

# Differential gene regulation

---

Master Thesis in Biostatistics (STA495)

by

Joël Meili  
14-679-393

supervised by

Prof. Dr. Mark D. Robinson  
Dr. Simone Tiberi

Zurich, October 2022



# Differential gene regulation

Joël Meili

Version October 12, 2022



# Contents

<b>Preface</b>	iii
<b>Abstract</b>	1
<b>1 Introduction</b>	3
1.1 RNA sequencing . . . . .	3
1.2 Objective . . . . .	5
<b>2 Methods</b>	7
2.1 Alignment and quantification with alevin-fry . . . . .	7
2.2 Read-level simulation with minnow . . . . .	7
2.3 Differential methods . . . . .	8
2.4 Analysis of results . . . . .	11
<b>3 Results</b>	13
3.1 Exploratory Data Analysis . . . . .	13
3.2 Simulation study . . . . .	20
3.3 Null analysis on the mouse kidney data . . . . .	27
3.4 Computational benchmark . . . . .	28
3.5 Data availability . . . . .	29
3.6 Code availability . . . . .	29
<b>4 Discussion</b>	31
4.1 Conclusion . . . . .	31
4.2 Outlook . . . . .	31
<b>Bibliography</b>	31



# Preface

Joël Meili  
October 2022



# Abstract

Here will be the abstract ..



# Chapter 1

## Introduction

### 1.1 RNA sequencing

RNA sequencing (RNA-seq) is a technology for detecting and quantifying the mRNA molecules of a biological sample (Stark *et al.*, 2019). The invention of RNA-seq was a major breakthrough in the field of bioinformatics that replaced the use of microarray technology in the late 00's. In comparison to microarrays, RNA-seq allows for full sequencing of the whole transcriptome whereas microarrays only profile predefined transcripts through hybridization (Rao *et al.*, 2019). Further, various protocols have since been derived from the standard RNA-seq protocol, e.g. single-cell RNA sequencing (Stark *et al.*, 2019).

#### 1.1.1 Bulk RNA-seq

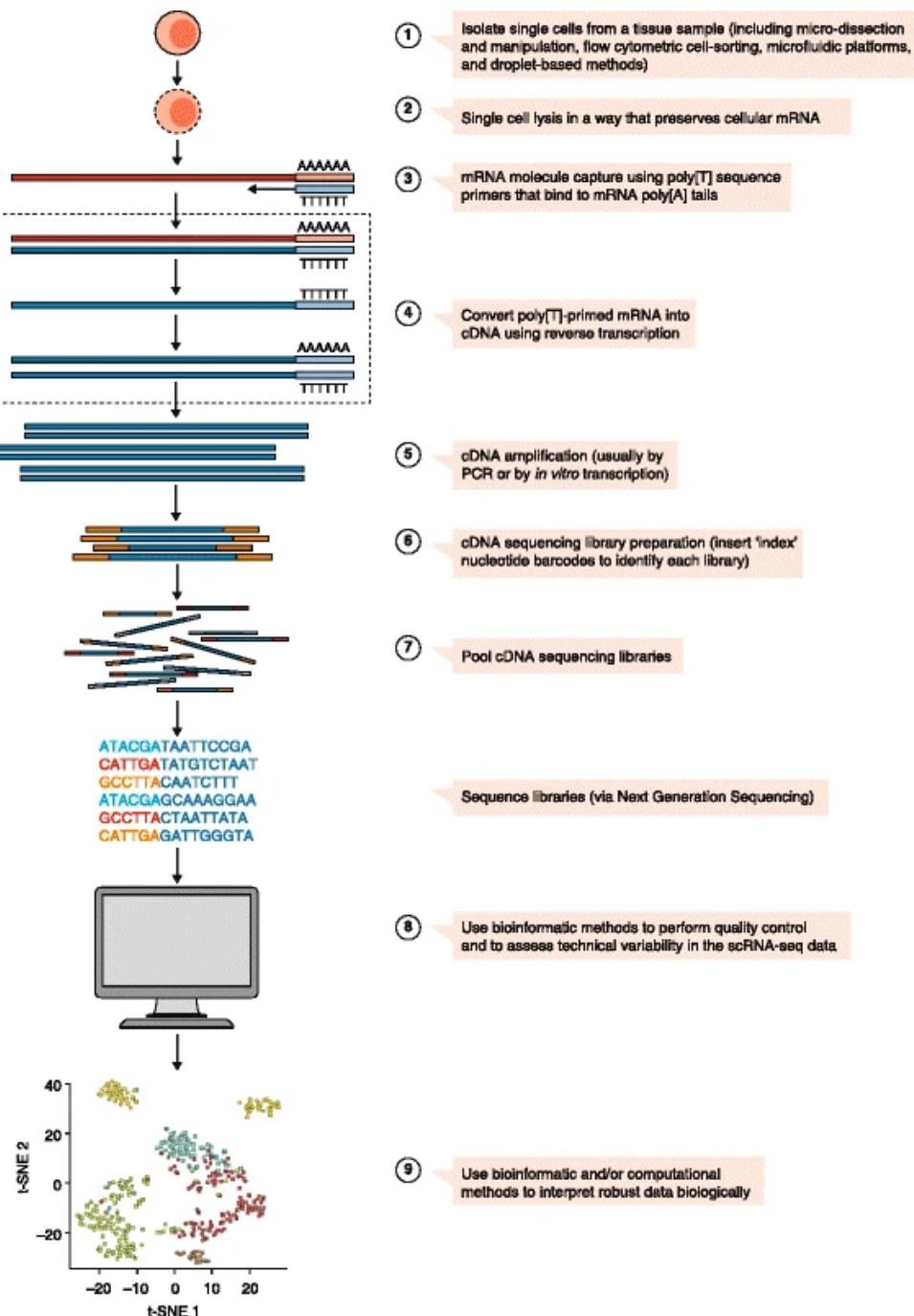
Bulk RNA sequencing allows detecting an aggregated signal across a mixture of cells. There are many applications for bulk RNA-seq. For example, it can be used to study the differences of expression profiles between tissues in healthy vs disease or across treatments (Stark *et al.*, 2019). However, with bulk RNA-seq one can only estimate the average expression of each gene across a population of cells without regard for the differences between cell types. RNA-seq has several use cases. It can be used to study which genes are turned on in a cell and what their level of transcription is. This allows researchers to understand the biology of a cell at a deeper level. Further, RNA-seq allows the identification of variants and allele specific expression. It is also possible to study the patterns of alternative splicing, which are important to understand their contribution to cell differentiation and human disease.

#### 1.1.2 Single-cell RNA-seq

Single-cell RNA sequencing was developed to overcome some of the limitations of bulk RNA sequencing. With scRNA-seq it is possible to estimate the distribution of expression levels for each gene across a population of cells. This allows answering new biological questions where cell-specific characteristics are important. However, there are some caveats with scRNA-seq (Haque *et al.*, 2017). scRNA-seq data in general is much more variable than bulk RNA-seq data due to both higher biological and technical variability at single-cell level (Haque *et al.*, 2017). Figure 1.1 shows the typical workflow of a scRNA-seq experiment. Said workflow is broadly summarized by the following steps:

1. RNA extraction
2. Reverse transcription into cDNA
3. Adapted ligation

4. Amplification
5. Sequencing
6. Downstream analysis using bioinformatics tools



**Figure 1.1:** General workflow of single-cell RNA-seq experiments ([Haque et al., 2017](#))

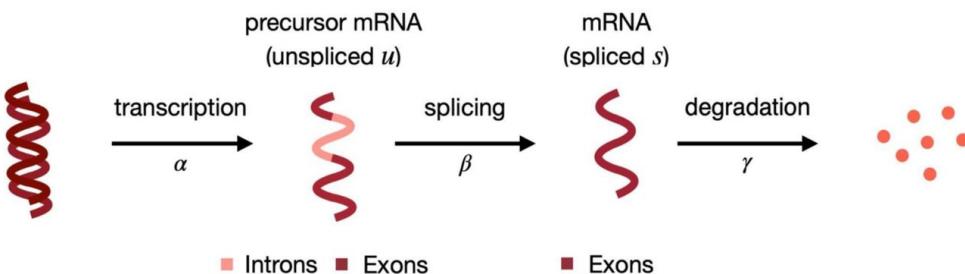
### 1.1.3 Quantification of single-cell RNA-seq data

scRNA-seq data has distinct characteristics that prevent it from being processed by widely used tools developed for bulk RNA-seq data (He *et al.*, 2022). In general, quantification works by aligning the reads generated from the RNA-seq to the reference genome. There are several tools that allow to do that, notably: STAR (Dobin *et al.*, 2013), kallisto | bustools (Melsted *et al.*, 2021) and alevin (Srivastava *et al.*, 2019). However, there is a difference between the first tool and the other two. STAR is an aligner, whereas the other two tools are mapping tools (pseudo-aligners). The difference between an full-aligner and a mapping tool is that the latter does not look for the exact location of the read, as a consequence pseudo-alignment is much faster than full-alignment. Here we focus on alevin-fry, and the method we have developed, which will be introduced later, has been built to work on the output of alevin-fry.

## 1.2 Objective

### 1.2.1 RNA velocity

We investigate spliced and unspliced reads from single-cell RNA-sequencing data. During transcription, DNA is decoded into precursor messenger RNA (pre-mRNA). Pre-mRNA contains both coding (exons) and non-coding regions (introns). In a next step, introns are removed from the pre-mRNA which leaves only the mature mRNA. Figure 1.2 shows the process from DNA to mature mRNA, where  $\alpha$  is the transcription rate,  $\beta$  is the splicing rate and  $\gamma$  is the degradation rate.



**Figure 1.2:** The transcription process from DNA to mature mRNA (Weiler *et al.*, 2021)

It was assumed that there is a signal (RNA velocity) detectable in scRNA-seq data that could reveal the rate and direction of change of an entire transcriptome (La Manno *et al.*, 2018). To quantify the relationship between the abundance of pre-mRNA and mature RNA, a simple system of ordinary differential equations was assumed (1.1): The solution of said system at equilibrium can then easily be estimated and used to explore the regulation of genes:

$$\begin{aligned} \frac{du}{dt} &= \alpha - \beta u \\ \frac{ds}{dt} &= \beta u - \gamma s \end{aligned} \tag{1.1}$$

The derivative of the spliced counts is then defined as the RNA velocity of cells. Thus, the balance of spliced and unspliced counts allows estimating whether a gene is up- or downregulated. If a larger fraction of unspliced counts are present than expected at equilibrium, a gene is likely upregulated. This is because within a short time interval, the newly spliced mRNA will exceed the amount of spliced mRNA which is degraded. Contrarily, if more spliced counts are present at equilibrium than expected, a gene is likely downregulated.

### 1.2.2 Differential regulation

The abundance of spliced and unspliced reads is directly linked to the regulation of genes and RNA velocities. Our idea is to examine how the abundance of spliced and unspliced counts changes between experimental conditions and biological replicates. We translate this intuition into the comparison of two experimental conditions, e.g. healthy vs. disease. Following the same intuitive rationale of RNA velocity, if a gene has a higher abundance of unspliced (spliced) counts in group A compared to group B, then this gene is likely being up-regulated (down-regulated) in group A compared to group B. Thus, we explore the differences in abundance of spliced and unspliced counts to study the differences in regulation between experimental conditions.

If the data contains multiple cell clusters (e.g. cell types), similarly to differential state analyses ([Crowell et al. \(2020\)](#) and [Tiberi et al. \(2021\)](#)) we will perform differential analyses in each cluster of cells, hence identifying cell-cluster/cell-type specific changes between conditions. The idea of performing differential analyses on the abundance of spliced and unspliced or exonic and intronic reads is not completely novel as there are at least two other methods that achieve that: `eisaR` and `BRIE2`.

### 1.2.3 Existing methods

`eisaR` ([Stadler et al., 2020](#)) is a R package implementation that allows for the split analysis between exons and introns. It allows one to measure changes in mature RNA and pre-mRNA across different experimental conditions. Ultimately, `eisaR` differential testing is based on `edgeR` ([Robinson et al., 2010](#)). `edgeR` is a R package that performs differential expression analyses between groups of samples. It implements statistical methods that are based on the negative binomial distribution as a model for count variability.

`BRIE2` ([Huang and Sanguinetti, 2021](#)) is a Bayesian hierarchical model that is implemented in Python and supports the analysis of splicing processes between spliced and unspliced RNA. There are two modes in which the tool can be used. First, the use of differential alternative splicing (DAS), where the aim is to quantify the proportions of alternative splicing isoforms. Second, the use of differential momentum genes (DMG), where the objective is to quantify the proportions of spliced and unspliced RNA in each gene and each cell. In this thesis we focus on the DMG mode as it performs differential testing on the relative abundance of spliced and unspliced reads.

Originally, `eisaR` and `BRIE2` were developed to analyse all cells, but can easily be adapted to perform cell-type specific differential analyses.

### 1.2.4 Mapping uncertainty

We can identify two main sources of mapping uncertainty concerning spliced and unspliced reads: i) multi-mapping reads across spliced and unspliced versions of a gene, and ii) reads compatible with multiple genes. In fact, it has been shown that many reads (5-40%) map to multiple genes ([Dharshini et al. \(2020\)](#), [McDermaid et al. \(2018\)](#)). In our real data analyses (see Section 3), we found approximately 20-30% of such multi-mapping reads across genes. We additionally found that a significant fraction of reads (6-19%) are compatible with both S and U versions of a gene. Therefore, the estimated spliced and unspliced counts carry a substantial amount of uncertainty, which should be accounted for in downstream analyses. However, both `eisaR` and `BRIE2` use estimated spliced and unspliced counts and neglect the mapping uncertainty. In this thesis, we propose two approaches that account for said mapping uncertainties.

