - z_j||^2), i, j = 1, 2, ...K$, and z_i is the center (medoid) of cluster i. **intra** is the average of distance of a point to the center of its corresponding cluster, which measures the compactness of clusters. **inter** is the average distance between centers, which measures the separation between clusters. In the original paper **inter** was defined as minimum distance between medoids (Ray and Turi, 2000). Here, we instead use the average for smoothness. By minimizing `validity` (contrary to what the name suggests) we aim for small intra-cluster distance and large inter-cluster distance. We find empirically that `validity` is monotonic decreasing with K and this trend stabilizes when K is sufficiently big. We select the first K satisfying $\texttt{validity}_K < \epsilon$. We set the default value of ϵ to be 1, as we found this yields good performance in simulation.

A.9 Double Dirichlet Mixture

In this section, we give proofs for the properties of DDM in Section 2.3 of the main paper. Using notation from the main paper, we have density functions:

$$p_\pi(\phi, \psi) = q_\pi(\Phi_\pi, \Psi_\pi) \prod_{b\in\pi}\left[p(\widetilde{\phi}_b)p(\widetilde{\psi}_b)\right]$$



Appendix Figure A8: Comparison between random weighting scheme and Dirichlet-process procedure. Top: heatmap of probabilities that two elements belong to the same class given the whole data. Bottom: scatterplot of these posterior probabilities (left), and adjusted RAND index comparing to the underlying true class label (right).

with

$$q_\pi(\Phi_\pi, \Psi_\pi) = \frac{\Gamma(\sum_{b\in\pi}\beta_b)}{\prod_{b\in\pi}\Gamma(\beta_b)}\left[\prod_{b\in\pi}\Phi_b^{\beta_b-1}\right] 1\left[\Phi_\pi = \Psi_\pi\right]$$

and

$$p(\widetilde{\phi}_b) = \frac{\Gamma(\sum_{k\in b}\alpha_k)}{\prod_{k\in b}\Gamma(\alpha_k)}\prod_{k\in b}\widetilde{\phi}_k^{\alpha_k-1}, \qquad p(\widetilde{\psi}_b) = \frac{\Gamma(\sum_{k\in b}\alpha_k)}{\prod_{k\in b}\Gamma(\alpha_k)}\prod_{k\in b}\widetilde{\psi}_k^{\alpha_k-1}.$$

These serve as key components for proving DDM properties.

Proof. [Proof of Property 1] When ϕ and ψ only satisfy the coarsest constraint: $\sum_{i=1}^{K}\phi_i = \sum_{i=1}^{K}\psi_i = 1$, then ϕ and ψ are independently Dirichlet distributed. Finer constraints will lead to dependency between ϕ and ψ as there is a proper subset b of π such that $\sum_{i\in b}\phi = \sum_{i\in b}\psi$, which make $P(\phi|\psi) \neq P(\phi)$.

Proof. [Proof of Property 2] By the law of total expectation, $E_\pi(\phi_k) = E_\pi(E_\pi((\phi_k|\Phi_b)) = E_\pi(E_{\widetilde{\phi}_b}(\widetilde{\phi}_k)) = E_{\widetilde{\phi}_b}(\widetilde{\phi}_k)E_\Phi(\Phi_b)$ where b is the block containing subtype index k. Since $\widetilde{\phi}_b \sim \text{Dirichlet}_{N(b)}[\alpha_b^1]$ and $\Phi_\pi \sim \text{Dirichlet}_{N(\pi)}[\beta_\pi]$, we have $E_{\widetilde{\phi}_b}(\widetilde{\phi}_k) = \frac{\alpha_k^1}{\sum_{k'\in b}\alpha_{k'}^1}$, $E_\Phi(\Phi_b) = \frac{\beta_b}{\sum_{b'\in\pi}\beta_{b'}}$ and $E_\pi(\phi_k) = \frac{\alpha_k^1}{\sum_{k'\in b}\alpha_{k'}^1}\frac{\beta_b}{\sum_{b'\in\pi}\beta_{b'}}$. The case for $E_\pi(\psi_k)$ is similar.

Proof. [Proof of Property 3] t^1/t_π^1 is independent of t^2/t_π^2 conditioning on t_π^1 and t_π^2 by the neutrality property of Dirichlet distribution

Proof. [Proof of Property 4] For $j = 1, 2$, let T_b^j be the vector of t_k^j such that $k \in b$. Recall $t_b^j = \sum_{k\in b} t_k^j$. Without loss of generality, we consider the case condition $j = 1$. At the support of p_π, for different blocks, $T_b^1|\widetilde{\phi}_b$ are mutually independent. Then we have factorization:

$$p_\pi(t^1|t_\pi^1, y) = \prod_{b\in\pi} p(T_b^1|t_b^1, y)$$

and right hand side prior predictive function can be obtained by integrating out $\widetilde{\phi}_b$. Namely

$$\begin{aligned} p(T_b^1|t_b^1, y) &= \int_{\widetilde{\phi}_b} p(T_b^1|\widetilde{\phi}_b)p(\widetilde{\phi}_b)d\widetilde{\phi}_b \\ &= \left\{\left[\frac{\Gamma(t_b^j+1)}{\prod_{k\in b}\Gamma(t_k^j+1)}\right]\left[\frac{\Gamma(\sum_{k\in b}\alpha_k^j)}{\prod_{k\in b}\Gamma(\alpha_k^j)}\right]\left[\frac{\prod_{k\in b}\Gamma(\alpha_k^j+t_k^j)}{\Gamma(t_b^j+\sum_{k\in b}\alpha_k^j))}\right]\right\} \end{aligned}$$

given the prior Dirichlet$[\alpha_b^1]$ of $\widetilde{\phi}_b$ and that $p(T_b^1|\widetilde{\phi}_b)$ is a multinomial$(\widetilde{\phi}_b)$ distribution.

Proof. [Proof of Property 5] t_π^1 and t_π^2, given the condition label y, are independent and identically distributed with $t_\pi^1|\Phi \sim \text{multinomial}(\Phi)$. Thus

$$\begin{aligned} p_\pi(t_\pi^1, t_\pi^2|y) &= \int_\Phi p(t_\pi^1|\Phi)p(t_\pi^2|\Phi)p(\Phi)d\Phi \\ &= \left[\frac{\Gamma(n_1+1)\Gamma(n_2+1)}{\prod_{b\in\pi}\Gamma(t_b^1+1)\Gamma(t_b^2+1)}\right]\left[\frac{\Gamma(\sum_{b\in\pi}\beta_b)}{\prod_{b\in\pi}\Gamma(\beta_b)}\right]\left[\frac{\prod_{b\in\pi}\Gamma(\beta_b+t_b^1+t_b^2)}{\Gamma(n_1+n_2+\sum_{b\in\pi}\beta_b)}\right]. \end{aligned}$$

As prior of Φ is Dirchlet$[\beta]$ and $n_j = \sum_{b\in\pi} t_b^j$ for $j = 1, 2$.

To prove Property 6, we need a fact about dimensionality of the intersection of two A_π's.

Lemma .0.1. *If π_2 is not a refinement of π_1 then $A_{\pi_1} \cap A_{\pi_2}$ is a lower dimensional subset of A_{π_2}.*

Proof. [Proof of Lemma 1] To formalize the problem in linear algebra, we consider the vector space R^{2K}, and define a map from block to vector in R^K :$g(b) = v_b$, where ith component of v_b is 1 if $i \in b$ and 0 otherwise.

Let V_1, V_2 denote the orthogonal space of $\phi - \psi$ when $(\phi, \psi) \in \cap A_{\pi_2}, A_{\pi_2}, A_{\pi_1},$. Notice that $\dim(A_{\pi_1} \cap A_{\pi_2}) = \dim(\phi - \psi) + \dim(\psi) = K - \dim(V_1) + K - 1 = 2K - \dim(V_1) - 1$, $\dim(A_{\pi_2}) = 2K - \dim(V_2) - 1$, $\dim(V_2) = N(\pi_2)$. Assuming $\pi_1 = \{b_1^1, \cdots, b_s^1\}$, and $\pi_2 = \{b_1^2, \cdots, b_t^2\}$. The corresponding vectors are $v_1^1, \cdots, v_s^1$ and $v_1^2, \cdots, v_t^2$. We claim there must be a $b_i^1 \in \pi_1$ whose corresponding vector v_i^1 is linear independent with $v_1^2, \cdots, v_t^2$. If not, for every v_i^1 there exists $\alpha_1^i, \cdots, \alpha_t^i$ such that

$$v_i^1 = \sum_{j=1}^{t} \alpha_j^i v_j^2 \qquad (*)$$

If $b_j^2 \cap b_i^1 \neq \emptyset$, then $v_i^1 * v_j^2 > 0$ and we multiply v_j^2 on both sides of (*), we obtain $v_i^1 * v_j^2 = \alpha_j^i (v_j^2)^2$, as $v_p^2 * v_q^2 = 0$ if $p \neq q$. This implies $\alpha_j^i > 0$. Consider $x = g(b_j^2 \setminus b_i^1)$. We have $x * v_i^1 = 0$ and multiply x on both sides of (*) to obtain $\alpha_j^i v_j^2 * x = 0$. Thus x must be the zero vector and $b_j^2 \setminus b_i^1 = \emptyset$, which implies $b_j^2 \subset b_i^1$. That is to say when $b_j^2 \cap b_i^1 \neq \emptyset$, b_j^2 must be a subset of b_i^1. So b_i^1 is the union of some blocks in π_2. This implies π_2 is a refinement of π_1, which is a contradiction. Consequently, there exists $b \in \pi_1$ whose v_b is linear independent with $v_{b'}, \forall b' \in \pi_2$. Thus the $\dim(V_1)$ is is at least $N(\pi_2) + 1, \dim(A_{\pi_1} \cap A_{\pi_2}) < \dim(A_{\pi_2})$.

Proof. [Proof of Property 6] For any π, $P(A_\pi, |y, z) = \sum_{\widetilde{\pi}\in\Pi} \int_{A_\pi} \omega_{\widetilde{\pi}}^{\text{post}} d\phi d\psi$, notice the support of $\omega_{\widetilde{\pi}}^{\text{post}}$ is $A_{\widetilde{\pi}}$. By Lemma 1, we know if $\widetilde{\pi}$ does not refine π, then $\int_{A_\pi} \omega_{\widetilde{\pi}}^{\text{post}} d\phi d\psi$ is an integral on lower dimension set and vanishes. if $\widetilde{\pi}$ refines π, then $\int_{A_\pi} \omega_{\widetilde{\pi}}^{\text{post}} d\phi d\psi = \int_{A_{\widetilde{\pi}}} \omega_{\widetilde{\pi}}^{\text{post}} d\phi d\psi = \omega_{\widetilde{\pi}}^{\text{post}}$. We have $P(A_\pi, |y, z) = \sum_{\widetilde{\pi}\in\Pi} \omega_{\widetilde{\pi}}^{\text{post}} 1[\widetilde{\pi} \text{ refines } \pi]$.

