Senior Thesis in Mathematics

Normalization of RNA-Seq data in the case of asymmetric differential expression

Author:
Ciaran Evans

Advisor:
Dr. Jo Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

May 5, 2016

Abstract

Use of RNA sequencing (RNA-Seq) to study organisms' genomes has become widespread in scientific research. The development of this technology has led to a substantial field of research into the statistical and computational methods needed to process and analyze RNA-Seq data, and one of the most important steps is normalizing the data so that measurements from different samples can be compared. However, statistical research on normalization is ongoing, and as yet no consensus has been reached on how to correctly normalize RNA-Seq data. In this thesis we examine how gene expression data can contain asymmetry, and investigate the performance of normalization methods in the case of asymmetric differential gene expression.

0.1 Acknowledgments

Thank you to Professor Jo Hardin for all her help with research and thesis work, and for always giving great advice. Thank you to Professor Dan Stoebel for providing the biological motivation and background. And thank you to Professor Vin de Silva for overseeing Senior Seminar and answering questions about thesis.

Contents

0.1 Acknowledgments . . . . . .

1 Introduction **1**

2 RNA-Seq and Differential Expression Analysis **3**
2.1 Overview of gene expression 4
2.2 Step 1: Measuring the quantity of mRNA 5

3 Normalization **7**
3.1 The need for normalization 7
3.2 Overview of normalization techniques 8
3.3 Review of Normalization Comparisons 16

4 Symmetry and Differential Expression **18**
4.1 Symmetric and Asymmetric Expression 18
4.1.1 Generalizing DESeq normalization to deal with asymmetry 21
4.2 Asymmetric DE and the FDR 23

5 Conclusions **27**

6 Appendix **31**
6.1 The False Discovery Rate 31
6.2 Differential expression analysis procedures 32
6.2.1 DESeq and DESeq2 32

List of Figures

2.1 mRNA fragments. 6
2.2 Sequenced mRNA fragments (reads), colored according to which gene they come from. There are three genes (blue, green, brown) in this example. . . . 6
2.3 Reads mapped back to the reference genome. There are three genes (blue, green, brown) in this example. 6
2.4 The first three rows of a read count matrix from an RNA-Seq experiment with three conditions (A, B, and C) and two samples per condition. 7

4.1 Symmetric differential expression. Blue = over-expressed, orange = under-expressed. 20
4.2 Asymmetric differential expression. Blue = over-expressed, orange = under-expressed. 20

List of Tables

4.1 Average (SE) empirical FDR for symmetric, partially asymmetric, and completely asymmetric simulated data with five different normalization methods. 25

6.1 Discoveries and false discoveries when testing m null hypotheses [3]. 31

Chapter 1

Introduction

The introduction of microarrays at the beginning of this century provided the ability to study many genes in an organism under different biological conditions, with a dramatic reduction in expense and time from previous methods [27]. More recently, high-throughput sequencing has become an affordable and effective way of examining gene behavior and has been applied to study a wide range of biological settings. For example, very specific questions about transcriptomes and splicing can now be addressed [18], and the study of techniques for the analysis of high-throughput sequencing data continues to be a hot topic, involving researchers from biology, statistics, computer science, and machine learning.

High-throughput sequencing with RNA, commonly referred to as RNA-Seq, involves mapping sequenced fragments of RNA. In RNA-Seq, the RNA is cut into many small fragments. These fragments are then sequenced, and aligned back to a pre-sequenced reference genome or transcriptome [2,18,35], or in some cases assembled without the reference [35]. The number of reads mapped to a gene is used to quantify its expression, providing information about how that gene functions under the experimental conditions. This RNA-Seq technology offers several advantages over microarrays that have contributed to its enormous popularity in recent years, to the extent that in many places it has replaced microarrays [32]. For example, a drawback of microarray studies, which use nucleic acid hybridization, is that they can only be performed on known sequences. RNA-Seq, on the other hand, can be used in *de novo* transcript assembly and so does not require prior sequence knowledge for all studies [27,32]. Furthermore, the nature of microarray measurements can impose upper and lower limits on the measured values as a result of probe saturation and noise respectively [32], leading to difficulty in accurately quantifying extreme levels of expression. This problem is somewhat addressed by RNA-Seq, although it has been pointed out that since a total number of RNA fragments are produced in any given sample from an RNA-Seq experiment, very highly expressed genes in that sample can take up most of those fragments and result in a reduced number to be shared among the other genes [24].

While the observational units in an RNA-Seq study need not be genes, in this thesis we will follow the example of others, such as Anders and Huber [1], and use *gene* as a general term to refer to the experimental units. This is a suitable simplification because one of the most common uses of an RNA-Seq experiment is to investigate the differential expression of an organism's genes under different biological conditions [18], and statistical and computational methods for differential expression analysis are the focus of this work. A

gene is said to be *differentially expressed* across different biological conditions if there is a difference in its true expression under these conditions; as there will always be a difference in the *observed* expression of a gene in different samples, statistical models are built that make assumptions about the underlying true expression and are used to perform hypothesis tests to detect significant differences in observed expression. Gene expression is measured in RNA-Seq using the number of *reads* (sequenced transcripts) aligned to each gene under each biological condition [35]. However, a naive comparison of *read counts* for a given gene under the different conditions is problematic for two reasons. First, the number of reads aligned to a given gene in a given sample can be considered a discrete random variable [1], and so read count comparisons must take into account the variability of these random variables; an observed difference in count could simply be due to random chance. Second, the total number of reads can vary across samples [18], and so a large difference in a gene's read count between different conditions may simply be the result of different coverage, rather than of differential expression. Therefore, *normalization* of read counts is required before differential expression analysis can be performed [2, 18].

In the past several years, a diverse range of methods have been developed to perform differential expression analysis. These methods use statistical and computational techniques to test for differential expression. Analysis generally begins with a *read count matrix*, which stores the read count for each gene under each condition; it is generally assumed that the previous data-gathering and data-organizing steps have been performed before these methods are applied. As our focus is on the methods and not the technical steps required to produce the data, we shall also assume throughout this thesis that a read count matrix has been properly produced for the experiment at hand. Following data collection, the pipeline continues with normalization, followed by differential expression analysis using the normalized data or information provided by the normalization procedure.

There is a vast array of both normalization and differential expression procedures available. While some normalization methods were either developed together with a differential expression analysis procedure (e.g., `DESeq` [1]) or are closely associated with one (such as TMM/`edgeR` [23, 24]), many others stand alone. In general, most normalization procedures can be applied independently of the choice of differential expression analysis procedure, even those which are closely tied to a specific package. A comprehensive overview of normalization techniques is presented in this thesis. Popular differential expression analysis methods for RNA-Seq data include the `R` packages `DESeq` [1] and its successor `DESeq2` [15], `edgeR` [23], and `limma` with the `voom` function (henceforth referred to as `limma-voom`) [11]. `DESeq`, `edgeR`, and `DESeq2` model the read count distribution for each gene and each condition as negative binomial, then use parameter estimates for the negative binomial distributions to carry out hypothesis testing. `limma-voom` uses the `voom` transformation to convert RNA-Seq data into a form usable by methods (like `limma`) which were originally developed for microarrays.

Because of the importance of differential expression analysis, the reliability of existing methods continues to be an important subject of research and has received much attention in recent literature. Many studies [7, 10, 20, 21, 25, 26, 28] have been devoted to the comparison of different methods, with various results. However, a common theme in many of these comparisons is that existing methods fail to perform correctly in some, or even all, situations. For example, Rocke *et al.* [26] finds inflated false positives in negative-binomial based methods like `DESeq2`, and `edgeR`; interestingly, while `DESeq` is also a negative-binomial

method, the original `DESeq` procedure is actually conservative [25] in that it has *too few* false positives.

As suggested by these examples, problems identified have to do with the number of *false positives* produced by a method. When "too many" false positives are produced, the results of differential expression analysis are not trustworthy, and so it is of considerable interest to control the rate at which false positives occur. The popular choice in genomics [30] is the *false discovery rate* (FDR), an error rate developed by Benjamini and Hochberg [3] which is the expected proportion of type I errors out of all discoveries (genes called significant by the differential expression testing). As discussed by Rocke *et al.* [26], one way that existing methods can fail to control the false discovery rate is through biased parameter estimation. Another cause has been identified by Soneson and Delorenzi [28] as incorrect normalization of RNA-Seq data in the case of *asymmetric differential expression*, which can occur when most differentially expressed genes are more highly expressed in one experimental condition than another. In this thesis, we will explore the need for normalization and examine a number of normalization techniques that are found in the literature. We will consider how these normalization techniques deal with asymmetric differential expression (if they do), and their performance in the case of asymmetry. In particular, we will investigate the impact of asymmetry on control of the false discovery rate when using different normalization techniques.

Chapter 2

RNA-Seq and Differential Expression Analysis

Of great interest in biology and medicine is the behavior of biological processes at a molecular level. These processes are intimately linked with nucleic acids, which come either in the form of DNA (deoxyribonucleic acid) or RNA (ribonucleic acid), and both forms play important roles. Many organisms encode their genetic information in the form of DNA, and this information is accessed through gene *expression*. A simplified view of gene expression is that each gene codes for a polypeptide, which is then incorporated into a protein that is involved in biological processes regulated by that gene. This flow of information, from nucleic acids to proteins, is referred to as the central dogma of molecular biology.

Integral in gene expression is RNA, which plays many different roles in the cell; there is messenger RNA (mRNA), ribosomal RNA (rRNA), and transfer RNA (tRNA), to name just a few. The entire collection of RNA molecules is referred to as the *transcriptome*, while the set of genes is the *genome*. As both genes and RNA transcripts play crucial roles in

an organism, their study can lead to greater understanding of the processes that govern an organism. For example, as discussed by Wang, Gerstein, and Snyder [35], the study of the transcriptome is essential for understanding the genome and for investigating biological processes at a molecular level.

One approach to studying the roles of genes in an organism is to examine the function of genes under different experimental conditions [2,18]. For example, a drug trial might compare the function of genes in the control groups vs. the treatment group. A biologist studying the bacterium *E. coli* may be interested in investigating how the bacterium's genes behave differently under different concentrations of a regulatory protein. To compare the function of genes under the different experimental conditions, we look for significantly different levels of expression under those different conditions, by testing for *differential gene expression* [18].

Previous technology for investigating transcriptomes and differential expression have included microarrays and low-throughput sequencing [35]. However, a number of challenges exist with these methods [35], and the preferred approach for such analysis is becoming high-throughput RNA sequencing (RNA-Seq) [2, 35], which works by creating millions of small RNA fragments and determining the base-pair sequence of these fragments. In this chapter, we give an overview of the background behind RNA-Seq experiments and handling RNA-Seq data.

