POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Technical variability versus biological heterogeneity in single-cell RNA-sequencing data

Politecnico di Torino

Relatori
Prof. Enrico Bibbona
Prof. Gianluca Mastrantonio

Candidato
Giulia Della Croce di Dojola

Anno Accademico 2020-2021

Summary

In the last few decades, studies on gene expression have increasingly grown. The development of new powerful techniques has more and more allowed to monitor such expression levels by quantifying, for each gene, the corresponding abundance of mRNA fragments within a cell. These quantities act as "signatures" that enable to deeply understand the molecular behaviour of many biological processes and can provide new biomarkers for specific conditions. This is the environment in which the present work is set. Indeed, it aims at identifying genuine heterogeneity of gene expression in a seemingly homogeneous population of cells, by simultaneously taking into account the technical variation introduced with the experiments.
In particular, in the first chapter of this work, we provide a biological introduction that outlines the fundamental mechanisms the cell undertakes to regulate gene expression. In the second chapter, some well-known techniques for sequencing the mRNA inside the cells are described. Starting from the less recent ones, we show the improvements that have been gradually reached and then present a modern powerful technique called Single-cell RNA-sequencing, which allows to profile the mRNA transcripts directly from single cells. In the third chapter, we introduce a Bayesian approach proposed in literature, that aims at identifying a potential set of highly variable genes (HVG) from a homogeneous population of cells. Here, we also present the MCMC algorithm that we have implemented. Finally, the results are depicted in the final chapter. First, a comparison between our results and the ones found in literature is made, finding them in accordance. Then, after remarking an identifiability issue regarding the so-called capture efficiency parameters, a modified version of the model is proposed. Conclusions on the correct classification of highly variable genes are then investigated.

Acknowledgments

Thanks to everyone that has been there for me.
To my dad, for his constancy and stability, and for helping me finding my own way.
To my mum, for her affection and for bringing lightness and joy to every situation.
To my sister, for the things we have always shared, and for her support and optimism.
To my grandpas, that knew the person I was.
To my grandmas, that support the person I am.
To my uncles Dedo, Francesco and Nicola that are important examples of life, each one in his own way.
And finally to my friends, to the ones that have been there, to the ones that still are, and to the ones that will come.

List of Figures

1.1 DNA packaging, from [20] . . . 8
1.2 Structure of a gene, from [15] . . . 9
1.3 Transcription elongation, from [2] . . . 11
1.4 The triplet code, from [2] . . . 12

2.1 Single-cell versus bulk analysis, from [26] . . . 17

3.1 Graphical representation of the model, by [13] . . . 24

4.1 Traceplot of parameters θ and δ_1 . . . 34
4.2 Traceplot of parameters s_1 and μ_1 . . . 34
4.3 Traceplot of parameter ν_1 . . . 34
4.4 Traceplot of parameter ϕ_1 . . . 35
4.5 Results obtained for the δ_i's: this work (left) and [13] (right) . . . 35
4.6 Results obtained for the s_j's: this work (left) and [13] (right) . . . 35
4.7 Results obtained for the μ_i's: this work (left) and [13] (right) . . . 36
4.8 Results obtained for the ν_j's: this work (left) and [13] (right) . . . 36
4.9 Results obtained for the ϕ_j's: this work (left) and [13] (right) . . . 36
4.10 Results obtained for θ: this work (left) and [13] (right) . . . 37
4.11 Posterior distribution histogram of parameter θ . . . 37
4.12 Posterior distributions of the ϕ_j's: boxplot . . . 38
4.13 Posterior distributions of the s_j's: boxplot . . . 38
4.14 Posterior distributions of the s_j's: median and 95% percentile . . . 39
4.15 Posterior distribution boxplots of the s_j's: $Gamma(1,1)$ prior, 25% data . . . 40
4.16 Posterior distribution histograms of the s_j's: $Gamma(1,1)$ prior, 25% data . . . 41
4.17 Shape difference between priors: $Gamma(1,1)$ (blue) and $Gamma(7,1)$ (red) . . . 41
4.18 Posterior distribution boxplots of the s_j's: $Gamma(7,1)$ prior, 25% data . . . 42
4.19 Posterior distribution histograms of the s_j's: $Gamma(7,1)$ prior, 25% data . . . 42
4.20 Posterior distributions of $s_1, ..., s_n$: boxplot . . . 45
4.21 Posterior distributions of $s_2, ..., s_n$: boxplot . . . 46
4.22 Posterior distributions histograms of s_2, s_6, s_{23}, s_{37} . . . 46

4.23 Posterior distribution histogram of parameter θ with the new model 47
4.24 Comparison between μ_i's in the two models 47
4.25 Difference between μ_i's in the two models: boxplot 47
4.26 Comparison between the probabilities of being LVG in the two models . . . 48
4.27 Comparison between the probabilities of being HVG in the two models . . . 49
4.28 Detection of LVG: original model 50
4.29 Detection of LVG: new model 50
4.30 Detection of HVG: original model 51
4.31 Detection of HVG: new model 51

Contents

List of Figures **3**

1 Gene expression mechanisms **7**
1.1 Cells and DNA . . . 7
1.2 The genetic code . . . 8
1.2.1 Structure of a gene . . . 8
1.3 The RNA . . . 9
1.4 Transcription . . . 10
1.5 Measuring gene expression . . . 11

2 Tracking mRNA: experiments pipeline **13**
2.1 Gene expression quantification techniques . . . 13
2.1.1 Northern Blotting . . . 13
2.1.2 SAGE . . . 14
2.1.3 RNA-sequencing . . . 15
2.1.4 Single-cell RNA-sequencing . . . 16
2.2 Estimation of the technical noise: spike-in genes . . . 18

3 Mathematical Model **19**
3.1 Mathematical background . . . 19
3.1.1 Bayes Theorem . . . 19
3.1.2 Monte Carlo Methods . . . 20
3.1.3 MCMC: Markov Chain Monte Carlo . . . 20
3.2 The Dataset . . . 22
3.3 The Model . . . 23
3.3.1 Variance decomposition . . . 24
3.3.2 Detection of highly and lowly variable genes . . . 26
3.4 Methods: MCMC algorithm . . . 27
3.4.1 Prior specification . . . 28
3.4.2 Full-conditional distributions . . . 28

3.4.3 Proposal distributions for the Metropolis algorithm 30

4 Results **33**
4.1 MCMC algorithm results and comparison 33
4.2 Identifiability of capture efficiency parameters 39
4.3 Modification of the model 43

5 Conclusions **53**

A Implemented R script **55**

Bibliography **61**

Chapter 1

Gene expression mechanisms

The purpose of the following chapter is to provide the reader with an overview of the main mechanisms the cell undertakes in order to regulate gene expression. To this extent, we will briefly introduce the biological context from which the present work has stemmed.

1.1 Cells and DNA

The building block of all living matter is the cell. Indeed, it is the smallest structure provided with autonomy and it guarantees the correct functioning of simple and more complex organisms. Cells can be divided in two main distinct types: prokaryotic and eukaryotic cells. Prokaryotic cells, typical of bacteria, are usually much smaller and simpler than eukaryotic cells, and even though several differences can be pointed out among these two domains, the major distinction consists in the DNA (*deoxyribonucleic acid*) location. On the one hand, in a prokaryotic cell the DNA is located in a region called *nucleoid*, that is not membrane enclosed; on the other hand, in a eukaryotic cell the DNA is located in an organelle bounded by a double membrane called *nucleus* [2]. The current dissertation will consider the latter type.
The importance of the DNA relies on the fact that it contains the necessary information to regulate the cell's function. It is a nucleic acid, that is a polymer of many molecular subunits, called *nucleotides*. Each nucleotide is, in turn, composed of a nitrogenous base (*adenine* (A), *cytosine* (C), *guanine* (G) or *thymine* (T)), a sugar called *deoxyribose*, and a phosphate group. The DNA molecule is organised as a long double strand of nucleotide bases, and it can be thought of as a twisted, or helical, ladder [4]. The sides of the ladder are given by alternating molecules of sugar and phosphate, while the "rungs" are constituted by pairs of complementary nitrogenous bases, that is adenine is paired with thymine, and cytosine with guanine, through weak hydrogen bonds. The information is generated by the sequence in which nucleotides appear along the DNA molecule.
In order to fit inside a nucleus' cell, the DNA must be very tightly packed. This is possible

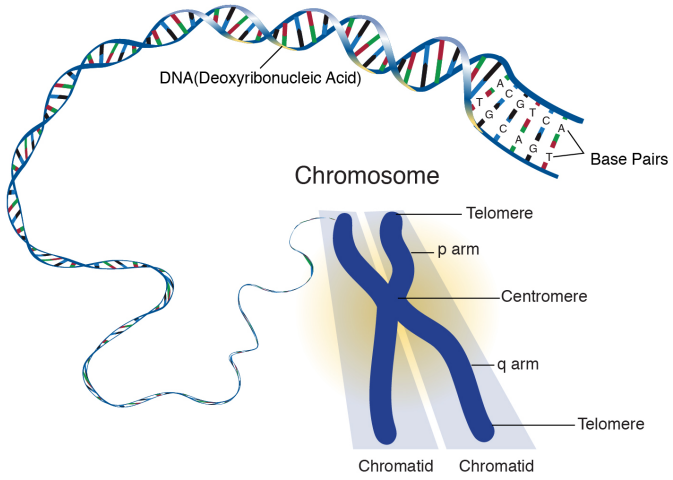

Figure 1.1: DNA packaging, from [20]

thanks to the *histones*, proteins around which the DNA is first wrapped, giving place to structures called *nucleosomes*. The nucleosomes further coil the DNA into *chromatin*, that is, in turn, condensed into the *chromosome* [11], as shown in Figure 1.1.

1.2 The genetic code

As previously mentioned, the DNA contains the necessary information for the regulation of the cell. More in detail, this information is specified by the ordering of base pairs, i.e. by the order in which the four letters A, C, G, T appear along the DNA molecule.
Segments of DNA that code for a particular product are called *genes* [4]. Indeed, genes are sequences of nucleotides that bring the information needed for the production of a specific protein. The genes of an organism are the fundamental units of heredity and, together, constitute its *genome*. The information they carry is referred to as the *genetic code*.

1.2.1 Structure of a gene

The structure of a gene is characterised by three regions: the promoter, the coding region and the termination sequence [14].

- The **promoter** is a sequence identifying the beginning of a gene;
- The **coding region** is the central region of a gene and it is constituted by a sequence of *introns* and *exhons*. An intron is a part of a gene sequence that does not code for amino acids, while exhons are the gene's portions that are expressed in the protein;

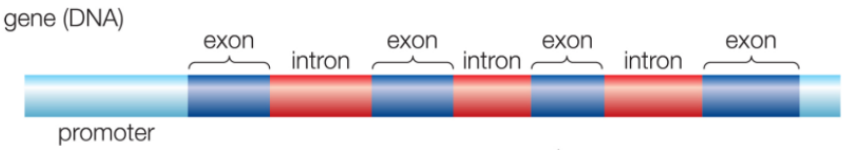

Figure 1.2: Structure of a gene, from [15]

- The **termination sequence**, located after the promoter and the coding region, represents the sequence signaling the end of the gene.

An example of gene structure is depicted in Figure 1.2.
As we will see later on, in the process of converting the genetic information into proteins, the introns are discarded. First of all, though, let us focus briefly on the molecule playing a fundamental role in this process: the RNA.

1.3 The RNA

The RNA (*ribonucleic acid*) molecule is constituted of a single strand of nucleotides. It does not contain the genetic information, however it carries out a crucial role in transferring it outside the cell's nucleus, in the protein synthesis and in the regulation of gene expression [18]. Apart from the fact that it consists of a single strand, it differs from the DNA for two reasons: firstly, the sugar in the backbone is *ribose* and not deoxyribose; secondly, because the base *Uracil* U is present in the place of T. Cells usually contain a quantity of RNA equal to two-to-eight times the amount of DNA [4].
There are three different types of RNA, each one playing an important part in the cell's functioning:

- **messenger RNA (mRNA)**: it is a single-stranded molecule of RNA that corresponds to the genetic sequence of a gene [23]. Its job is to carry outside of the nucleus the code necessary to synthesize proteins. Since DNA cannot be directly decoded into proteins, it needs to be first transcripted (copied) into mRNA. Each mRNA molecule encodes the information for one protein, where each triplet of nucleotides refers to a single specific amino acid [16]: mRNA molecules are fundamental in the process of gene transcription.

- **transfer RNA (tRNA)**: it is a molecule composed of RNA that carries amino acids to the ribosomes, structures in the cell where the proteins are manufactured. The tRNA acts as a physical link between mRNA and proteins [24].

- **ribosomal RNA (rRNA)**: this is a non-coding type of RNA that forms part of the ribosome. The rRNA helps translating the message carried out by the mRNA into proteins [17]. The rRNA plays an essential role in the gene translation process.

In the next section we will focus on the process of gene transcription and, therefore, on the encoding task performed by the mRNA.

1.4 Transcription

Transcription is the process of synthesizing RNA using the information contained in the DNA. It can be split into three main phases: initiation, elongation and termination.

- **Initiation** takes place with the binding of the *RNA polymerase* enzyme to the promoter of a gene. The DNA unwinds so that the RNA polymerase can read the bases in one of the DNA strands. These bases will work as a template for the enzyme, in order to build a complementary mRNA strand. Gene promoters are very important control sequences, since they specify to the RNA polymerase, not only where to begin the transcription, but also which strand of the DNA to read and which direction to take.

- **Elongation** consists in adding nucleotides to the mRNA strand (see Figure 1.3). As the RNA polymerase moves along the DNA strand, the mRNA molecule is progressively formed, and the DNA crossbridges reform [4]. The elongation of the chain continues until the termination phase.

- **Termination** begins when the RNA polymerase reaches a specific termination sequence in the gene. The RNA polymerase stops and disengages from the DNA, leaving the mRNA strand also detached.

The mRNA strand is now bearer of the genetic information. We remark that the introns are non-coding DNA sequences and thus are translated into mRNA and then discarded through the process of *splicing*. The final mRNA molecule can be seen as made of *codons*, nucleotide triplets, each one coding for a specific amino acid. Indeed, it has been assessed that the information flow from a gene to a protein is based on a triplet code: the necessary instructions to build a polypeptide chain consist of a series of non-overlapping, three-nucleotide words [2].
In order to actually manufacture proteins, transcription needs to be followed by another process, called *Translation*. In fact, the result of transcription is a single-stranded copy of the gene, which has to be then translated into a protein. In the translation process, the mRNA molecule is read, codon by codon, and used as a template to assemble the chain of amino acids that will result in the corresponding protein. Figure 1.4 provides an overview of transcription and translation, clarifying the dynamics behind the whole process, that is

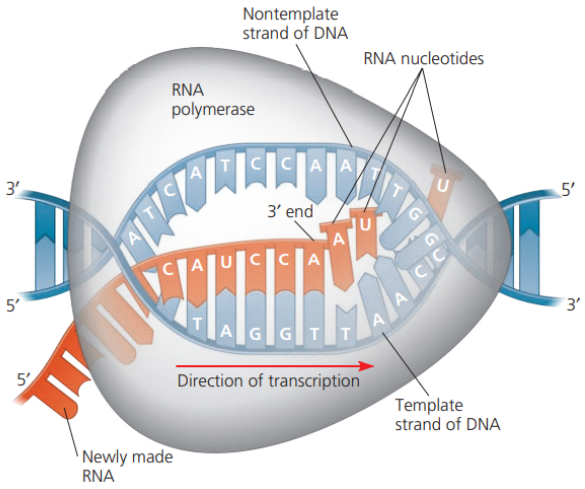

Figure 1.3: Transcription elongation, from [2]

referred to as *Gene expression*.