Proof. [Proof of Theorem 3] Recall the DDM prior: $p(\phi, \psi) = \sum_{\pi\in\Pi} p_\pi(\phi, \psi)$. By Bayes's rule $p(\phi, \psi|y, z) \propto p(\phi, \psi, y, z) = \sum_{\pi\in\Pi} p(y, z|\phi, \psi) p_\pi(\phi, \psi)\omega_\pi$ and the 1-1 map from (ϕ, ψ) to $(\widetilde{\phi}, \widetilde{\psi}, \Phi)$, we have

$$p(y, z|\phi, \psi)p_\pi(\phi, \psi) = p(y, z|\widetilde{\phi}, \widetilde{\psi}, \Phi_\pi)p(\widetilde{\phi})p(\widetilde{\psi})p(\Phi_\pi)$$

when $(\phi, \psi) \in A_\pi$. Let us denote right hand side of the above equation as U_π, then

$$U_\pi = \omega_\pi A_1 A_2 A_3 \prod_{k=1}^{K} (\widetilde{\phi}_k)^{t_k^1+\alpha_k^1} (\widetilde{\psi}_k)^{t_k^2+\alpha_k^2} \prod_{b\in\pi} (\Phi_b)^{t_b^1+t_b^2+\beta_b},$$

where A_1 is the product of normalizing terms from multinomial distribution of z^1 and z^2, $A_1 = \frac{\Gamma(n_1+1)\Gamma(n_2+1)}{\prod_{j=1}^{2}\prod_{k=1}^{K}\Gamma(t_k^j+1)}$, and A_2 is the product of normalizing terms from Dirichlet distribution of $\widetilde{\phi}$ and $\widetilde{\psi}$, $A_2 = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k^1+1)\Gamma(\sum_{k=1}^{K}\alpha_k^2+1)}{\prod_{j=1}^{2}\prod_{k=1}^{2}\Gamma(\alpha_k^j+1)}$, and A_3 is the normalizing term from Dirichlet distribution of Φ_π, $A_3 = \frac{\Gamma(\sum_{b\in\pi}\beta_b+1)}{\prod_{b\in\pi}\Gamma(\beta_b+1)}$. Looking at the indices of $\widetilde{\phi}, \widetilde{\psi}$ and Φ, we can decompose U_π as a product of three Dirichlet densities with a normalizing term. Namely $U_\pi = C_\pi * f_1 f_2 f_3$, where $f_1 \sim$ Dirichlet$[\alpha^1 + t^1]$, $f_2 \sim$ Dirichlet$[\alpha^2 + t^2]$ and $f_3 \sim$ Dirichlet$[\beta + t^1 + t^2]$. Considering the normalizing factors for densities f_1, f_2 and f_3, and multiplying them with A_1, A_2 and A_3, we have $C_\pi = p_\pi(t^1|t_\pi^1, y)\, p_\pi(t^2|t_\pi^2, y)\, p_\pi(t_\pi^1, t_\pi^2|y)\omega_\pi$. Consequently, we have

$$(\phi, \psi)|y, z \sim \text{DDM}\left[\omega^{\text{post}} = (\omega_\pi^{\text{post}}), \alpha^1 + t^1, \alpha^2 + t^2\right] \text{ and } \omega_\pi^{\text{post}} \propto p_\pi(t^1|t_\pi^1, y)\, p_\pi(t^2|t_\pi^2, y)\, p_\pi(t_\pi^1, t_\pi^2|y)\, \omega_\pi.$$

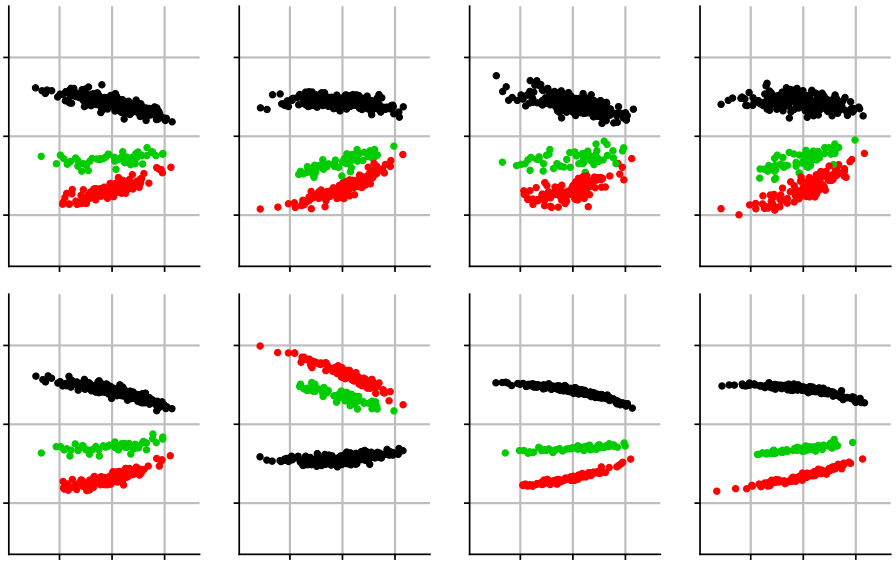
Notice in DDM, we restricted $\beta = \alpha^1 + \alpha^2$.

A.10 Numerical Experiments

Synthetic Data

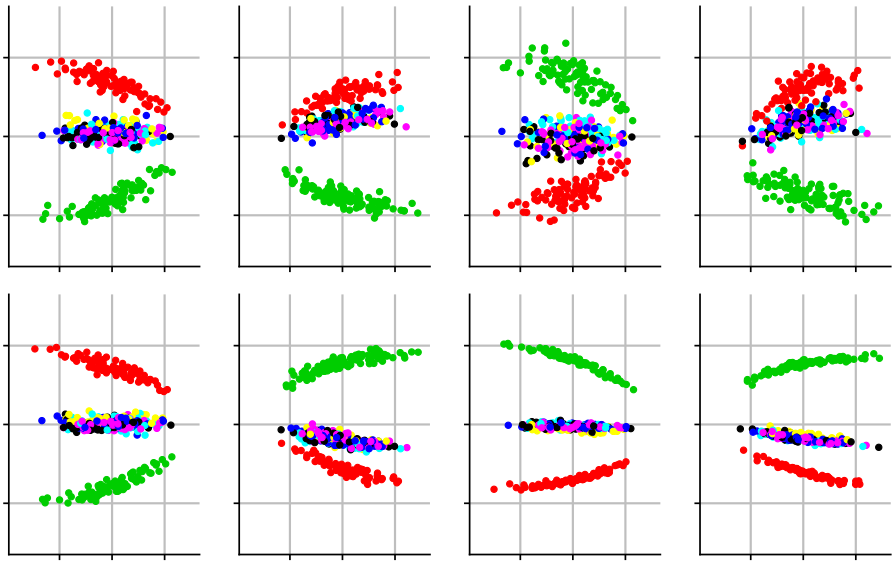
In this section we look more closely at the synthetic data generated using `splatter`. We use PCA plots to show the subtle changes underlying each subtype of simulated data and we demonstrate consistency of estimated distributional changes based on `scDDboost` and Wasserstein distance. Finally, ROC curves illustrate that `scDDboost` has favorable operating characteristics.

We first look at the PCA plots of the simulated data (Appendix Figure A9, A10, A11). For $K = 7$ and 12, in each scenario there were some subtypes nested in the 2d PCA projection and the distributional change of transcripts becomes difficult to detect. `scDDboost` benefits from the compositional structure and is more sensitive to those subtle changes.

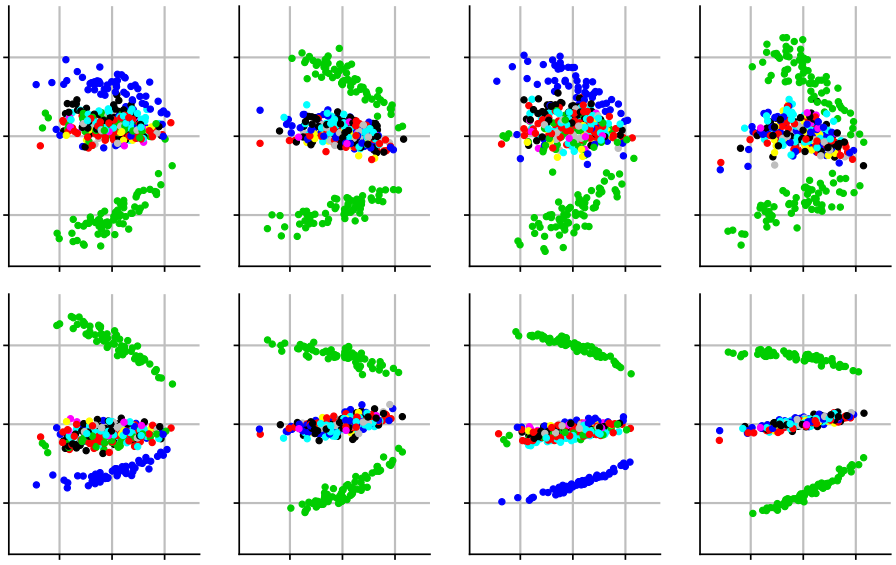


Appendix Figure A9: First two principal components of transcripts under different parameters for simulated data. Horizontal axis refers to first component, vertical axis refers to second component. Different parameters resulted in different degree of separation of subtypes. We have 4 different settings for hyper-parameters of simulation, each setting has ten replicates. From left to right, the associated hyper-parameters are (0.1,0.4), (-0.1,0.3), (0.3,0.5), (-0.1,1). Here we have 3 subtypes

We observed consistent measurements of distributional change by `scDDboost` and Wasserstein distance (Appendix Figure A12). Lower probabilities of equivalent distributed are associated

