Exact Tests of Overdispersed Counts for RNA-Seq

Jakob Walter

February 2022

Daily Supervisor:
Name: Dr. J.J. Goeman
E-Mail: J.J.goeman@lumc.nl
Institute: Biomedical Data Sciences, LUMC
Function: Professor of Biostatistics

Second Supervisor:
Name: Szymon Kielbasa
E-Mail: S.M.Kielbasa@lumc.nl
Institute: Biomedical Data Sciences, LUMC
Function: Assistant Professor

Abstract:
In the last decade, High-throughput RNA-Sequencing became an indispensable tool for researching molecular mechanisms. However, statistical analysis of the overdispersed count-data that arises from these experiments is complicated. Furthermore, we usually have few replicates per condition and thousands of potential genes to test. We thus face difficulties in estimation and testing: We cannot rely on asymptotic properties of tests and face a huge multiple-testing burden. In this thesis we will explore the theoretical and empirical properties of exact tests when applied to RNA-Seq data. This will be done on both simulated and real data-sets. A special focus will be given to permutation methods. Among others, the applicability of a novel permutation-test based on sign-flipping score contributions will be analysed [14].

Contents

1 Introduction

In the last 15 years Next-Generation Sequencing (NGS) technologies became an invaluable tool for the analysis of the transcriptome. The technological possibility to generate massive amounts of sequencing data reshaped the field of transcriptomics. The technology is now readily available in many laboratories and software packages allow almost automatic statistical analysis of the data. However, the data arising from the technology is highly complex. The ease of statistical analysis conceals the complexity of the models, the various assumptions the models make and the resulting drawbacks.

The key properties of all statistical tests are their Type-I error control and their power. Lack of Type-I error control can be especially problematic when multiply tests are performed, as this effect can be amplified when estimating the False Discovery Rate (FDR). In fact, multiple papers have shown inaccuracies of FDR estimation of many of the most popular statistical packages [36][5][3][22]. Possible reasons for this are, that the methods currently in use either make incorrect distributional assumptions or rely on asymptotic properties of tests that are invalid in the small-sample sizes typical for RNA-Seq experiments. In this thesis, we will investigate the possible reasons for the lack of Type-I error control of existing parametric methods, and propose the use of alternative more robust methods. We assume that one of the most likely reasons is the difficulty of estimating the variance when only limited data is available. We thus investigate the performance of permutation tests, which either do not require the estimation of the variance or are robust to misspecification of the variance. Besides the use of two classical permutation tests, we will propose the use of a recently developed statistical test for based on sign-flipping score contributions of GLMs [14]. The test thus allows the incorporation of co-variates into the model.

The thesis will be structured as follows: In Section 2 we will motivate the biological interest in RNA, outline the technology behind RNA-Seq and thereby explain the origin of some of the peculiarities of RNA-Seq data. Finally, we will introduce a dataset that will be used throughout this thesis to illustrate the methods. In Section 3 we will introduce some of the statistical concepts necessary for the rest of the thesis. In Section 4 we will review the statistical models underlying three of the most commonly used packages for RNA-Seq data. Then, in Section 5 we will introduce permutation tests in general, and the permutation test based on sign-flipping score-contributions. In Section 6 we will perform an extensive simulation study comparing the performance of all discussed methods. Finally, in Section 7, we will conclude with a discussion.

2 RNA-Seq: A primer

In this section we are motivating our interest in RNA, and what we hope to achieve by an improved understanding of how a disease or treatment affects RNA. For this, we explain some key biological terms in Section 2.1. In Section 2.2 we are then outlining the technology behind obtaining RNA-Seq Data. In Section 2.2.3 we are describing the unique characteristics of RNA-Seq Data. Lastly, in Section 2.3 we will introduce the dataset with which we will illustrate the methods and on which our simulation study will be based.