# Chapter 2

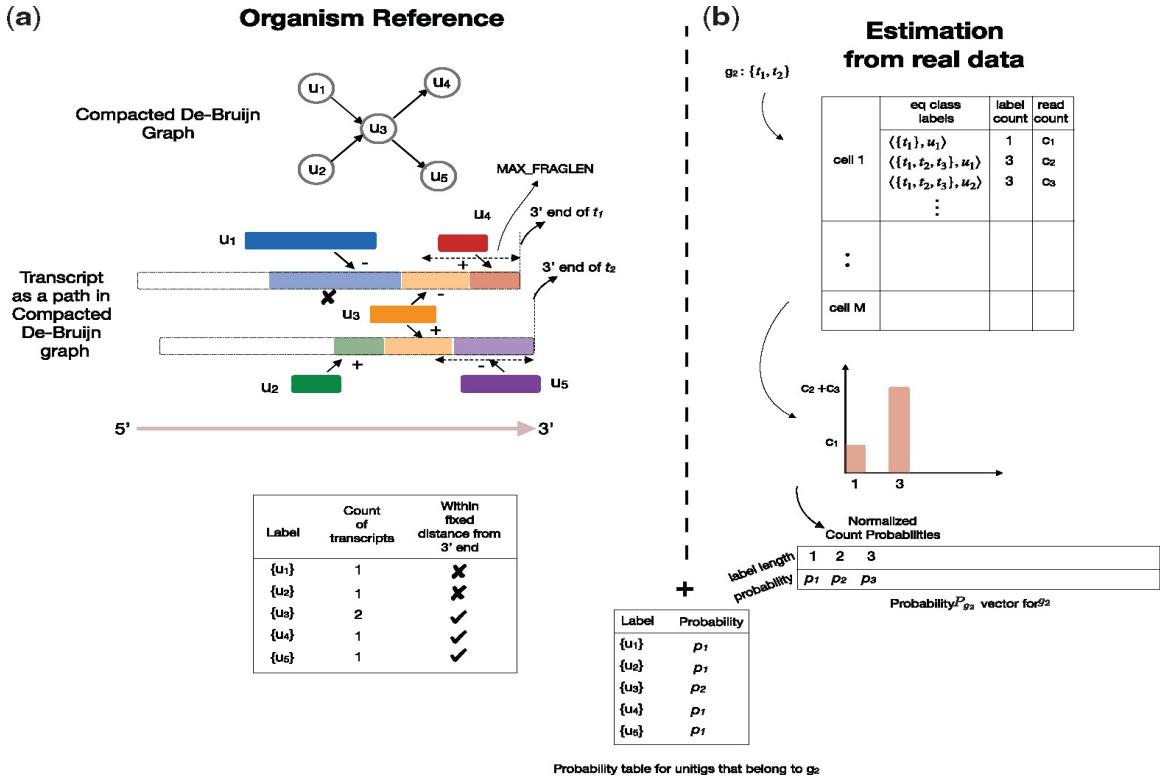
## Methods

### 2.1 Alignment and quantification with alevin-fry

Alevin was developed to tackle the computational challenges that come with scRNA-seq data and to provide a tool that supports technologies other than 10x Genomics. Alevin works in two steps. First, it parses a read file which contains the cellular barcode and a unique molecule identifier to generate a frequency distribution of observed barcodes. Second, it maps the reads to the transcriptome and generates a cell-by-gene count matrix. Alevin-fry ([He et al., 2022](#)) was designed to be the successor to alevin and achieves similar accuracy at significantly lower computational costs. It generates a permit list for cellular barcodes that will be quantified in subsequent steps. By using a multi-thread approach, alevin-fry filters and collates the mapping records for permitted cellular barcodes to produce a representation optimized for quantification ([He et al., 2022](#)).

### 2.2 Read-level simulation with minnow

*minnow* is a read level simulator for droplet based scRNA-seq data that accounts for important sequence-level characteristics and model effects ([Sarkar et al., 2019](#)). It matches the gene-level ambiguity characteristics that are present in real scRNA-seq experiments. With *minnow* it is possible to demonstrate the effect of gene-level sequence ambiguity on accurate quantification, which is used in this thesis to simulate mapping uncertainty between spliced and unspliced counts. It achieves this by either simulating sequences from the underlying de-Bruijn graph of the reference transcriptome or from the reference transcriptome directly ([Sarkar et al., 2019](#)). The *minnow* framework essentially works in three steps: (i) selection of transcript, (ii) simulation of cell barcode (CB) and unique molecular identifier (UMI) tagging and (iii) simulation of PCR, fragmentation and sequencing. First, *minnow* uses a gene-count matrix as input that provides an estimated number of distinct molecules corresponding to each gene and cell in the sample. *minnow* treats the normalized values of a particular cell as a multinomial distribution, then samples such molecules from that distribution ([Sarkar et al., 2019](#)). Figure 2.1 illustrates the process from input to simulated reads.



**Figure 2.1:** Summary of the two possible pathways of *minnow* from start to simulated reads (Sarkar *et al.*, 2019)

## 2.3 Differential methods

### 2.3.1 *eisaR*

Exon-intron split analysis (EISA) is a computational approach to measure mature RNA and pre-mRNA reads across different experimental conditions (Gaidatzis *et al.*, 2015). The method has been developed to detect changes in up- or down-regulated gene expression. *eisaR* works in multiple steps: first, both exonic and intronic counts are normalized to the mean library size and log-transformed with the addition of a pseudocount; second, genes with very low counts in either exons or introns are removed, such that there is a fixed set of quantifiable genes; third, the statistical analysis of the counts is conducted with the help of the *edgeR* framework (Robinson *et al.*, 2010). The *edgeR* framework is used to examine differential gene expression of replicated count data where the counts are modelled by a negative binomial distribution (2.1):

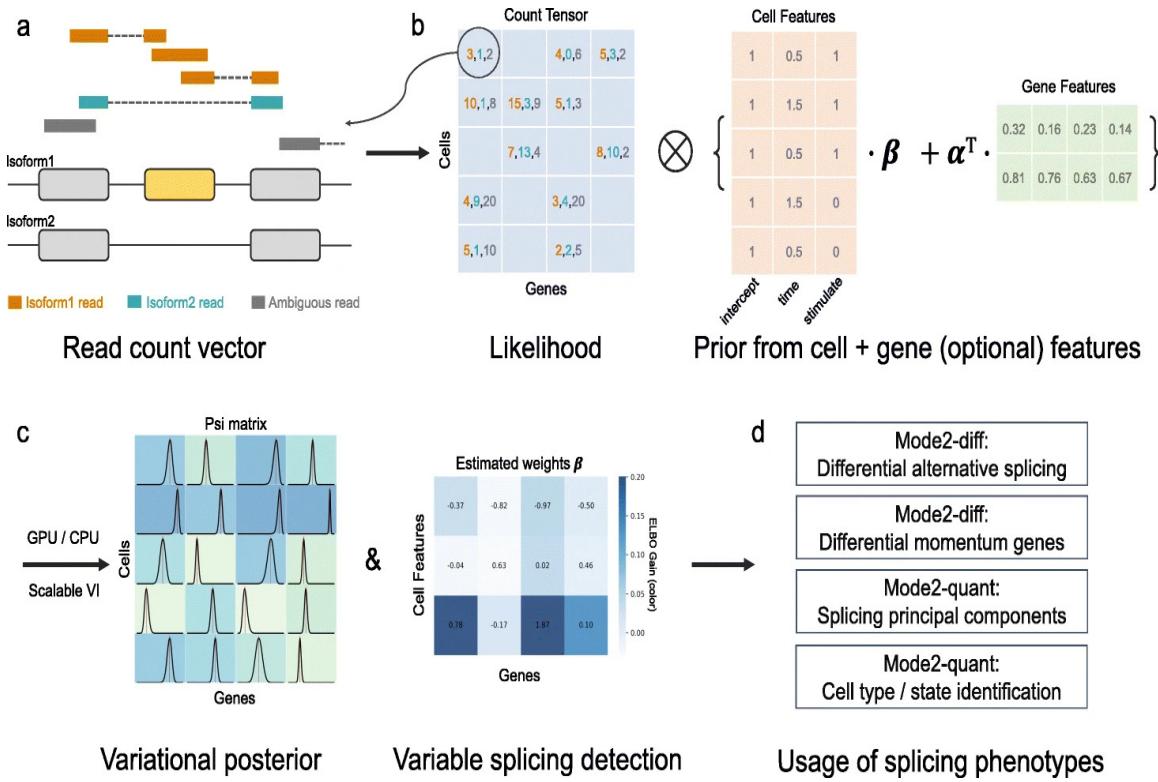
$$Y_{gi} \sim NB(M_i \rho_{gj}, \phi_g) \quad (2.1)$$

where  $g$  is the gene;  $i$  is the sample;  $M_i$  is the library size (total counts);  $\phi_g$  is the dispersion and  $\rho_{gj}$  is the relative abundance of gene  $g$  in experimental group  $j$  which sample  $i$  belongs to;  $NB(a, b)$  denotes the negative binomial distribution with mean  $a$  and dispersion  $b$ .

### 2.3.2 *BRIE2*

*BRIE2* is a scalable computational method that regresses single-cell RNA-seq data against cell-level features (Huang and Sanguinetti, 2021). Unavoidable difficulty in the quantification arises from the fundamental ambiguity of the data, because a majority of reads cannot be mapped unambiguously to a single isoform. *BRIE1* tackled this issue by regressing percentage of spliced-in (PSI) values through a Bayesian regression approach. However, *BRIE1* is not well suited

to quantify differential splicing across cell types because sequence features are usually the same between individual cells (Huang and Sanguinetti, 2021). *BRIE2* again starts from a latent regression framework, however, it differs from *BRIE1* in two important ways: first, it augments the set of regressor features by including cell-specific information e.g. cell type; second, the added complexity considerably increases the computational cost. Therefore, because of its elevated computational complexity, *BRIE2* was developed to be used in conjunction with advanced software e.g. Tensorflow and graphics processing units (GPUs), which significantly increases computational acceleration. Figure 2.2 summarizes the process from input to output in the *BRIE2* pipeline.



**Figure 2.2:** Summary of the estimation process of *BRIE2* (Huang and Sanguinetti, 2021)

As mentioned in the introduction, we focused on the differential momentum genes mode of *BRIE2*. As in this mode *BRIE2* does not use sequence level features, different splicing events are modelled independent under the model, therefore allowing comparison on an event by event basis. Equations (2.2) and (2.3) are the two posterior distributions where  $t$  denotes all cell features except feature  $t$ . In our case we only have one cell feature which is the group allocation, as we run *BRIE2* on all cell types individually. The two posterior distributions represent the two models - Model 0 with no effects and Model 1 with significant effects.

$$M_0 : p_g^{(0)} = \prod_{c=1}^M p(s_{c,g}|z_{c,g}) N(z_{c,g}|y_{c,t} \cdot 0 + y_{c,t}^\top \beta_{g,t}, \sigma_g^2) \quad (2.2)$$

$$M_1 : p_g^{(1)} = \prod_{c=1}^M p(s_{c,g}|z_{c,g}) N(z_{c,g}|y_{c,t} \cdot \beta_{g,t} + y_{c,t}^\top \beta_{g,t}, \sigma_g^2) \quad (2.3)$$

### 2.3.3 DEXSeq

*DEXSeq* (Anders *et al.*, 2012) is a statistical method originally proposed to test for differential exon usage in RNA-seq data, which has been widely adopted in other contexts too, such as differential transcript usage (Love *et al.*, 2018). The model is based on the negative binomial distribution and allows for covariates such as batch effects to be taken into account to offer reliable control of false discoveries (Anders *et al.*, 2012). In its original implementation *DEXSeq* inputs how many reads map to each exon, but the method has also been used on transcript level counts. Equation (2.4) shows that the read counts follow a negative binomial distribution where  $\alpha$  is the dispersion parameter. Further, a generalized linear model is used to predict the mean via a log-linear link:

$$K_{ijl} \sim \text{NB}(\text{mean} = s_j \mu_{ijl}, \text{dispersion} = \alpha_{il}) \quad (2.4)$$

$$\log(\mu_{ijl}) = \beta_i^G + \beta_{il}^E + \beta_{i\rho_j}^C + \beta_{i\rho_j l}^{EC} \quad (2.5)$$

where  $\text{NB}(a, b)$  denotes the negative binomial distribution with mean  $a$  and dispersion  $b$ ,  $\alpha_{il}$  is the dispersion parameter;  $s_j$  is the size factor;  $\mu_{ijl}$  is the expected value of the concentration of cDNA fragments;  $\beta_i^G$  is the baseline expression strength of gene  $i$ ;  $\beta_{il}^E$  is the logarithm of the expected fraction of the reads mapped to gene  $i$ ;  $\beta_{i\rho_j}^C$  is the logarithm of the fold change in overall expression of gene  $i$  under condition  $\rho_j$ ;  $\beta_{i\rho_j l}^{EC}$  is the effect that condition  $\rho_j$  has on the fraction of reads.