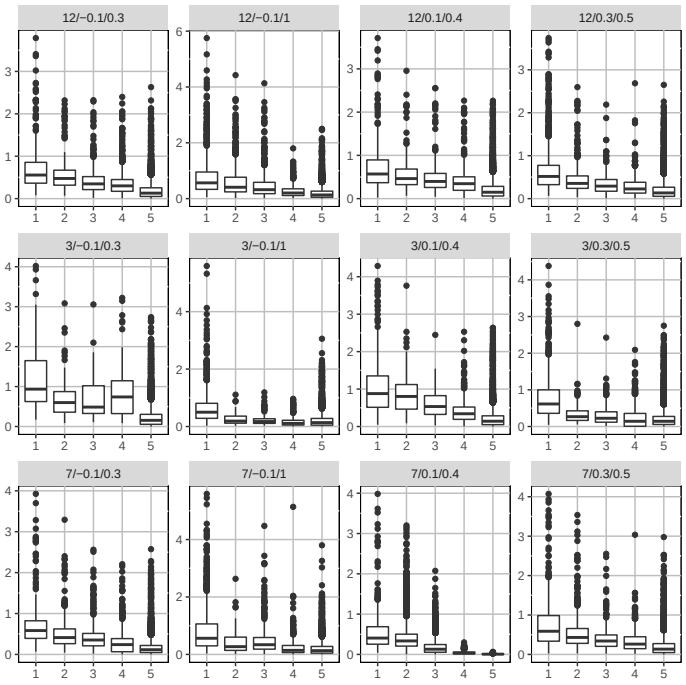


Appendix Figure A10: Similar plots as Appendix Figure A9, but for 7 subtypes



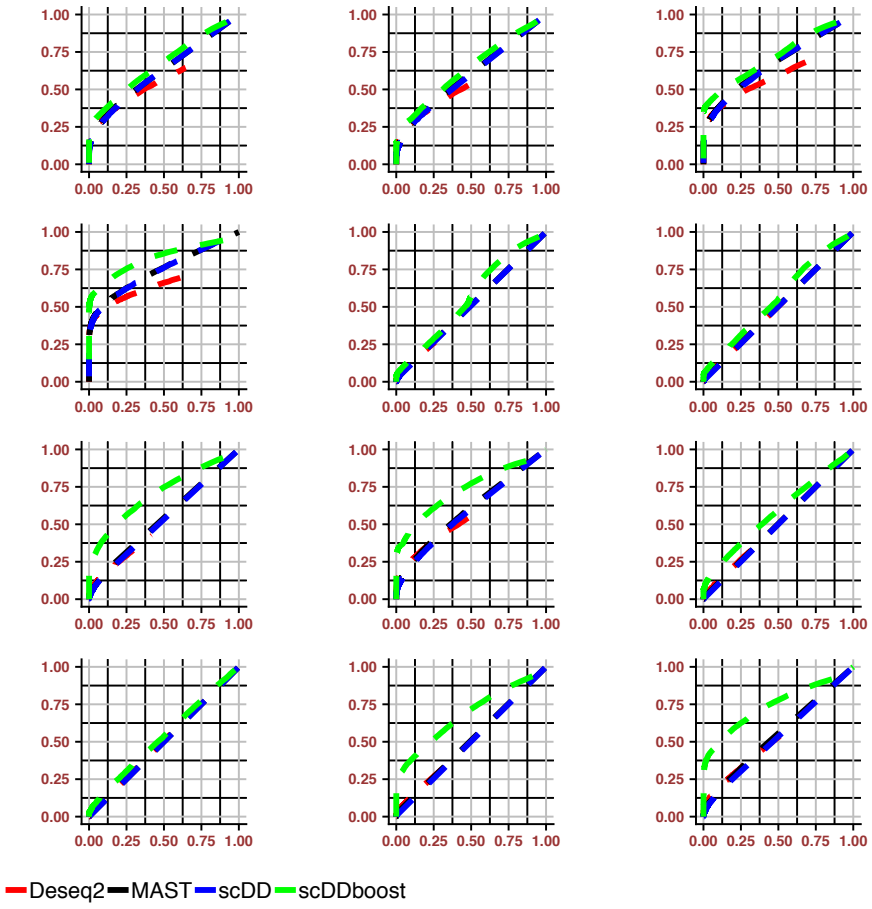
Appendix Figure A11: Similar plots as Sumpplementary Figure A9, but for 12 subtypes

with bigger distances.



Appendix Figure A12: $P(ED_g|X, y)$ given by scDDboost (horizontal) versus empirical Wasserstein distance (vertical). Genes associated with boxes from left to right having $P(ED_g|X, y)$ range from 0 - 0.2, 0.2 - 0.4, 0.4 - 0.6, 0.6 - 0.8, 0.8 - 1. For simulation cases with parameters in the format: number of clusters / shape / scale

ROC curves for the simulated data in Appendix Figure A13. Each sub-figure is averaged over ten replicates under the same parameters setting. `scDDboost` tends to outperform other methods .



Appendix Figure A13: Roc curve of the 12 simulation settings, under each setting, TPR and FPR are averaged over ten replicates, generally `scDDboost` performs better than other methods

Empirical Study

In this section, we provide details of the empirical datasets and also demonstrate consistency to Wasserstein distance on one dataset FUCCI (Leng et al., 2015).

Data sets Details for the datasets used in the empirical studies with the estimated number of subtypes K are shown in Appdenix Table A7.

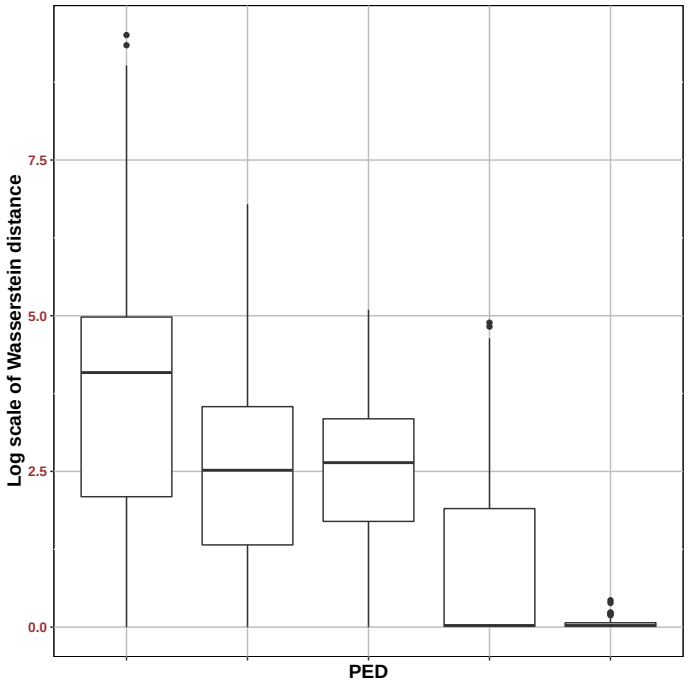
Data set	Conditions	Number of cells/condition	Organism	Ref	K
GSE94383	0 min unstim vs 75min stim	186,145	human	(Lane et al., 2017)	9
GSE48968-GPL13112	BMDC (2h LPS stimulation) vs 6h LPS	96,96	mouse	(Shalek et al., 2014)	4
GSE52529	T0 vs T72	69,74	human	(Trapnell et al., 2014c)	7
GSE74596	NKT1 vs NTK2	46,68	mouse	(Engel et al., 2016b)	7
EMTAB2805	G1 vs G2M	95,96	mouse	(Buettner et al., 2015)	6
GSE71585-GPL13112	Gad2tdTpositive vs Cux2tdTnegative	80,140	mouse	(Tasic et al., 2016)	4
GSE64016	G1 vs G2	91,76	human	(Leng et al., 2015)	6
GSE79102	patient1 vs patient2	51, 89	human	Kiselev et al. (2017)	4
GSE45719	16-cell stage blastomere vs mid blastocyst cell	50, 60	mouse	(Deng et al., 2014)	4
GSE63818	Primordial Germ Cells, developmental stage: 7 week gestation vs Somatic Cells, developmental stage: 7 week gestation	40,26	mouse	(Guo et al., 2015)	6
GSE75748	DEC vs EC	64, 64	human	(Chu et al., 2016)	5
GSE84465	neoplastic cells vs non-neoplastic cells	1000, 1000	human	(Darmanis et al., 2017)	9

Appendix Table A7: Datasets used for empirical study

For the first 11 datasets in Appendix Table A7 we use all the cells within that condition under same batch. The last one is the largest dataset we explored containing 3589 cells and comparing neoplastic cells (1091 cells) vs non-neoplastic cells (2498 cells). We randomly sampled 1000 cells from each condition, because it takes a lot of time for `DESeq` and `scDD` to compute when using all the samples and we conjecture that 1000 cells each condition would be enough to represent the

heterogeneity. We run the comparison on those subsamples instead and found `DESeq` identified significantly smaller numbers of positives than other methods. It is intuitive that we are more likely to encounter subtle changes when we have large samples, and only considering mean shifts would have limited power.

We also observed consistent distributional change measurements by `scDDboost` and Wasserstein distance (Appendix Figure A14).



Appendix Figure A14: $P(ED_g|X, y)$ given by scDDboost versus empirical Wasserstein distance. Genes associated with boxes from left to right having $P(ED_g|X, y)$ range from 0 - 0.2, 0.2 - 0.4, 0.4 - 0.6, 0.6 - 0.8, 0.8 - 1, data used: FUCCI

.

Datasets used for generating the Null cases are shown in Appendix Table A8.

Data set	Conditions	Number of cells/condition	Organism
GSE63818null	7 week gestation	20,20	mouse
GSE75748null	DEC	32, 32	human
GSE94383null	T0	93, 93	human
GSE48968-GPL13112null	BMDC (2h LPS stimulation)	48,48	mouse
GSE74596null	NKT1	23,23	mouse
EMTAB2805null	G1	48,48	mouse
GSE71585-GPL13112null	Gad2tdTpositive	40,40	mouse
GSE64016null	G1	46,45	human
GSE79102null	patient1	26, 25	human

Appendix Table A8: Datasets used for null cases, as cells are coming from same biological condition, there should not be any differential distributed genes, any positive call is false positive

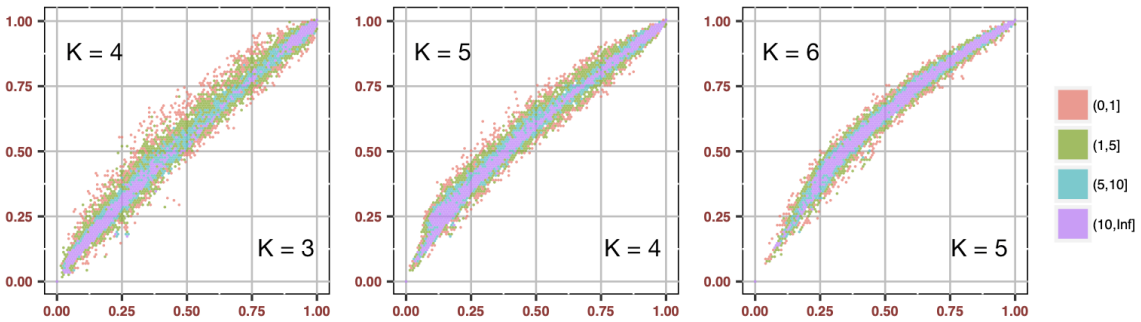
A.11 Robustness

In this section, we demonstrate change of the posterior probability of DD under different K, and also the robustness provided by random weighting. We also give an example where very large K inflates FDR.

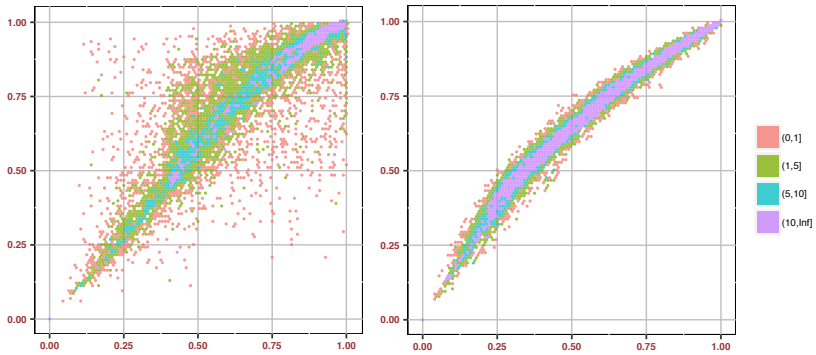
The number of subtypes K is an important parameter. Taking K too small may end up underfitting such that cells within same subtype can still be very different, the mean expression change among subtypes is incapable to capture the marginal distribution change. This would lead to reduced power. Too large K may end up overfitting such that two subtypes can be very similar. Given that we have a fixed number of cells, allowing more clusters will not only increase the burden of computation but decrease the certainty of our inference on DE pattern. Empirically we find that taking $K \leq 10$ is often sufficient (Appendix Table A7). In any case, we note here that K affects the posterior probability of DD (PDD).