1.5 Measuring gene expression

For every gene, the amount of corresponding mRNA contained within the cell acts as an indicator of the level of gene expression for that cell. In fact, the abundance of the gene-specific mRNA transcript is directly linked to the expression level of that gene. Possible variations of these levels denote changes in the corresponding gene's activity, thus providing insights in many biological processes. Gene expression patterns linked to a specific biological state can be assumed as biomarkers for that condition, for instance in case of normal progression such as disease development. Moreover, variations in gene expression levels can be used to identify patients at higher risk for a certain medical condition or to analyze the consequences associated with a certain treatment [12].
Therefore, gene expression levels can be intended as "signatures" that characterise the distinct tissues of an organism. By interrogating these signatures, it is possible to understand how a tissue is governed at the molecular level. Even in case of a simple tissue, characterised by homogeneous cells, one can find heterogeneity in gene expression levels that might be related to different factors. For example, it could refer to different cell-cycle stages, but it could also point out the presence of co-expressed genes or lead to the discovery of novel subpopulations of cells, among others [13].
Strictly speaking, the term "gene expression" comprises the whole process, i.e. from the moment the gene is activated and transcribed to the moment when the protein is ready to

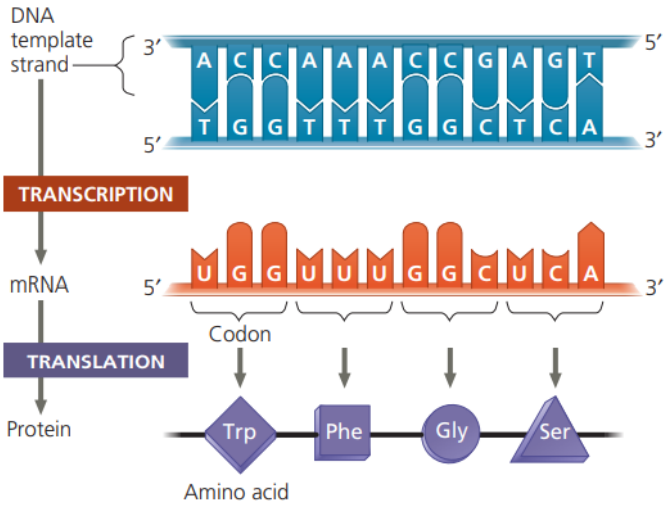

Figure 1.4: The triplet code, from [2]

perform its function; however, as previously mentioned, studies on gene expression usually aim at measuring the level of a gene in terms of mRNA instead of using other indicators [9]. Nowadays, many techniques make it possible to evaluate, more and more precisely, the quantity of mRNA contained inside a cell. This advancement has allowed gene expression studies to develop and significantly improve. As we will see in the next chapter, one of the most powerful techniques up to date is called *Single-cell RNA-sequencing (scRNA-seq)* and it enables to report information, in terms of mRNA quantity, at the single-cell level.

Chapter 2

Tracking mRNA: experiments pipeline

In the recent years, research on gene expression variations has increasingly grown, for many reasons. First of all, clinical samples and methods for measuring gene expression from a variety of tissues have drastically grown. Secondly, huge databases of gene expression data are becoming more accessible. Lastly, the most current technologies have nowadays reached a high degree of affordability and have broader applicability [12].

2.1 Gene expression quantification techniques

In this section, we will outline some of the most common laboratory methods that have been employed to quantify gene expression levels, in order to give the reader an idea of how RNA quantification is performed, together with the improvements that have been reached with the passing of time.

2.1.1 Northern Blotting

Northern Blotting is a laboratory method that can be used to analyse a sample of mRNA from a certain tissue or cell type, so that it is possible to quantify the RNA expression of particular genes. This technique consists in different subsequent steps that we will outline as they are described in [21] and [22]:

- First, the cells in the sample have to be exposed to an enzyme, called *protease*, that has the task of degrading the cells membranes, thus releasing the genetic material within the cells; after that, the mRNA is separated from the rest of the cellular content.
- *Gel electrophoresis* is applied; this technique is used to separate the molecules of

mRNA from each other, by passing an electrical current through a gel that contains them.

- After separation, the mRNA is transferred on a blotting membrane; this membrane is then carrier of all the mRNA fragments that were on the gel.
- Lastly, in order to identify the mRNA transcripts that refer to a specific gene, the membrane is treated with a *probe*, i.e. a short piece of single-stranded DNA or RNA. This probe is complementary to a specific sequence of mRNA that is in the sample; therefore, it will bind to a specific mRNA fragment on the membrane. Since the probe is labelled with a radioactive molecule, this permits to detect the mRNA fragment of interest and to quantify how much of it was in the sample, thus revealing the strength of the corresponding gene expression.

It is important to notice how Northern Blotting actually allows to look only for one or very few genes at a time. Fortunately, many advances have been made in this direction. Indeed, new technologies make large-scale studies of gene expression now possible [21]. For example, one of these is called *SAGE (Serial Analysis of Gene Expression)* and its procedure will be described in the next section.

2.1.2 SAGE

"It is now recognized that phenotypic changes in a tissue are the result of changes in the spatial and temporal expression of dozens or even hundreds of genes" [10]. Nowadays, it has become more and more evident how, in order to understand the molecular basis of a tissue, one needs to study the expression level variations that are related not only to individual genes, but to a variety of them. In this regard, SAGE has been applied to the gene expression profiling of a wide number of diseases [10].
SAGE is a complex protocol, structured as follows:

- First, mRNA is isolated from other cellular contents and then converted into complementary DNA sequences (*cDNA*), through a process called *Reverse Transcription*. This is done because mRNA is generally more fragile than DNA, and hence more difficult to handle [21]. The cDNA is then transformed into a double-stranded cDNA.
- Next, with the help of a cutting enzyme, segments of nucleotides (*tags*) are cut at specific locations of each cDNA molecule. For each molecule, two tags are then combined, becoming an identifier of the corresponding mRNA, hence of the corresponding gene.
- Then, the different tags are linked together, forming long cDNA chains, the *concameters*. These chains will thus contain the mRNA identifiers from a group of genes.

- Subsequently, in order for the concameters to be processed by a sequencing machine, they are injected into bacteria and then copied million of times, thanks to the bacteria's own replication.
- Finally, it is possible to process the data with a sequencing machine, that compares the obtained tags with a sequence database. This allows to identify, at once, all genes where the tags come from, and to quantify the corresponding expression level by counting the times each tag appears.

SAGE is a powerful technique for gene expression analysis, that brings a significant advantage. Indeed, it allows to measure the expression levels both of known and unknown genes. Sometimes the sequencing machine is not able to understand where certain tags come from and this only means that probably the corresponding genes have not been yet studied. This is the reason why SAGE has been helpful in discovering new genes, that are associated with a variety of diseases [21]. SAGE is rival to another technique called *DNA microarray analysis*, that also allows to analyse and quantify the presence of many genes, at once, within a DNA sample. We will not cover the details of this technique; however, a significative benefit of SAGE with respect to DNA microarray analysis is the fact that it is able to profile gene expression without having prior sequence information. Still, the utility of SAGE is limited by the requirement of a large amount of mRNA as input [10]. Furthermore, it is noteworthy how both the techniques introduced, Northern Blotting and SAGE, are quantitative techniques, i.e. they enable not only the identification of the genes within a mRNA or cDNA sample, but also the quantification of their expression levels. And this extremely powerful aspect is not to be taken for granted in all gene expression analysis techniques up to date.
In the next section, we will outline a very popular and successful technology, namely *RNA-sequencing (RNA-seq)*, followed by its finer improvement: *single-cell RNA-sequencing (scRNA-seq)*.

2.1.3 RNA-sequencing

Over the past few decades, *RNA-sequencing* has become essential in the analysis of gene expression. Indeed, RNA-sequencing is widely considered superior to most of the other technologies in the same field. First of all, because RNA-sequencing (like SAGE) is not limited by a prior knowledge of the organism's genome and can thus be performed in species whose genomes are still not available. Secondly, because RNA-seq shows a greater sensibility for genes that are both lowly and very highly expressed. Furthermore, it results in lower technical variation [19].
More in detail, the general RNA-seq protocol is the following (see [5],[6],[7]):

- **RNA isolation**: it consists in separating the totality of mRNA from the other cellular contents; as already mentioned, it represents an extremely sensitive step, since the mRNA can break down easily.

- **Reverse transcription**: in this step, the mRNA is reverse transcribed into complimentary DNA fragments (a cDNA library); this process requires an enzyme called *reverse transcriptase* in order to generate cDNA from a RNA template.

- **cDNA fragmentation and amplification**: next, the cDNA is fragmented and *adapters* are added to each end of the fragments; the adapters are useful because they contain functional elements which permit sequencing. After that, the fragments of cDNA are then amplified through the *PCR (Polymerase Chain Reaction)*, a molecular biology technique that enables the amplification of nucleic acids' fragments whose initial and terminal nucleotide sequences are known. These two substeps are not necessary executed in this order and present many details and choices not discussed here (for an in-depth analysis see [6]). Either way, this step leads to the library preparation and, once the library is prepared, one can use a sequencing platform of their choice in order to sequence the cDNA library to the required and desired depth.

Finally, it is possible to map the obtained data to a reference genome or, if there is no reference available, assemble it de novo. This will enable to discover novel transcripts, other than the ones that are already known.
Despite being a very powerful technique, RNA-seq is actually only a starting point. Indeed, it does not allow to perform cell-type specific analyses, since the isolated mRNA comes from a tissue which may be constituted by different cell types. Nonetheless, this challenge has been overcome by a developed version of RNA-seq, namely *single-cell RNA-sequencing*. This advanced technology will be presented in the next subsection.

2.1.4 Single-cell RNA-sequencing

Single-cell RNA-sequencing is a recent technology that enables to profile the totality of mRNA transcripts from a large number of individual cells. Its strength relies on the fact that it is able to address complex tissues, i.e. constituted by different types of cells, and therefore to answer questions that could not be undertaken with bulk RNA-sequencing [1]. Indeed, traditional RNA-sequencing methods, whilst analysing the RNA of a population of cells, only supply bulk average measurements. On the contrary, scRNA-seq allows representing the transcriptome of each individual cell, better capturing the heterogeneity of the sample [25]. This also results, naturally, in much larger data than those provided by RNA-seq experiments.
However, together with many advantages, scRNA-seq carries some challenges. In fact, the need to sequence mRNA from a single cell is itself related to two main non-negligible problems: first, the necessity of capturing single cells quickly and accurately and, secondly, the issue of amplifying the minute amounts of mRNA within each cell [5]. Concerning single-cell isolation, it can be addressed by *micromanipulation techniques*, that allow capturing single cells from samples constituted by few cells, like the early embrio. For example, with a technique called *Laser capture microdissection (LCM)*, it is possible to capture single cells

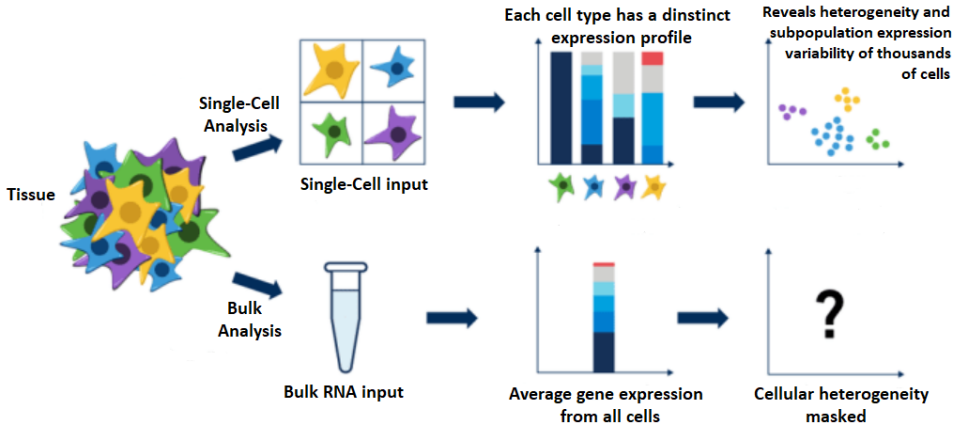

Figure 2.1: Single-cell versus bulk analysis, from [26]

from a tissue, by using a laser that attaches them to a thin film, which is then removed. Nonetheless, approaches like these are time expensive and have low throughput, therefore many strategies have been pursued in order to improve their efficiency. Additionally, more promising techniques, such as *Microdroplet-based microfluidics methods* are growing (details can be found in [8]).

In Figure 2.1, an overview on the differences brought by RNA-seq and scRNA-seq analyses. The scRNA-seq protocol is the same outlined for RNA-seq. Indeed, only the first step differs, in the sense that the mRNA fragments come from individual cells that have to be captured and lysed. Then, the rest of the procedure coincides, that is, in short: reverse transcription, PCR amplification and sequencing library preparation. It is important not to neglect that, in this process, there are sources of technical noise. Firstly, during reverse transcription, it is estimated that only a portion of the fragments (10-20%) is reverse transcribed, resulting in high technical noise, especially in the case of lowly expressed genes. Secondly, PCR amplification, as well as an alternative existing method called *in vitro transcription*, may induce biases [5], making it paramount to take into account these aspects during the analysis. *"Indeed, losses in cDNA synthesis and bias in cDNA amplification lead to severe quantitative errors"* [3]. Different recent strategies have been explored in order to improve the quantitative nature of scRNA-seq from the two points of view: cDNA synthesis efficiency and amplification bias. The first setting the limit of detection, the latter of quantitative accuracy.

Concerning the second issue, a solution supported by scRNA-seq technology consists in using *unique molecular identifiers (UMIs)*. UMIs are short barcodes or sequences that are attached to transcripts before amplification, thus enabling the identification of PCR duplicates and making it possible to obtain more accurate estimates of the gene expression

levels. Sequencing using UMIs can reduce the false-positive variant calls, distinguishing them from the true variants. Since in scRNA-seq it is necessary to exclude barcodes that are unlikely to represent intact individual cells, this is done by setting a dataset-specific threshold on the smallest number of accepted UMIs. Indeed, UMIs based datasets are expressed in terms of number of molecules and a number smaller than the chosen threshold could indicate that the corresponding barcode should not be considered as associated to an intact cell. UMI based scRNA-seq data can be represented by matrices, whose entries are the number of mRNA molecules that, in each cell, refer to a specific gene [13]. On the other hand, regarding the problem related to cDNA synthesis, it has also been possible to obtain some improvements. This has resulted in an increased mRNA capture efficiency, that otherwise would settle around 10% (details in [3]).
Single-cell RNA-sequencing analysis is a rapidly evolving field. Indeed, in the last years, it has encountered many upgrades and refinements that, together with the improvement of its performances, have reduced the time- and cost-consumption. Technical noise and bias have also been limited, but still not removed. In the next section, we will present a well-known and widely used strategy, that is applied in the sequencing experiments pipeline in order to better quantify and handle the technical noise.

2.2 Estimation of the technical noise: spike-in genes

Gene expression variability across cells can emerge due to biological factors, as well as to technical ones. In order to understand biological variability, it is paramount to estimate in some way the degree of the technical variability. So far, the most used approach consists in adding to each cell's lysate external spike-in mRNA molecules. Indeed, since the number of spike-in molecules added to each cell is known beforehand, it provides a standard to which empirical measurements can be compared, thus allowing to quantitatively assess the technical variation [13]. It is extremely important to add, for each spike-in gene, the same amount of molecules in each cell, in order not to introduce erroneous variability across cells. Furthermore, this quantity needs to be calibrated, so that the spike-in molecules do not incur in over- or under-representation.
A well-known example is given by the set derived by *ERCC (External RNA Controls Consortium)*. This set consists of 92 extrinsic spike-in RNA molecules. As we will see later on, in this work a portion of this set will be used.

Chapter 3

Mathematical Model

In this chapter, the problem under analysis will be introduced and, subsequently, the proposed approach to address it will be presented. But first, let us briefly discuss the mathematical background.

3.1 Mathematical background

In order to set the basis of the following part of the work, we will provide the reader with few important mathematical concepts.

3.1.1 Bayes Theorem

In statistics, *Bayes Theorem* is the milestone behind the bayesian approach. Given a specific model that explains the behaviour of one or more observed random variables, the bayesian approach consists in assuming that the parameters of the model are themselves random variables. One is, therefore, interested in understanding how does the parameters' distribution differ, after knowing the observations. More in detail, being y the vector containing the observed random variables, and θ the vector of parameters used to explain the behaviour of y, Bayes Theorem is stated as follows:

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{f(y)}$$

where:

- $f(\theta|y)$ is the posterior distribution of the parameters after having observed the data
- $f(y|\theta)$ is the likelihood of the data y
- $f(\theta)$ is the prior distribution of the parameters and it reflects the prior beliefs we have on those parameters

- $f(y)$ is the distribution of the data y; it is only a normalisation constant with respect to θ, that does not bring additional information and can be therefore neglected.

Hence, Bayes Theroem can be reformulated as:

$$f(\theta|y) \propto f(y|\theta)f(\theta)$$

When $f(\theta|y)$ is a known distribution, the prior distribution and the likelihood are said to be *conjugate*. However, this rarely happens and the majority of times the posterior distribution does not correspond to any known distribution. This is when *Monte Carlo Methods* come into hand, showing how, instead of the posterior distribution, one can use samples that are obtained from it.

3.1.2 Monte Carlo Methods

Let us consider a random variable $Y \sim F(\theta)$, where F is some distribution that depends on the parameters θ. If one would want to have an estimate of the random variable's expected value $\mu = E[Y] = \int yf(y;\theta)dy$, the *sample mean* should be used. Indeed, the sample mean is a correct estimator of the variable's expected value and it is expressed as:

$$\bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$$

where the y_i are realisations of Y, therefore samples from the variable's distribution. In the same spirit, given a generic function of Y, $h(Y)$, in order to estimate its expected value $E[h(Y)] = \int h(y)f(y;\theta)dy$, we can proceed as follows:

$$E[h(y)] \simeq \sum_{b=1}^{B} \frac{h(y^b)}{B}$$

where $y^1, ..., y^b$ are B samples from $f(y;\theta)$.
Monte Carlo Methods state, therefore, that even if a certain distribution is not attributable to any known distribution, it is still possible to compute many interesting quantities that refer to it, as long as one is able to sample from this distribution. This is extremely useful when dealing, for example, with the posterior distribution of some parameters. Indeed, being able to sample from $f(\theta|y)$, it is possible to compute $\int h(\theta)f(\theta;y)d\theta$ for any function 1-to-1 $h(\theta)$ of the parameters. When one is not able to sample from the desired distribution, some algorithms come into hand.