2.1 The Central Dogma of Molecular Biology

Let us first introduce some basic biological terminology. All life is constructed of basic units called cells. There are two cell-types: Prokaryotic cells and eukaryotic cells. Prokaryiotes are organisms consisting of a single cell - e.g. bacteria. Plants, animals and fungi are all eukariotic. In such complex organisms, cells specialize into different cell types. For humans, these include skin cells, muscle cells, neurons, blood cells and so on. Most importantly however, eukariotic cells consist of several compartments. One of which is the nucleus, which contains Desoxyribonucleic Acid (DNA). DNA is carrying genetic information on various biological processes, and is thus an essential part of the development, functioning and growth of all multi-cellular organisms. The exact role of DNA in this process is described by the Central Dogma of Molecular Biology, which we are going to review now.

(Re-)generation of cells requires a series of events known as the Cell Cycle. An essential part of this cycle is the transfer of genetic information. This transfer is described by the Central Dogma of Molecular Biology. The Dogma is illustrated in Figure 1.

The first step of the central Dogma is DNA-replication. During replication, the two strands of DNA are separated. For each of the two strands, a copy of its counterpart is then produced. The resulting copies are thus formed from one of the original strands and one synthesized strand. After DNA replication, a copy of the DNA is transcribed into RNA. Parts of the resulting DNA can encode proteins. These molecules are called messenger RNA (mRNA). Other segments of DNA copied into RNA are called non-coding RNAs (ncRNAs). The next step described by the Central Dogma is translation. Translation involves the decoding of mRNA in a Ribosome. After decoding, proteins can be synthesized, for this,

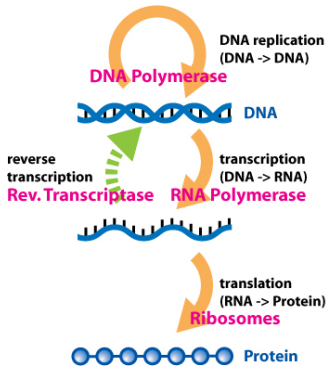


Figure 1: The central dogma of molecular biology [15].

a specific amino-acid chain (polypeptide) is produced. The polypeptide is then *folded* into an protein which can then perform its cellular functions. In summary, RNA thus encodes genetic information used in the synthesis of proteins. These proteins are indicative of underlying cellular processes. Understanding RNA can help us understand these processes. In the next chapter, we will take a closer look at what constitutes RNA and how we can quantify that information.

2.2 RNA and RNA-Seq Technology

2.2.1 RNA

Let us first explain a little bit more what RNA is.

RNA is a molecule that has various biological roles, some of which we mentioned above. It is assembled as a chain of nucleotides, which in turn consist of nucleobases. There are five primary nucleobases: Adenani (A), Cytosine (C), Guanine (G), Thymine (T) and Uracil (U). The are often referred to by their first letter. The nucleobases of A, G, C, and T are found in DNA while A, G, C, and U are found in RNA. These bases will become important when we attempt to quantify genetic information, which we will describe in the next section.

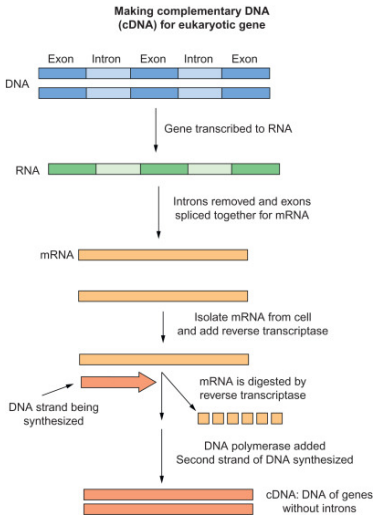


Figure 2: Formation of a cDNA Library [23].

2.2.2 RNA-Seq

There are various technologies for RNA-Seq. However, we will focus on describing the one used by our data. This approach is at the same time perhaps the most popular one. This technology relies on the Illumina-Platform for sequencing. However, many of the steps are very similar or identical for other technologies used. The process of sequencing RNA can be roughly divided into two steps: Library preparation and sequencing. Library preparation describes the process necessary to prepare the biological sample so that a machine can quantify its genomic material. The term library refers to a collection of DNA fragments. In the case of RNA-Seq, we want to obtain a complementary DNA (cDNA) library. The process of obtaining such a library is depicted in Figure 2. The steps are as follow

1. After the samples have been obtained, we first isolate the RNA from tissue. Furthermore, we chemically reduce the amount of genomic DNA in our sample by mixing it with Desoxyribonuclease. Generally, quality control

is performed to check the amount of degradation the RNA has suffered in this process. A quality measure can be incorporated into the downstream-analysis.