To demonstrate the change of PDD over different K, we present an example using dataset GSE75748. When we increase K, the variance of the differential term $\text{PDD}_{K+1} - \text{PDD}_K$ keeps decreasing and PDD keeps increasing. Our selection criterion (K = 5) happens to choose K such that change between PDD_{K+1} and PDD_K is small while not inflating PDD. We generally obtain stable validity score and PDD simultaneously (Appendix Figure A15). In addition, the random weighting scheme helps by smoothing PDD (Appendix Figure A16).

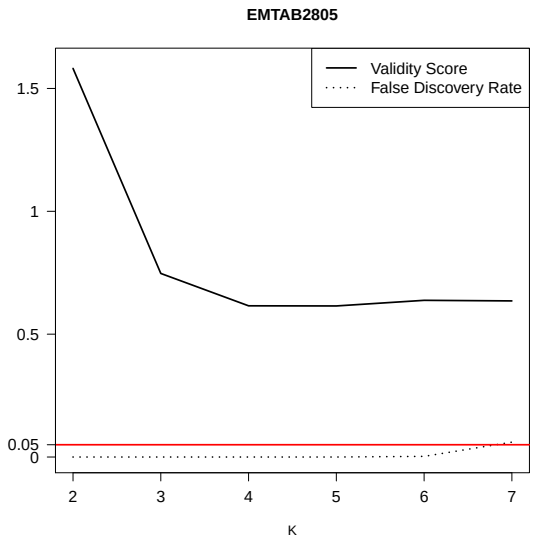
`scDDboost` as the potential to lose FDR control if K is not maintained at a sufficiently small value. Appendix Figure A17 shows what happens as K increases in one case, other factors staying constant. In our simulation study, we note that the validity score method was always conservative, and did not lead to overestimating K.



Appendix Figure A15: PDD change under different number of subtypes K for dataset DEC-EC (GSE75748). We select $K = 4$, which also stabilize the PDD.



Appendix Figure A16: PDD under $K = 5$ vs. $K = 6$ for dataset DEC-EC (GSE75748). PDD without randomization (left) vs. PDD with randomization (right). `scDDboost` gained robustness through random weighting.



Appendix Figure A17: Under NULL case, using dataset EMTAB2805, when using too big K we may lose FDR control (black dashed line shows proportion of false positive identified by scDDboost under 0.05 threshold, while validity score stabilized after $K > 2$

A.12 Posterior consistency

In this section, we prove Theorem 4 and we discuss a case when condition (12) fails. The density of DDM is computed by product or ratio over several gamma functions. We use a crucial lemma which gives us an approximation to the gamma function, namely

Lemma .0.2. *For $x \geq 1$, $\frac{x^{x-c}}{e^{x-1}} \leq \Gamma(x) \leq \frac{x^{x-1/2}}{e^{x-1}}$, where $c = 0.577215...$ is the Euler-Mascheroni constant.*

Proof. [Proof of Lemma 2] By (Li and ping Chen, 2007), we have $\frac{x^{x-c}}{e^{x-1}} \leq \Gamma(x) \leq \frac{x^{x-1/2}}{e^{x-1}}$ for $x > 1$ and now we added the case when $x = 1, \Gamma(x) = 1$ so that both sides will include the equality case.

We have another lemma.

Lemma .0.3. *If $(\phi, \psi) \in A_{\pi_1} \cap A_{\pi_2}$, follow the conditions in Theorem 1 then*

$$\frac{\omega_{\pi_1}^{post}}{\omega_{\pi_2}^{post}} \xrightarrow[n\to\infty]{a.s.} 0 \quad \text{if } N(\pi_1) < N(\pi_2)$$

Proof. [Proof of Lemma 3] Recall $\omega_\pi^{\text{post}} \propto p_\pi(t^1|t_\pi^1, y)\, p_\pi(t^2|t_\pi^2, y)\, p_\pi(t_\pi^1, t_\pi^2|y)\, \omega_\pi$ and RHS $= g(\pi, \alpha, \beta, n_1, n_2) f(\pi, t^1, t^2, \alpha, \beta)$ and $\frac{\omega_{\pi_1}^{post}}{\omega_{\pi_2}^{post}} = \frac{g(\pi_1,\alpha,\beta,n_1,n_2)}{g(\pi_2,\alpha,\beta,n_1,n_2)} \frac{f(\pi_1,t^1,t^2,\alpha,\beta)}{f(\pi_2,t^1,t^2,\alpha,\beta)}$ where

$$g(\pi, t^1, t^2, \alpha, \beta) = [\prod_{j=1}^{2}\prod_{b\in\pi} \frac{\Gamma(\Sigma_{k\in b}\alpha_k^j)}{\prod_{k\in b}\Gamma(\alpha_k^j)}] \frac{\Gamma(n_1+1)\Gamma(n_2+1)}{\prod_{b\in\pi}\Gamma(\beta_b)} \frac{\Gamma(\Sigma_{b\in\pi}\beta_b)}{\Gamma(n_1+n_2+\Sigma_{b\in\pi}\beta_b)}$$

$$f(\pi, t^1, t^2, \alpha, \beta) = [\prod_{j=1}^{2}\prod_{b\in\pi} \frac{1}{\prod_{k\in b}\Gamma(t_k^j+1)} \frac{\prod_{k\in b}\Gamma(\alpha_k^j+t_k^j)}{\Gamma(t_b^j+\Sigma_{k\in b}\alpha_k^j)}] \prod_{b\in\pi}\Gamma(\beta_b+t_b^1+t_b^2)$$

For notational simplicity, we use the abbreviation $g(\pi), f(\pi)$ to substitute $g(\pi, \alpha, \beta, n_1, n_2), f(\pi, t^1, t^2, \alpha, \beta)$. We take log on $\frac{\omega_{\pi_1}^{post}}{\omega_{\pi_2}^{post}}$, denote it as LR. $LR = \ln g(\pi_1) - \ln g(\pi_2) + \ln f(\pi_1) - \ln f(\pi_2)$. Denote $C(\pi_1, \pi_2, \alpha, \beta) = \ln g(\pi_1) - \ln g(\pi_2)$, $C(\pi_1, \pi_2, \alpha, \beta)$ does not change with sample size n_1, n_2 and is a constant determined by partition π_1, π_2 and hyper parameters α, β. For further convenience of notation let $h(x) = ln\Gamma(x)$ and $\gamma_b^j = \Sigma_{k\in b}\alpha_k^j$. Denote

$R(\pi_1, \pi_2, t^1, t^2, \alpha, \beta) = \ln f(\pi_1) - \ln f(\pi_2)$. And removing the common part of $f(\pi_1)$ and $f(\pi_2)$, we have

$$R(\pi_1, \pi_2, t^1, t^2, \alpha, \beta) = d(\pi_1, t^1, t^2, \alpha, \beta) - d(\pi_2, t^1, t^2, \alpha, \beta)$$

where

$$d(\pi, t^1, t^2, \alpha, \beta) = \sum_{b\in\pi} h(\beta_b + t_b^1 + t_b^2) - \sum_{j=1}^{2}\sum_{b\in\pi} h(t_b + \gamma_b^j)$$

Recall $\beta_b = \gamma_b^1 + \gamma_b^2$ and from Lemma 2, $(x-c)\ln(x) - x + 1 \le h(x) \le (x - 1/2)\ln(x) - x + 1$ we have

$$d(\pi, t^1, t^2, \alpha, \beta) \ge \sum_{b\in\pi} (\beta_b + t_b^1 + t_b^2 - c)\ln(\beta_b + t_b^1 + t_b^2) - \sum_{j=1}^{2}\sum_{b\in\pi} (t_b^j + \gamma_b^j - 1/2)\ln(t_b^j + \gamma_b^j) + N(\pi) \tag{7}$$

$$d(\pi, t^1, t^2, \alpha, \beta) \le \sum_{b\in\pi} (\beta_b + t_b^1 + t_b^2 - 1/2)\ln(\beta_b + t_b^1 + t_b^2) - \sum_{j=1}^{2}\sum_{b\in\pi} (t_b^j + \gamma_b^j - c)\ln(t_b^j + \gamma_b^j) + N(\pi) \tag{8}$$

$$\begin{aligned}\text{RHS of (4)} = \Sigma_b \big[&(t_b^1 + \gamma_b^1)\ln(1 + \frac{t_b^2 + \gamma_b^2}{t_b^1 + \gamma_b^1}) + (t_b^2 + \gamma_b^2)\ln(1 + \frac{t_b^1 + \gamma_b^1}{t_b^2 + \gamma_b^2}) \\ &+(1-c)\ln(\beta_b + t_b^1 + t_b^2) - 1/2(\ln(1 + \frac{t_b^2 + \gamma_b^2}{t_b^1 + \gamma_b^1}) + \ln(1 + \frac{t_b^1 + \gamma_b^1}{t_b^2 + \gamma_b^2}))\big] + N(\pi)\end{aligned}$$

By Taylor expansion at $x = 1$, $\ln(x+1) = \ln 2 + 1/2(x-1) - 1/8(x-1)^2 + g(\xi)(x-1)^3$, where $g(\xi)$ is the reminder term of form $\frac{1}{3(1+\xi)^3}$ for $0 < \xi < x$ For a fixed n_1, n_2, we have

$$\begin{aligned}\text{RHS of (4)} = (n_1 + n_2)\ln 2 - \Sigma_{b\in\pi}&(1/8(X_b^1 + X_b^2) \\ &+g(\xi_b)(Y_b^1 + Y_b^2)) + T(\pi) + N(\pi)\end{aligned}$$

where $X_b^1 = \frac{(t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)^2}{t_b^1 + \gamma_b^1}$, $X_b^2 = \frac{(t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)^2}{t_b^2 + \gamma_b^2}$, $Y_b^1 = \frac{(t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)^3}{(t_b^1 + \gamma_b^1)^2}$, $Y_b^2 = \frac{(t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)^3}{(t_b^2 + \gamma_b^2)^2}$
and $T(\pi) = \Sigma_{b\in\pi}\big[(1-c)\ln(\beta_b + t_b^1 + t_b^2) - 1/2(\ln(1 + \frac{t_b^2+\gamma_b^2}{t_b^1+\gamma_b^1}) + \ln(1 + \frac{t_b^1+\gamma_b^1}{t_b^2+\gamma_b^2}))\big]$
Similarly

$$\text{RHS of (5)} = (n_1+n_2)\ln 2 - \Sigma_{b\in\pi}(1/8(X_b^1 + X_b^2) + g(\xi_b)(Y_b^1 + Y_b^2)) + U(\pi) + N(\pi)$$

$$U(\pi) = \Sigma_{b\in\pi}\big[(2c-1/2)\ln(\beta_b + t_b^1 + t_b^2) - c(\ln(1 + \tfrac{t_b^2+\gamma_b^2}{t_b^1+\gamma_b^1}) + \ln(1 + \tfrac{t_b^1+\gamma_b^1}{t_b^2+\gamma_b^2}))\big]$$

Using above inequalities, we have

$$R(\pi_1,\pi_2,t^1,t^2,\alpha,\beta) \le U(\pi_1) - T(\pi_2) - 1/8(\Sigma_{b\in\pi_1}(X_b^1 + X_b^2) - \Sigma_{b\in\pi_2}(X_b^1 + X_b^2)) + \Sigma_{b\in\pi_1} g(\xi_b)(Y_b^1 + Y_b^2) - \Sigma_{b\in\pi_2} g(\xi_b)(Y_b^1 + Y_b^2)$$