3.1.3 MCMC: Markov Chain Monte Carlo

MCMC are methods used to obtain samples from a generic multivariate distribution. Let us consider a multivariate variable $\theta = (\theta_1, ..., \theta_p)$ with a joint distribution $f(\theta|y)$, from which we do not know how to sample. An MCMC method provides a way to obtain samples from this joint distribution. It is based on the following steps:

1. Initialisation of θ, by assigning some initial values $\theta_1^0, ..., \theta_p^0$
2. For $b = 1, ..., B$:
   - sample θ_1^b from the distribution $\theta_1|\theta_2^{b-1}, ..., \theta_p^{b-1}, y$
   - sample θ_2^b from the distribution $\theta_2|\theta_1^b, \theta_3^{b-1}, ..., \theta_p^{b-1}, y$
   - ...
   - sample θ_p^b from the distribution $\theta_p|\theta_1^b, ..., \theta_{p-1}^b, y$

At the end of the algorithm, one will obtain the B required samples from the joint distribution, represented by the rows of the following matrix:

$$\begin{bmatrix} \theta_1^1 & \theta_2^1 & ... & \theta_p^1 \\ \theta_1^2 & \theta_2^2 & ... & \theta_p^2 \\ ... & & & \\ \theta_1^B & \theta_2^B & ... & \theta_p^B \end{bmatrix}$$

It is common practice not to consider the first samples, since they could be strictly affected by the corresponding initial condition. These samples are therefore deleted, through an operation called *burnin*. Furthermore, since each sample has a knock-on dependence from the previous one, in order to avoid that, a *thin* operation is carried out. This last operation consists in fixing a number N and in keeping only one sample each N samples.
Additionally, a fundamental aspect of the algorithm are the distributions used for sampling during the procedure. Indeed, those distributions are called *full-conditional distributions*, since they are distributions depending on one single variable, conditioned to everything else is present in the model. The importance of such distributions is that they are univariate and it is therefore simpler to sample from them. Still, it is not granted to be able to sample from those and, based on the situation, there are two viable strategies. These are referred to as *Gibbs Sampling* and *Metropolis*.

Gibbs Sampling

Gibbs Sampling comes into play when the full-conditional distribution of a certain parameter is a known distribution, from which we are then able to sample. In this case, the corresponding step of the algorithm is said to be a *Gibbs step* and does not bring additional challenges.

Metropolis

When one is not able to directly sample from the required distribution, a *Metropolis step* needs to be performed.
Let us suppose that we want to sample from a generic distribution $f(x|z)$. Moreover, let us

consider a distribution Q, with density q, that we will call a *proposal distribution*. Then, the iterative algorithm of Metropolis consists in the following steps:

- A value x^* is proposed, by sampling from $Q(x^{b-1}, t)$, where x^{b-1} is the value of x at the preceding iteration of the MCMC, while t represents the parameters on which the proposal distribution depends; therefore x^* will depend on those parameters and on the previous value of x;
- The following ratio is computed, in order to determine α:

$$\alpha = min\left[1, \frac{f(x^*|z)q(x^{b-1}|x^*)}{f(x^{b-1}|z)q(x^*|x^{b-1})}\right] \in [0, 1]$$

- A random number u is generated from a Uniform distribution $U(0, 1)$ and then:
  1. if $u \leq \alpha$, then $x^b = x^*$ (the proposed new value is accepted)
  2. if $u > \alpha$, then $x^b = x^{b-1}$ (the value of the previous iteration is kept)

This allows to obtain samples from $f(x|z)$, when one does not know how to sample from it. Furthermore, one does not need to know the complete analytic expression of such distribution, but it is sufficient to consider a function proportional to it.

3.2 The Dataset

The Dataset that will be considered during the whole analysis is a scRNA-seq UMI-based dataset, introduced by [13]. It consists of gene expression measurements coming from mouse Embryonic Stem Cells (ESC) and of some spike-in genes from the extrinsic molecules set derived by the External RNA Controls Consortium. The data pre-processing step repeats what already done by the authors of [13], resulting in the following three data structures:

- A matrix X of expression counts, where X_{ij} represents the number of mRNA molecules referring to gene i in cell j; i represents the gene identifier and ranges between 1 and q, where the first q_0 genes are biological genes, while the remaining ones are spike-in; j is the cell identifier and it ranges between 1 and n. In the final rearranged version of the dataset we have: $q = 7941$ total genes, $q_0 = 7895$ biological genes (hence $q - q_0 = 46$ spike-in genes), $n = 41$ cells;
- A vector containing the true quantities of spike-in mRNA fragments that are added to each cell with the same amount (separately for each gene);
- A binary vector that identifies whether the corresponding gene is a spike-in gene or not (it will have the first q_0 entries equal to FALSE/0 and the remaining $q - q_0$ equal to TRUE/1);

With the dataset in mind, let us now introduce the model, as it is presented in literature.

3.3 The Model

The model introduced in [13] consists of a hierarchical structure that, through common parameters, jointly models both sets of genes, biological and spike-in. More in details, its mathematical definition is the following:

$$X_{ij}|\mu_i, \phi_j, \nu_j, \rho_{ij} \overset{ind}{\sim} \begin{cases} Poisson(\phi_j \nu_j \mu_i \rho_{ij}), & i = 1, ..., q_0, j = 1, ..., n; \\ Poisson(\nu_j \mu_i), & i = q_0 + 1, ..., q, \ j = 1, ..., n; \end{cases} \tag{3.1}$$

with:

$$\nu_j|s_j, \theta \overset{ind}{\sim} Gamma(1/\theta, 1/(s_j\theta))$$
$$\rho_{ij}|\delta_i \overset{ind}{\sim} Gamma(1/\delta_i, 1/\delta_i)$$

where:

- μ_i refers to the normalised expression rate of gene i in the cells' population (it represents the true concentration and it is known for the spike-in genes);
- ϕ_j and s_j are cell-specific normalising constants that are considered as additional model parameters and need, therefore, to be estimated. The parameter ϕ_j is used to adjust, in each cell j, the expression rate in terms of the total mRNA content (no need to be used for spike-in genes), while s_j is the so-called *capture efficiency* parameter and accounts for differences that can arise during the process of capturing and sequencing the single cells, in the dataset construction;
- ν_j is a random effect ($E[\nu_j|s_j, \theta] = s_j$, $Var(\nu_j|s_j, \theta) = s_j^2\theta$) that quantifies unexplained technical noise through the hyper-parameter θ and oscillates around the capture efficiency normalising constant s_j;
- ρ_{ij} is a random effect ($E[\rho_{ij}|\delta_i] = 1$, $Var(\rho_{ij}|\delta_i) = \delta_i$) that quantifies biological cell-to-cell heterogeneity through the gene-specific hyper-parameters δ_i; the random effects ν_j and ρ_{ij} are mutually independent.

The graphical representation of the model is depicted in Figure 3.1, in order to provide a deeper understanding of the relations between the different model components. The representation involves two cells, j and j', and two genes, biological (i) and spike-in (i'). Squared nodes represents known observed quantities (the expression counts X_{ij} and the added mRNA molecules μ_i for spike-in genes), while circular

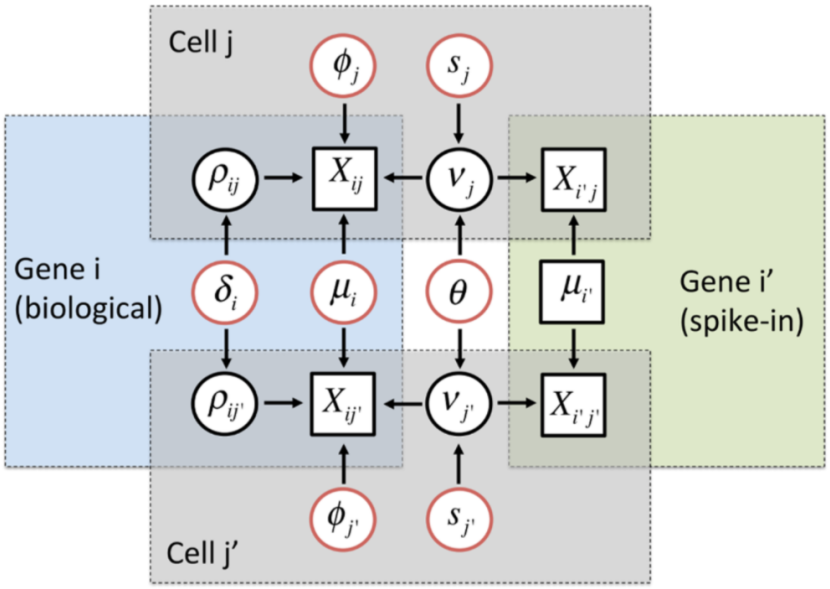

Figure 3.1: Graphical representation of the model, by [13]

nodes are the unknown elements. Black circular nodes refer to the random effects (intermediary layer of the structure) and the red ones to the parameters (top layer of the structure).
At last, in order for the ϕ_j's to be identified, an additional step is needed. Indeed, since the $\mu_1, ...\mu_{q_0}$ are unknown, the following additional constraint is imposed:

$$n^{-1} \sum_{j=1}^{n} \phi_j = \phi_0$$

This restriction can be obtained, by reparametrising the ϕ_j's in terms of $k_1, ..., k_n$ as follows:

$$\phi_j = \phi_0 \frac{e^{k_j}}{\sum_{j=1}^{n} e^{k_j}}, \; j = 1, ..., n, \quad k_1 = 0$$

The analysis is not affected by the value of ϕ_0, that will be considered equal to n.

3.3.1 Variance decomposition

By integrating out the ν_j's and the ρ_{ij}'s, it is possible to compute both expected value and variance of the marginal distribution of $X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta$ (expression count of gene i in cell j, not depending on ν_j and ρ_{ij}). As we will see, the variance decomposition will allow to define a criterium for the detection of highly and lowly

variable genes (as reported in [13]).
Indeed, we have:

$$\begin{aligned} i \leq q_0: \quad & E[X_{ij}|\nu_j, \rho_{ij}, \mu_i, \delta_i, \phi_j, s_j, \theta] = E[Poisson(\phi_j \nu_j \mu_i \rho_{ij})] = \phi_j \nu_j \mu_i \rho_{ij} \\ & \rightarrow E[X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta] = \phi_j E[\nu_j|s_j, \theta] \mu_i E[\rho_{ij}|\delta_i] = \phi_j s_j \mu_i \end{aligned}$$

$$\begin{aligned} i > q_0: \quad & E[X_{ij}|\nu_j, \mu_i, s_j, \theta] = E[Poisson(\nu_j \mu_i)] = \nu_j \mu_i \\ & \rightarrow E[X_{ij}|\mu_i, s_j, \theta] = E[\nu_j|s_j, \theta] \mu_i = s_j \mu_i \end{aligned}$$

This can be summarised in the next equation:

$$E[X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta] = \phi_j^{I_i} s_j \mu_i \tag{3.2}$$

$$\text{with } I_i = 1 \text{ when } i \leq q_0,\ I_i = 0 \text{ otherwise.}$$

Concerning the variance, the computation is less straightforward. Therefore, let us first recall three important properties, that will be useful for a better understanding of the subsequent steps.

1. $Var(aX) = a^2 Var(X)$, where $a \in R$, X random variable
2. $Var(XY) = Var(X)Var(Y) + Var(X)E[Y]^2 + Var(Y)E[X]^2$,
   with X, Y independent random variables
3. $Var(Y) = E[Var(Y|X)] + Var(E[Y|X])$ (*Law of total variance*),
   with X, Y random variables on the same probability space

Hence, let us consider $Y = X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta$, $X = \nu_j, \rho_{ij}$, and therefore $Y|X = X_{ij}|\nu_j, \rho_{ij}, \mu_i, \delta_i, \phi_j, s_j, \theta$.

Then, we will have:

$$Var(X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta) = E[Var(X_{ij}|\nu_j, \rho_{ij}, \mu_i, \delta_i, \phi_j, s_j, \theta)] + Var(E[X_{ij}|\nu_j, \rho_{ij}, \mu_i, \delta_i, \phi_j, s_j, \theta]),$$

where $X_{ij}|\nu_j, \rho_{ij}, \mu_i, \delta_i, \phi_j, s_j, \theta \sim Poisson(\mu_i \nu_j (\phi_j \rho_{ij})^{I_i})$, with I_i as in Equation 3.2.

Thus, we obtain:

$$\begin{aligned} i \leq q_0: \quad & Var(X_{ij}|\mu_i, \delta_i, \phi_j, s_j, \theta) = E[\mu_i \nu_j \phi_j \rho_{ij}] + Var(\mu_i \nu_j \phi_j \rho_{ij}) = \\ & = \mu_i \phi_j E[\nu_j] E[\rho_{ij}] + \mu_i^2 \phi_j^2 Var(\nu_j \rho_{ij}) = \end{aligned}$$

$$= \mu_i \phi_j s_j + \mu_i^2 \phi_j^2 \Big(Var(\nu_j) Var(\rho_{ij}) + Var(\nu_j) E[\rho_{ij}]^2 + Var(\rho_{ij}) E[\nu_j]^2 \Big) =$$
$$= \mu_i \phi_j s_j + \mu_i^2 \phi_j^2 \Big(s_j^2 \theta \delta_i + s_j^2 \theta + \delta_i s_j^2 \Big) =$$
$$= \mu_i \phi_j s_j + \mu_i^2 \phi_j^2 s_j^2 \Big(\theta \delta_i + \theta + \delta_i \Big)$$

$$i > q_0 : \quad Var(X_{ij} | \mu_i, \delta_i, \phi_j, s_j, \theta) = E[\mu_i \nu_j] + Var(\mu_i \nu_j) =$$
$$= \mu_i E[\nu_j] + \mu_i^2 Var(\nu_j) = \mu_i s_j + \mu_i^2 s_j^2 \theta$$

This can be summarised as in [13], leading to the following variance decomposition equation:

$$Var(X_{ij} | \mu_i, \delta_i, \phi_j, s_j, \theta) = \mu_i s_j \phi_j^{I_i} + \theta (\phi_j^{I_i} \mu_i s_j)^2 + I_i \delta_i (\theta + 1) (\phi_j^{I_i} \mu_i s_j)^2, \tag{3.3}$$

As shown in Equation 3.3, one can see how the observed variability can be decomposed into three components: the baseline Poisson variance, the variance inflation due to unexplained technical noise and the biological cell-to-cell variability. This decomposition will be the key point in order to identify genes that present genuine biological heterogeneity in the sample.

3.3.2 Detection of highly and lowly variable genes

The previous variance decomposition provides an intuitive criterium for the identification of highly and lowly variable genes (HVG and LVG, respectively), as introduced in [13]. Indeed, HVG can be identified as those for which a large fraction of the total expression variability can be explained by the biological heterogeneity component. Let us have a look at the ratio between the biological heterogeneity component of the variance and the total variance. This can be derived from Equation 3.3 and results in:

$$\frac{I_i \delta_i (\theta + 1) (\phi_j^{I_i} \mu_i s_j)^2}{\mu_i s_j \phi_j^{I_i} + \theta (\phi_j^{I_i} \mu_i s_j)^2 + I_i \delta_i (\theta + 1) (\phi_j^{I_i} \mu_i s_j)^2} = \frac{I_i \delta_i (\theta + 1)}{(\phi_j^{I_i} \mu_i s_j)^{-1} + \theta + I_i \delta_i (\theta + 1)}$$

Since HVG are searched among biological genes, $I_i = 1$, and they are thus characterised as those genes for which:

$$\sigma_i \equiv \frac{\delta_i (\theta + 1)}{[(\phi s)^* \mu_i]^{-1} + \theta + \delta_i (\theta + 1)} > \gamma_H, \tag{3.4}$$

where $(\phi s)^*$ is defined as $(\phi s)^* = \underset{j \in 1, \ldots, n}{median} (\phi_j s_j)$, in order to represent a typical cell

in the sample.
This means that a gene i is considered as highly variable, if the proportion of the total expression variability that is explained by the biological heterogeneity component exceeds a certain threshold γ_H. Equation 3.4 can be rewritten in terms of δ_i as follows:

$$\delta_i > \Big[\frac{\gamma_H}{1-\gamma_H}\Big]\Big[\frac{((\phi s)^*\mu_i)^{-1}+\theta}{1+\theta}\Big] \tag{3.5}$$

In the same way, LVG can be identified as those genes for which:

$$\sigma_i \equiv \frac{\delta_i(\theta+1)}{[(\phi s)^*\mu_i]^{-1}+\theta+\delta_i(\theta+1)} < \gamma_L, \tag{3.6}$$

Hence:

$$\delta_i < \Big[\frac{\gamma_L}{1-\gamma_L}\Big]\Big[\frac{((\phi s)^*\mu_i)^{-1}+\theta}{1+\theta}\Big] \tag{3.7}$$

Additionally, the evidence of a gene being highly or lowly variable can be quantified through the following posterior probabilities:

$$\pi_i^H(\gamma_H) = P(\sigma_i > \gamma_H | \{x_{ij} : i = 1, ..., q, j = 1, ..., n\}) > \alpha_H \tag{3.8}$$

$$\pi_i^L(\gamma_L) = P(\sigma_i < \gamma_L | \{x_{ij} : i = 1, ..., q, j = 1, ..., n\}) > \alpha_L \tag{3.9}$$

where the thresholds $\gamma_H, \gamma_L, \alpha_H$ and α_L can be optimally chosen as explained in [13]. Therefore, for specific threshold choices, a gene i is classified as HVG if σ_i is higher than γ_H and if this is supported by strong evidence ($P(\sigma_i > \gamma_H) > \alpha_H$). Similarly, LVG are those such that $\sigma_i < \gamma_L$ and $P(\sigma_i < \gamma_L) > \alpha_L$.