2. Next, if we are only interested in parts of the RNA, we can filter for parts of sequences. This process is called depletion. Commonly, we use Polyadenylation, which is the addition of a poly(A) tail to all RNA transcripts. This is done, as we will then have predominantly coding RNA (exons) in our sample.

3. Next, we need to synthesize cDNA. This is done as DNA is more stable and allows for amplification. This process is called reverse transcription and also occurs naturally as described by the central dogma.

After we prepared our library, the next step is the actual sequencing, which we will describe in the next paragraph.

For complementary cDNA, sequencing involves attaching clusters of cDNA onto a flowcell. The flowcell is a glass slide that is coated with nucleotides. The nucleotides can then be labelled using fluorescent chemicals and read out using high-resolution cameras [37]. After reading out the nucleotides, the dye is washed away. This is repeated for all four nucleotides. The DNA chains are then extended by one nucleotide and we can generate read lengths of 50-500 base pairs (bp) [8]. Each base pair consists of two nucleobases bound together by a hydrogen bond. Each of the reads will then consist of the order of the four nucleobases and will be of the required length. This process is illustrated in Figure 3.

For the sequencing, a decision can be made whether we want Single-end reads or paired reads. In single-end reads, the sequencer reads the fragments only in one direction. In paired-end sequencing, both directions are read which improves the accuracy of the reads. Furthermore, we need to make a decision about the desired length of reads. Longer reads improve the accuracy of the mapping to a reference genome. Thus, both decision affect the quality of the data. However, paired-end reads are generally more expensive and so are longer reads. Depending on the experiment a decision thus need to be made whether resources should be allocated to paired-reads, sequencing depth or increasing the sample size [34] [6].

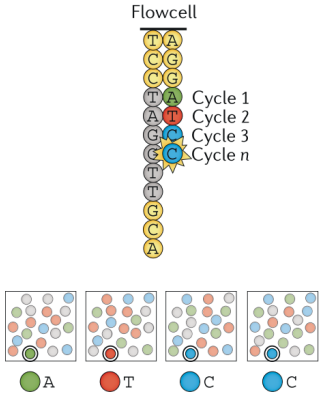


Figure 3: Sequencing using the Illumina workflow [37].

However, the resulting reads are still not ready for analysis. Most commonly, the reads are now mapped to a known transcriptome or an annotated genome. For each genomic location, we then count how many reads are mapped onto it. This count

is our measure of genetic abundance. A good measure of quality of our sample is the percentage of reads mapped. Ideally, we want 70%-90% of reads mapped for a human genome. When we map using a transcriptome, we expect fewer reads to map well. After having described the processes of library preparation and sequencing, we can now discuss what this means for the properties of our data.

2.2.3 Characteristics of RNA-Seq Data

The data-generating process causes the data to have specific properties. Most importantly, we have overdispersed counts that are not identically distributed but require some form of normalization. We will discuss these characteristics of the data now.

For each genomic location, we obtain a count. A common choice for count-data is the poisson-distribution which is parametrized by the rate λ. If X is a poisson random variable, we have $\mathbb{E}(X) = Var(X) = \lambda$. However, the assumption of equality of the Variance and the mean is not appropriate, as we often observe variances larger than the mean. This is called *overdispersion*. The reason for this are the multiple sources of variation in the data. More specifically, we have both biological as well as technical variation. Here, technical variation refers to the variation in the counts caused by the processes of library preparation and sequencing. Clearly there are sources of randomness in the preparation of the sample so that even if we have two exactly identical samples we would - in general - not observe the same counts. However, we also have biological variation, as we generally do not expect two samples of different individuals with the same condition to have exactly the same expression levels. We can model these different sources of variation using a hierarchical model. We will discuss an adequate distribution in Section 3.3. However, even if we would have an appropriate distribution readily available, our data is not identically distributed. This is because we do not have perfect control over how much genetic material is contained in a sample and how it is affected by the process of creating our libraries. We will thus need to *normalize* our data. Techniques of normalization are discussed in the next paragraph.