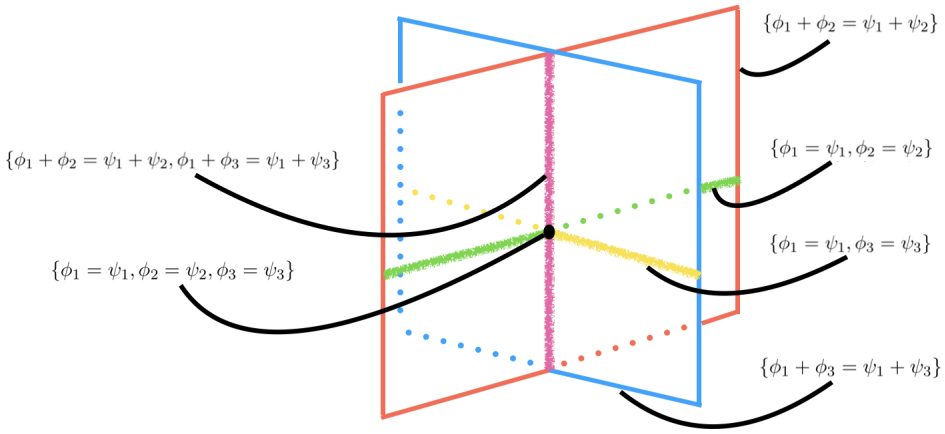
$Y_b^j = \frac{((t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)/\sqrt{n}))^3/\sqrt{n}}{((t_b^j + \gamma_b^j)/n)^2}$, by LLN the denominator goes to a constant and by CLT in the numerator $(t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)/\sqrt{n} \to (t_b^1 - t_b^2)/\sqrt{n} \to \sqrt{n}[(t_b^1/n - \Phi_b) - (t_b^2/n - \Psi_b)]$, which converges to a normally distributed random variable when $\Phi_b = \Psi_b$. So Y_b^j is $o_p(1)$. Similarly, $X_b^j = \frac{((t_b^1 - t_b^2 + \gamma_b^1 - \gamma_b^2)/\sqrt{n})^2}{t_b^j + \gamma_b^j/n}$ is asymptotically gamma (χ-square) distributed. $g(\xi_b)$ has bounded variance, $U(\pi_1) - T(\pi_2) = -\ln(n)$ if $N(\pi_2) < N(\pi_1)$ as $\ln(\beta_b + t_b^1 + t_b^2) - \ln(\beta_{b'} + t_{b'}^1 + t_{b'}^2) = \ln(\frac{\beta_b + t_b^1 + t_b^2}{n}) - \ln(\frac{\beta_{b'} + t_{b'}^1 + t_{b'}^2}{n}) \to O(1) \quad a.s.$, which completes the proof.

Proof. [Proof of Theorem 4]

Recall $\sum_{\pi\in\Pi} \omega_\pi^{\text{post}} = 1$ and $P(A_\pi|y,z) = \sum_{\widetilde{\pi}\in\Pi} \omega_{\widetilde{\pi}}^{\text{post}}\, 1[\widetilde{\pi} \text{ refines } \pi]$. If $(\phi,\psi) \notin$ Q, for all the A_π covers (ϕ,ψ) there is one finest π^* with the largest $N(\pi^*)$ and every other π that $(\phi,\psi) \in A_\pi$ is coarser than π^*. Theorem 4 now follows by Lemma 3.

Under some choices of (ϕ,ψ), condition (12) could fail.

In Appendix Figure A18, there are four subtypes, the rectangle with magenta boundary is a simplex



Appendix Figure A18: Four subtypes of cells, simplexes of (ϕ, ψ) satisfying different constraints.

$A_{\pi_1} = \{(\phi,\psi) : \phi_1+\phi_2 = \psi_1+\psi_2\}$, the rectangle with blue boundary is another simplex $A_{\pi_2} = \{(\phi,\psi) : \phi_1+\phi_3 = \psi_1+\psi_3\}$. The green line refers to $A_{\pi_3} = \{(\phi,\psi) : \phi_1 = \psi_1, \phi_2 = \psi_2\}$, the yellow line refers to $A_{\pi_4} = \{(\phi,\psi) : \phi_1 = \psi_1, \phi_3 = \psi_3\}$. the purple line refers to $O = \{(\phi,\psi) : \phi_1+\phi_2 = \psi_1+\psi_2, \phi_1+\phi_3 = \psi_1+\psi_3\}$, which is the intersection of A_{π_1} and A_{π_2}, and finally the black dot which is the intersection of those three lines refers to the simplex with finest partitions, $\phi_i = \psi_i, \forall i = 1, \cdots, 4$. When (ϕ, ψ) is from the purple line except the black dot, condition (12) would fail as there is not a finest π^* of $H(\phi, \psi)$. This may be of theoretical interest, but the practical implications of this finding are negligible as further computations have demonstrated.

A.13 EBSeq margin

A core computation is to evaluate the joint probability mass f of input data recordings $X = (X_1, \cdots, X_m)$ that have a common, unknown, mean, whose value is integrated out (i.e., continuous mixing). In EBSeq, we integrate a Negative Binomial mass function, $\text{NB}(q, \gamma)$, which has

mean $q\gamma/(1-q)$, against a Beta prior for q.

$$f(x) \quad = \quad \int_0^1 \prod_{i=1}^{m} [p(x_i|\gamma, q)]\ p(q|\alpha, \beta)\, dq$$

where $x|\gamma, q$ is a Negative Binomial(γ, q) distribution and $q|\alpha, \beta$ is a Beta(α, β) distribution. The result is reported in (3.1). In this representation, the expression data X are expected to be normalized and the shape parameter γ is common across samples. Thus the conditional mean μ is fully determined by q.

In EBSeq, hyperparameters are estimated for each unit, α is shared across all units. In general, different units may have different β values, but depending on the data structure, there will be blocking of units. For example, when the units are isoforms. Isoforms from the same host gene are treated as a block and share a common β. The global hyperparameters parameters (α, β), and the mixing rates $p_{\text{global},\pi}$ are estimated in an EM algorithm using data on all samples and all units (Dempster et al., 1977).

A.14 Proofs of Lemma 3.3.1 and Theorem 3.3.1

Proof of Lemma 3.3.1

Proof. Without loss of generality, we label groups so sample means are ordered, $\widehat{\mu}_1 \leq ... \leq \widehat{\mu}_K$, and the rank vector $r = (1, 2, ..., K)$, and therefore $A_r(b) = b$. Denote the number of blocks in π as $N(\pi)$. For any compatible π, if $N(\pi) > 1$, then by definition, any two distinct $b_j, b_k \in \pi$. We have b_j and b_k are not overlapping, which means either $\max b_j < \min b_k$ or $\max b_k < \min b_j$.

Without loss of generality, let $b_1, ..., b_{N(\pi)}$ be the blocks with $\max(b_1) < ... < \max(b_{N(\pi)})$. We say $v_i = \min(b_i)$ and $w_i = \max(b_i)$ for $i = 1, 2, ..., N(\pi)$. Then there is a way of assigning $\neq$ or = between $\widehat{\mu}_j$ and $\widehat{\mu}_{j+1}$ for $j = 1, ..., K-1$ to represent π (equation 3.2), namely if $N(\pi) > 1$, we assigning $\neq$ between $\widehat{\mu}_{w_i}$ and $\widehat{\mu}_{v_i}$ for $i = 1, ..., N(\pi) - 1$, and assign = for the rest slots. If $N(\pi) == 1$, we assign = to all slots.

Also for any assignment of $\neq$ and = between $\widehat{\mu}_j$ and $\widehat{\mu}_{j+1}$. There is a compatible partition corresponding to it. It is a 1-1 correspondence between compatible partitions and assignment between ranked estimated means. There is $K-1$ slots Thus there is 2^{K-1} possible ways.

To prove Theorem 3.3.1, let us further decompose predictive functions in Equation (3.1). Without loss of generality, we label groups '1' and '2' with and for $j = 1, 2$ denote by x_j the vector of n_j sample measurements in group j at the unit on test. Write:

$$\begin{aligned}
\log\{f(x_{1,2})\} &= \phi(\alpha + (n_1+n_2)\gamma) + \phi(\beta + n_1\widehat{\mu}_2 + n_2\widehat{\mu}_2) \\
&\quad - \phi(\alpha + (n_1+n_2)\gamma + \beta + n_1\widehat{\mu}_2 + n_2\widehat{\mu}_2) + \phi(\alpha) + \phi(\beta) - \phi(\alpha+\beta) \\
\log\{f(x_1)f(x_2)\} &= \phi(\alpha + n_1\gamma) + \phi(\beta + n_1\widehat{\mu}_1) - \phi(\alpha + n_1\gamma + \beta + n_1\widehat{\mu}_1) \\
&\quad + \phi(\alpha + n_2\gamma) + \phi(\beta + n_2\widehat{\mu}_2) - \phi(\alpha + \beta + n_2\gamma + n_2\widehat{\mu}_2) + 2(\phi(\alpha) + \phi(\beta) - \phi(\alpha+\beta)) \\
\log(D_{1,2}) &= \log\{f(x_1)f(x_2)\} - \log\{f(x_{1,2})\} \\
&:= g(n_1, n_2, \widehat{\mu}_1, \widehat{\mu}_2, \alpha, \beta, \gamma)
\end{aligned}$$

where $\phi(x) = \log\Gamma(x)$, and $\widehat{\mu}_j = \sum_i x_i 1[y_i = j]/n_j$.

Lemma .0.6, which is a key part of the proof, describes monotonicities of g with respect to different input variables. To prove Lemma .0.6, we have Lemma .0.4, which gives an approximation to the derivative of ϕ that is the key component in g, and Lemma .0.5, which provides useful inequalities to handle the logarithm.

Lemma .0.4. *Digamma function, defined as the (using natural base) derivative of the gamma function:*

$$\psi(x) = \frac{d}{dx}\phi(x) = \frac{d}{dx}\log(\Gamma(x)) \tag{9}$$

Through Binet's second integral for the gamma function, $\psi(x)$ can be rewritten as

$$\psi(x) = \log(x) - \frac{1}{2x} - 2\int_0^\infty \frac{tdt}{(t^2+x^2)(e^{2\pi t}-1)} \tag{10}$$

Further to handle the term, we have Lemma .0.5

Lemma .0.5. *For $x > 0$,*

$$\log(x) \geq x - \frac{x^2}{2} \tag{11}$$

$$\log(x) \leq x \tag{12}$$

Lemma .0.6. *Regularity conditions:*

A) $\gamma \geq C_0, C_0 = \int_0^\infty \frac{4t}{e^{2\pi t}-1} dt \approx 0.0075$

B) $\widehat{\mu}_2 - \widehat{\mu}_1 \geq C_1, C_1 = \frac{\gamma+\alpha+\beta}{n_1} + \frac{\gamma+\alpha}{n_2}$

C) $\gamma\widehat{\mu}_2 - \left(\gamma + \frac{\alpha}{n_2}\right)\widehat{\mu}_1 \geq \left(\frac{1}{n_1} + \frac{1}{n_2}\right)(\alpha + \gamma + C_2)$, $C_2 = \frac{2}{n_2}\left(1 + \frac{\alpha/n_2+\gamma}{\beta/n_2+\widehat{\mu}_2}\right)$

D) $\frac{1}{n_2}\left\{\frac{1}{n_2}\left(4\frac{\widehat{\mu}_2}{\widehat{\mu}_1} + 4\frac{\gamma}{\widehat{\mu}_1}\right) - 2\frac{\gamma\beta - \widehat{\mu}_1\alpha}{\gamma}(\widehat{\mu}_2 - \widehat{\mu}_1)\right\} < (\widehat{\mu}_2 - \widehat{\mu}_1)^2$

Under these regularity conditions, the two-group Bayes factor, g is:

a) Monotone decreasing for $\widehat{\mu}_1$, fix the others

b) Monotone increasing for $\widehat{\mu}_2$, fix the others

c) Monotone increasing for n_1, fix the others

d) Monotone increasing for n_2, fix the others

Proof.