3.4 Methods: MCMC algorithm

In this section, we will describe the methodology used to estimate the parameters of the model. Indeed, we build an MCMC algorithm in order to learn the posterior distribution of such parameters. More in detail, the authors of [13] have implemented an Adaptive Metropolis (AM) within Gibbs Sampling (GS) algorithm (available in a R package); however, still taking into account the prior distributions defined in the

article, we have decided to build our own MCMC algorithm in order to gain a deeper understanding of the problem and to compare the obtained results.

3.4.1 Prior specification

The prior distributions considered for the parameters of the model are the following (prior independence between all parameters is assumed):

- $\delta_1, ..., \delta_{q_0} \overset{iid}{\sim} Gamma(a_\delta, b_\delta)$
- $k_2, ..., k_n \overset{iid}{\sim} Normal(0, \sigma_k^2)$
- $s_1, ..., s_n \overset{iid}{\sim} Gamma(a_s, b_s)$
- $\theta \sim Gamma(a_\theta, b_\theta)$
- $\pi(\mu_1, ..., \mu_{q_0}) \propto \prod_{i=1}^{q_0} \mu_i^{-1}$

The latter one is an improper non-informative prior that leads to a uniform prior on the real line for each $log(\mu_i)$; even so, if there is no reliable prior information, it is strongly recommended to use it.
Furthermore, concerning the fixed hyper-parameters $a_\delta, b_\delta, \sigma_k^2, a_\theta, b_\theta, a_s, b_s$, the authors of [13] have shown how the choice of these hyper-parameters does not have major consequences in posterior inference; therefore, we will consider the following values: $a_\delta = b_\delta = \sigma_k^2 = a_\theta = b_\theta = a_s = b_s = 1$.

3.4.2 Full-conditional distributions

Let us now display the computation of the full-conditional distributions that will be needed to build the MCMC algorithm. As we will see, in some cases they correspond to known distributions, from which we are able to sample (Gibbs Sampling), in some other cases the corresponding expression is not related to any known distribution, therefore a Metropolis step will be necessary to sample from it.
The computation of the full-conditional distributions relies on Bayes Theorem and results in the following outcomes:

$$1: \ f(\rho_{ij}|X_{ij}, \mu_i, \nu_j, \phi_j, \delta_i) \propto f(X_{ij}|\rho_{ij}, \mu_i, \nu_j, \phi_j) f(\rho_{ij}|\delta_i)$$

$$\propto \rho_{ij}^{X_{ij}} e^{-\mu_i \nu_j \phi_j \rho_{ij}} \rho_{ij}^{1/\delta_i - 1} e^{-\rho_{ij}/\delta_i} \propto \rho_{ij}^{X_{ij} + 1/\delta_i - 1} e^{-\rho_{ij}(\mu_i \nu_j \phi_j + 1/\delta_i)}$$

$$\sim Gamma\Big(X_{ij} + \frac{1}{\delta_i}, \mu_i \nu_j \phi_j + \frac{1}{\delta_i}\Big), \quad i = 1, ..., q_0, j = 1, ..., n$$

$$2:\ f(\mu_i|X_{ij},\nu_j,\phi_j,\rho_{ij}) \propto \prod_{j=1}^{n}\Big[f(X_{ij}|\mu_i,\nu_j,\phi_j,\rho_{ij})\Big]f(\mu_i)$$

$$\propto \prod_{j=1}^{n}\Big[\mu_i^{X_{ij}}e^{-\mu_i\nu_j\phi_j\rho_{ij}}\Big]\frac{1}{\mu_i} = \mu_i^{\sum_{j=1}^{n}X_{ij}-1}e^{-\mu_i(\sum_{j=1}^{n}\nu_j\phi_j\rho_{ij})}$$

$$\sim Gamma\Big(\sum_{j=1}^{n}X_{ij},\sum_{j=1}^{n}\nu_j\phi_j\rho_{ij}\Big),\quad i=1,...,q_0$$

$$3:\ f(\delta_i|\rho_{ij}) \propto \prod_{j=1}^{n}\Big[f(\rho_{ij}|\delta_i)\Big]f(\delta_i)$$

$$\propto \prod_{j=1}^{n}\Big[\frac{\rho_{ij}^{1/\delta_i-1}e^{-\rho_{ij}/\delta_i}}{\delta_i^{1/\delta_i}\Gamma(1/\delta_i)}\Big]\delta_i^{a_\delta-1}e^{-\delta_i b_\delta}$$

$$= \Big(\prod_{j=1}^{n}\rho_{ij}\Big)^{1/\delta_i-1}e^{-(\sum_{j=1}^{n}\rho_{ij}/\delta_i)-\delta_i b_\delta}\ \delta_i^{a_\delta-1-n/\delta_i}\frac{1}{(\Gamma(1/\delta_i))^n}\ ,\quad i=1,...,q_0$$

$$4:\ f(k_j|X_{ij},\mu_i,\nu_j,\rho_{ij}) \propto \prod_{i=1}^{q_0}\Big[f(X_{ij}|k_j,\mu_i,\nu_j,\rho_{ij})\Big]f(k_j)$$

$$\propto \prod_{i=1}^{q_0}\Big[\phi_j^{X_{ij}}e^{-\mu_i\nu_j\phi_j\rho_{ij}}\Big]e^{-k_j^2/(2\sigma_k^2)} = \phi_j^{\sum_{i=1}^{q_0}X_{ij}}\ e^{-(\sum_{i=1}^{q_0}\mu_i\rho_{ij}\nu_j\phi_j)-k_j^2/(2\sigma_k^2)}$$

$$= \Big[\phi_0\frac{e^{k_j}}{\sum_{j=1}^{n}e^{k_j}}\Big]^{\sum_{i=1}^{q_0}X_{ij}}\ e^{-k_j^2/(2\sigma_k^2)-(\sum_{i=1}^{q_0}\mu_i\rho_{ij})\nu_j\,\phi_0\,e^{k_j}/(\sum_{j=1}^{n}e^{k_j})}\ ,\quad j=2,...,n$$

$$5:\ f(s_j|\nu_j,\theta) \propto f(\nu_j|s_j,\theta)f(s_j)$$

$$\propto \frac{e^{-\nu_j/(s_j\theta)}}{s_j^{1/\theta}}\ s_j^{a_s-1}\ e^{-s_j b_s}$$

$$= s_j^{a_s-1-1/\theta}\ e^{-\nu_j/(s_j\theta)-s_j b_s}\ ,\quad j=1,...,n$$

$$6:\ f(\theta|\nu_j,s_j) \propto \prod_{j=1}^{n}\Big[f(\nu_j|s_j,\theta)\Big]f(\theta)$$

$$\propto \prod_{j=1}^{n} \Big[\frac{\nu_j^{1/\theta} e^{-\nu_j/(s_j\theta)}}{(s_j\theta)^{1/\theta}\,\Gamma(1/\theta)} \Big]\, \theta^{a_\theta - 1}\, e^{-\theta b_\theta}$$

$$= \Big[\prod_{j=1}^{n} \frac{\nu_j}{s_j}\Big]^{1/\theta} e^{-(\sum_{j=1}^{n} \nu_j/(s_j\theta)) - \theta b_\theta}\, \theta^{a_\theta - 1 - n/\theta} \frac{1}{(\Gamma(1/\theta))^n}$$

At last, in order to derive the full-conditional distribution of ν_j, let us first condense the equation describing the model as follows:

$$X_{ij}|\mu_i, \phi_j, \nu_j, \rho_{ij} \overset{ind}{\sim} Poisson(\mu_i \nu_j [\phi_j \rho_{ij}]^{I_i}), \tag{3.10}$$

with I_i as in Equation 3.2.
Hence, the full-conditional distribution of ν_j is given by:

$$7: \; f(\nu_j|X_{ij}, \mu_i, \phi_j, \rho_{ij}, s_j, \theta) \propto \prod_{i=1}^{q} \Big[f(X_{ij}|\nu_j, \mu_i, \phi_j, \rho_{ij}) \Big] f(\nu_j|s_j, \theta)$$

$$\propto \prod_{i=1}^{q} \Big[\nu_j^{X_{ij}} e^{-\mu_i \nu_j [\phi_j \rho_{ij}]^{I_i}} \Big] \nu_j^{1/\theta - 1} e^{-\nu_j/(s_j\theta)}$$

$$= \nu_j^{(\sum_{i=1}^{q} X_{ij}) + 1/\theta - 1}\; e^{-\nu_j \Big[(\sum_{i=1}^{q} \mu_i [\phi_j \rho_{ij}]^{I_i}) + 1/(s_j\theta) \Big]}$$

$$\sim Gamma\Big(\sum_{i=1}^{q} X_{ij} + 1/\theta,\; \sum_{i=1}^{q} \mu_i [\phi_j \rho_{ij}]^{I_i} + \frac{1}{s_j\theta} \Big), \quad j = 1, ..., n$$

As one can see, the full-conditional distributions $1, 2, 7$ correspond to *Gamma* distributions, hence to known distributions from which one is able to sample. In such cases, the MCMC algorithm will use a Gibbs Sampling step (as previously described). However, the full-conditional distributions $3, 4, 5, 6$ are not known distributions, hence the Metropolis algorithm will be needed. More details about this and about the proposal distributions that will be chosen are in the next subsection.

3.4.3 Proposal distributions for the Metropolis algorithm

We have four types of parameters for which we need to define a proposal distribution; however, they can be split into two main cases:

- The k_j's, for which there is no domain restriction (they are defined over the real line);

- The δ_i's, s_j's and θ, which, on the contrary, can assume only positive values and present therefore a constraint in their domain;

The first case is the simpler one; indeed, we can use as proposal distribution a normal distribution whose mean is given by the previous value of the corresponding parameter, while its variance can be tuned in the algorithm, by verifying the proportion of acceptance of the proposed values. This is expressed by:

$$k_j^* \sim Normal(k_j^{b-1}, \sigma_1^2) \quad j = 2, ..., n,$$

where k_j^* is the proposed value for k_j^b and σ_1^2 is the variance that will be tuned.
The advantage of using as proposal of k_j^b a normal distribution with mean k_j^{b-1} is that this results in a density function that is symmetric with respect to k_j^* and k_j^{b-1}, leading to a simplification in the computation of Metropolis' rate of acceptance α. More in details, α will be computed as follows:

$$\alpha = min\left[1, \frac{f(k_j^*|z)q(k_j^{b-1}|k_j^*)}{f(k_j^{b-1}|z)q(k_j^*|k_j^{b-1})}\right],$$

where f is the distribution from which we want to sample (i.e. the full-conditional of k_j), depending on the other parameters represented by z, while q is the normal distribution used as proposal. Then, in this case we will have:

$$q(k_j^{b-1}|k_j^*) = (2\pi\sigma_1^2)^{-1/2} exp\Big(-\frac{(k_j^{b-1} - k_j^*)^2}{2\sigma_1^2}\Big)$$

$$q(k_j^*|k_j^{b-1}) = (2\pi\sigma_1^2)^{-1/2} exp\Big(-\frac{(k_j^* - k_j^{b-1})^2}{2\sigma_1^2}\Big)$$

The two expressions are the same, therefore they can be simplified from the ratio in the computation of α, leading to:

$$\alpha = min\left[1, \frac{f(k_j^*|z)}{f(k_j^{b-1}|z)}\right]$$

As it is evident, this simplifies the calculations and results in being extremely beneficial, especially when the expression of f is already cumbersome.
However, this solves the problem only for the k_j's, but one would want to have in some way the same benefit even for the other parameters that need Metropolis. Indeed, we cannot impose a normal distribution as proposal distribution for the δ_i's,

s_j's and θ, since the proposal distribution has to be defined over the same domain of the parameters. And in this case, while the normal distribution is defined over the real line, the parameters take values only over its positive portion.
To this purpose, a reparametrisation step comes into play. Let us define it for a generic parameter $\beta \in (0, +\infty)$. The following reparametrisation is defined:

$$\tau = log(\beta) \equiv t(\beta) \quad \longrightarrow \quad \beta = exp(\tau) \equiv g(\tau)$$

This relationship between the variables β and τ is such that β will always assume values greater than 0, hence respecting the domain in which it is defined, for every value assumed by $\tau \in R$. With this in mind, the solution to our problem is to use Metropolis on the parameter τ, instead of β, with the following proposal distribution:

$$\tau^* \sim Normal(\tau^{b-1}, \sigma_\tau^2),$$

leading to the same simplification of α:

$$\alpha = min\left[1, \frac{f_\tau(\tau^*|z)}{f_\tau(\tau^{b-1}|z)}\right]$$

One last issue needs to be taken care of, that is the expression of $f_\tau(\tau|z)$.
Indeed, this is computed as:

$$f_\tau(\tau) = f_\beta(g(\tau)) \, |\frac{\delta g(\tau)}{\delta \tau}| = f_\beta(\beta) \, exp(\tau) = f_\beta(\beta) \, \beta$$

Thanks to this reformulation, the Metropolis step is simplified for all parameters and, in conclusion, results in the following proposal distributions:

$$log(\delta_i^*) \sim Normal(log(\delta_i^{b-1}), \sigma_2^2) \quad i = 1, ..., q_0$$

$$log(s_j^*) \sim Normal(log(s_j^{b-1}), \sigma_3^2) \quad j = 1, ..., n$$

$$log(\theta^*) \sim Normal(log(\theta^{b-1}), \sigma_4^2),$$

with $\sigma_2, \sigma_3, \sigma_4$ to be tuned.
In our case, the values chosen for the MCMC algorithm are $\sigma_4 = 0.4$, $\sigma_3 = 0.6$, $\sigma_2 = 0.3$, $\sigma_1 = 0.03$, and they were tuned on the 25% of the data, before applying the algorithm to the whole dataset.

Chapter 4

Results

In this chapter, the computational results are discussed. We first depict the outcome given by the MCMC algorithm and compare it with the corresponding outcome reached in [13]; after that, we will focus on some additional analysis, proposing a modified version of the original approach.

4.1 MCMC algorithm results and comparison

Our MCMC algorithm is characterised by *burnin*= 500 and *thin*= 20. The total number of iterations needed corresponds to $N = 50000$, since the parameters ϕ_1 and ν_1 require a longer time for convergence. The following plots show the results obtained for some of the parameters, providing an additional zoom on the convergence region for those needing a higher number of iterations to converge. As one can see, all parameters, even if for ϕ_1 it is less smooth, reach convergence. Furthermore, the results obtained are in accordance with the ones obtained in [13]. This can be better verified by comparing the posterior median obtained in both cases. The comparison is depicted in the following images, where θ and the first six δ_i's, s_j's, μ_i's, ν_j's and ϕ_j's are considered. For each parameter are shown our results (on the left) in terms of quantiles of each variable, and the results obtained by the authors of [13] (on the right), where the numbers represent the posterior medians (50% quantiles) and the High Posterior Density intervals, containing 95% probability. As we can see, our 50% quantiles almost coincide with the posterior medians obtained by them; and also their 95% probability intervals (*lower* and *upper* columns) are fairly in accordance with our 2.5% and 97.5% quantiles, respectively. For the hyper-parameter θ, quantifying the strength of unexplained technical variability, we also depict (Figure 4.11) the posterior distribution histogram, that shows how the corresponding density function is roughly a bell curve with its mode around 0.4.