The dispersion parameter allows to model over-dispersed data (i.e. higher variance than mean). Here, we propose to use DEXSeq on estimated USA counts, and perform a differential usage test between conditions. This models ambiguous reads separately from spliced and unspliced or exonic and intronic, thus eliminating one of the main sources of mapping uncertainty. However, the uncertainty related to reads mapping to multiple genes is still neglected by this approach.

### 2.3.4 DifferentialRegulation

To address both sources of mapping uncertainty we propose our novel method. Similar to the idea above, we implemented a hierarchical Bayesian approach that models ambiguous counts separately from spliced and unspliced. Alternatively, one could use an estimate for the ambiguous reads, which however is non-trivial to obtain. Gene allocation is modelled as a latent state to address the gene-related mapping uncertainty. The model consists of two nested models: First, a multinomial model that models the relative abundance of genes for each sample individually (2.6). Second, we use a Dirichlet-multinomial model for the relative abundance of the USA counts in each gene (2.7).

$$\text{MN}(\pi_{\text{gene}}^{(1)}, \dots, \pi_{\text{gene}}^{(N_g)}) \quad (2.6)$$

$$\text{DIR-MN}(\pi_U^{(g)}, \pi_S^{(g)}, \pi_A^{(g)}, \phi) \quad (2.7)$$

Bayesian hierarchical modelling splits the inference problem into multiple steps, where the full model is made up of a series of sub-models. The Bayesian Hierarchical model links the sub-models together and uses a Metropolis-within-Gibbs MCMC approach to sample from the conditional distribution.

## 2.4 Analysis of results

### 2.4.1 Classification measurements

The true positive rate (TPR), false positive rate (FPR), and false discovery rate (FDR) are defined as:

$$\text{TPR} = \frac{|\text{TP}|}{|\text{TP} + \text{FN}|} \quad (2.8)$$

$$\text{FPR} = \frac{|\text{FP}|}{|\text{FP} + \text{TN}|} \quad (2.9)$$

$$\text{FDR} = \frac{|\text{FP}|}{|\text{TP} + \text{FP}|} \quad (2.10)$$

where TP, TN, FP, and FN indicate the sets of true positive, true negative, false positive, and false negative elements, respectively. TP are the number of elements that were correctly identified as the positive outcome. Similarly, TN are the number of elements that were correctly identified as the negative outcome. FP are those elements that were identified as the positive outcome, however they should have been identified as negative. In statistics, FP is usually referred to as Type 1 error. Similarly, FN are the elements that were incorrectly identified as negative - also referred to as Type 2 error in the realms of statistics. Figure 2.3 illustrates the meaning of these measurements quite clearly. The TPR (true positive rate) also referred to as sensitivity, measures the proportion of positive elements that were correctly identified - often alluded to as statistical power (2.8). The FPR (false positive rate) measures the proportion of negative outcomes that are identified as positive (2.9). On the other hand, FDR (false discovery rate) measures the expected proportion of false positive among all positive predictions (2.10).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

**Figure 2.3:** Confusion matrix reporting the performance of a binary classification problem ([Mohajon, 2020](#))

#### 2.4.2 ROC curve

The ROC curve is a performance measurement that is used for classification problems - in this case whether a gene is differently regulated or not. Essentially, the ROC curve plots the TPR (y-axis) against the FPR (x-axis). ROC curves above the diagonal indicate better performance than blind random guessing, denoted by the diagonal line. ROC curves are one of the performance evaluation methods used in this thesis.

#### 2.4.3 TPR v. FDR curves

Similar to the ROC curve the TPR v. FDR curve is used to assess performance in classification problems. It plots the TPR (y-axis) against the FDR (x-axis). In addition, one often plots the actual which was observed for some theoretical thresholds - usually 1, 5 and 10%. FDR values are essentially adjusted p-values which can be calculated in various ways from raw p-values. Arguably, the most popular one being the Benjamini-Hochberg correction ([Benjamini and Hochberg, 1995](#)). All the methods considered in this thesis, provide FDR adjusted p-values obtained via Benjamini-Hochberg correction.

When the observed FDR is lower than or equal to the specified threshold, the method controls for the FDR. However, if the observed FDR is greater than the threshold the method does not control for the FDR and there is an inflation of false positive predictions. In this thesis, the methods are evaluated by an adjusted p-value.

# Chapter 3

# Results

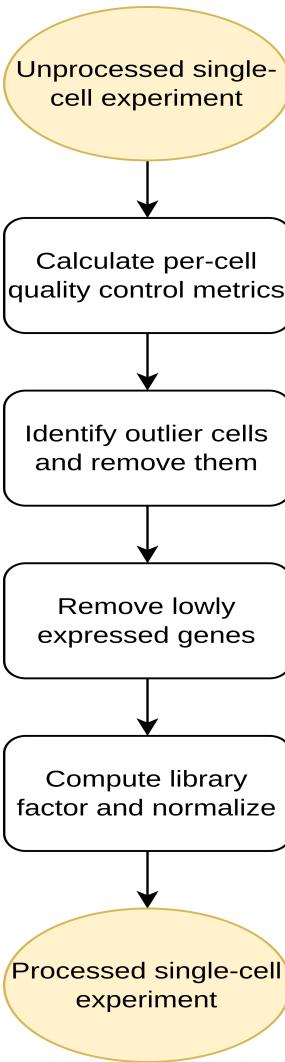
## 3.1 Exploratory Data Analysis

### 3.1.1 Mouse kidney cells

The first data set stems from a paper that investigates potential cellular targets of kidney disease in mice ([Park et al., 2018](#)). The authors isolated and sequenced a total of 57'979 cells from whole kidney cell suspensions (one kidney per mouse) derived from seven healthy male mice using droplet-based single-cell RNA sequencing. The samples were labelled as: normal1, normal2, normal3, normal4, Ksp-cre-GFP, Scl-cre-GFP and Pod-cre-GFP. For our work, we decided to use the raw data from the four samples that were labelled as normal to ensure the biological reproducibility between the samples.

We used the *alevin-fry* pipeline to quantify the raw single-cell RNA sequencing data for further use in the R programming environment. Quality control is a crucial stage in data pre-processing as low-quality libraries can contribute to misleading results in downstream analyses ([Amezquita et al., 2020](#)). Therefore, we filtered lowly abundant genes and low-quality cells to mitigate said problems to improve interpretability of the results.

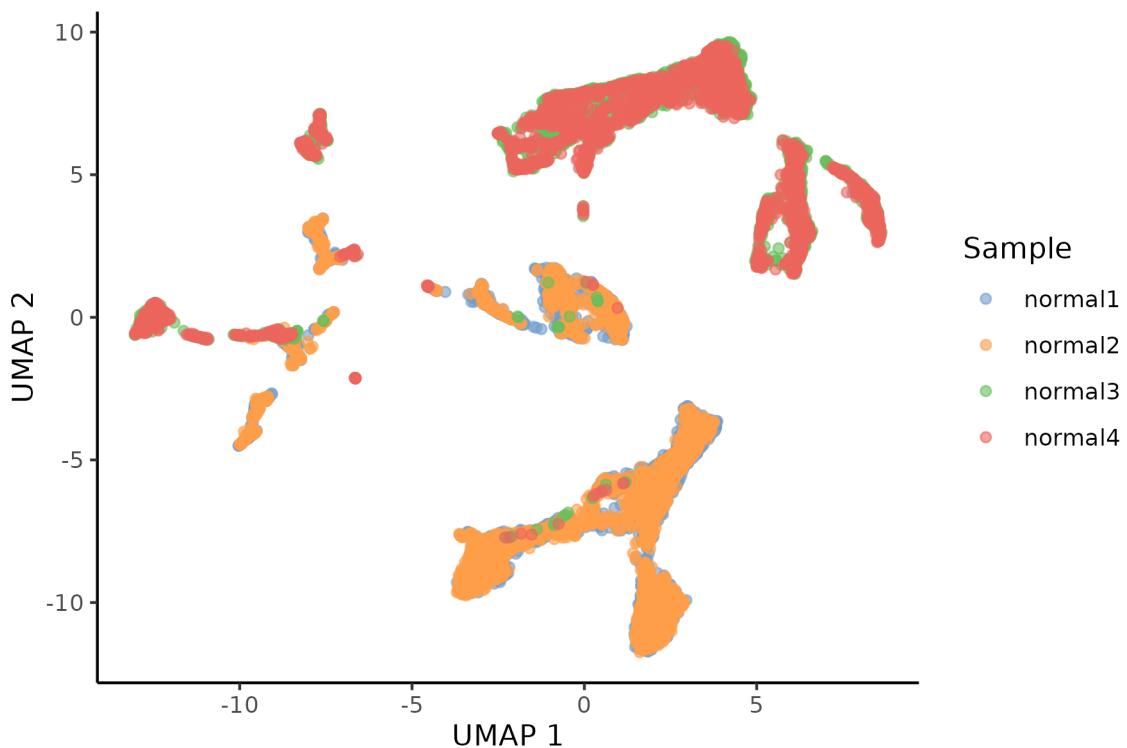
To identify low-quality cells, cell-specific QC metrics were calculated with the *perCellQC-Metrics* function from the *scater* R package ([McCarthy et al., 2017](#)). These metrics include the total number of expressed genes, the overall count across all genes, and the fraction of counts assigned to control genes such as mitochondrial genes. By setting a specific threshold on per-cell QC metrics, high-quality cells can be retained. In our setting, outliers are defined as cells with library sizes more than two median absolute deviations away from the median library size. Figure [3.1](#) summarizes the process from unprocessed to processed single-cell experiment.



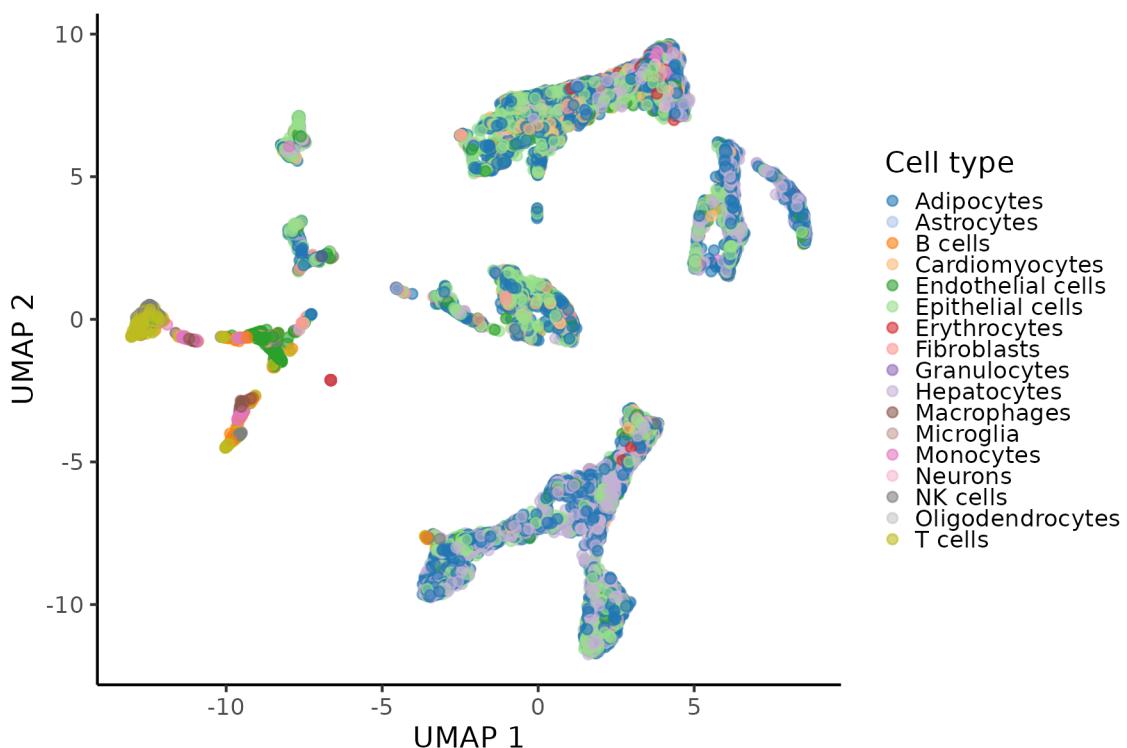
**Figure 3.1:** Quality control process from unprocessed, raw to processed, filtered single-cell experiment

After filtering, the data set consists of 23'543 cells and 18'537 genes. Next, we used the *singleR* function from the *singleR* R package (Aran *et al.*, 2019) for cell-type annotation. Cell-type annotation is important to determine what biological state is represented by cell clusters which helps the interpretability of the results and their implications (Amezquita *et al.*, 2020). *singleR* is a method that assigns labels based on the reference samples with the highest Spearman rank correlation while only using marker genes between pairs of labels to focus on the relevant differences between cell types (Aran *et al.*, 2019). Figure 3.2 shows the Uniform Manifold Approximation and Projection (UMAP) of the cells coloured by their respective sample id. From Figure 3.2 one can observe that the projection of the cells is very similar across the samples. Further, Figure 3.3 also illustrates that cells from the same cell-type cluster together, as one would expect.