2.1 Overview of gene expression

For the information encoded in a gene to be expressed, the gene must first be *transcribed* into the form of mRNA. Once in this form, the information carried by the mRNA is then *translated* into polypeptides. As mRNA is the intermediate step in the relation between genes (parts of the DNA) and proteins (the form in which they are expressed), we can measure the amount of gene expression by measuring the amount of mRNA produced under different biological conditions [35]. In particular, we can ask whether the amount of expression for each gene is significantly different across the different experimental conditions.

Definition 2.1 Differential Expression. Suppose we perform an experiment with c experimental conditions $A_1, ..., A_c$ on an organism with g genes in its genome. We say that a gene $i \in \{1, ..., g\}$ is **differentially expressed** across the experimental conditions if there is a difference in the true expression of gene i under the different conditions.

In practice, of course, the *true* expression of a gene under a specific condition is not known, nor is it even clear how we should think about it. Different methods and software packages make different choices in modeling the underlying expression of each gene, and for these methods the true expression would be the true value of the modeling parameter. In `DESeq`, for example, expression is modeled with an underlying *expression strength parameter*, and the amount of mRNA produced in the experiment is used to create estimates and hypothesis tests of the expression strength parameter for each condition [1]. More details on how `DESeq` models gene expression and performs hypothesis testing can be found in the Appendix (Section 6.2.1).

While the perspective on the underlying gene expression changes for different differential expression analysis packages, the approach to testing for differential expression does

not change: a method determines a gene to be differentially expressed when its observed expression is significantly different under the different experimental conditions, and we measure expression through the amount of mRNA produced [18]. The question is then how to measure the quantity of mRNA for each gene, and after doing so, how to use these measurements to determine which genes are differentially expressed. An overview of these steps can be found in [35] and in [2]. Moreover, existing `R` packages for performing analysis of RNA-Seq expression data are designed with a certain workflow. A summary of the basic steps follows in Definition 2.2.

Definition 2.2 The differential expression analysis procedure. Suppose we perform an experiment to examine differential expression in an organism under c different conditions. The process to determine which genes are differentially expressed has the following general outline:

1. Data collection: measure expression of each gene under each condition
2. Data normalization: account for differences between samples in the data collection procedure
3. Estimate parameters: using the model for gene expression, we estimate the parameters from the data
4. Hypothesis testing: using estimated model parameters, test each gene for differential expression

While in general `R` packages assume that Step 1 in Definition 2.2 has already been performed, it is useful to briefly describe the data-gathering process of an RNA-Seq experiment and this will be the topic of the remainder of this chapter. The focus of this thesis is on correctly performing Step 2 (Data Normalization), and details on different methods for carrying out normalization can be found in Chapter 3. Details on performing Steps 3 and 4 can be found in the Appendix (Chapter 6).

2.2 Step 1: Measuring the quantity of mRNA

In this section, we explain the data gathering process, which is Step 1 in Definition 2.2. We first cover how to measure the amount of mRNA for each gene and each sample, then how that information is stored. The material for this section is due to [2,18,35]. We first discuss the creation of sequenced mRNA fragments (called *reads*), then cover the form in which the data is stored.

The amount of mRNA produced by a gene under a given experimental condition is measured using high-throughput sequencing technology. The following steps provide a non-technical overview of the main components of the RNA-Seq data gathering procedure. We begin with a collection of mRNA transcripts, the results of transcription of our organism's genome, followed by:

1. Chop up the mRNA into small fragments, as depicted in Figure 2.1.

Figure 2.1: mRNA fragments.

2. Sequence each of the mRNA fragments; the process of sequencing coverts them to cDNA (DNA originating from mRNA). A sequenced fragment is called a *read*. Each read corresponds to a gene on the genome, so we can think of coloring each read according to which gene it comes from. Figure 2.2 depicts the sequenced reads.

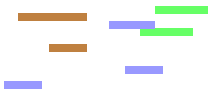

Figure 2.2: Sequenced mRNA fragments (reads), colored according to which gene they come from. There are three genes (blue, green, brown) in this example.

3. Map each read back to a pre-sequenced reference genome, as shown in Figure 2.3.

Figure 2.3: Reads mapped back to the reference genome. There are three genes (blue, green, brown) in this example.

4. The number of reads mapped to each gene in the reference genome is recorded in the read count matrix, as shown in Figure 2.4.

The mRNA sequencing is performed for each sample in the experiment. Because we are interested in measuring gene expression by the amount of mRNA produced, we count the number of reads aligned to each gene in each sample. The results can be stored in a matrix, which is referred to as the *read count matrix*.

Definition 2.3 The read count matrix. For each gene i, let k_{ij} denote the number of reads aligned to gene i under sample j. The **read count matrix** is the matrix $[k_{ij}]$, that is, whose (i, j) entry is k_{ij}.

An example read count matrix from a hypothetical experiment with three conditions and two samples per condition is displayed in Figure 2.4. The $(1, 5)$ entry in Figure 2.4, for example, is 47. So 47 reads were aligned to Gene 1 in Sample 5, and we can see from the example matrix that Sample 5 was collected under condition C.

Gene	Condition A		Condition B		Condition C	
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
1	37	42	67	19	47	29
2	2	3	4	11	2	2
3	2257	3300	2789	2692	5647	5737
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 2.4: The first three rows of a read count matrix from an RNA-Seq experiment with three conditions (A, B, and C) and two samples per condition.

Chapter 3

Normalization

The normalization step of RNA-Seq analysis involves transforming the read count matrix in some way so that the comparison of counts between different samples is valid. Though the analysis of microarray data has involved complex normalization techniques essentially since its inception (see, for example, [4]), with the advent of RNA-Seq technology it was initially believed that normalization would not be necessary [35]. As we shall see in the following section, however, normalization is an indispensable part of RNA-Seq analysis. Indeed, Bullard *et al.* [5] found that the normalization procedure used in a differential expression pipeline had the largest impact on the results of the analysis, even more than the choice of test statistic used in hypothesis tests for differential expression.

3.1 The need for normalization

Because there is inherent variability in the experimental collection of RNA-Seq data, even when two samples are collected under exactly the same biological conditions there may be differences in the sample which need to be removed in the analysis process through normalization. One cause of variability is the fact that different genes have different lengths. If fragments are roughly the same size, this means that longer genes will tend to have more reads aligned, which introduces bias into the expression measurements [19]. A second factor contributing to the need for normalization is library size/sequencing depth of each sample. Some samples can have more reads than others (a larger library size), either by having more highly differentially expressed genes, or by being sequenced at a different depth (different baseline levels of expression for the non-differentially expressed genes across the conditions).

A good example of library size and sequencing depth is found in Li *et al.* [12] and we present an adapted version here.

Example 3.1 Sequencing depth and library size discrepancies. Suppose that we perform an experiment with 1000 genes and two conditions A and B, with 100 of the genes differentially expressed, and one sample per condition. Each of the 900 non-differentially expressed genes has a count of 50 in condition A and a count of 100 in condition B, so the sequencing depth of the sample under condition A is half that of the sample under condition B. However, the 100 differentially expressed genes have counts of 1000 under condition A and 100 under condition B. This gives a total of 145,000 reads under condition A and 100,000 reads under condition B, and hence the sample performed under condition A has a larger library size but a smaller sequencing depth.

As demonstrated in Example 3.1, we cannot simply compare the number of reads aligned to a gene across conditions without adjusting for other factors. In the example, 900 genes in one sample had half the number of reads as in the other sample, but this was the result of a difference in sequencing depth rather than differential expression. On the other hand, quantifying the difference in sequencing depth is difficult because the sample under condition A appears to be sequenced *more* deeply as it has a larger library size. The implication of this example is that the salient information for normalization is found in the subset of genes which are not differentially expressed, and that differentially expressed genes hinder normalization. This insight has been used in many, though not all, normalization procedures. The following section contains a summary of a wide range of normalization procedures, covering the methods most commonly used and investigated in the literature, as well as some less-common methods and others that are more recent.

3.2 Overview of normalization techniques

A substantial number of normalization techniques are available and used by differential expression methods. In this section we present an overview of several that appear in the literature and that are meant to be representative of the range of existing methods.

Total Count: Total count normalization deals with the most observable difference in RNA-Seq samples: their library sizes. In total count normalization [6], read counts are normalized by dividing each count by the total number of reads in its sample. The goal of total count normalization is to account for differences in library size by simply dividing by library size in each sample.

RPKM: RPKM (reads per kilobase per million mapped reads) normalization [17] is an adaptation of total count normalization that attempts to normalize by gene length as well as the total number of reads in each sample. As the name suggests, in RPKM normalization each read count is normalized by dividing by the number of reads in the sample (in millions) and the gene length (in kilobases).

FPKM: FPKM (fragments per kilobase per million mapped fragments) normalization [34] is almost exactly the same as RPKM normalization, with the change of using cDNA molecules rather than RNA reads; each cDNA molecule corresponds to two reads, each starting at a different end of the fragment.

Quantile: Before the use of RNA-Seq experiments was common, a huge body of work was developed for the analysis of microarray data. Quantile normalization is the result of attempting to apply a normalization used in microarray analysis to RNA-Seq data. The basic algorithm is as follows, and is designed to make use of the fact that data vectors with the same distribution will have their quantiles plotted on the diagonal, by forcing the normalized data to have quantiles on the diagonal and hence have the same distribution [4]:

1. Order each column of the read count matrix; this causes each row to contain the same quantiles of each sample.
2. Replace each entry in the sorted read count matrix with the mean of that row.
3. Undo the sorting on the read count matrix, so that the entries are now back in the original order.

Using this algorithm, the read count matrix has been normalized so that each sample is forced to have the same distribution over all the genes. Other measures such as the median could be used in place of the mean of the quantiles.

Upper Quartile: Upper quartile normalization [5] is similar to quantile normalization but focuses on one specific quantile (the 75^{th} percentile). In upper quartile normalization, each read count is divided by the 75^{th} percentile of the read counts in its sample, where genes with read counts of 0 across all samples are excluded. Zyprych-Walczak *et al.* [36] also report a variant of Upper Quartile normalization in a rather complicated form that ultimately reduces to scaling each Upper Quartile normalization factor by the geometric means of the Upper Quartiles, so that the product of the normalization factors is 1.

Median: Median normalization [6] is essentially the same as Upper Quartile normalization, except that gene counts are scaled by the median of counts in their sample rather than the 75^{th} percentile.