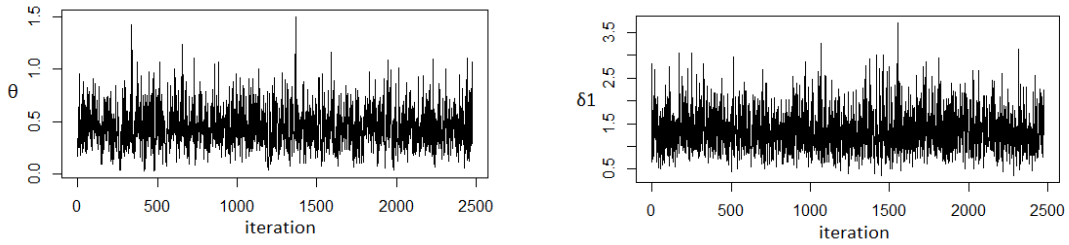

Figure 4.1: Traceplot of parameters θ and δ_1

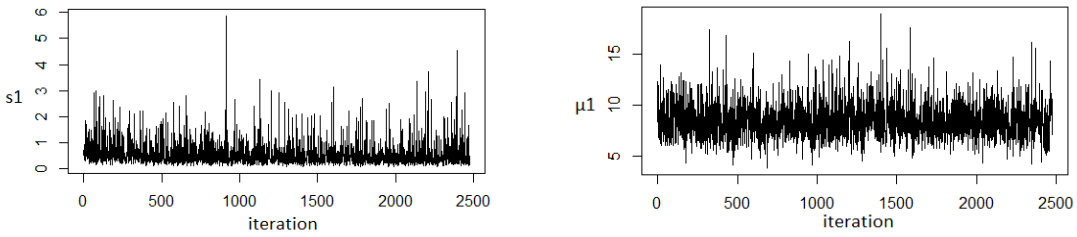

Figure 4.2: Traceplot of parameters s_1 and μ_1

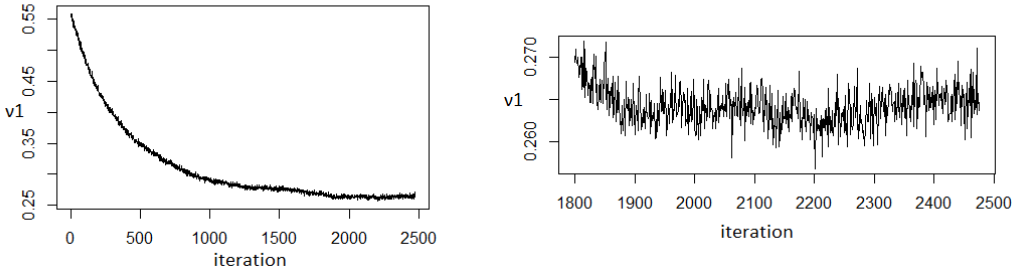

Figure 4.3: Traceplot of parameter ν_1

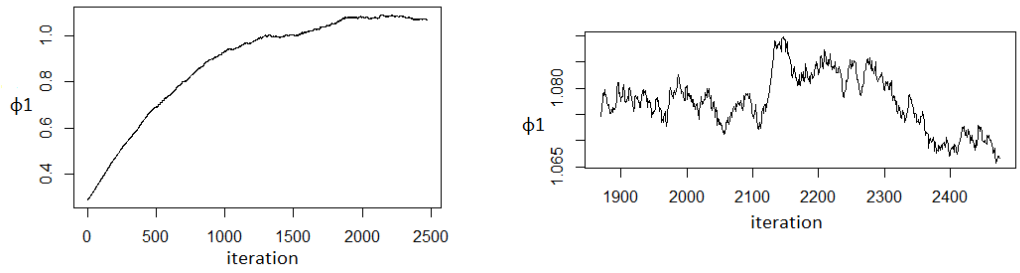

Figure 4.4: Traceplot of parameter ϕ_1

```
   Quantiles for each variable:

          2.5%    25%    50%    75%  97.5%
delta1 0.6368 1.0078 1.2735 1.5688 2.3674
delta2 0.4427 0.7083 0.8945 1.1009 1.6337
delta3 0.7114 1.1914 1.5244 1.9451 3.0022
delta4 0.7281 1.1948 1.4990 1.8717 2.8377
delta5 0.2911 0.4357 0.5342 0.6590 0.9539
delta6 0.1732 0.3128 0.4085 0.5207 0.8182
```

```
##          Delta     lower     upper
## var1 1.2357233 0.5687319 2.1534997
## var2 0.8934631 0.3701011 1.5468271
## var3 1.5197111 0.6129799 2.6846489
## var4 1.5245022 0.7231801 2.7465162
## var5 0.5316314 0.2601823 0.9470649
## var6 0.4084082 0.1619650 0.7947863
```

Figure 4.5: Results obtained for the δ_i's: this work (left) and [13] (right)

```
   Quantiles for each variable:

     2.5%    25%    50%    75% 97.5%
s1 0.1337 0.2748 0.4263 0.6896 1.906
s2 0.1310 0.2687 0.4044 0.6594 1.731
s3 0.1279 0.2594 0.3965 0.6439 1.843
s4 0.1264 0.2547 0.3849 0.6183 1.704
s5 0.1162 0.2404 0.3719 0.5815 1.541
s6 0.1245 0.2566 0.3863 0.6144 1.739
```

```
##              S      lower    upper
## var1 0.3812280 0.07777273 1.281444
## var2 0.4116489 0.07578549 1.342358
## var3 0.3877426 0.07452562 1.333508
## var4 0.3975925 0.08744929 1.383250
## var5 0.3756211 0.06693769 1.268793
## var6 0.3955919 0.07779230 1.309991
```

Figure 4.6: Results obtained for the s_j's: this work (left) and [13] (right)

Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
mu1	5.616	7.263	8.291	9.464	12.364
mu2	7.462	9.311	10.528	11.845	15.131
mu3	3.072	4.168	4.917	5.808	7.883
mu4	3.958	5.337	6.210	7.216	10.008
mu5	16.584	19.844	21.654	23.651	28.373
mu6	10.201	11.909	12.961	14.125	16.605

##	Mu	lower	upper
## var1	8.202876	5.139092	11.905359
## var2	10.667867	7.068047	14.456868
## var3	4.942627	2.955094	7.630980
## var4	6.144896	3.693061	9.256696
## var5	21.681496	16.290014	27.936334
## var6	12.838129	9.875983	16.443286

Figure 4.7: Results obtained for the μ_i's: this work (left) and [13] (right)

Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
nu1	0.2602	0.2627	0.2640	0.2656	0.2692
nu2	0.2878	0.2914	0.2936	0.2958	0.3002
nu3	0.2670	0.2710	0.2731	0.2753	0.2795
nu4	0.2696	0.2732	0.2753	0.2773	0.2812
nu5	0.2457	0.2488	0.2509	0.2530	0.2570
nu6	0.2738	0.2772	0.2792	0.2808	0.2846

##	Nu	lower	upper
## var1	0.2654422	0.2555881	0.2735322
## var2	0.2931209	0.2865417	0.2989842
## var3	0.2740734	0.2676038	0.2793736
## var4	0.2759257	0.2694894	0.2820111
## var5	0.2505074	0.2447978	0.2563018
## var6	0.2789405	0.2731178	0.2846139

Figure 4.8: Results obtained for the ν_j's: this work (left) and [13] (right)

Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
phi1	1.0678	1.0736	1.0774	1.0811	1.0879
phi2	1.1034	1.1199	1.1310	1.1406	1.1538
phi3	1.1338	1.1607	1.1717	1.1841	1.2056
phi4	1.0964	1.1140	1.1231	1.1327	1.1494
phi5	0.8272	0.8421	0.8514	0.8591	0.8722
phi6	1.0555	1.0676	1.0756	1.0840	1.1016

##	Phi	lower	upper
## var1	1.068651	1.0232189	1.1248908
## var2	1.132755	1.1004276	1.1613009
## var3	1.165209	1.1327453	1.1966068
## var4	1.120181	1.0919895	1.1512169
## var5	0.851961	0.8286403	0.8773731
## var6	1.075790	1.0482842	1.1064535

Figure 4.9: Results obtained for the ϕ_j's: this work (left) and [13] (right)

Let us now focus on the parameters s_j's. As previously said, they model the capture efficiency of each cell, hence one would expect for them to vary depending on j, since they represent a cell-specific characteristic. However, this fact does not seem to happen. Let us first have a look at the ϕ_j's. Indeed, as we can see in Figure 4.12, the boxplots representing the posterior distributions of those parameters suggest that

```
Quantiles for each variable:

          2.5%    25%    50%    75%  97.5%
theta 0.1159 0.3101 0.4190 0.5451 0.8541
```

```
##          Theta     lower     upper
## var1 0.4140722 0.1457055 0.8021185
```

Figure 4.10: Results obtained for θ: this work (left) and [13] (right)

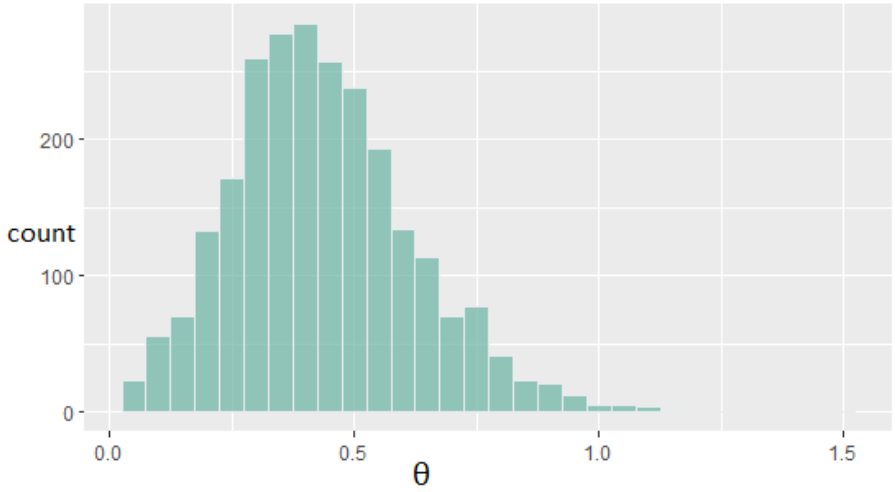

Figure 4.11: Posterior distribution histogram of parameter θ

there is a substantial heterogeneity in the total mRNA content per cell. This makes sense, since cell-specific measurements can vary in scale because of differences in total cellular mRNA content, and therefore this normalisation aspect needs to be taken into account for the performances of the model.
In the same perspective, capture efficiency parameters (s_j's) should also reflect a difference in their posterior distribution across cells, otherwise this would make the model unnecessarily complex. Still, both the results obtained by [13] and by our algorithm seem to confirm this contradiction. Indeed, taking a look at Figure 4.13 and 4.14, one can see the problem. Both figures represent the posterior distributions of the s_j's, Figure 4.13 through boxplots, Figure 4.14 in terms of medians and 95% probability percentiles. It is self-evident how the posterior distributions inferred for the s_j's seem to be substantially the same for every j. The boxplots are nearly completely overlapping along the y-axis in Figure 4.13, and the same can be seen more clearly for the posterior medians and 95% percentiles in Figure 4.14.
The lack of heterogeneity in the s_j's posterior distributions suggests that some further analysis needs to be done. Indeed, it seems that the current model's definition does not allow to correctly estimate and identify the s_j's. To this end, in the next section

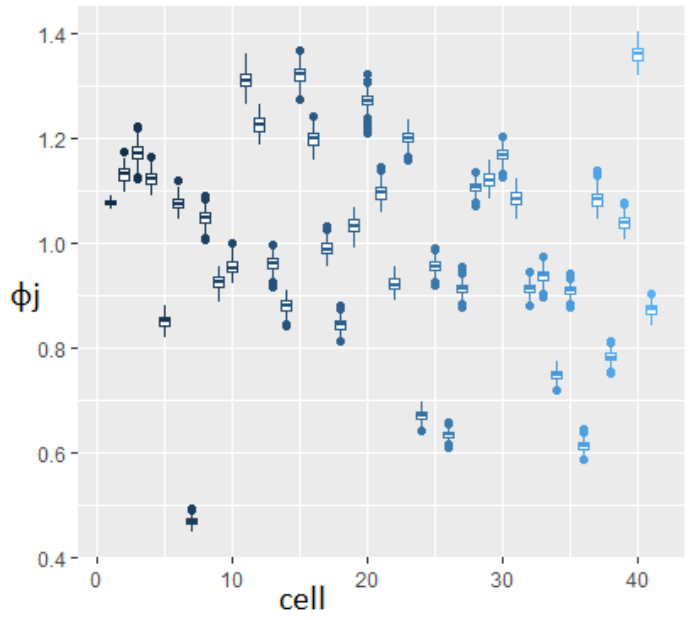

Figure 4.12: Posterior distributions of the ϕ_j's: boxplot

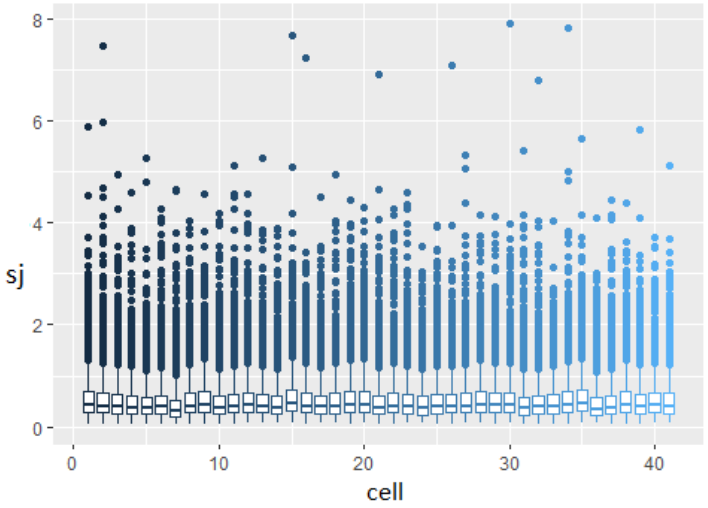

Figure 4.13: Posterior distributions of the s_j's: boxplot

we perform an additional analysis, in order to gain a better understanding of the situation.

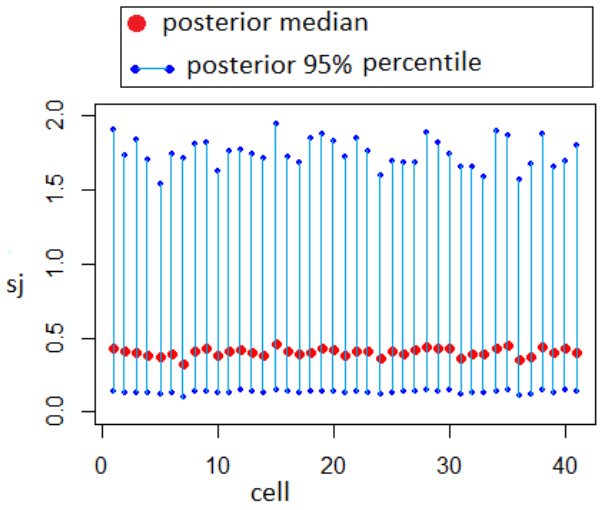

Figure 4.14: Posterior distributions of the s_j's: median and 95% percentile

4.2 Identifiability of capture efficiency parameters

In this section, we carry out an additional analysis on the capture efficiency parameters, the s_j's. For computational reasons, the following analysis will be performed on a 25% random stratified sample of the dataset, where the stratification relates to biological and spike-in genes.
As expected, without changing any previous assumption, the algorithm on the 25% of the dataset lead approximatively to the same results of Figure 4.13. Indeed, the posterior distribution boxplots of the s_j's are depicted in Figure 4.15 and are nearly the same for every j. However, this is not the only thing that catches the eye. In fact, by trying to represent through an histogram the posterior distribution of the s_j's, not only such distributions are all almost the same, but they also present substantially the shape of the prior distribution imposed. This is shown in Figure 4.16, where the posterior distributions of 4 among the n s_j's are represented through histograms. In each plot, such histogram is paired with the density of a $Gamma(1,1)$, i.e. the prior imposed for the capture efficiency parameters, that seems to define their shape.
In order to test this hypothesis, we decided to run the algorithm imposing a different prior on the s_j's. The prior distribution imposed is, in this case, a $Gamma(7,1)$, that results in a curve with a fairly different shape than the previous prior, thanks to the change in the first parameter value. The shape difference between the two prior distributions is visible in Figure 4.17.