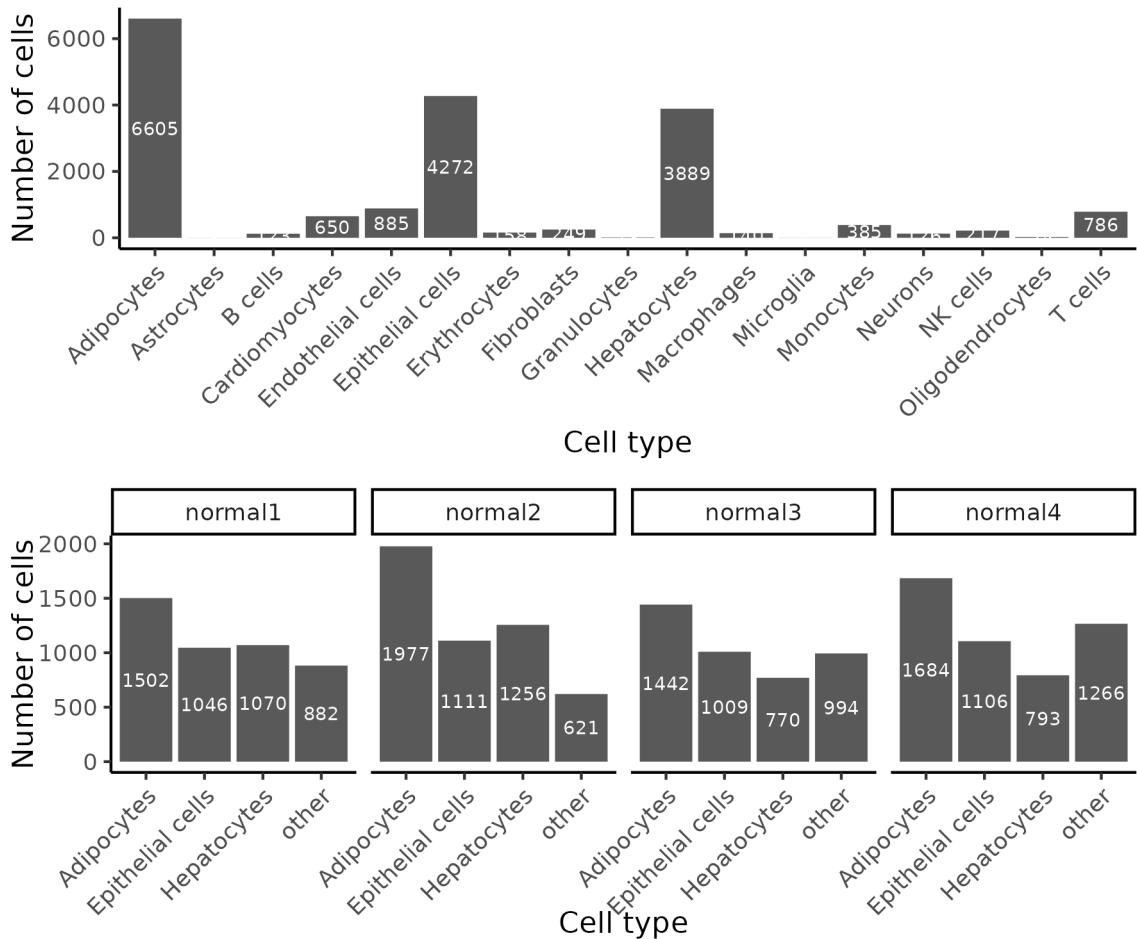
Figure 3.4 shows that the annotated cells were largely classified as either: Adipocytes, Epithelial cells and Hepatocytes. Additionally, one can observe that those three cell types are approximately evenly distributed across the four samples. For further analyses we focus on those cell types to investigate the performance of existing methods for the detection of differentially expressed genes and design our own simulation study.



**Figure 3.2:** UMAP representation of the mouse kidney cells coloured by sample id

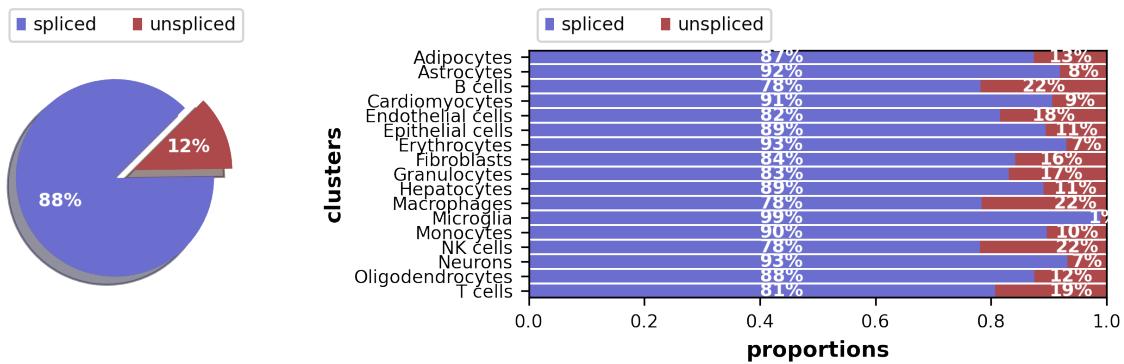
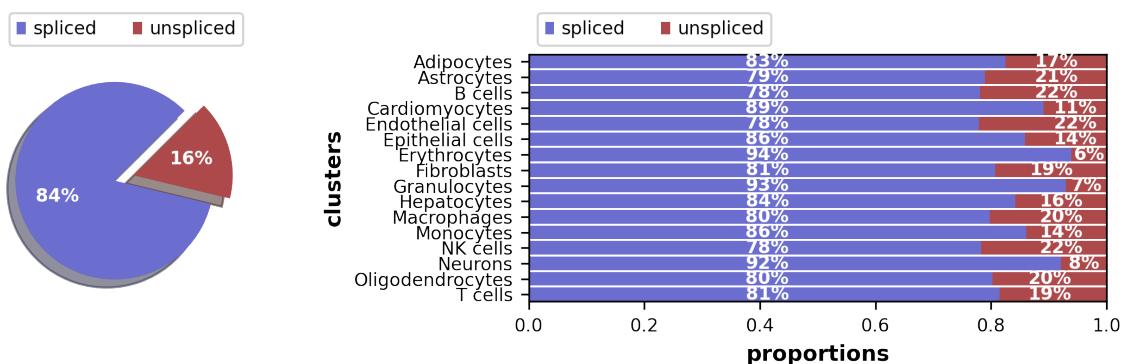
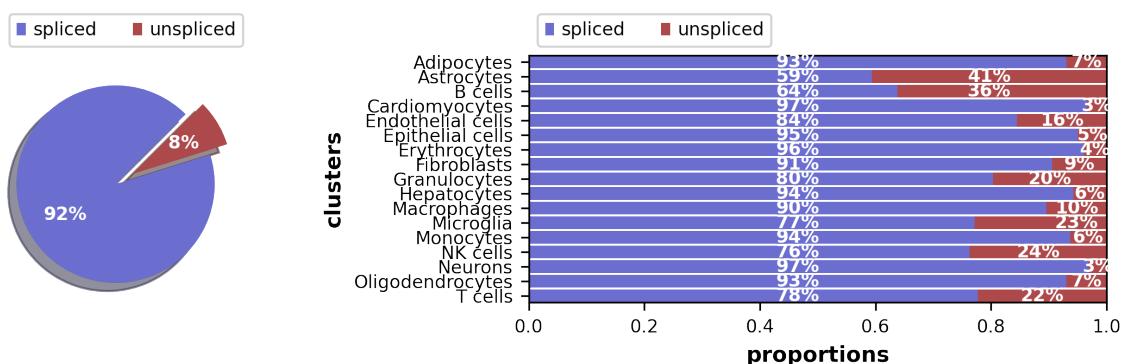


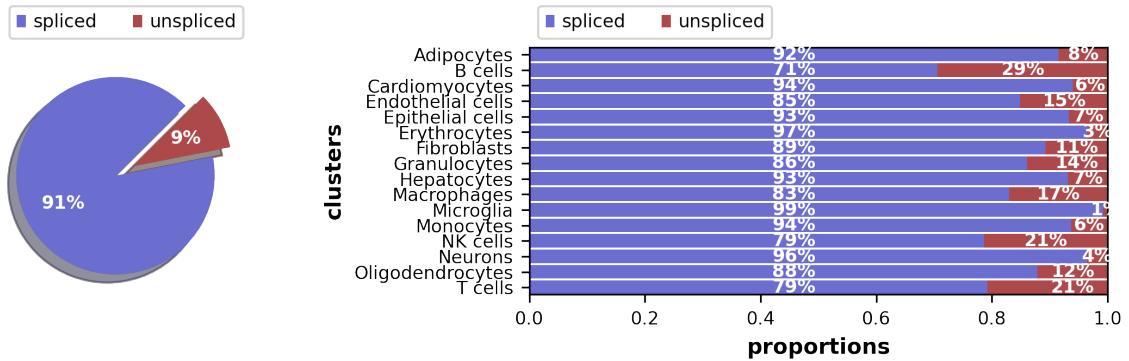
**Figure 3.3:** UMAP representation of the mouse kidney cells coloured by cell type



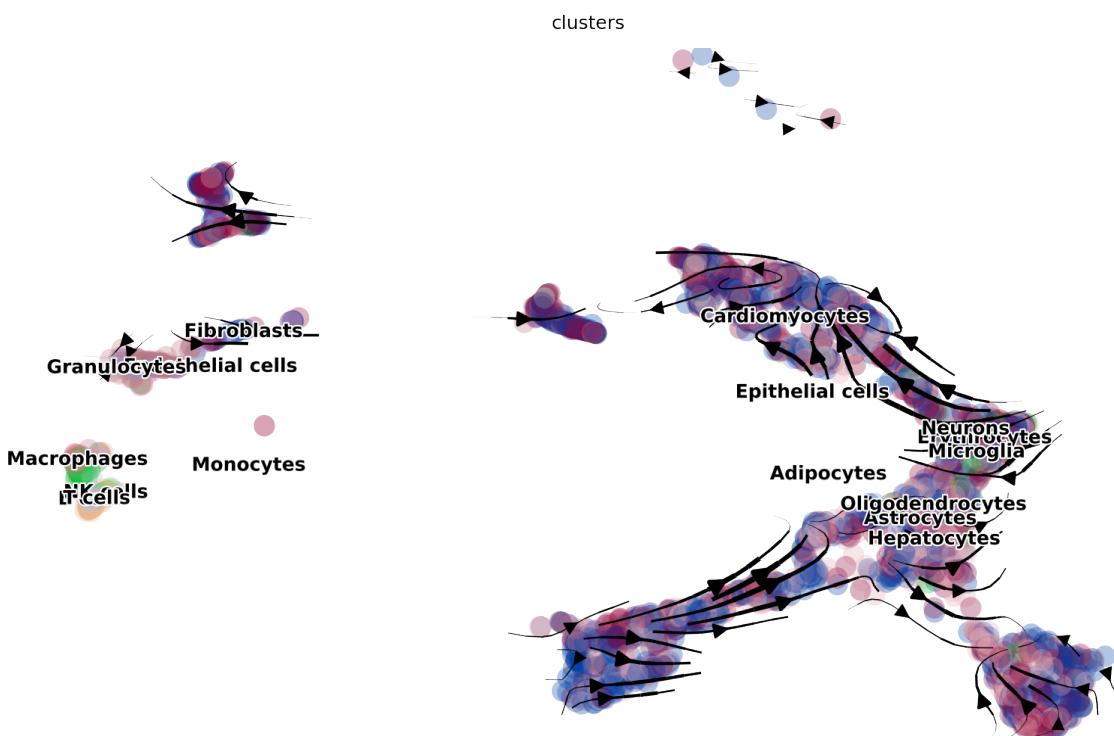
**Figure 3.4:** Frequency distribution of the cell types after quality control

After QC filtering and cell type annotation we investigated the RNA velocities for each sample. Initially, we expected similar patterns with changing trajectories. First, we explored the proportions of spliced and unspliced counts in each sample. Figures 3.5 to 3.8 show that the abundance of spliced counts is very high in comparison to unspliced counts. In samples 3 and 4 the abundance of unspliced counts approximately half, compared to samples 1 and 2. After exploring the velocity plots there were no clear patterns that were consistent between the biological replicates as shown in Figures 3.9 to 3.12. We initially thought that we could relate differentially regulated genes to differential velocity between experimental conditions. However, this does not seem possible. We therefore concluded that interpretation based solely on the RNA velocities is not possible and a different approach to tackle the objective is needed. Although, the two concepts are connected, we stick to differences between spliced and unspliced counts.