DESeq: The `DESeq` normalization strategy attempts to find a *size factor* for each sample, such that the ratios of size factors of different samples represent the ratio of their respective sequencing depths. Let k_{ij} be the number of reads aligned to gene i under sample j. The estimated size factor $\hat{s}_j$ for sample j is given by

$$\hat{s}_j = \text{median}_i \left\{ \frac{k_{ij}}{\left(\prod_{v=1}^{m} k_{iv} \right)^{1/m}} \right\}$$

where m is the total number of samples, across all conditions. The denominator $\left(\prod_{v=1}^{m} k_{iv}\right)^{1/m}$ serves as a pseudo-reference sample to which each sample can be compared. As discussed in [1], the rational behind the size factor estimation is that a good estimate for the ratio of sequencing depths of two samples should be the median of the ratios of their counts. This is generalized to multiple samples through the use of the pseudo-reference sample, which allows for transitivity in the calculated size factor ratios. In this thesis, for simplicity we will introduce the notation

$$e_{ij} = k_{ij} \left(\prod_{v=1}^{m} k_{iv}\right)^{-1/m}$$

and we will refer to the e_{ij} as the *relative expression* values.

TMM: TMM (Trimmed Mean of the M-values) [24] is a normalization strategy with a very similar approach to the `DESeq` size-factor estimate. TMM sets one of the samples as a reference sample, then compares the counts in each sample to the reference sample to estimate the ratio of sequencing depths between each sample and the reference. The procedure involves trimming genes twice, using both the fold-changes and expression levels between samples; the goal is to remove genes that are differentially expressed, so that the mean can be taken over genes that do not show differential expression. For these genes, we expect that the ratio of counts in one sample to the reference sample is represented by the ratio of the sequencing depths.

Let k_{ij} again denote the number of reads aligned to gene i under sample j. Let μ_{ij} be the true gene expression level of gene i under sample j, L_i the length of gene i, and N_j the total number of reads for sample j $\left(= \sum_i k_{ij}\right)$. Fixing one of the samples r as the reference sample, we define *gene-wise log fold changes*

$$M_{ij}^r = \log_2 \frac{k_{ij}/N_j}{k_{ir}/N_r}$$

and *absolute expression levels*

$$A_{ij}^r = \frac{1}{2} \log_2 \left(\frac{k_{ij}}{N_j} \cdot \frac{k_{ir}}{N_r}\right).$$

For sample j, the M_{ij}^r and A_{ij}^r values are trimmed independently (the default is 30% for the M_{ij}^r and 5% for the A_{ij}^r) to produce a set of genes G for which neither the M_{ij}^r nor A_{ij}^r value was removed. Using this set G, we calculate the scaling factor $TMM_j^{(r)}$ for sample j via a weighted mean:

$$\log_2(TMM_j^{(r)}) = \frac{\sum_{i \in G} w_{ij}^r M_{ij}^r}{\sum_{i \in G} w_{ij}^r}$$

where

$$w_{ij}^r = \frac{N_j - k_{ij}}{N_j k_{ij}} - \frac{N_r - k_{ir}}{N r k_{ir}}.$$

Note that in the calculation of the scaling factors, we divide by the library size of each sample. Thus, the $TMM_j^{(r)}$ scaling factors tell us the relative size of samples after we have normalized by library size, and to normalize so that read counts are directly comparable between samples we would divide each sample by $TMM_j^{(r)} \cdot \frac{N_j}{N_r}$ where N_r is the library size of the reference sample.

CuffDiff: Introduced by Trapnell *et al.* [33] as part of the CuffDiff 2 software, the CuffDiff normalization method is a slight modification of the `DESeq` method. The CuffDiff approach calculates two different normalization factors: an *internal scale* is used when comparing samples taken under the same biological conditions, while an *external scale* is used to compare samples across different biological conditions.

Calculation of the internal scale is simply a restriction of the `DESeq` normalization method to the read count sub-matrix for each set of replicates; in an experiment with two conditions A and B and three replicates per condition, for example, the `DESeq` method would be applied to both groups of replicates separately, taking three columns of the matrix with each application.

The external scale is calculated after the internal scale; in the case of 3 samples per condition and two conditions, the result would be two sets of three size factors. Let $\hat{s}_j$ denote the internal size factor for sample j. We then use the internal size factors to normalize each column (divide by the corresponding internal size factor). For each gene and each condition, we average the internal-scaled counts for the replicates in that gene and condition; let $\overline{k}_{i,A}$ and $\overline{k}_{i,B}$ denote these averages for gene i in the case of two conditions. That is, with k_{ij} again denoting the (i, j) entry of the full read count matrix,

$$\overline{k}_{i,A} = \frac{1}{m_A} \sum_{j:\rho(j)=A} \frac{k_{ij}}{\hat{s}_j}$$

and likewise for $\overline{k}_{i,B}$, where m_A is the number of samples performed under condition A and $\rho(j)$ denotes the condition under which sample j was performed. We then use the $\overline{k}_{i,\rho(j)}$ values to produce external size factor estimates

$$\eta_j = \text{median}_i \left\{ \overline{k}_{i,\rho(j)} \left(\prod_{\rho(v)} \overline{k}_{i,\rho(v)} \right)^{-1/c} \right\}$$

where c is the number of conditions. We can compare internal-scaled counts across different conditions using the external scale.

Median Ratio: Similarly to how CuffDiff normalization extends the `DESeq` normalization procedure, Median Ratio normalization (MRN) [16] is designed to be a more robust

adaptation of the TMM method. As in the TMM method, define k_{ij} to be the number of reads aligned to gene i under sample j and N_j the number of reads in sample j (its library size). And like the TMM method, the MRN method separates library size normalization and normalization of the samples after dividing by library size. Here, as in [16], we will describe MRN in the special case where there are two experimental conditions A and B, although the method can be generalized to more than two conditions.

MRN begins by taking the mean of library-normalized counts for each gene within each condition:

$$\overline{k}_{iA} = \frac{1}{m_A} \sum_{j:\rho(j)=A} \frac{k_{ij}}{N_j}$$

would define this mean for condition A, and the definition is analogous for condition B. Then, we calculate the ratio τ_i of these two means for each gene i:

$$\tau_i = \frac{\overline{k}_{iB}}{\overline{k}_{iA}}.$$

We define τ to be the median of these ratios across all genes. The intuition is that between two samples of the same experimental condition, the difference in sequencing depth can be determined directly by the difference in library size since there are no genes which can be differentially expressed within the same biological condition. Then, normalization by library size puts samples within the same condition on the same scale. Any remaining differences in normalized read counts within a replicate group are then due to randomness, and so we can remove some of that natural variability by averaging across samples within a replicate group. Then, τ represents the relative sizes of samples under each condition after accounting for library size; to get the normalization factor for the original read count matrix, we include the library size:

$$e_j = \begin{cases} N_j & \text{if } \rho(j) = A \\ \tau \cdot N_j & \text{if } \rho(j) = B \end{cases}$$

Then, dividing each column of the original read count matrix by its corresponding normalization factor will allow for direct comparison of reads across different samples and conditions. The final step is to make the product of the normalization factors be 1 by dividing by their geometric mean, which does not change the relationship between them but ensures that the normalized read counts will be on a similar scale as the originals. Let $\tilde{f} = \left(\prod_{v=1}^{m} e_v \right)^{-1/m}$ where m is the total number of samples across all conditions. Then, the final normalization factor for sample j is

$$f_j = \frac{e_j}{\tilde{f}}$$

PoissonSeq: We mentioned above that the information for normalization is found in the non-differentially expressed genes. The TMM explicitly aims to remove differentially

expressed genes through trimmed means, while methods like Upper Quartile normalization, `DESeq`, and MRN address the issue by examining a quartile of the data, or a transformed version of the data, that is expected to be reasonably representative of the non-differentially expressed genes. In the `PoissonSeq` method [12], developed as part of the `PoissonSeq` package, the idea of using the non-differentially expressed genes is taken a step further by directly performing a goodness-of-fit test to try to find a subset of non-differentially expressed genes.

Let K_{ij} be the random variable for the number of reads aligned to gene i under sample j. It is assumed in the `PoissonSeq` package that $K_{ij} \sim \text{Poisson}(\mu_{ij})$, although for the purposes of the normalization technique the most salient point is using μ_{ij} to denote the expectation of K_{ij}, and the actual distribution of K_{ij} is less important for normalization than for performing tests for differential expression. We model μ_{ij} using

$$\log(\mu_{ij}) = \log(d_j) + \log(\beta_i) + \gamma_{i,\rho(j)}$$

where d_j is the sequencing depth for sample j, β_i is the level of expression of gene i, and $\gamma_{i,\rho(j)}$ represents how associated the expression of gene i is with the condition $\rho(j)$ of sample j. If γ_i is 0 for all conditions, then there is no association between the expression of gene i and the biological conditions and hence gene i is not differentially expressed in the study. Under the null hypothesis that there is no association between gene i and the condition of sample j, $\gamma_{i,\rho(j)} = 0$.

We estimate the expression level of gene i as $\hat{\beta}_i = \sum_{v=1}^{m} k_{iv}$ where m is the total number of samples across all conditions. Since sequencing depth can be compared across samples using non-differentially expressed genes, given a set S of non-differentially expressed genes we can compute an estimate for the sequencing depth of sample j by the proportion of reads aligned to non-differentially expressed genes that come from sample j:

$$\hat{d}_j = \frac{\sum_{i \in S} k_{ij}}{\sum_{i \in S} \left(\sum_{v=1}^{m} k_{iv} \right)} = \frac{\sum_{i \in S} k_{ij}}{\sum_{i \in S} \hat{\beta}_i}.$$

For genes in S, $\gamma_{i,\rho(j)} = 0$ and so $\log(\mu_{ij}) = \log(d_j \beta_i)$. Hence, an estimate for $E(k_{ij})$ is $\hat{d}_j \hat{\beta}_i$ and we can create a goodness-of-fit statistic for each gene i:

$$GOF_i = \sum_{v=1}^{m} \frac{(k_{ij} - \hat{d}_j \hat{\beta}_i)^2}{\hat{d}_j \hat{\beta}_i}.$$

We ultimately want a good estimate of d_j, which means we want to identify S. To do so, we start with an initial estimate of d_j using the entire set of genes as S, then calculate GOF_i statistics and take the middle $(1 - 2\varepsilon) \cdot 100\%$ and re-calculate $\hat{d}_j$. We then alternate between estimating S and d_j until convergence. By default, `PoissonSeq` uses $\varepsilon = 0.25$. The final sequencing depths estimates $\hat{d}_j$ are then scaled so that their product is 1.

DEGES: This normalization approach [9], which stands for Differentially Expressed Gene Elimination Strategy, has a very similar approach to `PoissonSeq`. It alternates between

estimating normalization factors and using those normalization factors to determine which genes are differentially expressed. In the original paper, TMM was used for normalization and the `baySeq` [8] package was used for identifying differentially expressed genes. This was later expanded in the `TCC` package [31], which now allows the user to choose among several different methods for the normalization and differential expression testing steps. For this reason, we will describe the algorithm without relying on a specific strategy for normalization or testing.

1. Using all genes in the experiment, calculate normalization factors for each sample. For example, if we used `DESeq` normalization here, we would calculate the median of the relative expression values across all genes.