For simplicity, let $S_j = n_j\widehat{\mu}_j, R_j = n_j\gamma, j = 1, 2$. To prove a), the derivative for $\widehat{\mu}_1$ is

$$\begin{aligned}\frac{\partial g}{\partial \widehat{\mu}_1} = n_1 \{&\psi(\beta + S_1) - \psi(\alpha + R_1 + \beta + S_1) \\ &-\psi(\beta + S_1 + S_2) + \psi(\alpha + \beta + S_1 + S_2 + R_1 + R_2)\}.\end{aligned}$$

Using Lemma .0.4 and after some simplification and reorganizing, we have

$$
\begin{aligned}
\frac{\partial g}{\partial \widehat{\mu}_1} &= H_1 + H_2 + H_3 \\
H_1 &= \log\left\{(1 + \frac{n_1 n_2 r(\widehat{\mu}_1 - \widehat{\mu}_2) + R_2\beta - S_2\alpha}{(\alpha + R_1 + S_1 + \beta)(\beta + S_1 + S_2)}\right\} \leq 0 \\
H_2 &\leq -\frac{1}{2}\left\{\frac{(\alpha S_2 + R_1 S_2 - \beta^2)R_2 + (2R_1 + 2\alpha - \beta)S_1 S_2}{(\beta + S_1)(\beta + S_1 + S_2)(\alpha + \beta + R_1 + S_1)(\alpha + \beta + R_1 + R_2 + S_1 + S_2)}\right\} \leq 0 \\
H_3 &= -\int_0^\infty \frac{2t}{e^{2\pi t} - 1}\left\{\frac{\gamma}{t^2 + (\alpha + R_1 + R_2 + \beta + S_1 + S_2)^2} + \frac{\widehat{\mu}_1}{t^2 + (\beta + S_1)^2}\right. \\
&\quad \left. - \frac{\widehat{\mu}_1}{t^2 + (\beta + S_1 + S_2)^2} - \frac{\widehat{\mu}_1}{t^2 + (\alpha + R_1 + \beta + S_1)^2}\right\} dt.
\end{aligned}
$$

Overall, $\frac{\partial g}{\partial \widehat{\mu}_1}$ is dominated by H_1 and is non-positive. The intuition is that $H_1 \sim O_p(1)$, while H_2 and H_3 are $o_p(1)$, where O_p and o_p are the big O and small O notations.(Bachmann, 1894). Thus, H_1 goes to some constant and H_2, H_3 go to 0 when the sample size is large.

To prove b), the derivative for $\widehat{\mu}_2$ is

$$
\begin{aligned}
\frac{\partial g}{\partial \widehat{\mu}_2} &= n_2(\psi(\beta + S_2) - \psi(\alpha + R_2 + \beta + S_2) \\
&\quad - \psi(\beta + S_1 + S_2) + \psi(\alpha + \beta + S_1 + S_2 + R_1 + R_2))
\end{aligned}
$$

which is symmetric to $\frac{\partial g}{\partial \widehat{\mu}_1}$, and we have

$$
\begin{aligned}
\frac{\partial g}{\partial \widehat{\mu}_2} &= H_4 + H_5 + H_6 \\
H_4 &= \log\left\{1 + \frac{n_1 n_2 \gamma(\widehat{\mu}_2 - \widehat{\mu}_1) + R_1\beta - S_1\alpha}{(\alpha + R_1 + S_1 + \beta)(\beta + S_1 + S_2)}\right\} \geq 0 \\
H_5 &\leq -\frac{1}{2}\left\{\frac{(\alpha S_1 + R_2 S_1 - \beta^2)R_1 + (2R_2 + 2\alpha - \beta)S_1 S_2}{(\beta + S_2)(\beta + S_1 + S_2)(\alpha + \beta + R_2 + S_2)(\alpha + \beta + R_1 + R_2 + S_1 + S_2)}\right\} \leq 0 \\
H_6 &= -\int_0^\infty \frac{2t}{e^{2\pi t} - 1}\left\{\frac{\gamma}{t^2 + (\alpha + R_1 + R_2 + \beta + S_1 + S_2)^2} + \frac{\widehat{\mu}_2}{t^2 + (\beta + S_2)^2}\right. \\
&\quad \left. - \frac{\widehat{\mu}_2}{t^2 + (\beta + S_1 + S_2)^2} - \frac{\widehat{\mu}_2}{t^2 + (\alpha + R_2 + \beta + S_2)^2}\right\} dt.
\end{aligned}
$$

$\frac{\partial g}{\partial \widehat{\mu}_2}$ is dominated by H_4, when the regularity conditions are met. To prove c), the derivative for n_1 is

$$\frac{\partial g}{\partial n_1} = \gamma\psi(\alpha+R_1) + \widehat{\mu}_1\psi(\beta+S_1) - (\gamma+\widehat{\mu}_1)\psi(\alpha+R_1+\beta+S_1) - \gamma\psi(\alpha+R_1+R_2)$$
$$- \widehat{\mu}_1\psi(\beta+S_1+S_2) + (\gamma+\widehat{\mu}_1)\psi(\alpha+R_1+R_2+\beta+S_1+S_2)$$

Using Lemma .0.4 and after some simplification and reorganizing, we have

$$\begin{aligned}
\frac{\partial g}{\partial n_1} &= H_7 + H_8 + H_9 \\
H_7 &= \gamma \log\left(1+\frac{T_1-T_2}{1+T_2}\right) - \widehat{\mu}_1 \log\left(1+\frac{T_1-T_2}{T_1T_2+T_2}\right) \\
H_8 &= \frac{\gamma}{2}\left(\frac{T_1}{\alpha+\beta+R_1+R_2+S_1+S_2} - \frac{T_2}{\alpha+\beta+R_1+S_1}\right) \\
&\quad + \frac{\widehat{\mu}_1}{2}\left(\frac{1}{T_1(\alpha+\beta+R_1+R_2+S_1+S_2)} - \frac{1}{T_2(\alpha+\beta+R_1+S_1)}\right) \\
H_9 &= -\int_0^\infty \frac{2t}{e^{2\pi t}-1}\left\{\frac{\gamma}{t^2+(\alpha+R_1)^2} + \frac{\widehat{\mu}_1}{t^2+(\beta+S_1)^2}\right. \\
&\quad - \frac{\gamma+\widehat{\mu}_1}{t^2+(\alpha+R_1+\beta+S_1)^2} \\
&\quad + \frac{\gamma}{t^2+(\alpha+R_1+R_2)^2} + \frac{\widehat{\mu}_1}{t^2+(\beta+S_1+S_2)^2} \\
&\quad \left. - \frac{\gamma+\widehat{\mu}_1}{t^2+(\alpha+\beta+R_1+R_2+S_1+S_2)^2}\right\} dt,
\end{aligned}$$

where $T_1 = \frac{\beta+S_1+S_2}{\alpha+R_1+R_2}$, $T_2 = \frac{\beta+S_1}{\alpha+R_1}$. Using Lemma .0.5, we have

$$H_7 \geq r\left\{\frac{T_1-T_2}{1+T_2} - \frac{1}{2}\left(\frac{T_1-T_2}{1+T_2}\right)^2\right\} - \widehat{\mu}_1\frac{T_1-T_2}{T_1T_2+T_2}$$

Notice $T_1 - T_2 = \frac{\widehat{\mu}_2-\widehat{\mu}_1-(\frac{\beta}{n_1}-\frac{\alpha\widehat{\mu}_2}{R_1})}{(r+\frac{\alpha+R_1}{n_2})(\frac{\alpha}{R_1}+1)} \geq 0$. By regularity condition B). Using this, we have

$$\begin{aligned}
H_7 &\geq \frac{T_1-T_2}{(1+T_1)^2}\frac{1}{\alpha+R_1+R_2}\left\{\frac{\gamma}{2}n_2(\widehat{\mu}_2-\widehat{\mu}_1) + (\gamma\beta-\widehat{\mu}_1\alpha) + O\left(\frac{1}{n_1}\right)\right\} \\
&= \frac{n_2(\widehat{\mu}_2-\widehat{\mu}_1)(\frac{\gamma}{2}n_2(\widehat{\mu}_2-\widehat{\mu}_1)+\gamma\beta-\widehat{\mu}_1\alpha)}{(\alpha+\beta+R_1+R_2+S_1+S_2)^2(\frac{\alpha}{R_1}+1)} + o\left(\frac{1}{n_1^2}\right)
\end{aligned}$$

with positive coefficient. The first part of H_8,

$$\frac{\gamma}{2}\left(\frac{T_1}{\alpha+\beta+R_1+R_2+S_1+S_2}-\frac{T_2}{\alpha+\beta+R_1+S_1}\right)$$
$$=\frac{\gamma}{2}\frac{R_1^2S_2-2R_2R_1S_1}{(\alpha+R_1)(\alpha+R_1+R_2)(\alpha+\beta+R_1+R_2+S_1+S_2)(\alpha+\beta+R_1+S_1)}+o\left(\frac{1}{n_1^2}\right)$$

The second part of H_8

$$\frac{\widehat{\mu}_1}{2}\left(\frac{1}{T_1(\alpha+\beta+R_1+R_2+S_1+S_2)}-\frac{1}{T_2(\alpha+\beta+R_1+S_1)}\right)$$
$$=\frac{-S_2R_1^2-2S_2S_1R_1}{(\alpha+\beta+R_1+R_2+S_1+S_2)(\beta+S_1+S_2)(\beta+S_1)(\alpha+\beta+R_1+S_1)}+o\left(\frac{1}{n_1^2}\right).$$

So we have

$$H_8=\frac{-2S_2S_1R_1-2R_2S_1R_1}{(\alpha+\beta+R_1+R_2+S_1+S_2)(\beta+S_1+S_2)(\beta+S_1)(\alpha+\beta+R_1+S_1)}+o\left(\frac{1}{n_1^2}\right)$$

By regularity condition D, we have $H_7+H_8\sim O(\frac{1}{n_1^2})\geq 0$, and $H_9\sim o(\frac{1}{n_1^2})$. So the derivation is positive when n_1 is large. Proof of d) is similar and details are omitted.

The regularity conditions essentially require the difference between the sample means of two groups being bigger than some threshold, that is $\widehat{\mu}_2-\widehat{\mu}_1>O(\frac{1}{min(n_1,n_2)})$. Once there is a sufficient difference, Lemma .0.6 states that two-group Bayes factor is monotone function with respect to those inputs $\widehat{\mu}_1,\widehat{\mu}_2,n_1,n_2$. Roughly speaking, the two-group Bayes factor is monotone increasing with the difference between the sample means and the sample sizes.

Proof of Theorem 3.3.1.