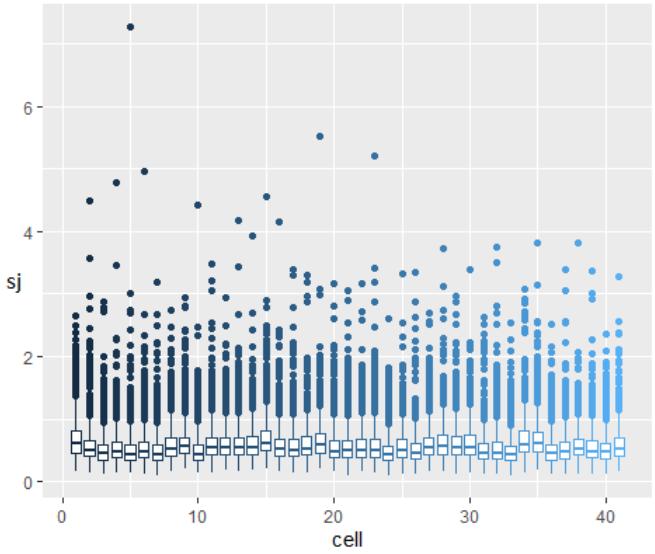

Figure 4.15: Posterior distribution boxplots of the s_j's: $Gamma(1,1)$ prior, 25% data

Hence, after imposing the second prior, one would expect to still find that the posterior distributions are roughly the same for every s_j (overlapping boxplots), but that this time such distributions take the shape of a $Gamma(7,1)$ (the current prior distribution imposed). This is exactly what we obtain: in Figure 4.18 the posterior distribution boxplots are nearly completely overlapping and in Figure 4.19 the resemblance between the posterior distributions and the prior imposed is undeniable.
By taking a closer look to the model's definition, the reason behind this behaviour seems to be more clear. It appears that there is some kind of identifiability issue, given by the relationship that links each ν_j to each s_j. Indeed, when constructing the posterior (full-conditional) distribution of s_j, there are two parts that contribute to its definition: the likelihood of ν_j and the prior distribution of s_j. However, the likelihood is based only on one ν_j, that is, on one single data. This means that, when updating the posterior distribution of s_j at each iteration of the MCMC, one takes into account the contribution given by the prior distribution and the one given by a single data (ν_j). This is why the s_j's, in the end, look all alike and are completely determined by the prior distribution: one ν_j is not enough to contrast the weight given by the prior when updating the posterior distribution of each s_j. To this purpose, in the next section we present a modification of the model that solves the problem.

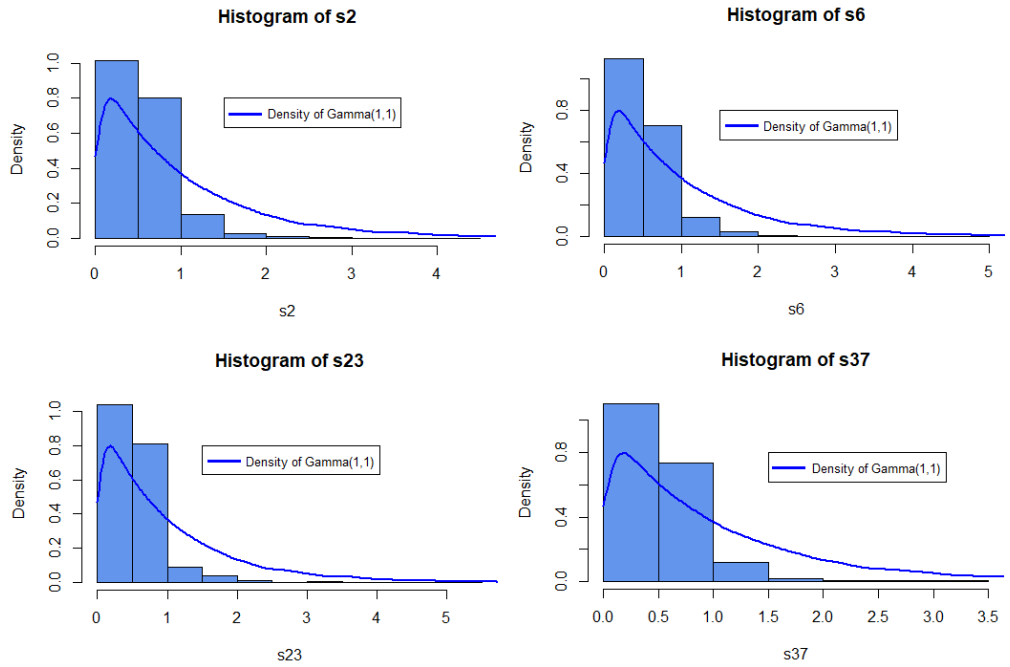

Figure 4.16: Posterior distribution histograms of the s_j's: *Gamma*(1,1) prior, 25% data

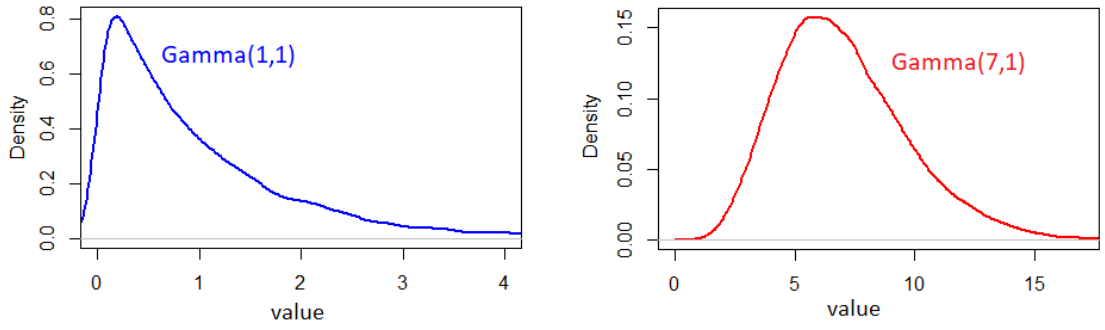

Figure 4.17: Shape difference between priors: *Gamma*(1,1) (blue) and *Gamma*(7,1) (red)

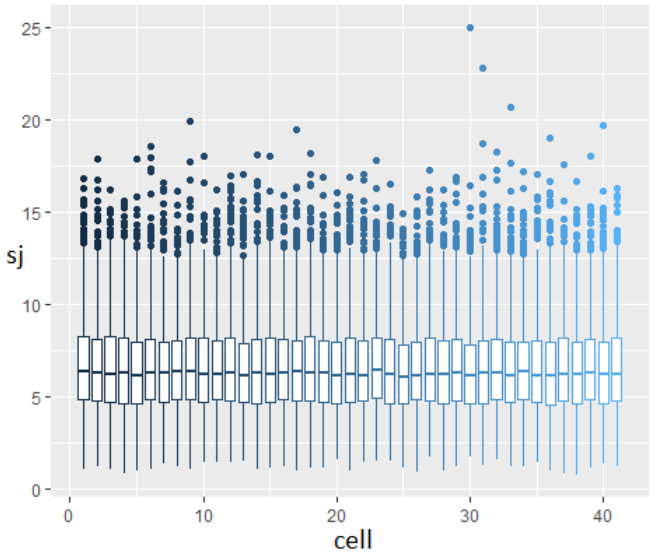

Figure 4.18: Posterior distribution boxplots of the s_j's: *Gamma*(7,1) prior, 25% data

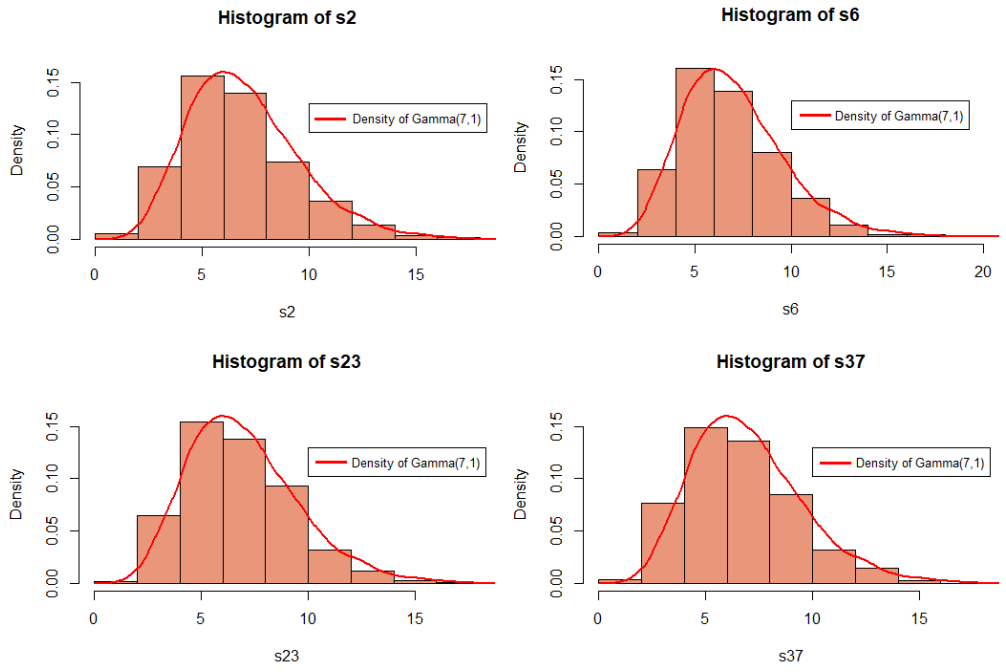

Figure 4.19: Posterior distribution histograms of the s_j's: *Gamma*(7,1) prior, 25% data

4.3 Modification of the model

The proposed modified version of the model involves the definition of the random effect described by ν. That is, instead of making it only cell-dependent, we add a dependence on the gene, hence having to deal with the ν_{ij}'s instead of the ν_j's. The model definition is the same as in Equation 3.1, apart from the characterisation of ν, that is the following:

$$\nu_{ij} \overset{ind}{\sim} Gamma(\frac{1}{\theta}, \frac{1}{s_j\theta})$$

Naturally, this leads to a modification in the expression of the full-conditional distributions, that are now defined as below.

$$1: \ f(\rho_{ij}|X_{ij}, \mu_i, \nu_{ij}, \phi_j, \delta_i) \propto f(X_{ij}|\rho_{ij}, \mu_i, \nu_{ij}, \phi_j) f(\rho_{ij}|\delta_i)$$
$$\propto \rho_{ij}^{X_{ij}} e^{-\mu_i\nu_{ij}\phi_j\rho_{ij}} \rho_{ij}^{1/\delta_i - 1} e^{-\rho_{ij}/\delta_i} \propto \rho_{ij}^{X_{ij}+1/\delta_i-1} e^{-\rho_{ij}(\mu_i\nu_{ij}\phi_j+1/\delta_i)}$$
$$\sim Gamma\Big(X_{ij} + \frac{1}{\delta_i}, \mu_i\nu_{ij}\phi_j + \frac{1}{\delta_i}\Big), \quad i = 1, ..., q_0, j = 1, ..., n$$

$$2: \ f(\mu_i|X_{ij}, \nu_{ij}, \phi_j, \rho_{ij}) \propto \prod_{j=1}^{n} \Big[f(X_{ij}|\mu_i, \nu_{ij}, \phi_j, \rho_{ij})\Big] f(\mu_i)$$
$$\propto \prod_{j=1}^{n} \Big[\mu_i^{X_{ij}} e^{-\mu_i\nu_{ij}\phi_j\rho_{ij}}\Big] \frac{1}{\mu_i} = \mu_i^{\sum_{j=1}^{n} X_{ij} - 1} e^{-\mu_i(\sum_{j=1}^{n} \nu_{ij}\phi_j\rho_{ij})}$$
$$\sim Gamma\Big(\sum_{j=1}^{n} X_{ij}, \sum_{j=1}^{n} \nu_{ij}\phi_j\rho_{ij}\Big), \quad i = 1, ..., q_0$$

$$3: \ f(\delta_i|\rho_{ij}) \propto \prod_{j=1}^{n} \Big[f(\rho_{ij}|\delta_i)\Big] f(\delta_i)$$
$$\propto \prod_{j=1}^{n} \Big[\frac{\rho_{ij}^{1/\delta_i - 1} e^{-\rho_{ij}/\delta_i}}{\delta_i^{1/\delta_i}\Gamma(1/\delta_i)}\Big] \delta_i^{a_\delta - 1} e^{-\delta_i b_\delta}$$
$$= \Big(\prod_{j=1}^{n} \rho_{ij}\Big)^{1/\delta_i - 1} e^{-(\sum_{j=1}^{n} \rho_{ij}/\delta_i) - \delta_i b_\delta} \ \delta_i^{a_\delta - 1 - n/\delta_i} \frac{1}{(\Gamma(1/\delta_i))^n}, \quad i = 1, ..., q_0$$

$$4:\ f(k_j|X_{ij},\mu_i,\nu_{ij},\rho_{ij}) \propto \prod_{i=1}^{q_0}\Big[f(X_{ij}|k_j,\mu_i,\nu_{ij},\rho_{ij})\Big]f(k_j)$$

$$\propto \prod_{i=1}^{q_0}\Big[\phi_j^{X_{ij}} e^{-\mu_i\nu_{ij}\phi_j\rho_{ij}}\Big]e^{-k_j^2/(2\sigma_k^2)} = \phi_j^{\sum_{i=1}^{q_0} X_{ij}} e^{-(\sum_{i=1}^{q_0}\mu_i\rho_{ij}\nu_{ij}\phi_j)-k_j^2/(2\sigma_k^2)}$$

$$= \Big[\phi_0\frac{e^{k_j}}{\sum_{j=1}^{n} e^{k_j}}\Big]^{\sum_{i=1}^{q_0} X_{ij}}\ e^{-k_j^2/(2\sigma_k^2)-(\sum_{i=1}^{q_0}\mu_i\rho_{ij}\nu_{ij})\,\phi_0\, e^{k_j}/(\sum_{j=1}^{n} e^{k_j})},\quad j=2,...,n$$

$$5:\ f(s_j|\nu_{ij},\theta) \propto f(\nu_{ij}|s_j,\theta)f(s_j)$$

$$\propto \prod_{i=1}^{q}\Big[\frac{e^{-\nu_{ij}/(s_j\theta)}}{s_j^{1/\theta}}\Big]\ s_j^{a_s-1}\ e^{-s_jb_s}$$

$$=\ e^{-(\sum_{i=1}^{q}\nu_{ij}/(s_j\theta))-s_jb_s}\ s_j^{a_s-1-q/\theta},\quad j=1,...,n$$

$$6:\ f(\theta|\nu_{ij},s_j) \propto \prod_{j=1}^{n}\prod_{i=1}^{q}\Big[f(\nu_{ij}|s_j,\theta)\Big]f(\theta)$$

$$\propto \prod_{j=1}^{n}\prod_{i=1}^{q}\Big[\frac{\nu_{ij}^{1/\theta}e^{-\nu_{ij}/(s_j\theta)}}{(s_j\theta)^{1/\theta}\,\Gamma(1/\theta)}\Big]\ \theta^{a_\theta-1}\ e^{-\theta b_\theta}$$

$$= \Big[\prod_{j=1}^{n}\prod_{i=1}^{q}\nu_{ij}\Big]^{1/\theta}\ e^{-(\sum_{j=1}^{n}\sum_{i=1}^{q}\nu_{ij}/(s_j\theta))-\theta b_\theta}\ \theta^{a_\theta-1-n*q/\theta}\ \frac{1}{(\Gamma(1/\theta))^{n*q}}\ [\prod_{j=1}^{n} s_j]^{-q/\theta}$$

$$7:\ f(\nu_{ij}|X_{ij},\mu_i,\phi_j,\rho_{ij},s_j,\theta) \propto f(X_{ij}|\nu_{ij},\mu_i,\phi_j,\rho_{ij})f(\nu_{ij}|s_j,\theta)$$

$$\propto \nu_{ij}^{X_{ij}}\ e^{-\mu_i\nu_{ij}[\phi_j\rho_{ij}]^{I_i}}\ \nu_{ij}^{1/\theta-1}\ e^{-\nu_{ij}/(s_j\theta)}$$

$$= \nu_{ij}^{X_{ij}+1/\theta-1}\ e^{-\nu_{ij}\Big[\mu_i[\phi_j\rho_{ij}]^{I_i}+1/(s_j\theta)\Big]}$$

$$\sim\ Gamma\Big(X_{ij}+1/\theta,\ \ \mu_i[\phi_j\rho_{ij}]^{I_i}+\frac{1}{s_j\theta}\Big),\quad i=1,...,q, j=1,...,n$$

Keeping the original prior distributions, if we run the new MCMC algorithm with the full-conditional distributions as above, we can notice how the results change (for the implemented R script see Appendix A). In Figure 4.20, the boxplots representing the