2. Using the normalization factors from Step 1, perform differential expression hypothesis testing and identify a set of non-differentially expressed genes.

3. Re-calculate normalization factors using the set of genes identified in Step 2.

The algorithm alternates between Steps 2 and 3 a prespecified number of times, the idea being to iteratively improve normalization. The final normalization factors can then be used in an official differential expression analysis.

Negative Control Genes: As we have discussed, the information needed for normalization is contained in the read counts of the non-differentially expressed genes. If one can identify *a priori* a set of negative control genes which will not be differentially expressed, these could be used for normalization purposes. For example, Bullard *et al.* [5] investigates the use of housekeeping genes, specifically POLR2A, to perform normalization, and the Remove Unwanted Variation method (below) also provides specific techniques for using negative control genes. The term "gene" can also be loosened here, as more recent studies have examined the possibility of using spike-in controls [22], which are designed to not be differentially expressed across any biological conditions.

Remove Unwanted Variation: Adapted from previous work on normalization of microarray data, the Remove Unwanted Variation [22] (RUV) method aims to remove variation between samples that is not the result of the biological covariates of interest. The notation associated with this method will differ from that used in the other normalization procedures described above, as the method is sufficiently complicated that it is easiest to communicate by being consistent with the original paper.

Suppose an RNA-Seq experiment is performed with J genes and n samples, and p covariates of interest. We will restrict our examination of this method to the classic case of a differential study with two conditions. In this case, $p = 2$.

- Let $Y \in \mathsf{M}_{n\times J}$ be the read count matrix (note that this is the transpose of the matrix given in Definition 2.3), so Y_{ij} corresponds to the number of reads aligned to gene j in sample i.

- Let $X \in \mathsf{M}_{n\times p}$ denote the design matrix for the experiment. In our restricted case, the design matrix has a column for the intercept and each entry in the second column is

an indicator for whether the sample corresponding to that row is under condition A or condition B.

- Let $W \in \mathsf{M}_{n\times k}$ be a matrix related to k factors of unwanted variance (this method requires choosing a specific value of k beforehand).
- Let $\alpha \in \mathsf{M}_{k\times J}$ be the coefficients corresponding to the factors of unwanted variance in W.
- Let $\beta \in \mathsf{M}_{p\times J}$ be the coefficients which represent the relationship between each gene and each covariate of interest.
- Let $O \in \mathsf{M}_{n\times J}$ be a matrix reflecting sequencing depth offsets; the authors suggest using Upper Quartile normalization, though of course other methods would also work in its place.

Then, we assume the log-linear model

$$\log E[Y|W, X, O] = W\alpha + X\beta + O. \tag{3.1}$$

The RUV method provides three different sub-procedures to approach normalization given this model, with varying assumptions. RUVg assumes that a set of negative control genes (which can be spike-in controls) is known. RUVr uses the residuals of a first-pass fit to the log-linear model in Equation 3.1 and does not require knowledge of negative control genes, though does assume that the factors of unwanted variation are uncorrelated with the biological conditions. RUVs creates negative control samples by comparing samples within replicate groups, and also relies on negative control genes and the factors of unwanted variation being uncorrelated with the biological conditions in the experiment. The difference between RUVs and RUVg is that RUVs is designed to be more robust to the choice of negative control genes, and the authors state that the method can still perform reasonably even when the entire set of genes is used.

The three RUV paths are reasonably similar, and so for sake of brevity only one (RUVg) will be described in this thesis; notation is borrowed from Risso *et al.* [22]. We begin by assuming that there is a set of J_c negative control genes. When the matrices in Equation 3.1 are restricted to these negative control genes, we will use the subscript c.

1. Define $Z_c = \log Y_c - O_c$, so that we have accounted for offsets in the experimental data. This should make samples of different sequencing depths comparable. Then let Z_c^* be the column-centered version of Z_c. After accounting for sequencing depth, the only variation of negative control genes across samples is from factors of unwanted variation. By subtracting the mean of each column, the measurement of the expression of each gene in Z_c^* is centered at 0, which also allows the intercept term to be 0 in β_c. Since none of the genes are associated with the biological covariates of interest, the other coefficients in β_c will be 0 as well, yielding $Z_c^* = W\alpha_c$.
2. Next, perform the singular value decomposition of Z_c^*, so $Z_c^* = U\Lambda V^T$ where Λ is the rectangular diagonal matrix of singular values of Z_c^*.

3. For a given number k of factors of unwanted variation, we are interested in determining the impact of those factors so we reduce to only the k largest singular values. Denote by Λ_k the $n \times J_c$ matrix obtained from Λ by setting all singular values but the k largest to 0. We estimate W by $\hat{W} = U\Lambda_k$ where we have removed columns of 0s to ensure that $\hat{W} \in \mathsf{M}_{n\times k}$. Under the assumption that the factors of unwanted variation for the negative control genes span the same space as the factors of unwanted variation for all genes (in the linear algebra sense, since columns of W are factors of unwanted variation and $W\alpha$ is a linear combination of the columns of W), then $\hat{W}$ will estimate W.

4. Substituting $\hat{W}$ back into Equation 3.1, and with knowledge of the design matrix X, GLM regression can be used to estimate the remaining parameters α and β, and then differential expression analysis can be performed. Though the authors do not recommend obtaining normalized counts separately from the differential expression analysis procedure, it is possible to use RUVg to normalize by performing OLS regression of $Z = \log Y - O$ on $\hat{W}$. The residuals of this regression are the normalized read counts.

3.3 Review of Normalization Comparisons

Several papers have investigated the different normalization procedures described in the previous section. A general consensus is that Total Count normalization and RPKM/FPKM normalization should not be used. Dillies *et al.* [6] found that length bias was still present after RPKM normalization, and Total Count normalization led to bias when the data had a few highly expressed genes. Similarly, Oshlack and Wakefield [19] found that while normalization by gene length accounts for length bias in the number of reads aligned to each gene, it also *introduces* a length bias in the read count variance. As demonstrated in Figure 2b of [19], dividing by gene length causes longer genes to have too low variance. As in Dillies *et al.*, Bullard *et al.* [5] also found that Total Count normalization performed poorly as a result of bias from a small number of genes with a large number of reads. Maza *et al.* [16] also find problems with using FPKM, although did find reasonable performance using Total Count normalization. Lin *et al.* [14] examined both Total Count and RPKM, finding poor performance which they attributed to the fact that half of the reads for the male *Drosophila* in their experiment were aligned to only 45 genes, while in the females half of the reads were aligned to only 186 genes. In contrast, Li *et al.* [13] found that all conventional normalization methods they examined, including RPKM, performed equivalently.

Quantile normalization generally seemed to do roughly as well as other normalization methods, with perhaps slightly worse performance due to an increase in variability. Dillies *et al.* [6] found that the assumption that all samples should have the same read count distribution causes increased within-condition variability (see, for example, their Figure 1b). Lin *et al.* [14] found reasonably comparable performance between Quantile normalization and other normalization procedures, though like most of the normalization procedures Quantile normalization was not able to detect extreme values well. Bullard *et al.* [5] found similar results to Dillies *et al.*, with Quantile normalization potentially introducing slightly more variation in the data but otherwise producing comparable results.

Many of the other normalization methods have similar motivation and generally produce

reasonably similar results. Upper Quartile, Median, `DESeq`, and TMM are based on the same idea of using a quantile or measure of center on the data, or a transformed version of the data, to get at the information in the non-differentially expressed genes: Upper Quartile uses the 75^{th} percentile, Median and `DESeq` use the median, and TMM uses a trimmed mean. These appear to be some of the most popular methods to examine in papers that compare normalization procedures, and oftentimes they are compared to modifications of the original methods. The CuffDiff and MRN strategies described above are respectively based on `DESeq` and TMM, so one can consider them to be in the same class as Upper Quartile, Median, `DESeq`, and TMM. Overall, the literature has found that this class of methods performs comparably, and papers that compare them usually find that `DESeq` and TMM perform best. This was the conclusion of Dillies *et al.* [6], whose study included Upper Quartile, Median, `DESeq`, and TMM and concluded that `DESeq` and TMM performed better than the others as they were the only methods to combine power and error control. Maza *et al.* [16] also found equivalent performance of `DESeq` and TMM, which did somewhat better than Median and Upper Quartile normalization. Maza *et al.* also compared these methods with the MRN method they developed, and found that MRN performed best (an unsurprising conclusion given that this result is reported by the paper in which they introduced the MRN method). Similarly, Lin *et al.* [14] demonstrated decent performance with Upper Quartile, Median, TMM, and `DESeq`, and saw best performance by `DESeq` and TMM. Zyprych-Walczak *et al.* [36] compared `DESeq`, Upper Quartile, and TMM from this class of methods and agree that `DESeq` performs well, and better than Upper Quartile. Interestingly, they also found that TMM performed worst out of all the methods they evaluated, which is a difference between their results and those of other authors. In contrast with other studies, Rapaport *et al.* [20] found no differences in normalization performance of different methods. Their paper examined the overall differential expression analysis procedure for the packages CuffDiff, `edgeR` (which uses TMM), `DESeq`, `PoissonSeq`, `baySeq` (which uses Upper Quartile by default), and `limma` (which uses TMM). Likewise, Risso *et al.* [22] found that all conventional normalization methods performed equivalently with Upper Quartile normalization (and did not perform as well as RUV), and Li *et al.* [13] found no difference in any standard normalization methods.

The `PoissonSeq` and DEGES methods both attempt to find a subset of non-differentially expressed genes, while the use of negative control genes attempts to use knowledge about which genes should be non-differentially expressed, and RUV relies on negative control genes in two of its variants (RUVg and RUVs). Bullard *et al.* [5] argue that the use of negative control genes is not a feasible strategy in general, since without prior investigation it is impossible to know for sure which genes will not be differentially expressed. This difficulty may soon be obviated by the use of spike-in controls as they are designed to function as negative control genes in any sample, though Lin *et al.* [14] were unable to use these spike-ins with RUVg as the spike-ins were too variable across samples. RUVg is also sensitive to the choice of negative control genes, and so RUVs was proposed as a more robust way to perform RUV normalization when the data included replicate samples [22]. For the methods which attempt to determine non-differentially expressed genes for use in normalization, the developers of DEGES found that it performed better than the individual normalization methods that can be used in Steps 1 and 3 of DEGES [9]. `PoissonSeq` performed better than every method compared by Zyprych-Walczak *et al.* except for `DESeq` [36], while again Rapaport *et al.* [20]

found `PoissonSeq` did just as well as the other methods under consideration in their paper.