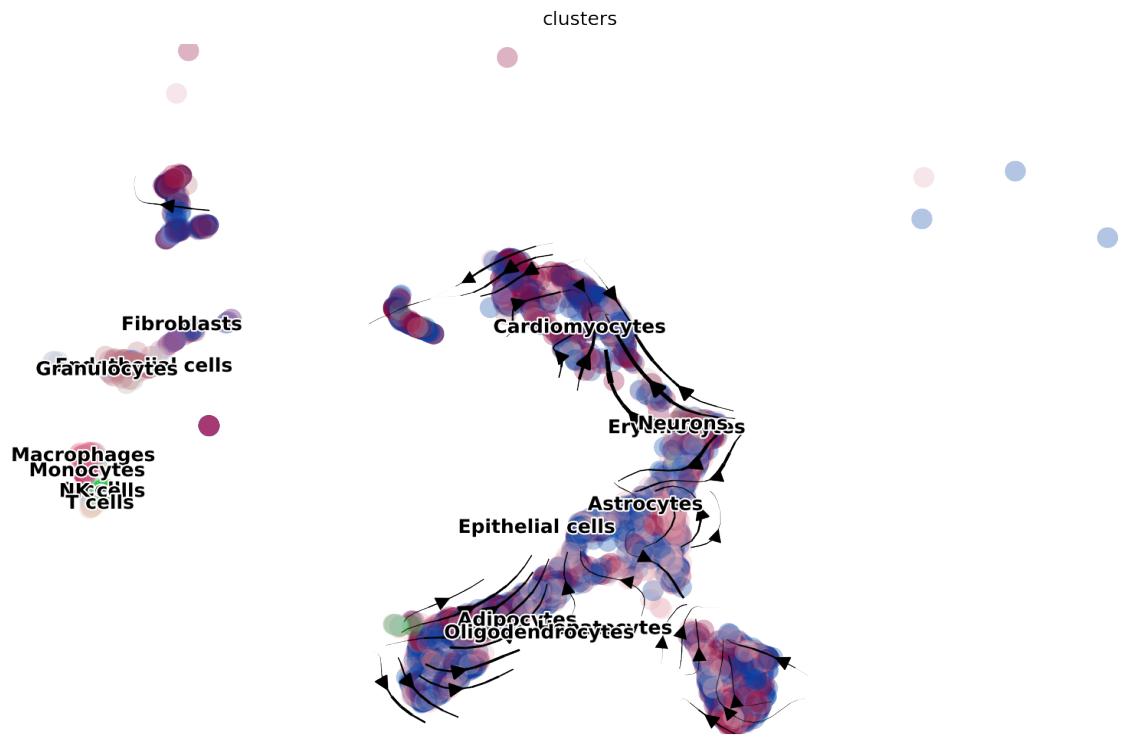
**Figure 3.5:** Abundance of spliced and unspliced counts in sample normal1**Figure 3.6:** Abundance of spliced and unspliced counts in sample normal2**Figure 3.7:** Abundance of spliced and unspliced counts in sample normal3



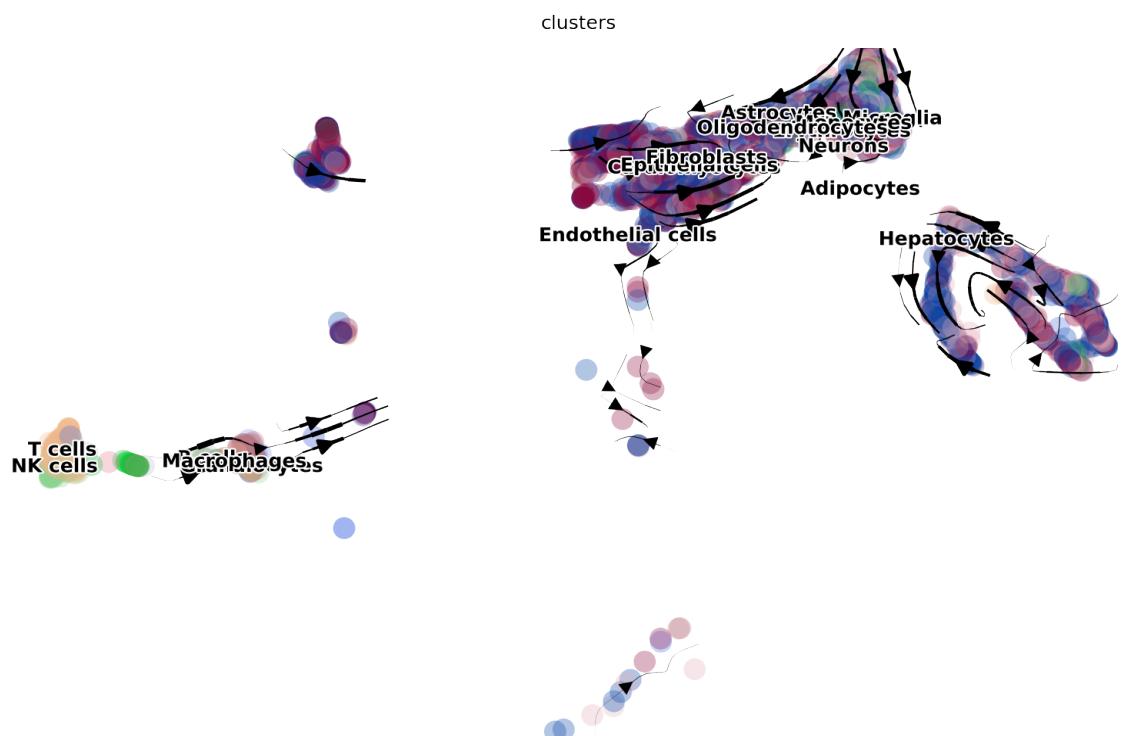
**Figure 3.8:** Abundance of spliced and unspliced counts in sample normal4



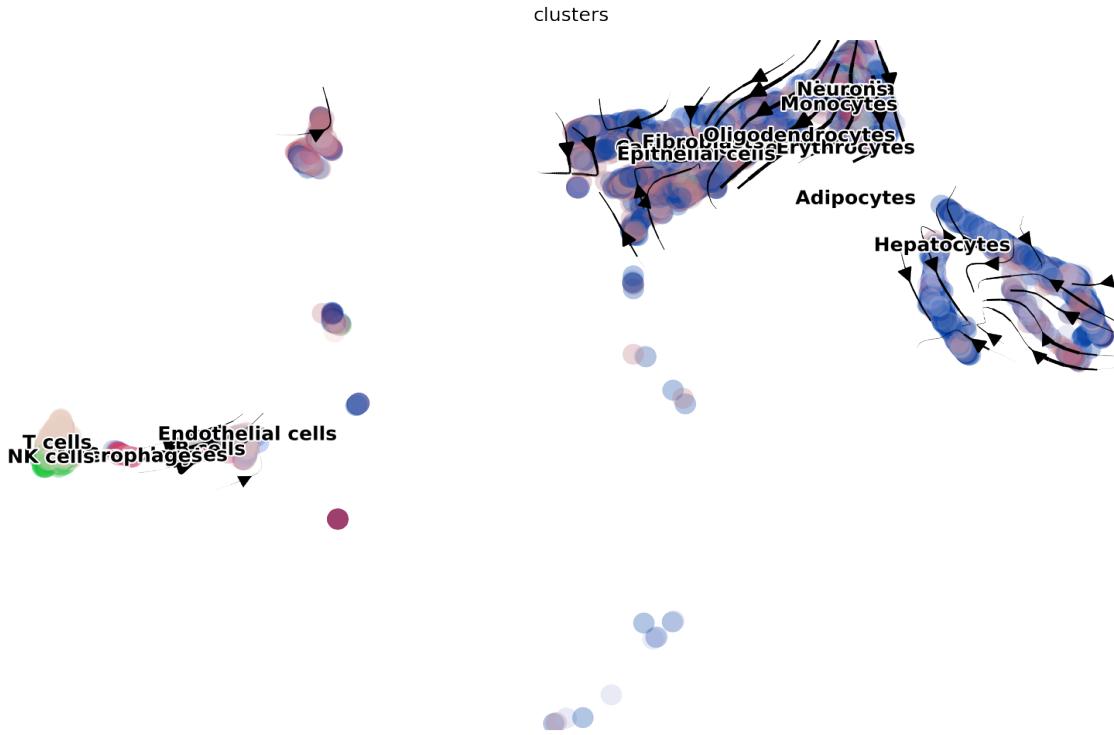
**Figure 3.9:** Dynamical model showing RNA velocity of sample normal1



**Figure 3.10:** Dynamical model showing RNA velocity of sample normal2



**Figure 3.11:** Dynamical model showing RNA velocity of sample normal3



**Figure 3.12:** Dynamical model showing RNA velocity of sample normal4

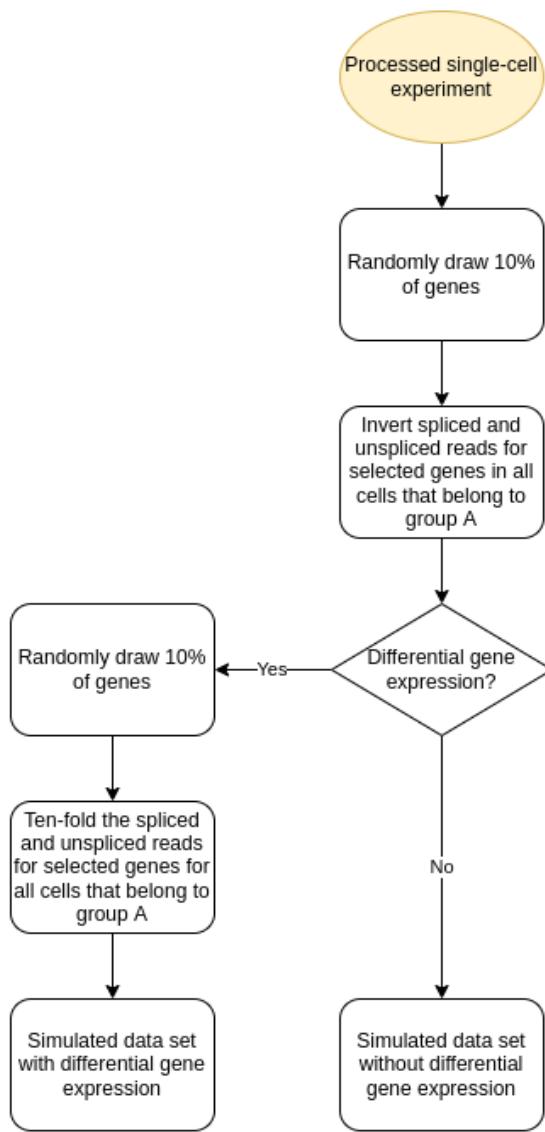
## 3.2 Simulation study

### 3.2.1 Simulation strategy

Initially, the simulation strategy was to invert the spliced and unspliced counts for 10% of genes for all cells that belong to an arbitrary group A. The group allocation was made based up on the consideration of the UMAP from figure 3.2 where it is visible that samples tend cluster in pairs: 1 with 2, and 3 with 4. Therefore, the group allocation of  $Group_A = (sample_1, sample_3)$  and  $Group_B = (sample_2, sample_4)$  was chosen to best reflect a homogeneous representation of the groups. The set of genes, whose counts were to be inverted, was randomly drawn by a sampling algorithm without replacement (hypergeometric distribution). There are many different ways to introduce a differential effect, however nailing down on inverting the spliced and unspliced counts seemed like a neat way to achieve this without actually modifying the originally estimated counts. This procedure was done separately for each cell type, so that there are differential genes across cell types. Additionally, differential gene expression was introduced in 10% of genes in all cells that belong to said arbitrary group A. Differential gene expression was added to introduce a nuisance parameter to the simulation we do not want to detect. In this way, we essentially ran two simulations: one with differential regulation only and one with differential regulation and differential gene expression additionally. In the differential gene expression simulation, we additionally multiply the counts by 10 (ten-fold gene expression) for 10% randomly drawn genes in group A. Again, the set of genes was randomly drawn by the same sampling algorithm as before.

Starting from a real data set as an anchor data set, then artificially introducing a differential effect, we essentially created two semi-simulated data sets. Compared to full simulations, semi-simulated approaches have the advantage of having a realistic structure, because it was indeed taken from real data. The genes and cells that were subject to change were stored as ground truth for further evaluation in the downstream analyses. Figure 3.13 illustrates the simulation

process from the original mouse kidney data set to two simulated data sets.



**Figure 3.13:** Simulation process from original mouse data set to simulated data sets

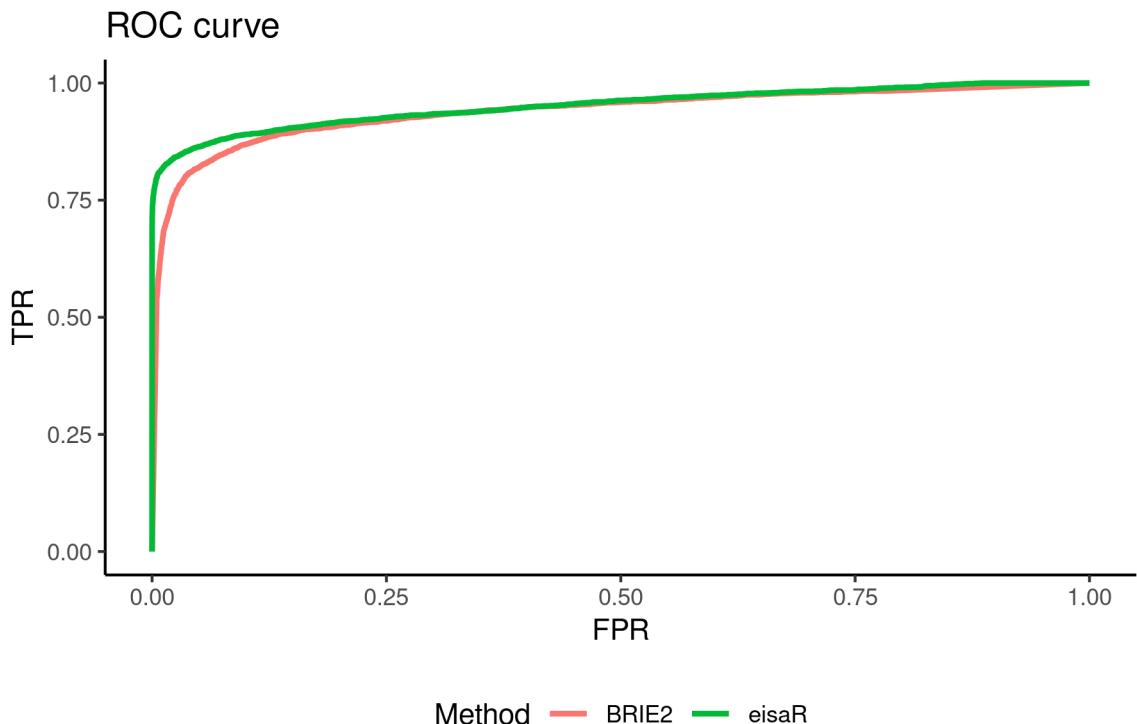
The goal of this thesis was to determine how well the aforementioned methods perform on detecting differentially regulated genes on read-level simulated data sets. To achieve this two groups of methods were postulated as *eisaR* and *BRIE2* cannot take into account ambiguous reads. For this reason the classification performance of *eisaR* and *BRIE2* are compared with each other with the help of ROC and TPR v. FDR curves. As *alevin-fry* allows the estimation of ambiguous counts as well as spliced and unspliced, it was possible to assign the ambiguous counts to both spliced and unspliced counts (50-50 split) so that *eisaR* and *BRIE2* can be applied. In a similar manner, *DEXSeq* and our own method *DifferentialRegulation* were used to detect differentially regulated genes. However, in this case the ambiguous counts were used as an additional information rather than discarding it. As a next step, *minnow* was used to introduce mapping uncertainty into the simulated data sets. After alignment and quantification of the new data sets the same protocol was used to assess the performance of all four methods.