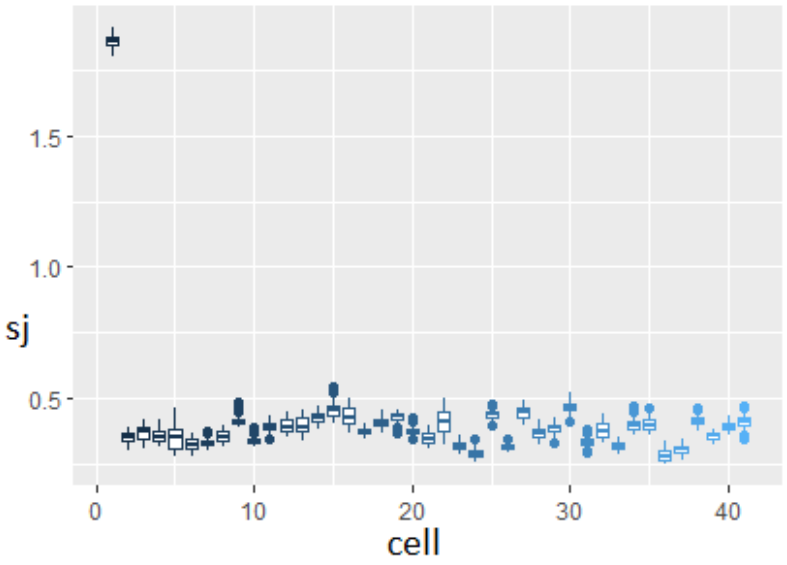

Figure 4.20: Posterior distributions of $s_1, ..., s_n$: boxplot

posterior distributions of the s_j's are depicted. This time, even if the s_j's oscillate around similar values apart from s_1 (that needs more iterations to converge), we can see a difference in their posterior distributions, both in term of the posterior median value and especially in the shape of the distribution. This is highlighted in Figure 4.21, where s_1 is excluded from the plot, hence the variations among the other s_j's are more evident. Furthermore, the posterior distribution histograms of the four s_j's previously considered can be also drawn (see Figure 4.22), in order to show how this time the histograms are not all significantly alike, and especially how they do not resemble to the prior distribution imposed ($Gamma(1, 1)$).
This is not the only difference in the results that one can observe. Indeed, regarding for example parameter θ, this time its posterior distribution histogram is centered around much smaller values than before. With the original approach, its mode was around 0.4, while now it is around 0.127. Furthermore, the distribution is condensed in a smaller range of values and the bell curve appears lower and more compact.
Concerning the parameter μ_i's (for the biological genes), we can also remark a difference in the corresponding obtained values (we are considering the posterior distribution median). This is shown in Figure 4.24, where the μ_i's obtained with the original approach are plotted against the μ_i's obtained with the new model. The bisector line (red) is also drawn, in order to better point out the difference in the values depicted. As we can see, the values obtained with the original model tend to be larger

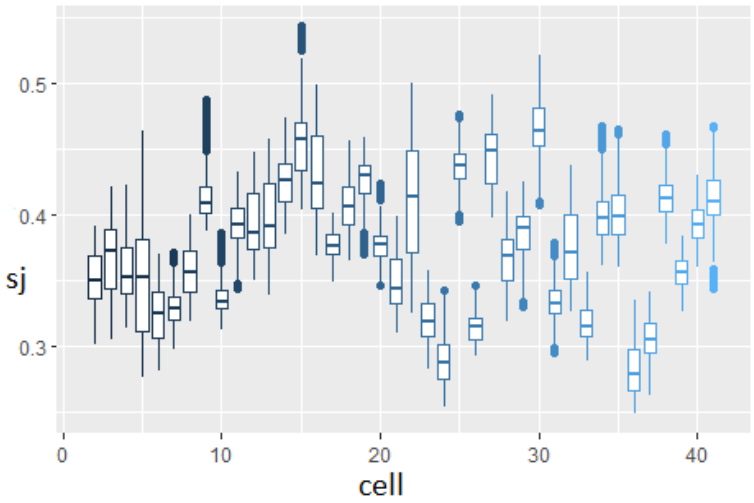

Figure 4.21: Posterior distributions of $s_2, ..., s_n$: boxplot

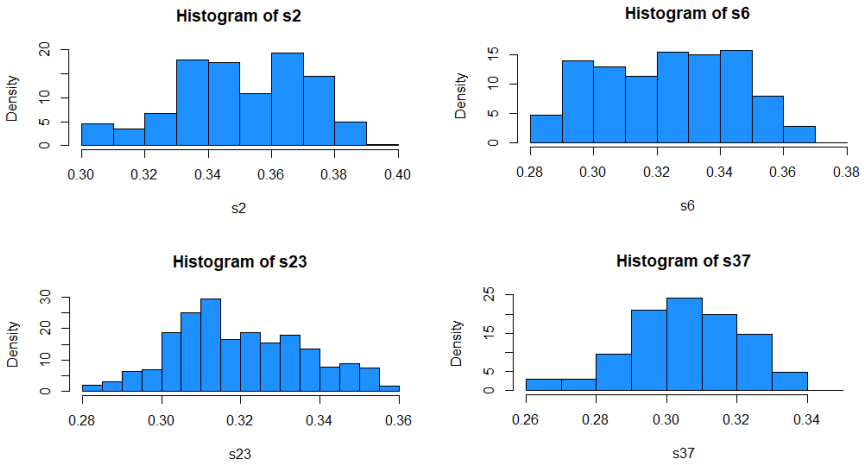

Figure 4.22: Posterior distributions histograms of s_2, s_6, s_{23}, s_{37}

than the ones achieved by the new approach. Indeed, if we consider the quantity $diff_{\mu_i} = \mu_{i,\nu_j} - \mu_{i,\nu_{ij}}$, it is possible to depict its boxplot, that provides a quantitative evaluation of such difference (see Figure 4.25). For visualisation purposes, the values of $diff_{\mu_i}$ that are extremely large and far from the median are not included in the plot.
In light of such differences between the two models, the classification of highly and lowly variable genes will probably vary. Indeed, one would expect that with the

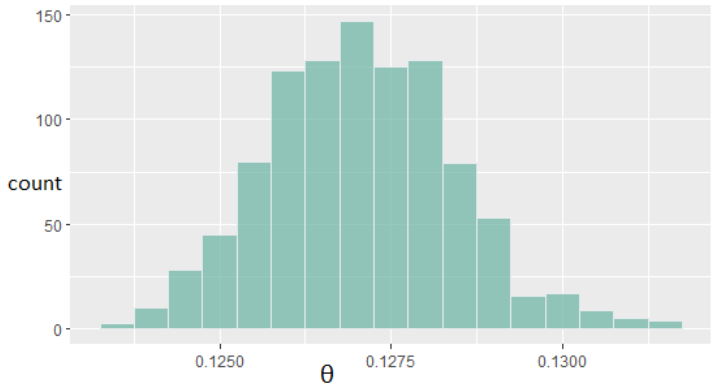

Figure 4.23: Posterior distribution histogram of parameter θ with the new model

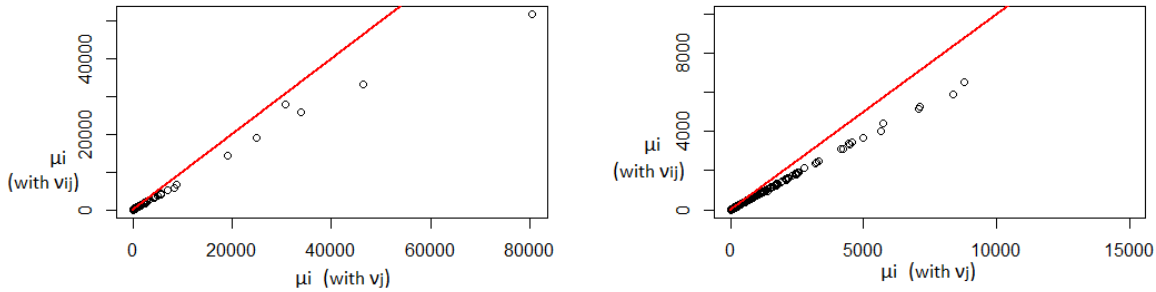

Figure 4.24: Comparison between μ_i's in the two models

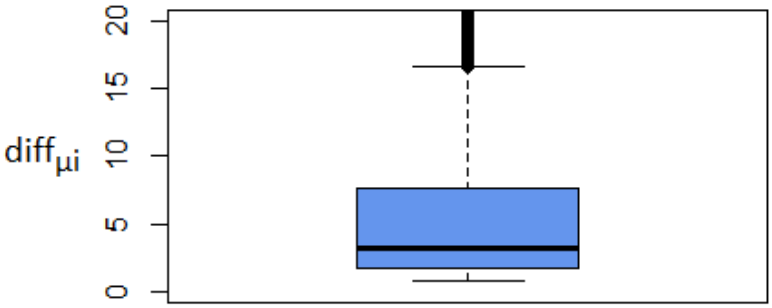

Figure 4.25: Difference between μ_i's in the two models: boxplot

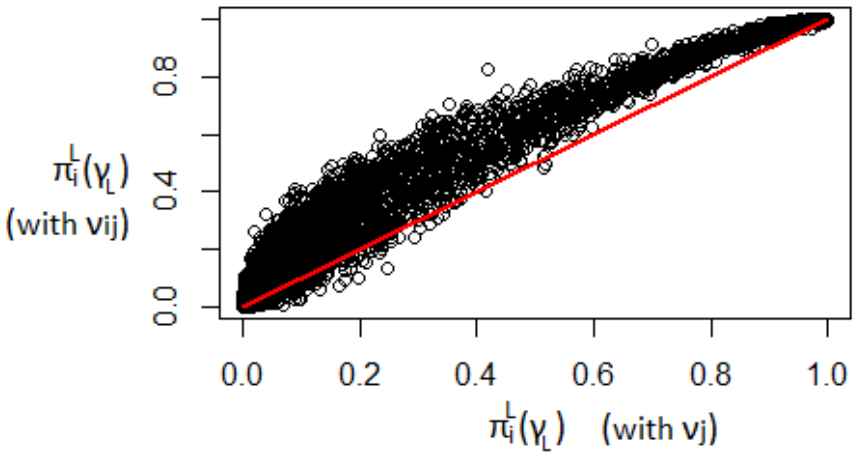

Figure 4.26: Comparison between the probabilities of being LVG in the two models

original approach we are, actually, neglecting a part of the technical variance, therefore obtaining a larger amount of biological heterogeneity. On the contrary, with the new model we explain a part of this variance through the s_j's, hence expecting to find less highly variable genes. To verify this assumption, we decided to compute, for both models, the quantities $\pi_i^H(\gamma_H)$ and $\phi_i^L(\gamma_L)$, defined in Equations 3.8 and 3.9. As previously stated, such quantities define the evidence in favour of a gene being highly or lowly variable, respectively. By choosing $\gamma_H = 0.79, \gamma_L = 0.41, \alpha_H = 0.7925$ and $\alpha_L = 0.7650$ as in [13], we can plot the values of $\pi_i^H(\gamma_H)$ obtained with the original model, versus the same values obtained with the new approach. The same can be done for $\phi_i^L(\gamma_L)$ (see Figures 4.27 and 4.26). As one can see from the plots, Figure 4.26 validates our expectations: setting a value for the threshold α_L, there are much more genes that are considered as LVG with the new approach, than those that were considered LVG with the original model. Analogously, the number of HVG seems to be higher with the old model, with respect to those found with the new one, corroborating the hypothesis that the new model would lead to the estimation of smaller values for the biological heterogeneity component. In addition, this last image reveals some surprising information. Setting a value for the thereshold α_H, we can see that, apart from a number of genes that are HVG for both models, some of the genes that result in being HVG for one model, are actually not considered as HVG for the other model and viceversa. This can be better seen if we take a look at the following images. Indeed, by comparing Figures 4.28 and 4.29, first, and then Figures 4.30 and 4.31, the comparison between LVG and HVG in the two models should be more clear. The first two images depict the values of parameters μ_i's (log scale of posterior

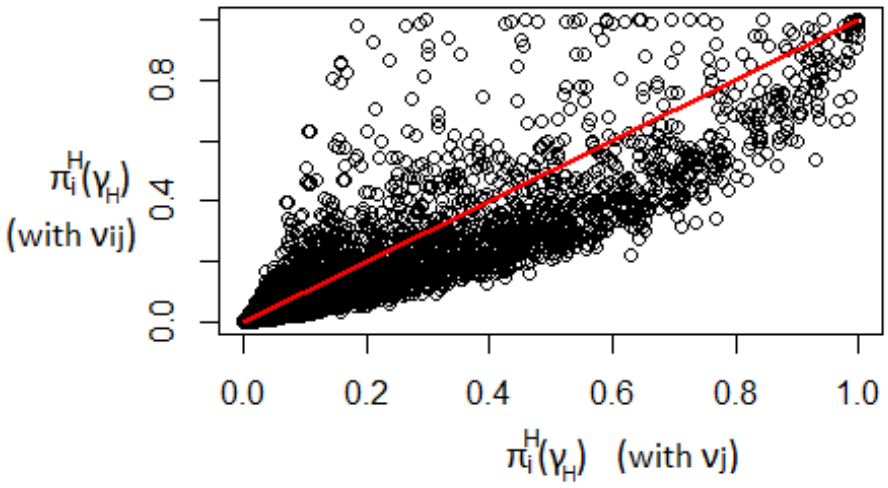

Figure 4.27: Comparison between the probabilities of being HVG in the two models

median) against the evidence for a gene in favour of being LVG, separately for the two models. As one can see, with the new approach (Figure 4.29), the blue region, representing the LVG is much denser, especially in the top left region. This confirms what previously already pointed out, that is that with the new model we explain a larger part of the variability as technical (not biological), hence resulting in a larger number of LVG. In the same perspective, Figures 4.30 and 4.31 plot the values of parameters μ_i's against the evidence for a gene in favour of being HVG and show how, for the new model, the red region representing the HVG appears more sparse, hence indicating a smaller number of genes identified as HVG.
Quantitatively speaking, with the old model we identify 641 LVG and 131 HVG (589 and 133 in [13]), while with the new approach the number of LVG rises to 862 and the number of HVG goes down to 110. Concerning HVG, some more details are in Table 4.1. As one can see, of the 110 genes classified as HVG with the new model, only 59 correspond to HVG with the old one. This means that approximately only a half (53.6%) of the HVG identified with the new approach is also identified with the original approach, while the remaining part of the HVG (51 genes, 46.4%) are exclusively detected by the new model. Furthermore, the amount of common HVG decreases if we look at it with respect to the old model, where they correspond only to 59 out of 131, resulting in a percentage of the 45%.

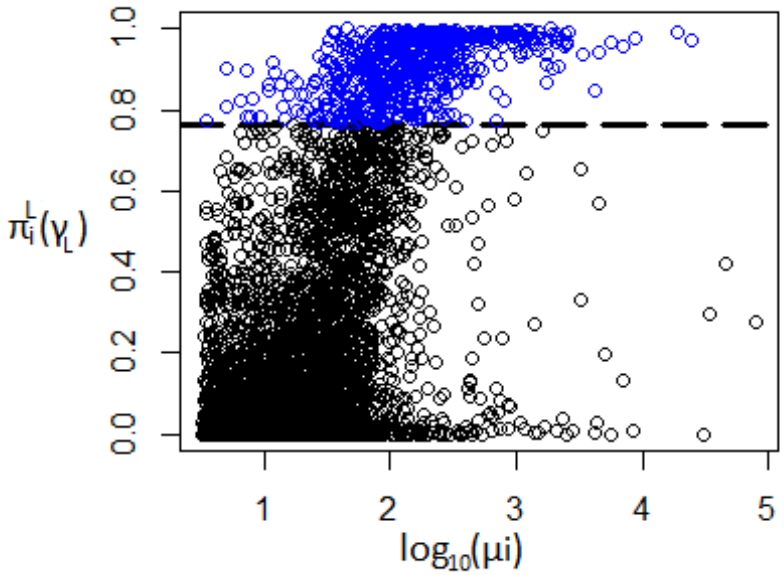

Figure 4.28: Detection of LVG: original model

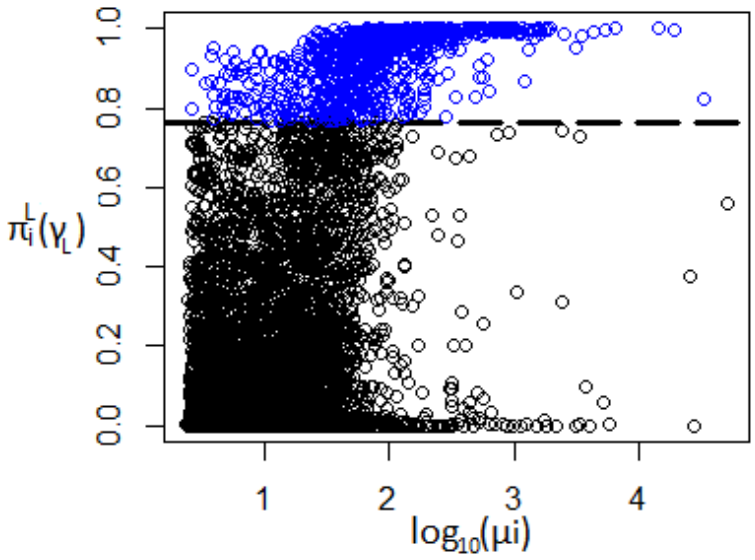

Figure 4.29: Detection of LVG: new model

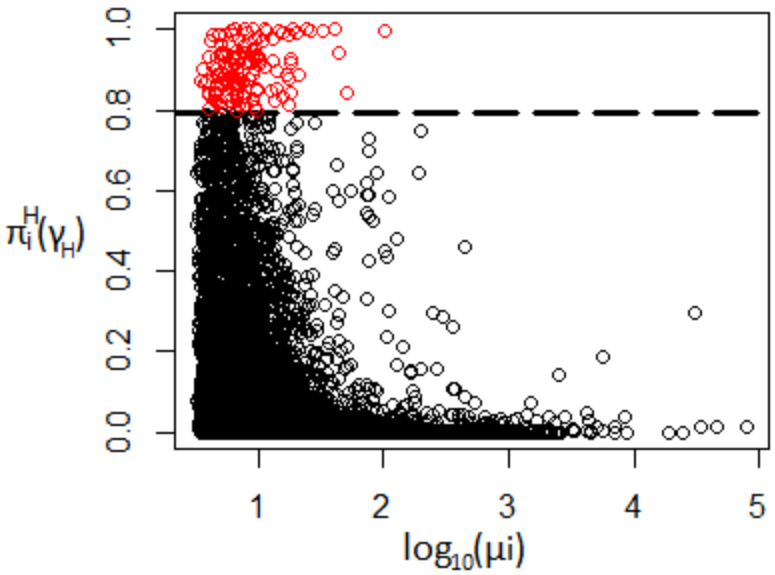

Figure 4.30: Detection of HVG: original model

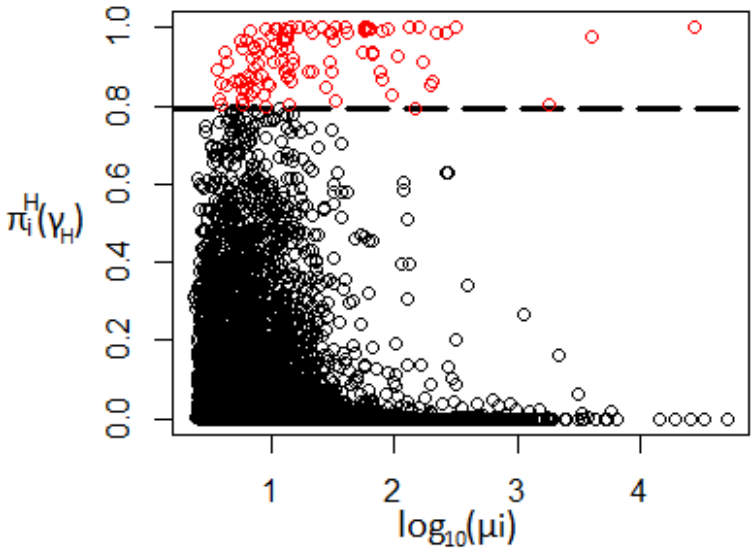

Figure 4.31: Detection of HVG: new model

	Model with ν_j's	Model with ν_{ij}'s
Total number of HVG	131	110
Number of common HVG	59	59
Number of characteristic HVG	72	51
Percentage of common HVG	45%	53.6%
Percentage of characteristic HVG	55%	46.4%