The literature discussed above has compared many different normalization procedures for RNA-Seq data. However, each paper examines a different subset of methods, and more recently developed methods appear in fewer such comparisons. This makes synthesizing the results of different studies and normalization procedures difficult, especially as some have directly contradictory results. Overall, it appears that Total Count and RPKM/FPKM normalization do not perform well and should be avoided. The core group of standard methods and their extensions - Upper Quartile, Median, `DESeq`, TMM, CuffDiff, and MRN - are reasonably similar, and in particular TMM and `DESeq` often produce decent results. More involved normalization procedures that search for non-differentially expressed genes (`PoissonSeq` and CuffDiff) may perform better for data in circumstances where the distribution of read counts makes it difficult to retrieve information on the non-differentially expressed genes, but appear less often in the literature. All of these normalization procedures may ultimately prove unnecessary if a set of negative control genes can be identified, either using housekeeping genes or spike-in controls. Finally, all methods described so far in this paragraph perform normalization with only one number for an entire sample (e.g., the `DESeq` size factor estimates). More advanced techniques such as RUV allow for normalization at the gene level instead of the sample level, and attempt to account for more than one factor of unwanted variation, but there does not appear to be a large body of literature analyzing the performance of newer methods like RUV. As discussed in Bullard *et al.* [5], normalization is an essential step in the differential expression analysis process, and if it is not performed correctly the ability to detect differentially expressed genes can be severely compromised. With this in mind, the remainder of this thesis will focus on the ability to perform normalization when there is asymmetric differential expression.

Chapter 4

Symmetry and Differential Expression

4.1 Symmetric and Asymmetric Expression

In Example 3.1, the differentially expressed genes were all much more highly expressed in one condition than the other. This leads to a type of asymmetry in the read count data, as the expression of the differentially expressed genes is not balanced between conditions. For our purposes, we will introduce the terms *up-regulated* and *down-regulated*. A word of warning: in the usual biological context, these terms often refer to the departure of a gene from its "normal" levels of expression. In this thesis, however, we will use the terms only as

relative descriptors, having meaning only in the context of the other experimental conditions and in relation to them.

Definition 4.1 Up-regulation and down-regulation. We say a gene i is **up-regulated** in condition A relative to condition B if we expect the expression of gene i under condition A to be higher than under condition B. Conversely, we say that gene i is **down-regulated** in condition A relative to condition B if we expect the expression of gene i under condition A to be lower than under condition B.

With our definitions of up- and down-regulation, we can define symmetric differential expression using the underlying regulation of genes.

Definition 4.2 Symmetric and asymmetric expression. Consider an experimental condition A. If the proportion of genes which are up-regulated under A relative to the other experimental conditions is equal to the proportion which are down-regulated relative to the other experimental conditions, then we say that the differential expression under A is **symmetric** in the experiment. Otherwise, it is **asymmetric**.

There are two important notes about Definition 4.2. First, symmetry (or asymmetry) of the differential expression is meaningful only at the level of each *condition* in an RNA-Seq experiment. In experiments of three or more conditions, it is possible to have symmetric expression of one condition and asymmetric expression of another. For example, consider an experiment with three conditions A, B, and C, and 1000 genes. In 100 genes, condition A is more highly expressed than conditions B and C, which are equally expressed. In a distinct set of 100 genes, condition B is more highly expressed than conditions A and C, which are equally expressed. In the other 800 genes, expression is equal across the three conditions. In this toy example, A is over-expressed in half of the differentially expressed genes, and under-expressed (relative to B, that is) in the other half. So differential expression under A is symmetric in this experiment, as is differential expression under B. However, genes under condition C are never more highly than in other conditions, but are sometimes less highly expressed. This means that differential expression under condition C is asymmetric in this experiment.

On the other hand, in the special case of two experimental conditions, symmetry in one condition does imply symmetry in the other. Suppose we have two conditions, A and B. Notice that if a gene is more highly expressed under A, by default it must be less highly expressed under B, and vice-versa. Hence to have symmetric differential expression under A, we must have an equal proportion of over- and under-expressed genes under each condition. In Figures 4.1 and 4.2, we can see respective examples of symmetric and asymmetric differential expression. In these figures, we have a simple experiment with 10 genes and 2 different conditions, A and B. We make a diagram to denote the relative expression of the genes under the two conditions. A box is colored orange if the gene is more highly expressed under that condition, blue if it is less highly expressed, and not colored if the level of expression is equal under the two conditions. As we can see, in the symmetric case (Figure 4.1) the proportion of over-expressed genes is equal to the proportion of under-expressed genes in each condition, and with two conditions, symmetric expression under one condition implies symmetric expression under the other condition. Similarly, asymmetry in one condition must

imply asymmetry in the other.

Gene	Condition A	Condition B
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Figure 4.1: Symmetric differential expression. Blue = over-expressed, orange = under-expressed.

Gene	Condition A	Condition B
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Figure 4.2: Asymmetric differential expression. Blue = over-expressed, orange = under-expressed.

For many of the normalization methods described in the previous chapter, symmetric differential expression is a necessary condition to guarantee that they work completely correctly. For example, `DESeq` normalization find the median of the relative expression values e_{ij} to estimate the size factor for each sample. This works exactly when

$$\text{median}_i\{e_{ij} : i \text{ not DE}\} = \text{median}_i\{e_{ij}\},$$

that is, when the non-differentially expressed median is represented by the median of the entire sample. This requires that for differentially expressed genes, e_{ij} lie above and below the median of the non-DE e_{ij} with equal probability. And since having a higher e_{ij} is the result of up-regulation in that gene, this condition is equivalent to having the same proportion of up- and down-regulated genes (symmetric differential expression).

Part of the issue of asymmetric differential expression is that even if the differential expression in one condition is symmetric relative to the others, it is the relationship between normalization factors that is important and normalization factors must be calculated for each sample. Thus, calculating normalization factors "correctly" for some samples and "incorrectly" for others means that the relationship between the estimates will not be representative of the relationship between the true normalization factors. This further implies that normalization factor estimation works correctly only when *all* normalization factors are estimated well.

It is not reasonable to assume that differential expression is perfectly symmetric, and this issue can be avoided by using only non-differentially expressed genes for normalization. One can attempt to identify non-differentially expressed genes as in `PoissonSeq` and DEGES, the latter of which was specifically developed to address the issue of asymmetric differential expression. It is also useful to see what changes would need to be made to the standard normalization techniques (the group which includes `DESeq` and TMM) to allow them to work with asymmetric differential expression. Since `DESeq` generally performs as well or better

than these standard techniques, and uses a simpler method than others such as TMM, in the next section we will develop a theoretical generalization of `DESeq` to asymmetric differential expression.

4.1.1 Generalizing DESeq normalization to deal with asymmetry

Suppose that differential expression is asymmetric under condition A, and let j be a sample under this experimental condition. The `DESeq` normalization process uses the size factor estimate to approximate the median value $\text{med}_{0,j}$ of $e_{ij} = k_{ij}\left(\prod_{v=1}^{m} k_{iv}\right)^{-1/m}$ for the *non-differentially expressed genes*. However, as we have seen, the size-factor estimate works completely correctly only in the case of symmetric differential expression for the condition $\rho(j)$ of sample j. Under the assumption that differential expression is asymmetric under $A = \rho(j)$, then we must modify our approach to size factor estimation.

Let $\pi_{\text{above},j}$ denote the expected proportion of genes for which $e_{ij} > \text{med}_{0,j}$ and $\pi_{\text{below},j}$ denote the expected proportion of genes for which $e_{ij} < \text{med}_{0,j}$. As discussed in the previous section, the condition of symmetric expression is equivalent to the statement $\pi_{\text{below},j} = \pi_{\text{above},j}$. On the other hand, if there is asymmetric differential expression under condition $\rho(j)$, then we can capture both the direction and magnitude of the asymmetry with a *proportion of asymmetry* Δ_j:

Definition 4.3 Proportion of Asymmetry. The **proportion of asymmetry** in sample j is given by

$$\Delta_j = \pi_{\text{above},j} - \pi_{\text{below},j}. \tag{4.1}$$

The interpretation of the quantity Δ_j is straightforward. First, as Δ_j is the difference of two proportions which sum to 1 ($\pi_{\text{above},j} + \pi_{\text{below},j} = 1$), then $\Delta_j \in [-1, 1]$. We also cannot technically have $|\Delta_j| = 1$, as this would imply that we expect all e_{ij} to lie to one side of $\text{med}_{0,j}$ which can only happen when all genes are differentially expressed and in that situation $\text{med}_{0,j}$ is meaningless. Furthermore, $|\Delta_j|$ captures the lower bound on the proportion of genes which are differentially expressed; the expected difference between $\pi_{\text{above},j}$ and $\pi_{\text{below},j}$ is 0 if no genes are differentially expressed. If $\Delta_j < 0$, then more genes are up-regulated under condition $\rho(j)$, and similarly if $\Delta_j > 0$ then more genes are down-regulated under condition $\rho(j)$.

Given our definition of the proportion of asymmetry, it is now clear how to use Δ_j to get back $\text{med}_{0,j}$. As $\pi_{\text{above},j}$ is the proportion of genes having $e_{ij} > \text{med}_{0,j}$ and $\pi_{\text{below},j}$ is the proportion having $e_{ij} < \text{med}_{0,j}$, and $\pi_{\text{above},j} = \pi_{\text{below},j} + \Delta_j$, then

$$\text{med}_{0,j} = E\left(q_{i,\frac{(1-\Delta_j)}{2}}\{e_{ij}\}\right) \tag{4.2}$$

where $q_{i,\frac{(1-\Delta_j)}{2}}$ denotes the $(1-\Delta_j)/2$ quantile. Given knowledge of Δ_j, this then leads to a natural size factor estimate that generalizes the `DESeq` estimate, as under symmetric expression $\Delta_j = 0$ and hence the relevant relative expression normalization quantile becomes the median.

Definition 4.4 General size factor estimate. The **general size factor estimate** for a sample j with proportion of asymmetry Δ_j is

$$\hat{s}_j = q_{i, \frac{(1-\Delta_j)}{2}}\{e_{ij}\}. \tag{4.3}$$

The generalized size factor estimate proposed in Definition 4.4 provides the framework for improving the `DESeq` normalization procedure, but requires the ability to estimate Δ_j. Though this estimation has proved difficult, the generalized framework still proves useful for understanding how the `DESeq` method can fail in the presence of asymmetric differential expression (use of the wrong quantile), and provides a way to correctly estimate size factors if a good estimate of Δ_j is derived. Furthermore, we note that if a different method is used to accurately estimate the normalization factors, one can determine their corresponding quantiles in the e_{ij} values for each sample. This would allow determination of Δ_j, which would be unnecessary for normalization (since it has already been correctly performed) but may still be helpful as a summary statistic of the differential expression data.

Finally, we present a generalization to the `DESeq` method that would allow correct estimation of size factors in the case of asymmetric (even globally asymmetric) differential expression, provided a specific assumption holds.