2.2.4 Normalization

For two samples, the total number of mapped reads are different. This total number is referred to as the *library size* and we need to normalize our data such that the counts are approximately identically distributed. A straightforward technique is to normalize by total library size. For example, in a GLM the library size can be included as an offset term. However, when we expect some of the genes to be differentially expressed this approach can lead to erroneous inference. To illustrate this, suppose we observe counts on two genes from which the first is differentially expressed between two groups. We write $\mathbb{E}(X_j^k)$, $k \in \{A, B\}$ for the expectation of the count of a sample from group k that mapped to gene $j \in 1, \ldots, J$. Thus, $\mathbb{E}(X_1^A) \neq \mathbb{E}(X_1^B)$ but $\mathbb{E}(X_2^A) = \mathbb{E}(X_2^B)$.

We are interested in testing $H_0 : \mathbb{E}(X_j^A) = \mathbb{E}(X_j^B)$ for both genes. Thus, the null hypothesis is false for the first gene and true for the second. Now suppose we observe the following counts:

$$\mathbf{X} = \left[\begin{array}{cc|c} X_{11}^A & X_{12}^A & s_1 \\ X_{21}^A & X_{22}^A & s_2 \\ X_{31}^B & X_{32}^B & s_3 \\ X_{41}^B & X_{42}^B & s_4 \end{array}\right] = \left[\begin{array}{cc|c} 0 & 14 & 14 \\ 0 & 20 & 20 \\ 51 & 17 & 65 \\ 39 & 12 & 59 \end{array}\right] \tag{2.1}$$

where s_i denotes our library size of sample $i \in 1, \ldots, N$. We can see that when we normalize our samples so that they sum to the same constant we distort our data. In our adjusted data, the evidence of differential expression in Gene 1 becomes less extreme. Furthermore, we create false evidence in Gene 2 by effectively halving the counts for group B. We thus need more intelligent techniques of normalization that are robust to differential expression of some genes. Our discussed methods use of of two techniques: TMM and RLE. We summarize TMM [31], as it will introduce us to the general thought-process behind designing a robust normalization procedure and as we will use it for almost all of our discussed methods. For RLE - the normalization method used by DESeq2 - see [25]. For a comparison see [26].

TMM starts by computing two quantities. The gene-wise log-fold changes M_j between all samples and the absolute expression levels A_j. These are defined as follows

$$M_j(i, i') = \log_2 \frac{X_{ij}/s_i}{X_{i'j}/s_{i'}} \qquad A_j = \frac{1}{2} \log_2 \frac{X_{ij} X_{i'j}}{s_i s_{i'}} \tag{2.2}$$

In practice we specify one reference sample s' and compute the normalization factors of all samples against that reference sample. For each of the $n - 1$ samples, we thus obtain M and A-values for all J genes. Both values are then trimmed by discarding high and low quantiles of the observed values. Finally, weights w_{ij} are computed using an estimated variance of the values. Write J^* for the set of genes which are not trimmed. Then, our scaling factors a are computed as follows:

$$\log_2(a(i, i')) = \sum_{j \in J^*} w_j(i, i') M_j(i, i') \Big/ \sum_{j \in J^*} w_j(i, i') \tag{2.3}$$

For the computation of weights and other details see [31] and [28]. For our purposes it suffices to note that we get a reasonable robust normalization procedure as long as we do not expect the majority of genes to be differentially expressed. When we expect many genes to be differentially expressed we can adjust the quartiles of trimming so that more genes genes are trimmed. Lastly, note that there is also normalization by gene length, so that counts between

genes are comparable. This is relevant for some research questions where comparisons between genes are made. However, it is not relevant here, as we are looking at differential expression within a gene. We thus discussed methods of obtaining RNA-Seq and one way to normalize it for detecting differential-expression. We will now discuss methods of filtering that are sometimes used, before introducing the dataset used throughout the thesis.