### 3.2.2 Simulation without mapping uncertainty

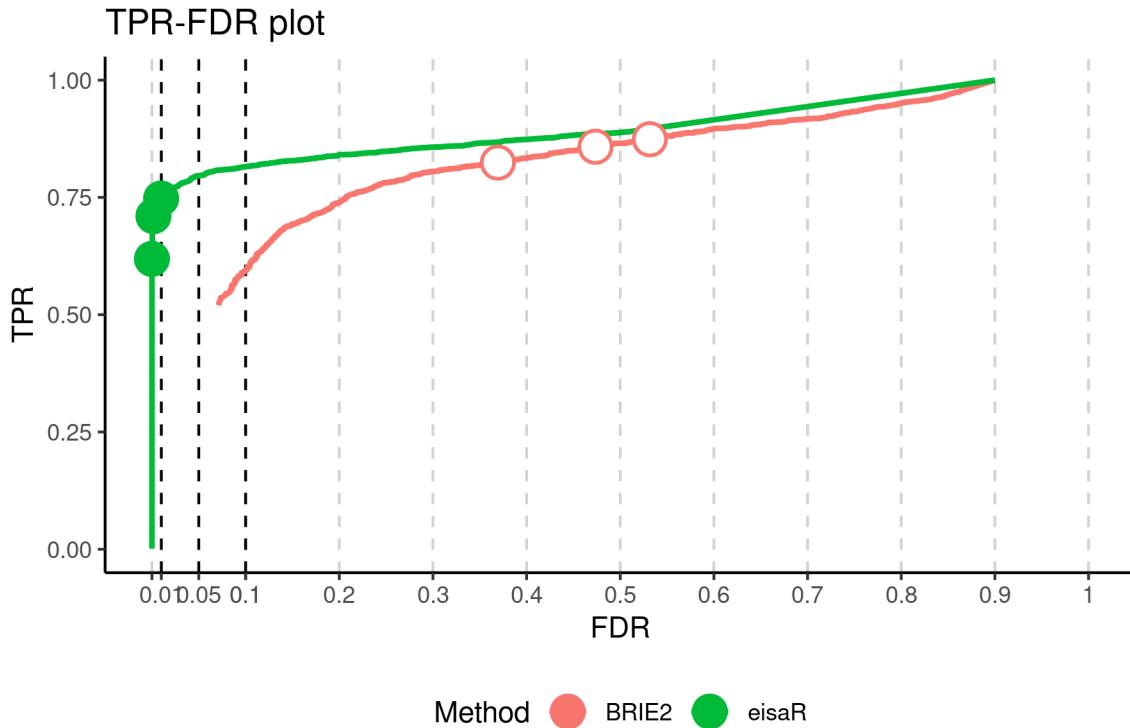
As a first step, we looked at the results from the naive simulation for both data sets with and without differential gene expression. Figures 3.14 and 3.15 show that both *eisaR* and *BRIE2* perform quite well in detecting differential genes. *eisaR* has a slightly higher TPR as shown in the ROC curve. Although, both methods have a similar performance pattern in the ROC curve, the TPR v. FDR plot looks quite different between the two methods. From Figures 3.16 and 3.17 it is shown that *eisaR* is well calibrated for FDR, whereas *BRIE2* is not.

After introducing differential gene expression in the second data set performance drops quite substantially as shown in Figure 3.16. TPR is almost halved in both methods as it is shown in the ROC curve of Figure 3.16. However, *eisaR* is still well calibrated for FDR. Similar to the data set without differential gene expression, *BRIE2* is not well calibrated for FDR. It seems that both methods are heavily affected by the introduction of differential gene expression as performance dropped to almost half. However, we introduced a very strong effect - 10 fold change - therefore, it is to be expected that performance drops. Nevertheless, we wanted to investigate how the performance changes in an extreme scenario, hence the strong effect size.

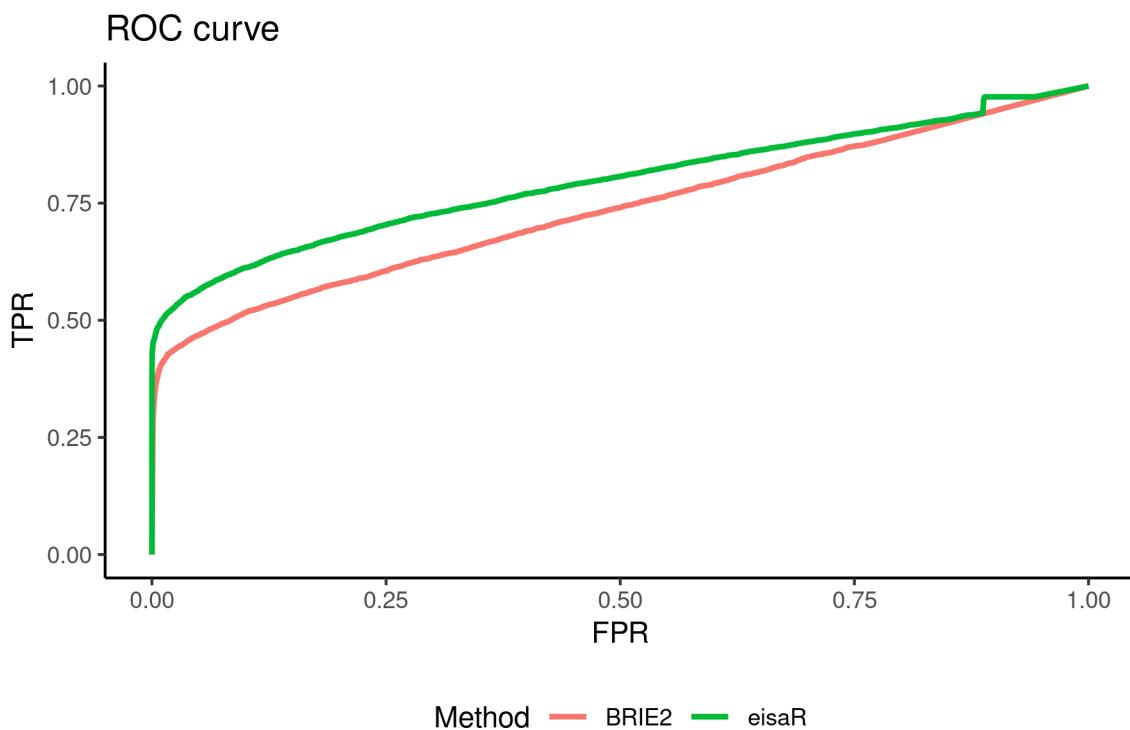
From this naive simulation we concluded that *BRIE2* has inflated FDR in both cases - with and without differential gene expression. On the other hand, *eisaR* was well calibrated for FDR in both cases. However, TPR decreases by almost half after introducing differential gene expression. In the next step, we use *minnow* to introduce multi-mapping uncertainty in to both data sets to make the simulation more realistic. It is to be expected that performance will further decrease after the introduction of multi-mapping uncertainty.



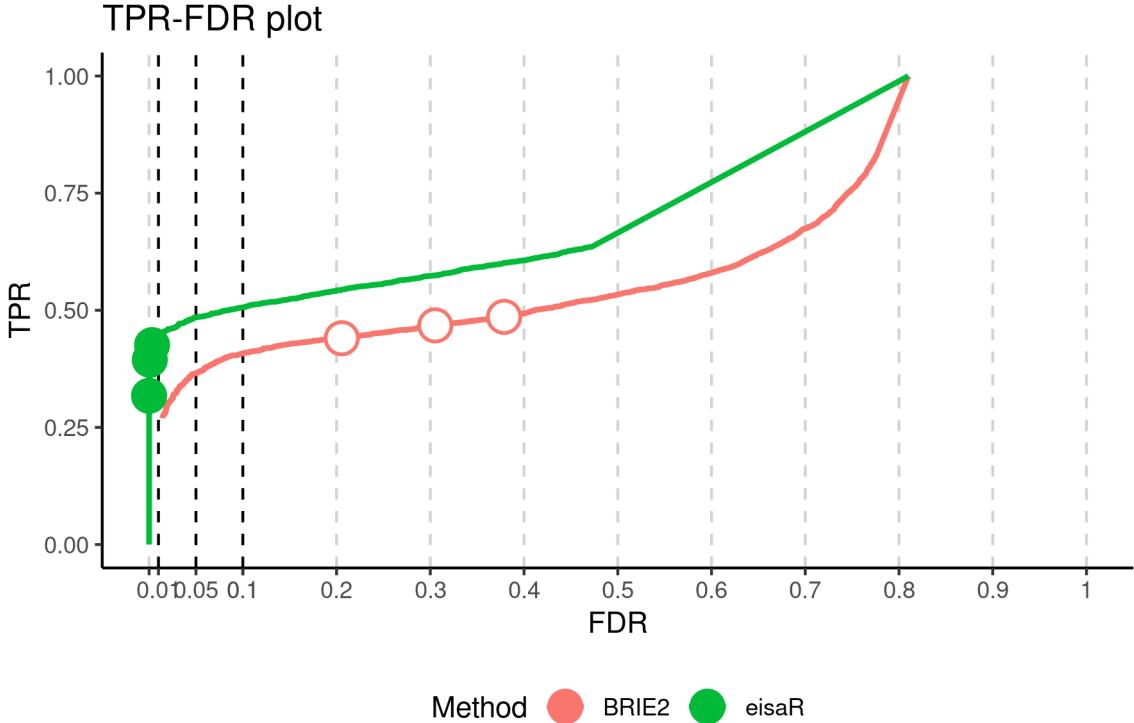
**Figure 3.14:** ROC curve of *BRIE2* and *eisaR* in the initial simulation without differential gene expression



**Figure 3.15:** TPR v. FDR plot of *BRIE2* and *eisaR* in the initial simulation without differential gene expression



**Figure 3.16:** ROC curve of *BRIE2* and *eisaR* in the initial simulation with differential gene expression



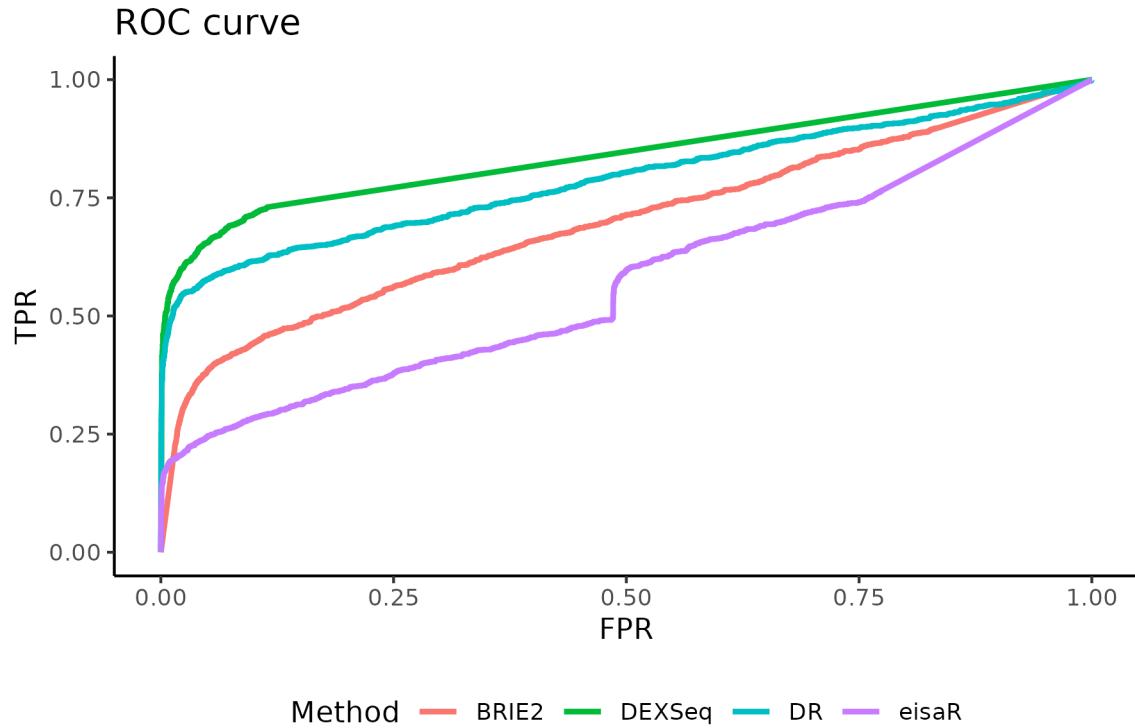
**Figure 3.17:** TPR v. FDR plot of *BRIE2* and *eisaR* in the initial simulation with differential gene expression