Definition 4.5 Mean-constant assumption. Suppose we have an experiment with multiple replicates per condition. Let s_j denote the true size factor for sample j. The **mean-constant assumption** is that the geometric means of the true size factors within each replicate group are the same. For example, in an experiment with two conditions A and B we have

$$\left(\prod_{j:\rho(j)=A} s_j\right)^{1/m_A} = \left(\prod_{j:\rho(j)=B} s_j\right)^{1/m_B}.$$

where m_A and m_B are respectively the number of samples under conditions A and B.

Note that the mean-constant assumption is similar to some of the assumptions needed by RUVs normalization. Both need replicates for each condition, and the mean-constant assumption assumes a constant geometric mean which is a slightly more restrictive case of the normalization factors being uncorrelated with the biological conditions of the experiment. However, without requiring negative control genes we can use this assumption to estimate size factors.

Given this assumption, size factors calculated within replicate groups (without using samples from a different condition) should have the same relationship as the true differences in sequencing depth. This is because size factor estimation relies on dividing by a pseudo-reference sample that is calculated as a geometric mean of read counts across conditions. When there is differential expression in a gene, read counts from the other condition can make the e_{ij} value too high or too low, but restricting size factor estimation to within replicate groups ensures that there will be no differential expression between the samples. Thus we estimate the size factor s_j with the *internal size factor* $\hat{s}_j$ of CuffDiff normalization. For the purposes of this thesis, we will refer to this normalization strategy as Mean-Constant

Internal (MCI) normalization. We note that because all comparisons are between samples in the same replicate group and hence are not subject to differential expression, the method will work as long as the mean-constant assumption holds, even in the case of globally asymmetric differential expression ($|\Delta_j| = 1$).

While the mean-constant assumption imposes additional constraints on the data, MCI normalization allows more freedom in the experiment than other normalization procedures when the assumption does hold. For example, the DEGES and `PoissonSeq` methods will not work without a set of non-differentially expressed genes to detect whereas MCI allows for global differential expression. Negative control genes could be forced in a sample through the use of spike-in controls, but MCI does not require them and will work without controls, which is beneficial when experiments do not have controls or they are too variable to be used for normalization. Finally, MCI is independent of the differential expression testing procedure and can be used in multiple hypothesis testing methods, whereas the authors of RUV normalization do not recommend using RUV-normalized counts in a different testing procedure.

4.2 Asymmetric DE and the FDR

So far, we have discussed a number of different normalization techniques and comparisons of them that are found in the literature. We have also delved into the meaning of asymmetric differential expression and seen why normalization strategies like `DESeq` will have difficulty as the proportion of asymmetry increases. In this section we will evaluate the performance of different normalization techniques under varying amounts of asymmetry, using a selection of normalization procedures representative of the variety of normalization procedures available. First, we include the MCI procedure discussed in the previous section as it is deliberately designed to address asymmetric differential expression. Since `PoissonSeq` and DEGES actively aim to identify a set of differentially expressed genes, they are included because their approach has promise even in the case of asymmetric differential expression. We choose `DESeq` as a representative of the standard class of normalization procedures, as it generally performs as well or better than the other methods in its class and is the basis for MCI. Finally, as we will use simulated data, we include normalization with the true sequencing depths since these are known in the simulation; this procedure will be referred to as Oracle normalization. To assess the performance of these different methods, we will evaluate the ability to correctly control hypothesis testing error rates when using the results of the methods in downstream differential expression testing.

When conducting a differential expression analysis, the general approach is to perform at least one hypothesis test per gene for a statistically significant difference in expression between conditions. The `DESeq` and `DESeq2` packages are fairly representative of existing differential expression methods and have proven popular in experiments, so the `DESeq2` package will be used to perform differential expression analysis hypothesis testing (details on how the testing procedure works can be found in the Appendix, Section 6.2.1). With thousands of genes in a genome, such an analysis needs to consider multiple testing issues. Suppose, for example, that we perform each hypothesis test at a significance level α. Then, we would expect $\alpha \cdot 100\%$ of the true null hypotheses to be rejected; since most hypotheses

could be null, we would expect that most of our "differentially expressed" genes in that case would be false positives.

To deal with this issue, it is common to control the *false discovery rate* (FDR) [3], which is the expected proportion of rejections which are false positives. The idea behind controlling the FDR is that we are willing to accept false positives as long as most (for a specified value of "most") of the discoveries are trustworthy. In many RNA-Seq setting, the FDR is controlled using the Benjamini-Hochberg (BH) procedure [3], which we will also use in this thesis. The specifics of the false discovery rate and FDR control via BH are not necessary for understanding the main body of this thesis. They are, however, helpful for a thorough understanding of the issue at hand and so are included in the Appendix, Section 6.1.

The purpose of this section is to demonstrate the effect of asymmetric differential expression on false discovery rate control when using the normalization procedures listed at the beginning of this section: `DESeq`, MCI, `PoissonSeq`, DEGES, and Oracle (true normalization factors). To evaluate the effect of asymmetry, we will perform a simulation using each of these normalization methods in conjunction with the `DESeq2` testing procedure. DEGES will be used by applying the `TCC` package with the default settings for the `calcNormFactors` function. `PoissonSeq` will be used with the default settings for the `PS.Est.Depth` function.

Simulation: To evaluate the performance of different normalization procedures, they will be tested on simulated data with different amounts of asymmetry and differential expression. Performance is measured by the ability to control the FDR at a specified level. To perform the simulations, we have adapted the simulation code used by Law *et al.* [11] with the following parameters:

Number of genes: 1000

Number of conditions and samples: 2 conditions (A and B), 10 replicates per condition

Sequencing depths: drawn independently from $U(0.5, 2)$ distribution. The relative sequencing depth of two samples is the ratio of their corresponding entries in the random sample. The sequencing depths are divided by the geometric mean of the depths for their condition, to ensure that the mean-constant assumption holds.

Differential expression genes and asymmetry: four different proportions of differentially expressed genes (10%, 20%, 30%, and 50%). We simulate data under 3 (a)symmetry scenarios: symmetry (half of the DE genes are up-regulated in A and the other half in B), partial asymmetry (three-fourths of DE genes are up-regulated in A, the rest are up-regulated in B), and "complete" asymmetry (all DE genes are up-regulated in A). Each combination of proportion of expression and amount of asymmetry is simulated.

Fold change for DE genes: 2 (differentially expressed genes are simulated to have twice the baseline expression in the up-regulated condition than in the down-regulated condition).

Normalization methods: as listed above, we use Oracle, `DESeq`, MCI, `PoissonSeq`, and DEGES

FDR control for hypothesis testing: use level of 0.05, and use the Benjamini-Hochberg procedure to control.

Number of runs: each combination of proportion of expression and amount of asymmetry is used to generate 50 sets of data. The mean empirical false discovery rate (observed proportion of false discoveries) is recorded for each normalization method, along with the sample standard deviation of the 50 eFDR measures.

Scenario	Method	10%	20%	30%	50%
Symmetric	Oracle	0.0730 (0.0253)	0.0584 (0.0148)	0.0481 (0.0126)	0.0317 (0.00728)
	`DESeq`	0.0734 (0.0263)	0.0587 (0.0155)	0.0474 (0.0118)	0.0311 (0.00748)
	MCI	0.0723 (0.0290)	0.0580 (0.0152)	0.0478 (0.0122)	0.0319 (0.00713)
	`PoissonSeq`	0.0740 (0.0255)	0.0579 (0.0155)	0.0491 (0.0110)	0.0340 (0.00994)
	DEGES	0.0733 (0.0261)	0.0581 (0.0148)	0.0473 (0.0114)	0.0323 (0.00756)
Partial	Oracle	0.0802 (0.0257)	0.0576 (0.0170)	0.0446 (0.0114)	0.0307 (0.00817)
	`DESeq`	0.0984 (0.0295)	0.1143 (0.0214)	0.1629 (0.0200)	0.2867 (0.00980)
	MCI	0.0802 (0.0268)	0.0572 (0.0170)	0.0450 (0.0114)	0.0312 (0.00807)
	`PoissonSeq`	0.0837 (0.0293)	0.0615 (0.0161)	0.0632 (0.0138)	0.2109 (0.0600)
	DEGES	0.0813 (0.0255)	0.0564 (0.0170)	0.0442 (0.0114)	0.0314 (0.00748)
Complete	Oracle	0.0684 (0.0205)	0.0565 (0.0184)	0.0484 (0.0138)	0.0312 (0.00744)
	`DESeq`	0.1421 (0.0332)	0.2929 (0.0235)	0.4590 (0.0141)	0.4998 (0.00392)
	MCI	0.0680 (0.0182)	0.0568 (0.0187)	0.0486 (0.0138)	0.0314 (0.00756)
	`PoissonSeq`	0.0747 (0.0221)	0.0821 (0.0212)	0.1202 (0.0265)	0.5264 (0.0332)
	DEGES	0.0675 (0.0187)	0.0555 (0.0176)	0.0501 (0.0141)	0.5957 (0.260)

Table 4.1: Average (SE) empirical FDR for symmetric, partially asymmetric, and completely asymmetric simulated data with five different normalization methods.

‘

Results: The results from our simulations are shown in Table 4.2. There are two important trends that must be kept in mind when interpreting the data. First, we notice that even with the Oracle normalization method, the empirical false discovery rate is not controlled at the desired level of 0.05 when the proportion of differentially expressed genes is 10% or 20%. This is a result of a separate issue in the `DESeq2` package. Rocke *et al.* [26] demonstrate that even with no differentially expressed genes, the `DESeq2` package leads to inflated false positives, and they suggest this to be the result of flawed dispersion estimation in the `DESeq2` model. Second, the empirical false discovery rate decreases as the proportion of differentially expressed genes increases. As discussed in the Appendix, Section 6.1, if we specify a level α for FDR control, the Benjamini-Hochberg procedure actually controls at $\alpha\frac{m_0}{m}$ where $\frac{m_0}{m}$ is the proportion of null hypotheses. As the proportion of differentially expressed genes increases, the proportion of null hypotheses must necessarily decrease, hence the decrease in observed FDR. Because of these differences, an evaluation of the normalization procedures

is obtained by comparison with the empirical FDR under the Oracle procedure; since the Oracle procedure uses the true sequencing depths for normalization, and the sequencing depths are the only source of unwanted variability in the data, no normalization procedure can hope to produce better results than the Oracle procedure.