2.2.5 Filtering

Often, not all genes are included in the testing pipeline. This has multiple reasons. First, by excluding genes that are unlikely to be differentially expressed we decrease the multiple-testing burden of our procedure. We will explain this in Section 3. Secondly, very low counts might be purely due to technical noise.Genes with low counts might thus not actually be expressed in a given sample. A common approach is therefore to filter out genes for which not a sufficient number of genes are expressed. Thus, a threshold is defined using two pre-defined numbers: The minimum number of counts for which we view a gene as expressed and the minimum number of samples for which we want a gene to be expressed. Genes that do not meet this threshold are filtered out. However, it is unclear how and if the filtering should be viewed as part of the statistical model. We view filtering out as a failure to reject the null hypothesis. Ideally we want our models to be able to control the Type-I error rates on the complete set of unfiltered genes, such that filtering is an optional step. However, we will see that some methods only work reliable when apply filtering. We will thus compare the methods for both filtered and unfiltered data.

Let us now introduce the dataset we will be using to illustrate the models, and as a basis for our simulation study.

2.3 Introduction to Illustratory Dataset

We chose to use data from the Cancer Genome Atlas (TCGA) [39]. The TCGA database holds more than 20,000 samples of more than 33 cancer types [38]. There are two reasons for this choice. First, the large sample size allows subsetting and thus analysing results for different (smaller) sample sizes. Secondly, cancer is perhaps the disease for which the analysis of RNA-Seq is the most promising, as it might help to develop novel therapies and diagnostic methods.

From TCGA, we chose the TCGA-LIHC dataset of Liver Hepatocellular Carcinoma (HCC). Important studies of RNA-Seq analysis of HCC are [16][18]. The TCGA-LIHC Dataset consists of 421 samples, from which 50 are paired observations. While it is interesting and often often advantageous to use paired data, it is not the focus of this thesis. We will thus delete 50 samples and obtain 371 samples, from which 50 are liver-cells without and 321 come from the tumor. The dataset comes with additional clinical data, such as the sex, age and race of the patients. This information can be incorporated into the model as covariates, as it also might affect genetic expression. However, we will focus

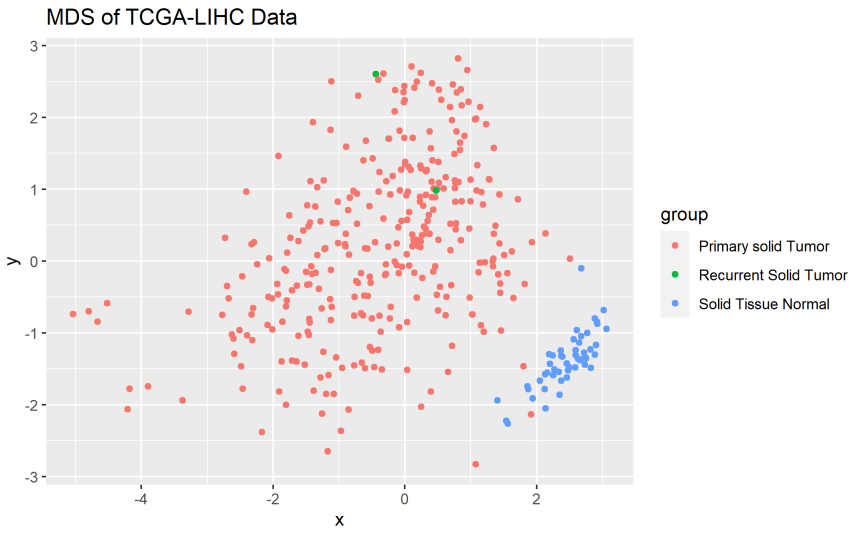


Figure 4: MDS plot of distances between genes. The distances represent the leading log2-fold-changes, which are defined as being the top-k largest log2-fold changes between those two samples. We chose k to be 500.

on two-group comparisons between cancer and control and thus discard other clinical data.

Our data was sequenced using the Illumina platform. Paired-ends were used. Alignments was done using STAR [10] to reference genome hg38.

A first impression of the dataset can be obtained from the Multi-Dimensional-Scaling (MDS) Plot (Figure 4). The data is scaled so that distances between samples approximate the leading $\log_2$-fold-change. The fold changes computed are equivalent to the computation of the M-values described in equation 2.2. However, only the top 500 fold changes are used. The distance measure is the root-mean-squared-average of these 500 fold changes. First, we notice that the two groups indeed form clusters, where the cluster of the normal tissue is much more homogenous than the cluster of the primary solid tumor. However, we have more samples from the tumor-group. MDS minimizes the *strain* i.e. attempts to represent the observed distances as well as possible in two-dimensions. The heterogeneity of the tumor group might thus be a result of having a larger sample-size for that group and the MDS-algorithm consequently focussing on representing the distances between these samples as accurately as possible. There is a third group in the data, representing recurrent tumors. We remove this group for further analysis as we are only interested in a two-group comparison. We will now move on to the preliminaries, where we define some statistical concepts necessary for the rest of this thesis.