### 3.2.3 Simulation with mapping uncertainty

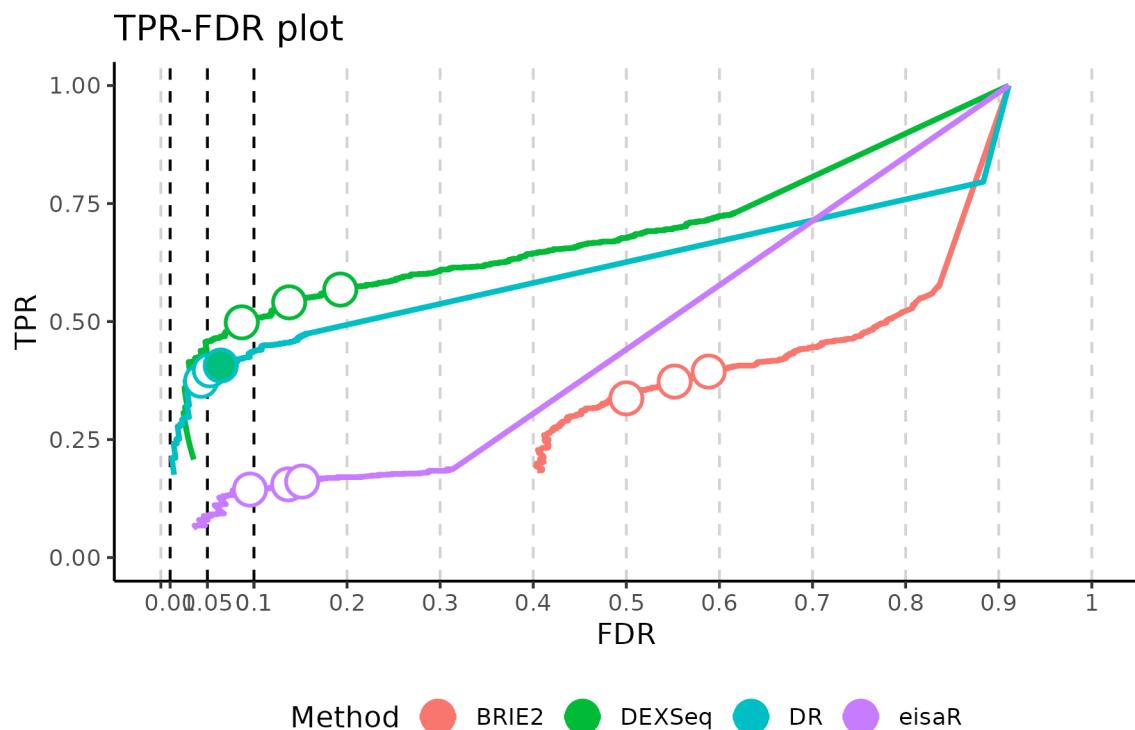
As a next step, multi-mapping uncertainty was introduced in to the two simulated data sets. First, *minnow* was used to simulate new reads from the previously simulated data sets. Second, we used *alevin-fry* to align the generated read files to the reference genome and for quantification of the reads. The newly generated data sets were then run on the aforementioned methods to identify differential genes.

Figure 3.18 shows the performance of all four methods without differential gene expression. From the ROC curve we observe that *DEXSeq* and *DifferentialRegulation* have a similar TPR, although *DEXSeq* perform slightly better. Both *BRIE2* and *eisaR* have a lower TPR than the other two methods. From Figure 3.19 we observe that again *DEXSeq* and *DifferentialRegulation* have a similar performance profile. However, *DifferentialRegulation* controls better for FDR than *DEXSeq*. Contrarily to before, *eisaR* controls quite well for FDR, however has half the TPR compared *DEXSeq* and *DifferentialRegulation*. On the other hand, *BRIE2* does not control well for FDR as there is strong inflation.

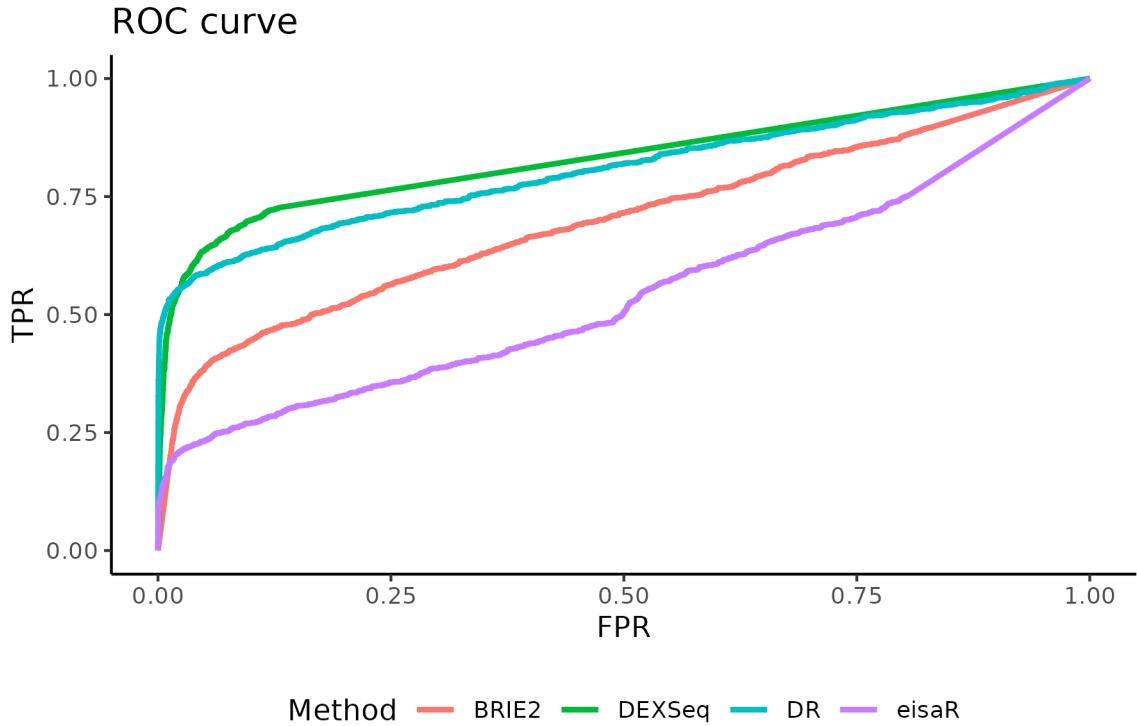
In the next step, we investigated how the performance changes after introduction of differential gene expression into the data set. Figure 3.20 shows the ROC curve for all four methods. From the figure we observe that the difference in TPR between *DEXSeq* and *DifferentialRegulation* is smaller than before. On the other hand, the performance based on TPR is approximately the same as before for *BRIE2* and *eisaR*. From Figure 3.21 it is shown that *DifferentialRegulation* still controls well for FDR, whereas the other three methods do not. The introduction of differential gene expression as a nuisance parameter seems to affect the performance of *DEXSeq* and *eisaR* quite substantially in terms of FDR calibration as FDR is almost doubled to before. The FDR calibration for *BRIE2* was not affected by differential gene expression, although it was not well calibrated to begin with.



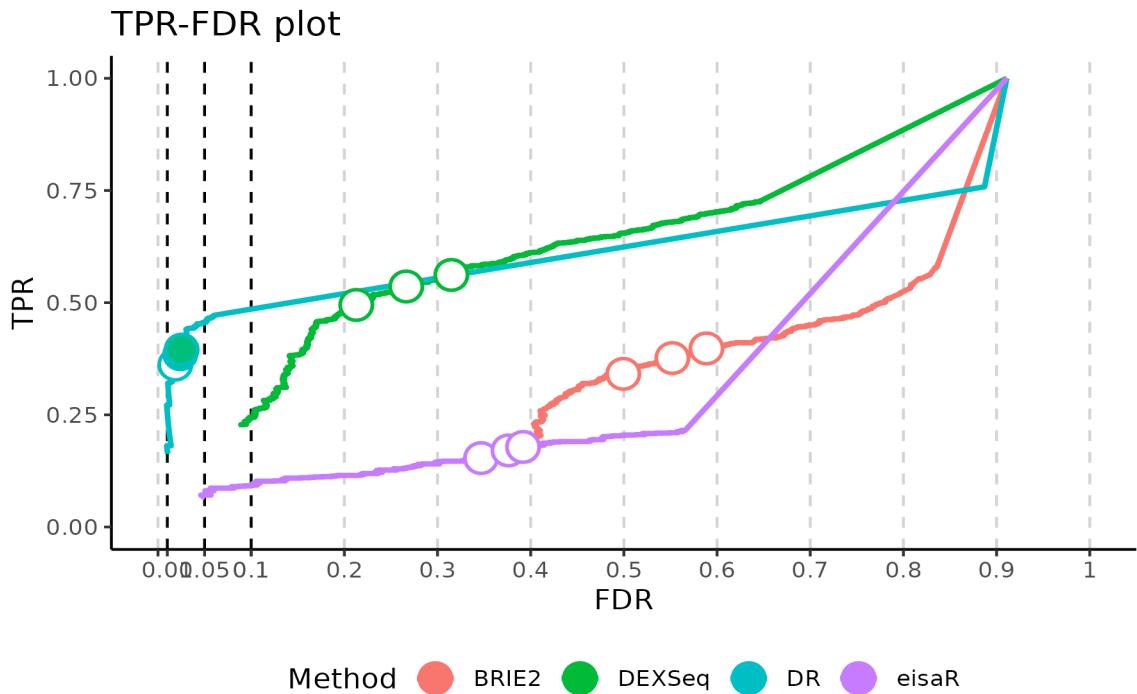
**Figure 3.18:** ROC curve of *BRIE2*, *DEXSeq*, *DifferentialRegulation* and *eisaR* in the sophisticated simulation without differential gene expression



**Figure 3.19:** TPR v. FDR plot of *BRIE2*, *DEXSeq*, *DifferentialRegulation* and *eisaR* in the sophisticated simulation without differential gene expression



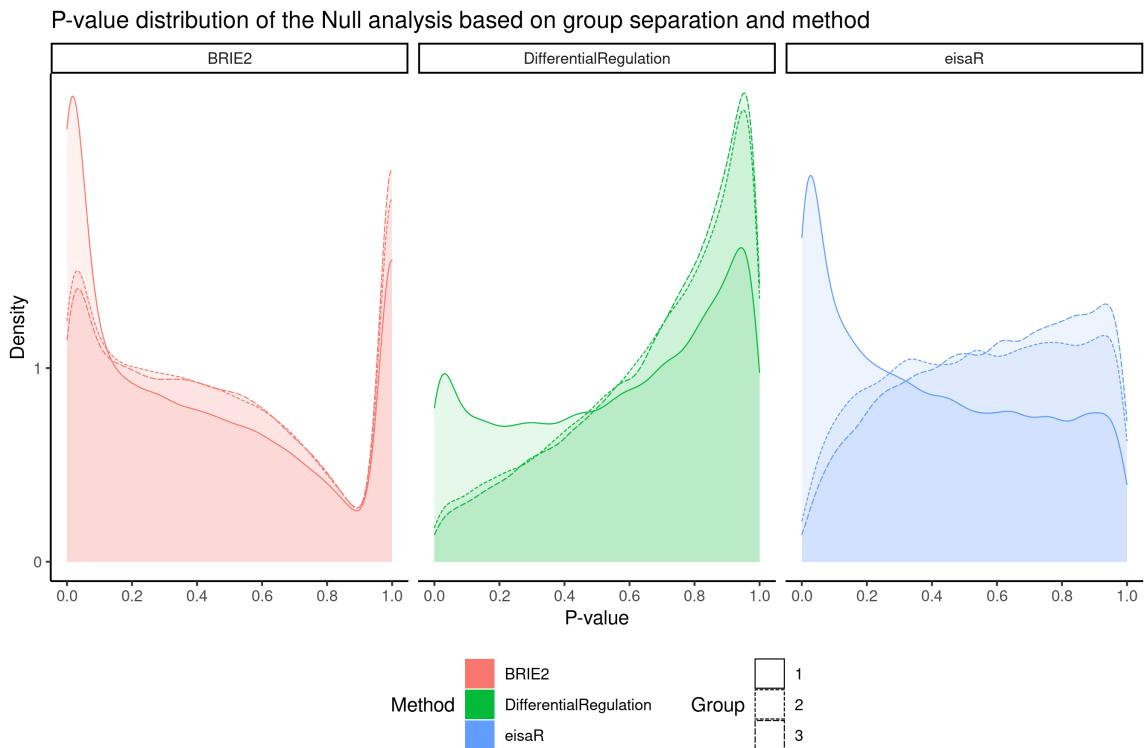
**Figure 3.20:** ROC curve of *BRIE2*, *DEXSeq*, *DifferentialRegulation* and *eisaR* in the sophisticated simulation with differential gene expression



**Figure 3.21:** TPR v. FDR plot of *BRIE2*, *DEXSeq*, *eisaR* and *DifferentialRegulation* in the sophisticated simulation with differential gene expression

### 3.3 Null analysis on the mouse kidney data

As a last step, a Null analysis was conducted on the original mouse kidney data set to evaluate the methods on a real world data set. For that, all three possible group allocations were considered. Under the Null hypothesis  $H_0$  the p-values would be uniformly distributed between zero and one, because all samples belong to the same experiment condition (i.e. normal). In particular, we were interested in checking for false positive detections which are indicated by inflated p-values towards zero. Figure 3.22 shows that the p-values are slightly inflated for the first group separation, which leads to a marginal separation between groups, as visible from the UMAP 3.2. Further, from Figure 3.22 it is shown that for *DifferentialRegulation* FPs are never inflated, hence our method is a bit conservative compared to the other two methods. *eisaR* is only inflated for the first group separation, but overall approximately uniformly distributed. On the other hand, *BRIE2* demonstrates inflated FPs for all three group separations, which is also consistent with the results from the simulation study.