Examining the results for the symmetric scenario, we can see that all methods perform equally well. This is good verification that in the case of symmetric differential expression the methods work as intended, which we expect. As we increase the amount of asymmetry in the data, however, some of the normalization methods lose control of the false discovery rate. As expected, `DESeq` is the first method to lose control. Even with partial asymmetry and a relatively small proportion of differentially expression, `DESeq` normalization exhibits higher empirical false discovery rates. The results with partially and completely asymmetric data demonstrate that the proportion of asymmetry is the driving factor in errors with `DESeq` normalization. Since partially asymmetric data is simulated to have a lower proportion of asymmetry than completely asymmetric data, we expect to see better performance of `DESeq` normalization for partially asymmetric data vs. completely asymmetric data with the same proportion of differentially expressed genes. This is indeed the case, and in fact we can match up partially asymmetric and completely asymmetric simulations that should have the same proportion of asymmetry. For example, the partially asymmetric simulation with 30% differential expression has 22.5% of all genes up-regulated in condition A and 7.5% up-regulated in condition B. The discrepancy between up-regulation and down-regulation (15 percentage points) is similar to 10% difference in completely asymmetric expression with 10% differentially expressed genes. Unsurprisingly, the mean empirical false discovery rates are similar for `DESeq` normalization under these cases (0.1629 vs. 0.1421). Note that these percentage point differences are not the exact proportions of asymmetry. The proportion of asymmetry is defined by the expected difference in the amount of relative expression values e_{ij} lying above and below the median value for the non-differentially expressed genes. However, even with completely asymmetric expression one might expect that some truly up-regulated expressed genes would not be observed to have e_{ij} above the median.

All simulations indicate that as long as the mean-constant assumption holds, MCI performs accurately no matter the proportion of asymmetry or the proportion of differentially expressed genes. In all cases, MCI tracks with the Oracle method and produces almost identical mean empirical FDR values. This is expected, as the data were simulated under the mean-constant assumption that is the basis of the MCI method. Hence the simulations provide confirmation that the MCI method is the correct approach in the case of equivalent geometric means of the sequencing depths between conditions.

The final two methods considered, `PoissonSeq` and DEGES, both aim to identify a set of non-differentially expressed genes and use these genes to perform normalization. At low to medium proportions of differential expression and asymmetry these methods both work well, but both reach breakdown points as the proportion of asymmetry is increased. For `PoissonSeq`, we see decent (although not perfect) performance with completely asymmetric differential expression and proportions of differentially expressed genes of 10% and 20%. However, `PoissonSeq` failed to control the false discovery rate with complete asymmetry and 30% or more differentially expressed genes. DEGES performed better than `PoissonSeq`, with the mean empirical FDR very close to that of the Oracle method until 50% differential expression is reached. At this point, FDR control is completely lost and DEGES performs no

better than `PoissonSeq` and `DESeq`. Furthermore, we must keep in mind that with global, completely asymmetric differential expression it would be impossible to use the DEGES and `PoissonSeq` methods as there would be no non-differentially expressed genes to identify.

These simulations identify the need for correct normalization when there is asymmetric differential expression, and demonstrate that when the mean-constant assumption holds, the MCI method is well-suited to provide sequencing depth normalization.

Chapter 5

Conclusions

The use of RNA-Seq experiments to study organisms' genomes is becoming ubiquitous, and the explosion in the use of sequencing technology has led to a related explosion in the development of statistical methods for processing and analyzing RNA-Seq data. As previous research has demonstrated [5], proper normalization is an essential step in the analysis pipeline. The need for normalization arises from the inherent variability in the collection of RNA-Seq data, and a variety of normalization methods have been devised to combat this variability. The methods discusses in Chapter 3 cover a range of approaches to normalization and include those methods discussed most in the literature. As we saw in the same chapter, this literature has not reached a consensus on which normalization method to use, though there do seem to be common opinions on which normalization methods should not be used. The importance of normalization and the lack of consensus on how to perform it indicate that normalization warrants further study.

In this thesis, we examine the effect of asymmetry on the performance of several representative normalization methods. We have seen that asymmetric differential expression can lead to loss of control of the false discovery rate as the proportion of asymmetry increases. One potential approach to normalization that in theory would not be affected by asymmetry is the use of negative control genes such as spike-in controls. These controls would have constant expression across samples, and so would provide the essential information each normalization method attempts to obtain. However, in the case that spike-in controls are not available or are too variable to allow for normalization, there is a need for methods capable of dealing with asymmetric differential expression. To this end we propose the Mean-Constant Internal (MCI) normalization method, which uses CuffDiff internal size factor estimates to compare across conditions under the mean-constant assumption. Through a simulation study, we have demonstrated that MCI normalization performs comparably with Oracle normalization (perfect knowledge of the true sequencing depths) and outperforms even normalization methods which attempt to detect a set of negative control genes through differential expression testing. Unfortunately, the mean-constant assumption is fairly stringent, and future work

would investigate the possibility of relaxing this assumption and still achieving acceptable performance.

Bibliography

[1] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome biol*, 11(10):R106, 2010.

[2] P. L. Auer, S. Srivastava, and R. Doerge. Differential expressionthe next generation and beyond. *Briefings in functional genomics*, page elr041, 2011.

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

[4] B. Bolstad, R. Irizarry, M. Astrand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[5] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinformatics*, 11(94), 2010.

[6] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, et al. A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Briefings in bioinformatics*, 14(6):671–683, 2013.

[7] Y. Guo, C.-I. Li, F. Ye, and Y. Shyr. Evaluation of read count based rnaseq analysis methods. *BMC genomics*, 14(Suppl 8):S2, 2013.

[8] T. Hardcastle and K. Kelly. bayseq: empirical bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, 11(422), 2010.

[9] K. Kadota, T. Nishiyama, and K. Shimizu. A normalization strategy for comparing tag count data. *Algorithms for molecular biology*, 7(5), 2012.

[10] V. M. Kvam, P. Liu, and Y. Si. A comparison of statistical methods for detecting differentially expressed genes from rna-seq data. *American journal of botany*, 99(2):248–256, 2012.

[11] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth. Voom: precision weights unlock linear model analysis tools for rna-seq read counts. *Genome Biol*, 15(2):R29, 2014.

[12] J. Li, D. Witten, I. Johnstone, and R. Tibshirani. Normalization, testing, and false discovery rate estimation for rna-sequencing data. *Biostatistics*, 13(3):523 – 538, 2012.

[13] P. Li, Y. Piao, H. Shon, and K. Ryu. Comparing the normalization methods for the differential analysis of illumina high-throughput rna-seq data. *BMC Bioinformatics*, 16(347), 2015.

[14] Y. Lin, K. Golovnina, Z. Chen, H. Lee, Y. L. Serrano Negron, H. Sultana, B. Oliver, and S. Harbison. Comparison of normalization and differential expression analyses using rna-seq data from 726 individual drosophila melanogaster. *BMC Genomics*, 17(28), 2016.

[15] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biol*, 15(12):550, 2014.

[16] E. Maza, P. Frasse, P. Senin, M. Bouzayen, and M. Zouine. Comparison of normalization methods for differential gene expression analysis in rna-seq experiments: A matter of relative size of studied transcriptomes. *Communicative & Integrative Biology*, 6(6):e25849, 2013.

[17] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[18] A. Oshlack, M. D. Robinson, M. D. Young, et al. From rna-seq reads to differential expression results. *Genome biol*, 11(12):220, 2010.

[19] A. Oshlack and M. J. Wakefield. Transcript length bias in rna-seq data confounds systems biology. *Biology Direct*, 4(14), 2009.

[20] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel. Comprehensive evaluation of differential gene expression analysis methods for rna-seq data. *Genome Biol*, 14(9):R95, 2013.

[21] P. D. Reeb and J. P. Steibel. Evaluating statistical analysis models for rna sequencing experiments. *Frontiers in genetics*, 4, 2013.

[22] D. Risso, J. Ngai, T. Speed, and S. Dudoit. Normalization of rna-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, 32:896–902, 2014.

[23] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[24] M. D. Robinson, A. Oshlack, et al. A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biol*, 11(3):R25, 2010.

[25] J. A. Robles, S. E. Qureshi, S. J. Stephen, S. R. Wilson, C. J. Burden, and J. M. Taylor. Efficient experimental design and analysis strategies for the detection of differential expression using rna-sequencing. *BMC genomics*, 13(1):484, 2012.

[26] D. M. Rocke, L. Ruan, Y. Zhang, J. J. Gossett, B. Durbin-Johnson, and S. Aviran. Excess false positive rates in methods for differential gene expression analysis using rna-seq data. *bioRxiv*, page 020784, 2015.

[27] J. Shendure. The beginning of the end for microarrays? *Nature Methods*, 5:585–587, 2008.

[28] C. Soneson and M. Delorenzi. A comparison of methods for differential expression analysis of rna-seq data. *BMC bioinformatics*, 14(1):91, 2013.

[29] J. D. Storey. The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035, 2003.

[30] J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9440–9445, 2003.

[31] J. Sun, T. Nishiyama, K. Shimizu, and K. Kadota. Tcc: an r package for comparing tag count data with robust normalization strategies. *BMC Bioinformatics*, 14(219), 2013.

[32] C. Tachibana. Transcriptomics today: Microarrays, rna-seq, and more. *Science*, 2015.

[33] C. Trapnell, G. Hendrickson, M. Sauvageau, L. Goff, J. Rinn, and L. Pachter. Differential analysis of gene regulation at transcript resolution with rna-seq. *Nature Biotechnology*, 31:46–53, 2013.

[34] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. van Baren, S. Salzberg, B. Wold, and L. Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isofrom switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

[35] Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[36] J. Zyprych-Walczak, A. Szabelska, L. Handschuh, K. Górczak, K. Klamecka, M. Figlerowicz, and I. Siatkowski. The impact of normalization methods on rna-seq data analysis. *BioMed Research International*, 2015.

Chapter 6

Appendix

6.1 The False Discovery Rate

Following the notation of Benjamini and Hochberg [3], suppose there is a family of m independent hypotheses to be tested, m_0 of which are truly null. We represent the different possible outcomes in Table 6.1.

	Declared non-significant	*Declared significant*	*Total*
True null hypotheses	**U**	**V**	m_0
Non-true null hypotheses	**T**	**S**	$m - m_0$
	$m-$**R**	**R**	m

Table 6.1: Discoveries and false discoveries when testing m null hypotheses [3].

Definition 6.1 The false discovery rate. From Table 6.1, the proportion of discoveries which are false is V/R, and the **false discovery rate** is defined to be

$$FDR = E(V/R)$$

where $V/R = 0$ whenever $R = 0$. In other words,

$$FDR = E(V/R|R > 0)P(R > 0).$$

To control the FDR at a desired level α, Benjamini and Hochberg proposed the following step-up procedure (henceforth referred to as BH) [3].