3 Preliminaries

In this section, we introduce some necessary statistical concepts. We will begin with defining Type-I and Type-II errors. Using these definitions, we define exact tests. We then outline methods for Multiple Testing. Then, we review probability distributions for count data and some of their properties. Furthermore, we review some concepts of GLMs. Lastly, we introduce the concept of shrinkage.

3.1 Type-I and Type-II errors

The definitions of Type-I and Type-II errors are well known. However, let us briefly recall them here. Rejecting H_0 when it is true is called a Type-I error. Failing to reject H_0 when it is false is called a Type II error. The relationship between the two becomes more clear when looking at the power function of the test. Let us briefly define it here.

Definition 1 (Power of a Test)**.** *Suppose we observe data* $\mathbf{X} = (X_1, \ldots, X_n)$ *Let* R *be the region for which we reject a test. The power-function of the test is defined as*

$$\beta(\theta) = \mathbb{P}_\theta(\mathbf{X} \in R) \tag{3.1}$$

Suppose we want to test $H_0 : \theta \in \Theta_0$ vs. $H_A : \theta \in \Theta_0^c$ where Θ_0^c denotes the complement of Θ_0. The ideal power function is then 0 for all $\theta \in \Theta_0$ and 1 for all $\theta \in \Theta_0^c$. Using the power-function, we can also make the probability of Type-I and Type-II errors more explicit

$$\mathbb{P}_\theta(X \in R) = \begin{cases} \text{Probability of Type I error} & \text{when } \theta \in \Theta_0 \\ 1 - \text{Probability of Type II error} & \text{when } \theta \in \Theta_0^c \end{cases} \tag{3.2}$$

Definition 2 (Size of a Test)**.** *A test with power function* $\beta(\theta)$ *is said to have level* α *when*

$$\sup_{\theta \in \Theta_0} \beta(\theta) \leq \alpha \tag{3.3}$$

However, we have not specified yet how the rejection Region R is obtained. In many parametric methods, X is assumed to come from a specific distribution. Using this assumption, a rejection region can be computed depending on the specified level of the test α. However, such a test is often only asymptotically valid i.e. write X_n for a sequence of random variables as before and $\beta_n(\theta)$ for its power-function. Then, a test is said to asymptotically have level α if

$$\lim_{n \to \infty} \sup_{\theta \in \Theta_0} \beta_n(\theta) \leq \alpha \tag{3.4}$$

When using the asymptotic distribution of a test-statistic to compute the rejection-region, the Type-I error rate might be above or below the *nominal*

level α. We call a test *conservative* if the rejection rate lies below the nominal level and *anti-conservative* if it lies above it. We call a test *exact* if the test maintains the Type-I error rate for all sample-sizes. Note that there is thus an important distinction between exact-tests and non-parametric tests. An exact test might rely on assumptions - such as the data coming from a parametric distribution. On the other hand, a non-parametric test might only have asymptotic guarantees. Robustness to violated assumptions, exactness and power are thus separate issues that also need to be investigated separately. Having defined the key properties of individual tests, we are now going to discuss how we can generalize control of Type-I error rate when we conduct many hypothesis tests in the same study.