**Figure 3.22:** P-value distribution of the methods based on the three possible group separations

Further, we looked into the performance of the models based on the proportion of p-values and FDR values smaller or equal than the conventional significance levels of 10%, 5% and 1%. From Table 3.1 it is shown that *BRIE2* does not control the p-values particularly well as the values are inflated for all three levels. On the other hand, both *DifferentialRegulation* and *eisaR* do not have inflated p-values, whereas *DifferentialRegulation* is a bit more conservative than *eisaR*. Table 3.2 paints a similar picture as *BRIE2* produces by far larger values than the other methods including *DEXSeq*. With FDR values one would usually want the values to be as small as possible. From that convention one can observe that *DifferentialRegulation* controls the FDR better than both *DEXSeq* and *eisaR*, whereas the latter two methods perform approximately equally as good.

**Table 3.1:** Proportion of p-values smaller than the proposed significance levels 10%, 5% and 1%

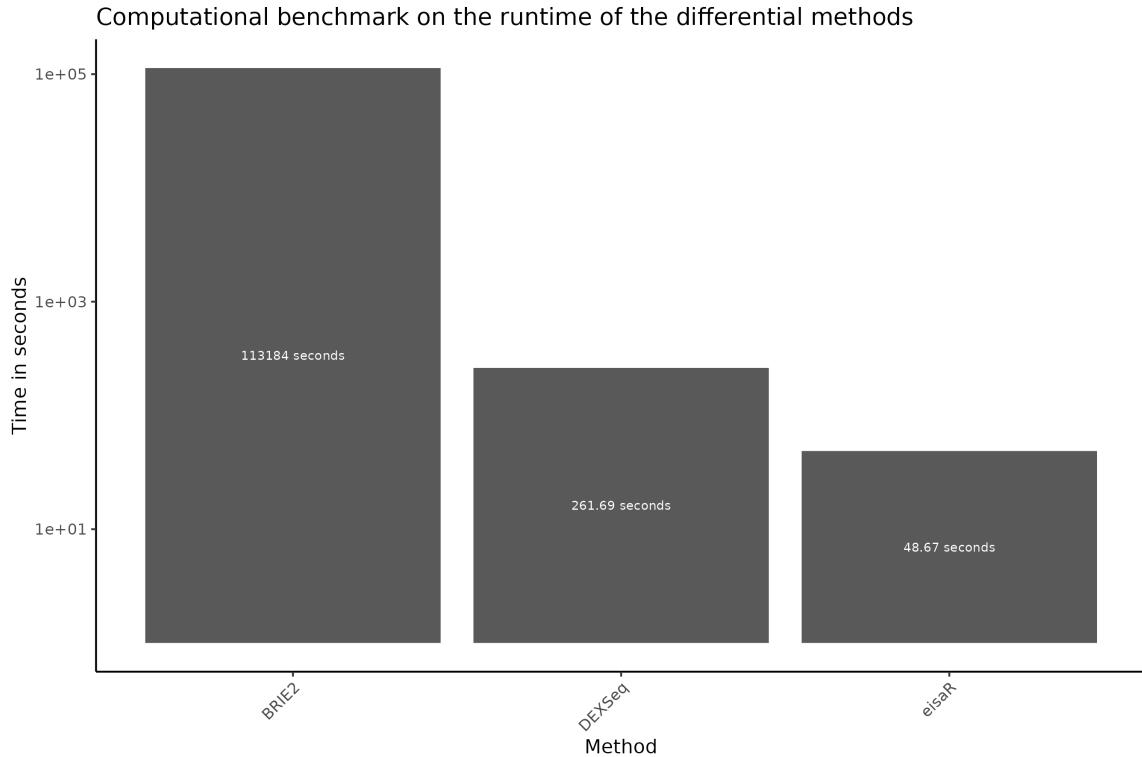
	10%	5%	1%
BRIE2	0.215	0.155	0.083
DifferentialRegulation	0.059	0.035	0.014
eisaR	0.108	0.065	0.026

**Table 3.2:** Proportion of FDR values smaller than the proposed significance levels 10%, 5% and 1%

	10%	5%	1%
BRIE2	0.080	0.061	0.038
DEXSeq	0.016	0.012	0.006
DifferentialRegulation	0.006	0.005	0.003
eisaR	0.017	0.010	0.004

### 3.4 Computational benchmark

We compared the computational burden of differential methods: *eisaR*, *BRIE2*, *DEXSEQ* and *DifferentialRegulation*, excluding alignment and quantification. All methods were run on 3 cores on the same machine (internal server). We ran the computational benchmark on the Null analysis for all group separations and averaged the runtime. Figure 3.23 shows the runtime of each method in seconds on a log-scaled axis as the numbers differ quite substantially between the methods.



**Figure 3.23:** Computational benchmark of the differential methods on the Null data averaged over the three possible group separations

## 3.5 Data availability

### Kidney mouse cells

The raw data can be downloaded from NCBI GEO (accession number GSE107585).

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE107585>

## 3.6 Code availability

All code for data preprocessing and analysis associated with the thesis is available at <https://github.com/joelmeili/DifferentialRegulation>. Any updates will also be published on GitHub.



# Chapter 4

## Discussion

### 4.1 Conclusion

In this thesis we investigated the performance of four differential gene detection methods on two semi-simulated data sets. The semi-simulated data sets were created by the use of real single-cell RNA sequencing data with four biological replicates. After alignment and quantification we applied quality control measures to the data such that the data is free of lowly expressed genes and outlier cells. In a first step, we introduced an arbitrary differential effect to a subset of genes and cells by inverting the counts of spliced and unspliced reads. Further, we added differential gene expression to a second subset of genes and cells as an additional nuisance parameter which we did not want detect in the differential gene analysis. Multi-mapping uncertainty was considered next to make the semi-simulated data sets more realistic as multi-mapping uncertainty is an important characteristic of single-cell RNA sequencing data. We achieved this by using the *minnow* pipeline where we used the two semi-simulated data sets as input. Again we used *alevin-fry* for alignment and quantification of the reads - such that we have estimated spliced, unspliced and ambiguous count matrices. We then analysed the performance of *BRIE2*, *eisaR*, *DEXSeq* and *DifferentialRegulation* in detecting the actual differential genes by comparing ROC and TPR v. FDR curves. From this analysis we found that *DifferentialRegulation* controls the FDR well for both the data sets with and without differential gene expression in comparison to the three other methods. As a last step, we did a Null analysis on the original data set where we compared the distribution of p-values for all three possible group separations. We found that *BRIE2* had inflated p-values for all three group separations, whereas *DifferentialRegulation* had no inflated p-values at all.

### 4.2 Outlook

**Bioconductor package** *DifferentialRegulation* ([Tiberi, 2022](#)) is a method for detecting differentially regulated genes between two groups of samples (e.g., healthy v. disease, or treated vs. untreated samples), by targeting differences in the balance of spliced and unspliced mRNA abundances, obtained from single-cell RNA-sequencing (scRNA-seq) data. *DifferentialRegulation* accounts for the sample-to-sample variability, and embeds multiple samples in a Bayesian hierarchical model. In particular, when reads are compatible with multiple genes or multiple splicing versions of a gene (unspliced spliced or ambiguous), the method allocates these multi-mapping reads to the gene of origin and their splicing version. Parameters are inferred via Markov chain Monte Carlo (MCMC) techniques (Metropolis-within-Gibbs).



# Bibliography

- Amezquita, R. A., Lun, A. T., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., Marini, F., Rue-Albrecht, K., Risso, D., Soneson, C., et al. (2020). Orchestrating single-cell analysis with bioconductor. *Nature methods*, **17**, 137–145. [13](#), [14](#)
- Anders, S., Reyes, A., and Huber, W. (2012). Detecting differential usage of exons from rna-seq data. [10](#)
- Aran, D., Looney, A. P., Liu, L., Wu, E., Fong, V., Hsu, A., Chak, S., Naikawadi, R. P., Wolters, P. J., Abate, A. R., Butte, A. J., and Bhattacharya, M. (2019). Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.*, **20**, 163–172. [14](#)
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289–300. [12](#)
- Crowell, H. L., Soneson, C., Germain, P.-L., Calini, D., Collin, L., Raposo, C., Malhotra, D., and Robinson, M. D. (2020). Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature communications*, **11**, 1–12. [6](#)
- Dharshini, S. A. P., Taguchi, Y.-H., and Gromiha, M. M. (2020). Identifying suitable tools for variant detection and differential gene expression using rna-seq data. *Genomics*, **112**, 2166–2172. [6](#)
- Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). Star: ultrafast universal rna-seq aligner. *Bioinformatics*, **29**, 15–21. [5](#)
- Gaidatzis, D., Burger, L., Florescu, M., and Stadler, M. B. (2015). Analysis of intronic and exonic reads in rna-seq data characterizes transcriptional and post-transcriptional regulation. *Nature biotechnology*, **33**, 722–729. [8](#)
- Haque, A., Engel, J., Teichmann, S. A., and Lönnberg, T. (2017). A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome medicine*, **9**, 1–12. [3](#), [4](#)
- He, D., Zakeri, M., Sarkar, H., Soneson, C., Srivastava, A., and Patro, R. (2022). Alevin-fry unlocks rapid, accurate and memory-frugal quantification of single-cell rna-seq data. *Nature Methods*, **19**, 316–322. [5](#), [7](#)
- Huang, Y. and Sanguinetti, G. (2021). Brie2: computational identification of splicing phenotypes from single-cell transcriptomic experiments. *Genome biology*, **22**, 1–15. [6](#), [8](#), [9](#)

- La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastriti, M. E., Lönnberg, P., Furlan, A., *et al.* (2018). Rna velocity of single cells. *Nature*, **560**, 494–498. [5](#)
- Love, M. I., Soneson, C., and Patro, R. (2018). Swimming downstream: statistical analysis of differential transcript usage following salmon quantification. *F1000Research*, **7**, . [10](#)
- McCarthy, D. J., Campbell, K. R., Lun, A. T. L., and Willis, Q. F. (2017). Scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R. *Bioinformatics*, **33**, 1179–1186. [13](#)
- McDermaid, A., Chen, X., Zhang, Y., Wang, C., Gu, S., Xie, J., and Ma, Q. (2018). A new machine learning-based framework for mapping uncertainty analysis in rna-seq read alignment and gene expression estimation. *Frontiers in genetics*, **9**, 313. [6](#)
- Melsted, P., Booeshaghi, A., Liu, L., Gao, F., Lu, L., Min, K. H. J., da Veiga Beltrame, E., Hjörleifsson, K. E., Gehring, J., and Pachter, L. (2021). Modular, efficient and constant-memory single-cell rna-seq preprocessing. *Nature biotechnology*, **39**, 813–818. [5](#)
- Mohajon, J. (2020). Confusion matrix for your multi-class machine learning model. [12](#)
- Park, J., Shrestha, R., Qiu, C., Kondo, A., Huang, S., Werth, M., Li, M., Barasch, J., and Suszták, K. (2018). Single-cell transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease. *Science*, **360**, 758–763. [13](#)
- Rao, M. S., Van Vleet, T. R., Ciurliionis, R., Buck, W. R., Mittelstadt, S. W., Blomme, E. A., and Liguori, M. J. (2019). Comparison of rna-seq and microarray gene expression platforms for the toxicogenomic evaluation of liver from short-term rat toxicity studies. *Frontiers in genetics*, **9**, 636. [3](#)
- Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140. [6, 8](#)
- Sarkar, H., Srivastava, A., and Patro, R. (2019). Minnow: a principled framework for rapid simulation of dscrna-seq data at the read level. *Bioinformatics*, **35**, i136–i144. [7, 8](#)
- Srivastava, A., Malik, L., Smith, T., Sudbery, I., and Patro, R. (2019). Alevin efficiently estimates accurate gene abundances from dscrna-seq data. *Genome biology*, **20**, 1–16. [5](#)
- Stadler, M. B., Gaidatzis, D., Burger, L., and Soneson, C. (2020). eisar: Exon-intron split anaalysis (eisa) in r. R package version 1.0. [6](#)
- Stark, R., Grzelak, M., and Hadfield, J. (2019). Rna sequencing: the teenage years. *Nature Reviews Genetics*, **20**, 631–656. [3](#)
- Tiberi, S. (2022). *DifferentialRegulation: Differentially regulated genes from scRNA-seq data*. R package version 1.0.7. [31](#)
- Tiberi, S., Crowell, H. L., Weber, L. M., Samartsidis, P., and Robinson, M. D. (2021). distinct: a novel approach to differential distribution analyses. [6](#)
- Weiler, P., Van den Berge, K., Street, K., and Tiberi, S. (2021). A guide to trajectory inference and rna velocity. [5](#)