Definition 6.2 BH procedure. Let $p_1, ..., p_m$ be the p-values resulting from tests of the m hypotheses, and $p_{(1)}, p_{(2)}, ..., p_{(m)}$ the p-values in increasing order. The **BH procedure** finds the largest index i such that

$$p_{(i)} \leq \alpha \frac{i}{m}$$

and then $p_{(1)}, ..., p_{(i)}$ are declared significant, and their associated hypotheses rejected. Equivalently, each p-value $p_{(i)}$ is adjusted by setting $p_{(i)} = \min\{\frac{m}{j} p_{(j)} : j \geq i\}$, then all p-values below the cutoff α are rejected.

Benjamini and Hochberg proved that this procedure controls the FDR at

$$FDR \leq \alpha \frac{m_0}{m} \leq \alpha$$

and furthermore that the cutoff $T = \max\{p_{(i)} : p_{(i)} \leq \alpha \frac{i}{m}\}$ can be less stringent than the cutoff given by FWER control, since FWER control implies FDR control but a procedure controlling the FDR need not necessarily control the FWER [3].

In the two decades since the introduction of the FDR, a number of alternative approaches have been suggested, including related errors like Storey's positive false discovery rate (pFDR) [29], and adaptive methods for controlling the FDR while attempting to maximize power, such as the one proposed by Storey and Tibshirani [30]. Other methods and procedures attempt to control FDR in more complicated scenarios. The common goal of all these varied methods is to maintain error control in different situations while conserving as much power as possible.

While more advanced methods than the BH procedure are demonstrably better at controlling FDR, in the sense of maintaining control while increasing power (the method proposed in [30] is one such example) the most common choice appears to still be BH, and is in fact the default in the `DESeq` package. For this reason, FDR control performed in simulations in this thesis will be done using BH.

6.2 Differential expression analysis procedures

In this section, we present basic details of performing differential expression analysis with the `DESeq` and `DESeq2` packages. While not a necessity for understanding the normalization procedures discussed in the main body of this thesis, a summary of these methods allows us to see what happens to normalized counts further down the pipeline of differential expression analysis. In this section, we will examine Steps 3 - 4 of Definition 2.2. As these steps are highly dependent on the specific method used to perform analysis, they will be examined in the context of each method considered.

6.2.1 DESeq and DESeq2

Both `DESeq` and its successor `DESeq2` are based on a negative binomial probability model of the read count data. In the early stages of RNA-Seq analysis a Poisson model seemed the natural choice for read count data, but turned out to have too small a variance (with variance equal to the mean), and so the negative binomial model was developed instead as an over-dispersed Poisson: if μ is the expectation of a negative binomial distribution, then the variance can be written as $\mu + \theta\mu^2$ with dispersion parameter θ.

The foundation of the `DESeq` and `DESeq2` methods is the same; the remainder of this section is entirely due to [1] and [15]. Suppose that there are two experimental conditions, A and B, and each sample has been collected under one of the two conditions. Let K_{ij} be a random variable for the number of reads aligned to gene i under sample j, and let $\rho(j)$ be the condition of sample j. Then $K_{iA} = \sum_{j:\rho(j)=A} K_{ij}$ and $K_{iB} = \sum_{j:\rho(j)=B} K_{ij}$ represent the number of reads aligned to gene i under conditions A and B respectively. The `DESeq` and `DESeq2`

models assume that K_{ij} follows a negative binomial distribution: $K_{ij} \sim NB(\mu_{ij}, \sigma_{ij}^2)$, with mean μ_{ij} and variance σ_{ij}^2. It is assumed in both `DESeq` and `DESeq2` that

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

Here $q_{i,\rho(j)}$ is the *expression strength* of gene i under the condition $\rho(j)$ of sample j. The term s_{ij} is the size factor for gene i and sample j, which reflects the sequencing depth of gene i under sample j. The default in `DESeq` and `DESeq2` is that $s_{ij} = s_j$, that is, that size factors are the same within each sample. Both packages estimate size factors under this assumption, but `DESeq2` does offer the option to input size factor estimates on a per-gene per-sample basis from other packages. In other words, the expected number of reads aligned to gene i in sample j is determined by the probability of a read being aligned to gene i in sample j, and the total quantity of reads in sample j.

Estimating expression strength. The expression strength parameter $q_{i,\rho(j)}$ is estimated as

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

where $m_{\rho(j)}$ is the number of samples performed under condition $\rho(j)$, k_{ij} is the observed number of reads aligned to gene i under sample j, and $\hat{s}_j$ is the size factor estimate for sample j.

Size factor estimation. As mentioned above, the size-factor estimates provided by `DESeq` and `DESeq2` are estimates of s_j, so that each gene has the same size factor for a given sample. A description of the size factor estimation process is found in the main body of this thesis.

DESeq

Variance estimation. In the `DESeq` model, it is assumed that

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

In the variance, $v_{i,\rho(j)}$ is a per-gene raw variance parameter for condition $\rho(j)$. The variance in the number of aligned reads is determined by the mean as well as the quantity of reads aligned and the raw variance. Details of the variance-estimation process can be found in [1].

Hypothesis testing. Since each K_{ij} is assumed to follow a negative binomial distribution, then K_{iA} and K_{iB} are also negative binomially distributed. Furthermore, assuming that the read counts are independent across samples, then $K_{iA} \sim NB(\mu_{iA}, \sigma_{iA}^2)$ where

$$\mu_{iA} = \sum_{j:\rho(j)=A} \mu_{ij} \quad \text{and} \quad \sigma_{iA}^2 = \sum_{j:\rho(j)=A} \sigma_{ij}^2.$$

Similarly, $K_{iB} \sim NB(\mu_{iB}, \sigma_{iB}^2)$.

Interest lies in testing whether each gene i is differentially expressed between conditions A and B. Because expression is measured with the expression strength parameter $q_{i,\rho(j)}$ then the relevant hypothesis is

$$H_0 : q_{iA} = q_{iB}.$$

Under the null hypothesis, DESeq estimates a pooled expression strength parameter $\hat{q}_{i0}$, which is then used to estimate the distributions of K_{iA} and K_{iB} under the null hypothesis:

$$\hat{\mu}_{iA} = \sum_{j:\rho(j)=A} \hat{s}_j \hat{q}_{i0} \quad \text{and} \quad \hat{\sigma}^2_{iA} = \sum_{j:\rho(j)=A} (\hat{s}_j \hat{q}_{i0} + \hat{s}^2_j \hat{v}_{i,\rho(j)}) \tag{6.1}$$

with analogous expressions for $\hat{\mu}_{iB}$ and $\hat{\sigma}^2_{iB}$. Using these estimated distributions and the assumption of independence, then the probability of observing any pair of counts (a, b) for K_{iA} and K_{iB} respectively is $P_i(a, b) = P(K_{iA} = a)P(K_{iB} = b)$ where each probability is calculated under the null distributions given in Eq. 6.1. By using the null distributions, p-values can be calculated to test H_0 for each gene. Let k_{iA} and k_{iB} be the observed values of K_{iA} and K_{iB} seen in the experimental data, and let $k_{iS} = k_{iA} + k_{iB}$. DESeq calculates the p-value p_i for gene i by considering all pairs (a, b) of nonnegative integers such that $a+b = k_{iS}$ and $P_i(a, b) \leq P_i(k_{iA}, k_{iB})$, out of the total set of pairs (a, b) such that $a + b = k_{iS}$:

$$p_i = \frac{\sum\limits_{\substack{a+b=k_{iS} \\ P_i(a,b) \leq P_i(k_{iA},k_{iB})}} P_i(a, b)}{\sum\limits_{a+b=k_{iS}} P_i(a, b)}. \tag{6.2}$$

With the p-values p_i in hand for each gene, then hypothesis testing proceeds as normal. In particular, DESeq makes use of the BH method to adjust the p-values and account for multiple tests.

DESeq2

Variance estimation. In the `DESeq2` model, we have

$$\sigma^2_{ij} = \mu_{ij} + \alpha_i \mu^2_{ij}$$

where α_i is the dispersion parameter for gene i. The details of dispersion estimation can be found in [15], but we will present an overview of the main steps:

1. Individual, maximum likelihood estimates are made for the dispersion of each gene.
2. A dispersion trend is fit.
3. The MLE dispersions are combined with the estimates from the trend fit to create final dispersion estimates.

Linear models. Whereas `DESeq` used discrete, count-based hypothesis testing to test for differential gene expression, `DESeq2` uses linear models with a logarithmic link to relate the expression strength parameter q_{ij} to the samples. Specifically,

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

where the x_{jr} are entries in the experiment's expanded design matrix (one column for each level of each explanatory variable and a column for the intercept) and the β_{ir} are coefficients. That is: for each gene i, we have a vector $\mathbf{q_i} = [q_{i1}, q_{i2}, ..., q_{im}]^T$ where each entry is the expression strength parameter for gene i under one of the samples. If $\mathbf{X}$ is the design matrix, with each column corresponding to an experimental condition (or the intercept), and $\beta_\mathbf{i}$ is a vector of coefficients, then

$$\log_2(\mathbf{q_i}) = \mathbf{X}\beta_i.$$

For example, consider a simple experiment to examine differential gene expression between two conditions, A and B. Six samples are performed, with the first three under condition A and the second three under condition B. Then,

$$\begin{bmatrix}\log_2(q_{i1})\\ \log_2(q_{i2})\\ \log_2(q_{i3})\\ \log_2(q_{i4})\\ \log_2(q_{i5})\\ \log_2(q_{i6})\end{bmatrix} = \begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 1 & 1 & 0\\ 1 & 0 & 1\\ 1 & 0 & 1\\ 1 & 0 & 1\end{bmatrix} \begin{bmatrix}\beta_{i0}\\ \beta_{i1}\\ \beta_{i2}\end{bmatrix}$$

By comparing the coefficients β_{i1} and β_{i2} in this example, we can compare the expression strength parameters for samples performed under the two different conditions: $\beta_{i1} - \beta_{i2}$ gives the logarithm of the ratio of expression strength parameters of samples performed under condition A to those of samples performed under condition B.

For this reason, contrasts are referred to as *log fold changes* (LFCs), and to determine if there is differential expression between two conditions we test whether their LFC is 0 (which would imply that the ratio of the expression strength parameters is 1 and hence there is no differential expression). In general, we can write a contrast of interest as

$$\beta_i^c = \mathbf{c}^T\beta_i$$

where $\mathbf{c}$ is the column vector specifying the contrast. In the above example, for instance, to get $\beta_{i1} - \beta_{i2}$ one would use the contrast vector $\mathbf{c}^T = [0\ 1\ -1]$.

Hypothesis testing. To test for pairwise differential expression between two conditions for a given gene i, we test whether the corresponding contrast is equal to 0:

$$H_0 : \beta_i^c = 0 \quad H_1 : \beta_i^c \neq 0.$$

This is performed using a *Wald test*, in which

$$\frac{\beta_i^c}{SE(\beta_i^c)}$$

is compared to a standard normal. As in `DESeq`, the resulting p-values are adjusted using the BH method.