Table 4.1: Results on HVG for the two models

Chapter 5

Conclusions

This work has stemmed from the study conducted in [13]. Its focus can be intended as divided into two main parts: firstly, the building of an MCMC algorithm that could provide results in accordance with those already found in literature; and secondly, the modification of the proposed model in order to overcome its limits.
More in detail, given the considered bayesian model, with the use of both Gibbs Sampling and Metropolis, we have built an MCMC algorithm to infer the posterior distributions of the parameters. Such distributions, and in particular the posterior medians, have been used as representative of the parameters in the variance decomposition criterium employed for the detection of HVG and LVG. Specifically, the parameters of the model aim at explaining the observed variability, hence taking into account both the technical component (introduced by the experiments) and the biological one, which we are interested in. Therefore, by decomposing the observed variance, it is possible to learn how much it is related to biological factors. Biological heterogeneity of gene expression is, indeed, a key aspect for ranking the genes based on their variability, thus detecting the ones that genuinely present more biological variation across cells than expected by chance.
However, while analysing the considered model, this work has detected some issues regarding the identifiability of the estimates of the so-called capture efficiency parameters. Such parameters refer to the ability of capturing single cells during the mRNA sequencing phase and, hence, are directly linked to the technical noise introduced by the experiments. In particular, what we have noticed with the original model is that the cell-specific capture efficiency parameters were leading to posterior distributions, not only significantly alike for every cell, but also completely determined by the prior distribution imposed in the algorithm. To this end, after carrying out some tests that have demonstrated our findings, we have proposed a modification of the original model. This modification consists in changing the dependence of the random effect ν (related to technical noise), making it, not only cell-, but also gene-

dependent, thus resulting in ν_{ij} instead of ν_j. Since the random effects ν_{ij}'s depend on the capture efficiency parameters, this adjustment has an impact on their identifiability, solving the problem encountered with the original model. Furthermore, after building the MCMC algorithm related to the new model, we have observed how the conclusions on the correct classification of HVG and LVG differ. As expected, since with the original model we were actually neglecting a part of the variance related to technical noise, with the new model, the number of detected LVG (presenting low biological variability) has increased. Simultaneously, the number of HVG has decreased for the same reason, going from 131 to 110. Moreover, we have noticed how, not only the HVG have decreased, but some of them have also changed, from one model to another. Indeed, only 59 (53.6%) of the genes classified as HVG with the new model corresponds to HVG detected with the original one, showing how the current modification of the model leads to significantly different outcomes.

Appendix A

Implemented R script

```
#Script to use once one has, from the pre-processing step
#described in [13], the following data structures:
# X=CountsQC
# mu_noti=SpikesInputQC
# genetype=TechQC


iterations=40000
burnin=500
thin=40
phi0=n
sigma2k=1
as=1
bs=1
adelta=1
bdelta=1
atheta=1
btheta=1

#variables to monitor the acceptance rate with Metropolis
#(hence to tune the variance of the normal proposal distributions)
theta_acc=0
s_acc=0
delta_acc=0
k_acc=0
```

```
N = floor((iterations-burnin)/thin) #number of samples saved

#matrices to memorise the results
theta = matrix(NA, ncol=1,nrow=N)
v = matrix(NA, ncol=1,nrow=N)
rho = matrix(NA, ncol=1,nrow=q0)
delta = matrix(NA, ncol=q0,nrow=N)
mu = matrix(NA, ncol=q,nrow=N)
k = matrix(NA, ncol=n,nrow=N)
phi = matrix(NA, ncol=n,nrow=N)
s = matrix(NA, ncol=n,nrow=N)


#initial conditions
theta_start=0.2
v_start=matrix(0.5,ncol=n,nrow=q)
s_start=matrix(1,ncol=n,nrow=1)
k_start=matrix(1,ncol=n,nrow=1)
k_start[1]=0
delta_start=matrix(0.5,ncol=q0,nrow=1)
mu_start=matrix(1,ncol=q,nrow=1)
mu_start[(q0+1):q]=mu_noti
rho_start=matrix(1/2,ncol=n,nrow=q0)

#allocation of the initial conditions:
#they are memorised in the variables that will contain
#the current parameter values during the algorithm
theta_c=theta_start #scalar
v_c=v_start #matrix q*n
delta_c=delta_start #vector 1*q0
mu_c=mu_start #vector 1*q, with the elements from q0+1 to q
              #equal to mu_noti
k_c=k_start #vector 1*n with first element = 0
phi_c=matrix(NA,ncol=n,nrow=1)
for (j in 1:n) {
  phi_c[j]=phi0*exp(k_c[j])/sum(exp(k_c))
}
s_c=s_start #vector 1*n
rho_c=rho_start #matrix q0*n
```

```
a=matrix(2,ncol=n,nrow=q-q0) #matrix (q-q0)*n, used later to
                             #adjust the dimensions

M=burnin

for(i1 in 1:N)
{
  for(i2 in 1:M)
  {
    #sample theta:
    theta_prop=exp(rnorm(1,log(theta_c),0.01))
    somma1=0
    for (i in 1:q) {
      somma1=somma1+sum(v_c[i,]/s_c)
    }
    LogNum=(atheta-1-(n*q)/theta_prop)*log(theta_prop)+
      1/theta_prop*sum(log(v_c))-theta_prop*btheta-
      (n*q)*lgamma(1/theta_prop)-somma1/theta_prop-
      q/theta_prop*sum(log(s_c))+log(theta_prop)
    LogDen=(atheta-1-(n*q)/theta_c)*log(theta_c)+
      1/theta_c*sum(log(v_c))-theta_c*btheta-
      (n*q)*lgamma(1/theta_c)-somma1/theta_c-
      q/theta_c*sum(log(s_c))+log(theta_c)
    Alpha = min(1,exp(LogNum - LogDen))
    # we decide if we accept:
    u_th = runif(1,0,1)
    if(u_th<Alpha) {
      theta_c = theta_prop
      theta_acc=theta_acc+1
    }

    for (j in 1:n) {
      #sample s:
      s_prop=exp(rnorm(1,log(s_c[j]),0.05))
      LogNum=-sum(v_c[,j])/(s_prop*theta_c)-s_prop*bs+
        (as-1-q/theta_c)*log(s_prop)+log(s_prop)
      LogDen=-sum(v_c[,j])/(s_c[j]*theta_c)-s_c[j]*bs+
        (as-1-q/theta_c)*log(s_c[j])+log(s_c[j])
      Alpha = min(1,exp(LogNum - LogDen))
```

```
  # we decide if we accept:
  u_s = runif(1,0,1)
  if(u_s<Alpha) {
    s_c[j] = s_prop
    s_acc=s_acc+1
  }

  #sample v:
  rho_use=rbind(rho_c,a)
  v_c[,j]=rgamma(nrow(v_c),shape=X[,j]+1/theta_c,
                 rate=mu_c[1,]*(phi_c[j]*rho_use[,j])^(1-genetype)+
                   1/(s_c[j]*theta_c))
}

#sample k:
k_c[1]=0
for (j in 2:n) {
  k_prop=rnorm(1,k_c[j],0.01)
  somma_k_1=sum(exp(k_c))-exp(k_c[j])+exp(k_prop)
  somma_k_2=sum(exp(k_c))
  LogNum=sum(X[1:q0,j])*log(phi0*exp(k_prop)/somma_k_1)-
    k_prop*k_prop/(2*sigma2k)-
    sum(mu_c[1:q0]*rho_c[,j]*v_c[1:q0,j])*phi0*
    exp(k_prop)/somma_k_1
  LogDen=sum(X[1:q0,j])*log(phi0*exp(k_c[j])/somma_k_2)-
    k_c[j]*k_c[j]/(2*sigma2k)-
    sum(mu_c[1:q0]*rho_c[,j]*v_c[1:q0,j])*phi0*
    exp(k_c[j])/somma_k_2
  Alpha = min(1,exp(LogNum - LogDen))
  # we decide if we accept
  u_k = runif(1,0,1)
  if(u_k<Alpha) {
    k_c[j] = k_prop
    k_acc=k_acc+1
  }
}

#transformation with phi:
for (j in 1:n) {
```

```
    phi_c[j]=phi0*exp(k_c[j])/sum(exp(k_c))
  }

  for (i in 1:q0) {
    #sample mu[1:q0]:
    mu_c[i]=rgamma(1,shape=sum(X[i,]),
                   rate=sum(v_c[i,]*phi_c*rho_c[i,]))

    #sample delta:
    delta_prop=exp(rnorm(1,log(delta_c[i]),1))
    LogNum=(1/delta_prop-1)*sum(log(rho_c[i,]))-
      sum(rho_c[i,])/delta_prop-delta_prop*bdelta+
      (adelta-1-n/delta_prop)*log(delta_prop)-
      n*lgamma(1/delta_prop)+log(delta_prop)
    LogDen=(1/delta_c[i]-1)*sum(log(rho_c[i,]))-
      sum(rho_c[i,])/delta_c[i]-delta_c[i]*bdelta+
      (adelta-1-n/delta_c[i])*log(delta_c[i])-
      n*lgamma(1/delta_c[i])+log(delta_c[i])
    Alpha = min(1,exp(LogNum - LogDen))
    # we decide if we accept
    u_d = runif(1,0,1)
    if(u_d<Alpha) {
      delta_c[i] = delta_prop
      delta_acc=delta_acc+1
    }

    #sample rho:
    rho_c[i,]=rgamma(ncol(rho_c),shape=X[i,]+1/delta_c[i],
                     rate=mu_c[i]*v_c[i,]*phi_c[1,]+1/delta_c[i])
  }
  mu_c[(q0+1):q]=mu_noti
}
M=thin

#we save the current values
theta[i1]=theta_c
s[i1,]=s_c
v[i1]=v_c[1,1] #only for one v
rho[i1]=rho_c[1,1] #only for one rho
```

```
    k[i1,]=k_c
    phi[i1,]=phi_c
    delta[i1,]=delta_c
    mu[i1,]=mu_c
}
```

Bibliography

[1] Andrews Tallulah S., Kiselev Vladimir Yu, McCarthy Davis, Hemberg Martin, "Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data", Nature Protocols 16, 1–9, 2021

[2] Cain Michael L., Wassermant Steven A., Minorsky Peter V., Jackson Robert B., Reece Jane B., Urry Lisa A., "Campbell Biology", Pearson, 2014

[3] Islam Saiful, Zeisel Amit, Joost Simon, La Manno Gioele, Zajac Pawel, Kasper Maria, Lönnerberg Peter, Linnarsson Sten, "Quantitative single-cell RNA-seq with unique molecular identifiers", Nature Methods 11, 163–166, 2014

[4] Keener James, Sneyd James, "Mathematical Physiology", Springer, 2009

[5] Kolodziejczyk Aleksandra A., Kim Jong Kyoung, Svensson Valentine, Marioni John C., Teichmann Sarah A., "The Technology and Biology of Single-Cell RNA Sequencing", Molecular Cell, Volume 58, Issue 4, Pages 610-620, 2015

[6] Kukurba Kimberly R., Montgomery Stephen B., "RNA Sequencing and Analysis", Cold Spring Harbor protocols, vol. 2015(11), 951–969, 2015

[7] Mackenzie Ruairi J., "RNA-Seq: Basics, Applications and Protocol", Technology Networks Genomic Research, April 6, 2018

[8] Mazutis Linas, Gilbert John, Ung W. Lloyd, Weitz David A., Griffiths Andrew D., Heyman John A., "Single-cell analysis and sorting using droplet-based microfluidics", Nature Protocols 8, 870–891, 2013

[9] Segundo-Val Ignacio San, Sanz-Lozano Catalina S., "Introduction to the Gene Expression Analysis", Methods in molecular biology (Clifton, N.J.), vol. 1434: 29-43, 2016

[10] Shui Qing Ye, "6 - Serial Analysis of Gene Expression in Human Diseases", Editor(s): M.A. Hayat, Handbook of Immunohistochemistry and in Situ Hybridization of Human Carcinomas, Academic Press, Volume 1, Pages 85-98, 2002

[11] Simpson Brittany, Tupper Connor, Al Aboud Nora M., "Genetics, DNA Packaging", StatPearls, 8 June 2021

[12] Singh Komal P., Miaskowski Christine, Dhruva Anand A., Flowers Elena, Kober Kord M., "Mechanisms and Measurement of Changes in Gene Expression", Biological research for nursing, vol. 20(4): 369-382, 2018

[13] Vallejos Catalina A., Marioni John C., Richardson Sylvia, "BASiCS: Bayesian Analysis of Single-Cell Sequencing Data", Plos Computational Biology, June 24, 2015

[14] AgBiosafety at UNL, "Biotech Basic Gene Regions", 2001, URL https://agbiosafety.unl.edu/education/gene.htm

[15] Britannica, The Editors of Encyclopaedia, "gene", Encyclopedia Britannica, 16 Jan. 2019, URL https://www.britannica.com/science/gene

[16] Britannica, T. Editors of Encyclopaedia, "messenger RNA", Encyclopedia Britannica, May 22, 2020, URL https://www.britannica.com/science/messenger-RNA

[17] Britannica, T. Editors of Encyclopaedia, "ribosomal RNA" Encyclopedia Britannica, May 22, 2020, URL https://www.britannica.com/science/ribosomal-RNA

[18] Enciclopedia Treccani, Dizionario di Medicina, "RNA", 2010, URL https://www.treccani.it/enciclopedia/rna_res-20af529e-9b5d-11e1-9b2f-d5ce3506d72e_(Dizionario-di-Medicina)/

[19] Functional genomics II, "RNA sequencing", 2021, URL https://www.ebi.ac.uk/training/online/courses/functional-genomics-ii-common-technologies-and-data-analysis-methods/rna-sequencing/

[20] National Human Genome Research Institute, "Chromosome", URL https://www.genome.gov/genetics-glossary/Chromosome

[21] Scitable by nature education, "Gene Expression Is Analyzed by Tracking RNA", 2014, URL https://www.nature.com/scitable/topicpage/gene-expression-is-analyzed-by-tracking-rna-6525038/

[22] Scitable by nature education, "Northern Blot", 2014, URL https://www.nature.com/scitable/definition/northern-blot-287/

[23] Wikipedia, "Messenger RNA", URL https://en.wikipedia.org/wiki/Messenger_RNA

[24] Wikipedia, "Transfer RNA", URL https://en.wikipedia.org/wiki/Transfer_RNA

[25] 10x Genomics, "Revolutionizing Gene Expression with Single Cell RNA-seq", 2021, URL https://www.10xgenomics.com/single-cell-technology

[26] 10x Genomics, "Single-Cell RNA-Seq: An Introductory Overview and Tools for Getting Started", 2017, URL https://www.10xgenomics.com/blog/single-cell-rna-seq-an-introductory-overview-and-tools-for-getting-started