3.2 Multiple Testing

Suppose we want to test a set of hypotheses $\mathcal{H} = (H_1, \ldots, H_m)$. We do not know how many of these hypotheses are true. Write $\mathcal{T} \subseteq \mathcal{H}$ for the set of true hypothesis and $\mathcal{F} = \mathcal{H} \backslash \mathcal{T}$ for the set of false hypothesis. Write R for the set of rejected hypothesis. Note that R is a random variable. Suppose our test is exact and we perform each test at level α. Then, the probability that we falsely reject at least one hypothesis grows with the number of tests we perform. This is easy to see when all null-hypothesis are wrong and when we assume that they are independent. Then, the probability of at least one false rejection is

$$\mathbb{P}(|\mathcal{R} \cap \mathcal{F}| > 0) = 1 - (1 - \alpha)^m \tag{3.5}$$

It is thus better to choose a criteria for the whole set of tested hypotheses rather than individual criteria. There are two common choices: Family-Wise error rate (FWER), which directly controls $\mathbb{P}(|\mathcal{R} \cap \mathcal{F}| > 0)$. However, when m gets very large we lose a lot of power. A popular alternative is the False Discovery Rate (FDR), which controls the expected proportion $\mathbb{E}\left(\frac{|\mathcal{R} \cap \mathcal{F}|}{|\mathcal{R}|}\right)$. We are briefly introducing these two approaches in the next two sections.

3.2.1 Controlling the FWER using the Bonferroni-Correction

Let $\mathcal{H} = (H_1, \ldots, H_m)$ be the family of hypotheses we want to test. Suppose we want to control $\mathbb{P}(|\mathcal{R} \cap \mathcal{F}| > 0) \leq \alpha$. We can control this by rejecting each individual hypotheses only when $p_i \leq \alpha/m$, this is called the Bonferroni Method.

Theorem 1. *Using the Bonferroni Method, the probability of falsely rejecting at least one null hypothesis is less than or equal to* α

Proof. Let $m_0 = |\mathcal{T}|$ be the size of the set of true null hypotheses, then:

$$\mathbb{P}(|\mathcal{R} \cap \mathcal{F}| > 0) = \mathbb{P}\left[\bigcup_{i=1}^{m} (p_i \leq \frac{\alpha}{m})\right] \leq \sum_{i=1}^{m} \mathbb{P}(p_i \leq \frac{\alpha}{m}) = \frac{m_0 \alpha}{m} \leq \alpha \tag{3.6}$$

Here, the first inequality follows from Boole's inequality. The rest follows because $m_0 \leq m$. □

When m is very large, and when the tests are very correlated, the Bonferroni correction can lead to a very conservative test. Instead, we can control the False Discovery Rate.

3.2.2 Controlling the FDR using the Benjamini-Hochberg

Instead of controlling the probability of at least one false rejection we can control the expected proportion of false rejections in the set of all rejections. The most popular method to do so is Benjamini-Hochberg [4].

Define the False Discovery Proportion as

$$FDP = \begin{cases} \frac{|\mathcal{R} \cap \mathcal{F}|}{|\mathcal{R}|} & \text{if } R > 0 \\ 0 & \text{if } R = 0 \end{cases} \tag{3.7}$$

Note that the FDP is an unknown, as it depends on $\mathcal{F}$, and random as it depends on $\mathcal{R}$. However, we can estimate it's expectation $\mathbb{E}(FDP)$, which is called the False Rejection Rate. In practice, we want a procedure that allows us to have control over this expectation. I.e. that for all $\alpha \in [0, 1)$, $\mathbb{E}(FDP) \leq \alpha$. This can be done using the Benjamini-Hochberg (BH) Method, which works as follows:

1. Let $p_{(1)} \leq \cdots \leq p_{(m)}$ denote the ordered $p-$values.
2. Find the largest i, such that $p_i \leq \frac{i\alpha}{m}$
3. Reject all null hypotheses corresponding to $p_{(1)}, \ldots, p_{(i)}$

Theorem 2. *Using the Benjamini-Hochberg method we have*

$$FDR = \mathbb{E}(FDP) \leq \frac{m_0 \alpha}{m} \leq \alpha \tag{3.8}$$

Proof. See [4] □

When the tests are not independent or when a large proportion of our hypotheses are false Benjamini-Hochberg becomes conservative. There are alternative, more complex testing procedures for both FWER as well as FDR that take this into account. For an overview see [12]. However, for this thesis more detail is not necessary. Let us now review common probability distributions for count data.