Proof. Without loss of generality, let $x_1,\cdots,x_K$ be vectors of data at one unit with group labels sorted from smallest to largest by the sample means. If for some $j<K$, we have the two-group Bayes factor favoring differential means between groups j and $j+1$,then $D_{j,j+1}>1$. Denote Π_e as the set of partitions assigning equivalent means between group j and $j+1$ and let $\Pi_d=\Pi\setminus\Pi_e$ be set of partitions assigning differential means. As for those partitions assigning equivalent means, we can merge x_j,x_{j+1} into one group and view Π_e as partitions for $K-1$ elements, $|\Pi_e|=B_{K-1}$,

while $|\Pi_d| = B_K - B_{K-1}$, and so $|\Pi_e| < |\Pi_d|$. There is an injection mapping from Π_e to Π_d. Namely, for any $\pi \in \Pi_e$, there is $\pi' \in \Pi_d$ having the following property: $\forall b \in \pi$,we have $b \in \pi'$ as well if $j \notin b$, let $b_1 \in \pi$ be the block containing j and $j+1$. We can always uniquely split b_1 into two disjoint blocks $b_2, b_3 \in \pi'$, where j is the element with maximal sample mean in b_2 and $j+1$ is the element with minimal sample mean in b_3.

Recalling condition labels in y, we have

$$\frac{P(x|M_{\pi'}, y)}{P(x|M_{\pi}, y)} = \frac{f(x_{b_2})f(x_{b_3})}{f(x_{b_2,b_3})} = \exp(g(\widehat{\mu}_{b_2}, \widehat{\mu}_{b_3}, n_{b_2}, n_{b_3}, \alpha, \beta, \gamma)). \tag{13}$$

where the left hand side ratio only depends on the data from blocks b_2 and b_3. We treat x_{b_2} and x_{b_3} as two "new groups", thus the ratio in the middle is also a two-group Bayes factor, where we have bigger difference in sample means: $\widehat{\mu}_{b_2} \leq \widehat{\mu}_j$ and $\widehat{\mu}_{b_3} \geq \widehat{\mu}_{j+1}$. Also we have more sample sizes: $n_{b_2} \geq n_j$ and $n_{b_3} \geq n_{j+1}$. By Lemma .0.6, we have $\frac{f(x_{b_2})f(x_{b_3})}{f(x_{b_2,b_3})} \geq D_{j,j+1} > 1$, which means for any $\pi \in \Pi_e$, there is $\pi' \in \Pi_d$ such that $P(x|M_{\pi'}, y) > P(x|M_{\pi}, y)$. Thus, partitions with maximal $P(x|M_{\pi}, y)$ must assign differential means between groups j and $j+1$.

A.15 Computational details

Number of groups K

For all the empirical examples used in Section 2, their group number K is determined via clustering analysis in scDDboost (Ma et al., 2019)

Hyperparameters of two-group Bayes factor

Under different hyperparameters the final selected partitions $\mathcal{S}$ may vary. To measure the similarity between two sets of partitions $\mathcal{S}_1$ and $\mathcal{S}_2$, we have the metric:

$$\mathcal{I} = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1 \cup \mathcal{S}_2|}$$

The default value of α is 0.4, and β is set to 2 for all units.

$\alpha\beta$	1	2	3
0.3	0.97	0.93	0.91
0.4	0.9	1	0.94
0.5	084	0.91	0.97

table$\mathcal{I}$ for comparing different hyperparamters to the default setting ($\alpha = 0.4, \beta = 2$). Results are averaged over data sets GSE74596 and GSE57872

The scale of α and β are determined by estimations from EBSeq.v1 applied on benchmark data sets built into it and an empirical data GSE45719.

Number of uncertain positions

We provide an option to limit the number of uncertain positions. Users can specify a upper bound U^* for $N_{\text{UC},g}$, which is shared by all the units. Let $K_g^* = \min(N_{\text{UC},g}, U^*)$. Then for unit g, we keep the smallest K_g^* uncertain positions in terms of the absolute value of two-group Bayes factor in the log scale.

A.16 simulation details

In the simulation, we set 200 samples per group, 20000 units (genes) in total. It is biologically plausible that a DE pattern can be shared by several units, so we assign blocks to units and units belong to the same block have same pattern. The blocks is generated from a Chinese restaurant process (CRP) with strength and discount parameter (α_0, α_1) (Aldous et al., 2006). Namely

$$\Pr(C_i = c \mid C_1, \ldots, C_{i-1}) = \begin{cases} \dfrac{\alpha_0 + B\alpha_1}{\theta + i - 1} & \text{if } c \in \text{new block}, \\ \\ \dfrac{|b| - \alpha_1}{\alpha_0 + i - 1} & \text{if } c \in b; \end{cases} \tag{14}$$

Where $C_1, ..., C_i$ are the cluster label for elements $1, ...i$. B is the number of blocks, $|b|$ is the number of elements inside that block. We set $\alpha_1 = 1$ and use α_0 to control the number of blocks. After we obtain the blocking structure of genes, the next step is to generate partitions of groups and assign them to those blocks of genes. One challenge is that the size of possible partitions increases rapidly, for example when $K = 15$, there is 1.38 billion partitions, makes it impossible to directly sample from the population. Again, we use CRP to generate partitions. The remaining problem is to assign partitions to blocks of genes. We observe that in the expression data, partitions π with big $p_{\text{local},\pi}$ for most genes have small number of blocks and thus are coarse. Therefore, we use a monotone mapping, for example, big block of units will be associated with coarse partitions and small block of units will get fine partitions. We use entropy to measure the complexity(fineness) of a partition. That is, for a partition π, we have the entropy: $-\Sigma_b \log(\frac{|b|}{\Sigma_b |b|}) \frac{|b|}{\Sigma_b |b|}$, it reaches minimum when all samples belong to a single block and reaches maximum when every sample forms a distinct block. We assign the patterns with low entropy to big blocks. At each unit, counts are sampled from a Beta mixture of Negative Binomial (NB) distributions.

References

Aldous, D. J., Ibragimov, I. A., and Jacod, J. (2006). *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983*, volume 1117. Springer.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology* **11,** R106–R106.

Bacher, R. and Kendziorski, C. (2016). Design and computational analysis of single-cell rna-sequencing experiments. *Genome Biology* **17,** 63.

Bachmann, P. (1894). *Die analytische zahlentheorie*, volume 2. Teubner.

Baños-Lara, M. D. R., Zabaleta, J., Garai, J., Baddoo, M., and Guerrero-Plata, A. (2018). Comparative analysis of mirna profile in human dendritic cells infected with respiratory syncytial virus and human metapneumovirus. *BMC research notes* **11,** 432.

Boost (2015). Boost C++ Libraries. `http://www.boost.org/`. Last accessed 2015-06-30.

Buettner, F., Natarajan, K. N., Casale, F. P., Proserpio, V., Scialdone, A., Theis, F. J., Teichmann, S. A., Marioni, J. C., 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,** 155 EP –.

Chen, W., Li, Y., Easton, J., Finkelstein, D., Wu, G., and Chen, X. (2018). Umi-count modeling and differential expression analysis for single-cell rna sequencing. *Genome Biology* **19,** 70.

Chu, L.-F., Leng, N., Zhang, J., Hou, Z., Mamott, D., Vereide, D. T., Choi, J., Kendziorski, C., Stewart, R., and Thomson, J. A. (2016). Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biology* **17,** 173.

Dahl, D. B. (2009a). Modal clustering in a class of product partition models. *Bayesian Anal.* **4,** 243–264.

Dahl, D. B. (2009b). Modal clustering in a class of product partition models. *Bayesian Anal.* **4,** 243–264.

Darmanis, S., Sloan, S. A., Croote, D., Mignardi, M., Chernikova, S., Samghababi, P., Zhang, Y., Neff, N., Kowarsky, M., Caneda, C., Li, G., Chang, S. D., Connolly, I. D., Li, Y., Barres, B. A., Gephart, M. H., and Quake, S. R. (2017). Single-cell rna-seq analysis of infiltrating neoplastic cells at the migrating front of human glioblastoma. *Cell reports* **21,** 1399–1410.

Delmans, M. and Hemberg, M. (2016). Discrete distributional differential expression (d3e) - a tool for gene expression analysis of single-cell rna-seq data. *BMC Bioinformatics* **17,** 110.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39,** 1–22.

Deng, Q., Ramsköld, D., Reinius, B., and Sandberg, R. (2014). Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343,** 193–196.

Dominguez, D., Tsai, Y.-H., Gomez, N., Jha, D. K., Davis, I., and Wang, Z. (2016). A high-resolution transcriptome map of cell cycle reveals novel connections between periodic genes and cancer. *Cell Research* **26,** 946 EP –.

Efron, B. (2005). Local false discovery rates.

Efron, B. (2007). Size, power and false discovery rates. *Ann. Statist.* **35,** 1351–1377.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science* **297,** 1183–1186.

Engel, I., Seumois, G., Chavez, L., Samaniego-Castruita, D., White, B., Chawla, A., Mock, D., Vijayanand, P., and Kronenberg, M. (2016a). Innate-like functions of natural killer t cell subsets result from highly divergent gene programs. *Nature immunology* **17,** 728–739.

Engel, I., Seumois, G., Chavez, L., Samaniego-Castruita, D., White, B., Chawla, A., Mock, D., Vijayanand, P., and Kronenberg, M. (2016b). Innate-like functions of natural killer t cell subsets result from highly divergent gene programs. *Nature Immunology* **17,** 728 EP –.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., Linsley, P. S., and Gottardo, R. (2015). Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome Biology* **16,** 278.

Gardner, M. (1978). Mathematical Games. *Scientific American* **238,** 24–30.

Guennebaud, G., Jacob, B., et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Guo, F., Yan, L., Guo, H., Li, L., Hu, B., Zhao, Y., Yong, J., Hu, Y., Wang, X., Wei, Y., Wang, W., Li, R., Yan, J., Zhi, X., Zhang, Y., Jin, H., Zhang, W., Hou, Y., Zhu, P., Li, J., Zhang, L., Liu, S., Ren, Y., Zhu, X., Wen, L., Gao, Y. Q., Tang, F., and Qiao, J. (2015). The transcriptome and dna methylome landscapes of human primordial germ cells. *Cell* **161,** 1437–1452.

Haghverdi, L., Buettner, F., and Theis, F. J. (2015). Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics* **31,** 2989–2998.

Heller, K. A. and Ghahramani, Z. (2005). Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304.

Hrvatin, S., Hochbaum, D. R., Nagy, M. A., Cicconet, M., Robertson, K., Cheadle, L., Zilionis, R., Ratner, A., Borges-Monroy, R., Klein, A. M., et al. (2018). Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. *Nature neuroscience* **21,** 120–129.

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J. I., Raj, A., Li, M.,

and Zhang, N. R. (2018). Saver: gene expression recovery for single-cell rna sequencing. *Nature Methods* **15,** 539–542.

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., et al. (2015). Orchestrating high-throughput genomic analysis with bioconductor. *Nature methods* **12,** 115.

Hwang, B., Lee, J. H., and Bang, D. (2018). Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine* **50,** 1–14.

Jara, A., Hanson, T., Quintana, F., Müller, P., and Rosner, G. (2011). Dppackage: Bayesian semi- and nonparametric modeling in r. *Journal of Statistical Software, Articles* **40,** 1–30.

Jensen, S. T., Erkan, I., Arnardottir, E. S., and Small, D. S. (2009). Bayesian testing of many hypotheses x many genes: A study of sleep apnea. *Ann. Appl. Stat.* **3,** 1080–1101.

Kaufman, L. and Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Kendziorski, C., Newton, M., Lan, H., and Gould, M. (2003a). On parametric empirical bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in medicine* **22,** 3899–3914.

Kendziorski, C. M., Newton, M. A., Lan, H., and Gould, M. N. (2003b). On parametric empirical bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine* **22,** 3899–3914.

Kim, T., Chen, I. R., Lin, Y., Wang, A. Y.-Y., Yang, J. Y. H., and Yang, P. (2018). Impact of similarity metrics on single-cell rna-seq data clustering. *Briefings in Bioinformatics* page bby076.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., and Hemberg, M. (2017). Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods* **14,** 483 EP –.

Korthauer, K. D., Chu, L.-F., Newton, M. A., Li, Y., Thomson, J., Stewart, R., and Kendziorski, C. (2016a). A statistical approach for identifying differential distributions in single-cell rna-seq experiments. *Genome Biology* **17,** 222.

Korthauer, K. D., Chu, L.-F., Newton, M. A., Li, Y., Thomson, J., Stewart, R., and Kendziorski, C. (2016b). A statistical approach for identifying differential distributions in single-cell rna-seq experiments. *Genome Biology* **17,** 222.

Lane, K., Van Valen, D., DeFelice, M. M., Macklin, D. N., Kudo, T., Jaimovich, A., Carr, A., Meyer, T., Pe'er, D., Boutet, S. C., and Covert, M. W. (2017). Measuring signaling and rna-seq in the same cell links gene expression to dynamic patterns of nf-b activation. *Cell Systems* **4,** 458–469.e5.

Lee, C.-J., Ahn, H., Jeong, D., Pak, M., Moon, J. H., and Kim, S. (2020). Impact of mutations in dna methylation modification genes on genome-wide methylation landscapes and downstream gene activations in pan-cancer. *BMC Medical Genomics* **13,** 1–14.

Leng, N., Chu, L.-F., Barry, C., Li, Y., Choi, J., Li, X., Jiang, P., Stewart, R. M., Thomson, J. A., and Kendziorski, C. (2015). Oscope identifies oscillatory genes in unsynchronized single-cell rna-seq experiments. *Nature Methods* **12,** 947 EP –.

Leng, N., Dawson, J. A., Thomson, J. A., Ruotti, V., Rissman, A. I., Smits, B. M. G., Haag, J. D., Gould, M. N., Stewart, R. M., and Kendziorski, C. (2013). Ebseq: an empirical bayes hierarchical model for inference in rna-seq experiments. *Bioinformatics* **29,** 1035–1043.

Leng, N., Dawson, J. A., Thomson, James A.and Ruotti, V., Rissman, A. I., Smits, B. M. G., Haag, J. D., Gould, M. N., Stewart, R. M., and Kendziorski, C. (2013). EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics* **29,** 1035–1043.

Leng, N. and Kendziorski, C. (2019). *EBSeq: An R package for gene and isoform differential expression analysis of RNA-seq data*. R package version 1.24.0.

Li, F. and Altieri, D. C. (1999). The cancer antiapoptosis mouse *survivin* gene. *Cancer Research* **59,** 3143.

Li, X. and ping Chen, C. (2007). Inequalities for the gamma function. In *2007), Art. 28. [ONLINE: http://jipam.vu.edu.au/ article.php?sid=842*.

Lin, P., Troup, M., and Ho, J. W. K. (2017). Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biology* **18,** 59.

Little, A. F., Maggioni, M., and Murphy, J. M. (2017). Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods* **15,** 1053–1058.

Louro, B., Marques, J. P., Manchado, M., Power, D. M., and Campinho, M. A. (2020). Sole head transcriptomics reveals a coordinated developmental program during metamorphosis. *Genomics* **112,** 592–602.

Love, M. I., Huber, W., and Anders, S. (2014a). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology* **15,** 550.

Love, M. I., Huber, W., and Anders, S. (2014b). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology* **15,** 550.

Ma, B. X., Korthauer, K., Kendziorski, C., and Newton, M. A. (2019). A compositional model to assess expression changes from single-cell rna-seq data. *bioRxiv* .

MacEachern, S. N. (1998). Computational methods for mixture of dirichlet process models. In *Practical nonparametric and semiparametric Bayesian statistics*, pages 23–43. Springer.

Marioni, J. C. and Arendt, D. (2017). How single-cell genomics is changing evolutionary and developmental biology. *Annual Review of Cell and Developmental Biology* **33,** 537–553. PMID: 28813177.

McDavid, A., Dennis, L., Danaher, P., Finak, G., Krouse, M., Wang, A., Webster, P., Beechem, J., and Gottardo, R. (2014). Modeling bi-modality improves characterization of cell cycle on gene expression in single cells. *PLOS Computational Biology* **10,** e1003696–.

Muralidharan, O. (2010). An empirical bayes mixture method for effect size and false discovery rate estimation. *Ann. Appl. Stat.* **4,** 422–438.

Navin, N. E. (2015). The first five years of single-cell cancer genomics and beyond. *Genome Research* **25,** 1499–1507.

Nawy, T. (2013). Single-cell sequencing. *Nature Methods* **11,** 18 EP –.

Newhouse, D. J., Hofmeister, E. K., and Balakrishnan, C. N. (2017). Transcriptional response to west nile virus infection in the zebra finch (taeniopygia guttata). *Royal Society open science* **4,** 170296.

Newton, M. A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics* **5,** 155–176.

O'Grady, T., Baddoo, M., and Flemington, E. K. (2017). Analysis of ebv transcription using high-throughput rna sequencing. In *Epstein Barr Virus*, pages 105–121. Springer.

Papalexi, E. and Satija, R. (2017). Single-cell rna sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology* **18,** 35 EP –.

Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., Gillespie, S. M., Wakimoto, H., Cahill, D. P., Nahed, B. V., Curry, W. T., Martuza, R. L., et al. (2014). Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344,** 1396–1401.

Peccoud, J. and Ycart, B. (1995). Markovian modeling of gene-product synthesis. *Theoretical Population Biology* **48,** 222–234.

Pierson, E. and Yau, C. (2015). Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology* **16,** 241.

Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H. A., and Trapnell, C. (2017). Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods* **14,** 979–982.

Quintana, F. A. and Iglesias, P. L. (2003). Bayesian clustering and product partition models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65,** 557–574.

Raj, A. and van Oudenaarden, A. (2008). Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell* **135,** 216–226.

Ray, S. and Turi, R. H. (2000). Determination of number of clusters in k-means clustering and application in colour image segmentation.

Sabbagh, M. F., Heng, J. S., Luo, C., Castanon, R. G., Nery, J. R., Rattner, A., Goff, L. A., Ecker, J. R., and Nathans, J. (2018). Transcriptional and epigenomic landscapes of cns and non-cns vascular endothelial cells. *Elife* **7,** e36187.

Saliba, A.-E., Westermann, A. J., Gorski, S. A., and Vogel, J. (2014). Single-cell rna-seq: advances and future challenges. *Nucleic acids research* **42,** 8845–8860.

Sanders, S. M. and Cartwright, P. (2015). Patterns of wnt signaling in the life cycle of podocoryna carnea and its implications for medusae evolution in hydrozoa (cnidaria). *Evolution & development* **17,** 325–336.

Shalek, A. K., Satija, R., Shuga, J., Trombetta, J. J., Gennert, D., Lu, D., Chen, P., Gertner, R. S., Gaublomme, J. T., Yosef, N., Schwartz, S., Fowler, B., Weaver, S., Wang, J., Wang, X., Ding, R., Raychowdhury, R., Friedman, N., Hacohen, N., Park, H., May, A. P., and Regev, A. (2014). Single-cell rna-seq reveals dynamic paracrine control of cellular variation. *Nature* **510,** 363 EP –.

Shekhar, K., Lapan, S. W., Whitney, I. E., Tran, N. M., Macosko, E. Z., Kowalczyk, M., Adiconis, X., Levin, J. Z., Nemesh, J., Goldman, M., et al. (2016). Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell* **166,** 1308–1323.

Sohr, S. and Engeland, K. (2008). Rhamm is differentially expressed in the cell cycle and downregulated by the tumor suppressor p53. *Cell Cycle* **7,** 3448–3460.

Son, J. C., Jeong, H. O., Park, D., No, S. G., Lee, E. K., Lee, J., and Chung, H. Y. (2017). mir-10a and mir-204 as a potential prognostic indicator in low-grade gliomas. *Cancer informatics* **16,** 1176935117702878.

Soneson, C. and Robinson, M. D. (2017). Bias, robustness and scalability in differential expression analysis of single-cell rna-seq data. *bioRxiv* .

Song, X., Tang, T., Li, C., Liu, X., and Zhou, L. (2018). Cbx8 and cd96 are important prognostic biomarkers of colorectal cancer. *Medical science monitor: international medical journal of experimental and clinical research* **24,** 7820.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3,** 583–617.

Svensson, V., Vento-Tormo, R., and Teichmann, S. A. (2018). Exponential scaling of single-cell rna-seq in the past decade. *Nature Protocols* **13,** 599–604.

Tasic, B., Menon, V., Nguyen, T. N., Kim, T. K., Jarsky, T., Yao, Z., Levi, B., Gray, L. T., Sorensen, S. A., Dolbeare, T., Bertagnolli, D., Goldy, J., Shapovalova, N., Parry, S., Lee, C., Smith, K., Bernard, A., Madisen, L., Sunkin, S. M., Hawrylycz, M., Koch, C., and Zeng, H. (2016). Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience* **19,** 335 EP –.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014a). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32,** 381–386.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014b). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32,** 381.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014c). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32,** 381–386.

Wagner, U. and Taudes, A. (1986). A multivariate polya model of brand choice and purchase incidence. *Marketing Science* **5,** 219–244.

Wang, T., Li, B., Nelson, C. E., and Nabavi, S. (2019). Comparative analysis of differential gene expression analysis tools for single-cell rna sequencing data. *BMC bioinformatics* **20,** 40.

Yakowitz, S. J. and Spragins, J. D. (1968). On the identifiability of finite mixtures**39,** 209–214.

Yang, F., Lv, S.-X., Lv, L., Liu, Y.-H., Dong, S.-Y., Yao, Z.-H., Dai, X.-x., Zhang, X.-H., and Wang, O.-C. (2016). Identification of lncrna fam83h-as1 as a novel prognostic marker in luminal subtype breast cancer. *OncoTargets and therapy* **9,** 7039.

Yoon, K.-J., Ringeling, F. R., Vissers, C., Jacob, F., Pokrass, M., Jimenez-Cyrus, D., Su, Y., Kim, N.-S., Zhu, Y., Zheng, L., et al. (2017). Temporal control of mammalian cortical neurogenesis by m6a methylation. *Cell* **171,** 877–889.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell rna sequencing data. *Genome Biology* **18,** 174.

Zhang, Q., Zeng, L.-P., Zhou, P., Irving, A. T., Li, S., Shi, Z.-L., and Wang, L.-F. (2017). Ifnar2-dependent gene expression profile induced by ifn-α in pteropus alecto bat cells and impact of ifnar2 knockout on virus infection. *PloS one* **12,**.