3.3 Probability Distributions of Count-Data

The perhaps best known distribution of count-data is the binomial distribution. It is the number of successes in a sequence of n independent Bernoulli-trials. Suppose we map a total of n-fragments to our genome, where n is fixed and known before-hand. We can then see that each for each individual gene, the count follows indeed a binomial distribution, i.e. $X_{ij} \sim Bin(p_j, n)$ where $n =$

$\sum_{j=1}^{J} X_{ij}$. Each sample thus follows a multinomial distribution with parameters $p_1, \ldots, p_J, n$. However, n is usually very large and p_j very small, n is itself a random variable. Thus, the poisson distribution might be a good approximation. To see this, note the following: Fix $\lambda = np$, as $n \to \infty$ we can approximate the distribution of each gene using a poisson distribution which has the following form:

Definition 3 (Poisson Distribution)**.** *A random variable X follows a Poisson Distribution with parameter $\lambda > 0$ if it's Probability Mass Function (PMF) is given by*

$$f(k; \lambda) = \mathbb{P}(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{3.9}$$

It's expected value and variance are given by $\lambda = \mathbb{E}(X) = Var(X)$

However, in many RNA-Seq experiments, the observed Variance of the counts is larger than it's mean. This phenomenon is called overdispersion. We already mentioned the likely cause of this overdispersion: The two separate sources of variation. It is thus natural to define a hierarchical model which models the rate λ as a random variable. The most natural mixture distribution is Gamma, which yields a Negative Binomial random variable.

Lemma 1 (Gamma-Poisson Mixture)**.** *Let λ be a Gamma random variable with shape r and scale $p/(1-p)$ and $Y|\lambda \sim Poisson(\lambda)$. Then $Y \sim NB_{(r,p)}(r, p)$.*

However, we reparametrize the NB-distribution to a more convenient form.

Definition 4 (Negative Binomial Distribution.)**.** *Let Y be an Negative Binomial (NB) random variable with mean μ and dispersion parameter ϕ, we write $Y \sim NB(\mu, \phi)$. It's probability mass function (pmf) is defined by*

$$f(y; \mu, \phi) = \mathbb{P}(Y = y) = \frac{\Gamma(y + \phi^{-1})}{\Gamma(\phi^{-1})\Gamma(y+1)} \left(\frac{1}{1+\mu\phi}\right)^{\phi^{-1}} \left(\frac{\mu}{\phi^{-1}+\mu}\right)^{y} \tag{3.10}$$

We then have $\mathbb{E}(Y) = \mu$ and $Var(Y) = \mu + \phi\mu^2$.

We will use this parametrization for the rest of the thesis. We can change parameterization using the following formulas:

$$\mu = \frac{pr}{1-p} \qquad \phi = 1/r \tag{3.11}$$

Using the parametrization in terms of p and r, let us note one last property of the NB-distribution.

Lemma 2 (Additivity of NB-Random Variables.)**.** *Let $X \sim NB(r, p)$ and $Y \sim NB(s, p)$ then, $X + Y \sim NB(r + s, p)$.*

Having defined the necessary probability distributions, we can now move on to recapitulate the exponential family of distributions and their role in Generalized-Linear Models (GLMs).

3.4 GLMs and the Exponential Family of Distributions

Recall that the random component of a GLM consists of a random variable whose distribution is a member of the exponential family defined below:

Definition 5. *A family of distributions is said to belong to a exponential family if it can be written in the form*

$$f(y;\theta,\phi) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right] \tag{3.12}$$

The parameter θ is called the natural parameter and ϕ is called the dispersion parameter. If $a(\phi) = 1$ the natural exponential family arises. This family has the form

$$f(y;\theta) = h(y)\exp[y\theta - b(\theta)] \tag{3.13}$$

Given such a distribution, a linear predictor $\eta = \mathbf{X}\beta$ and a link function g such that $\mathbb{E}(Y|X) = \mu = g^{-1}(\eta)$ we have the three elements of a GLM. Let us now give examples how the poisson and negative-binomial distribution can be shown to belong to this family and be modelled using a GLM.

Example 1 (Poisson in the Exponential Family)**.** *Recall that Y follows a Poisson distribution if it has PMf*

$$f(y;\mu) = \frac{e^{-\mu}\mu^y}{y} = \exp[y\log\mu - \mu - \log(y!)] \tag{3.14}$$

$$= \exp[y\theta - \exp(\theta) - \log(y!)] \tag{3.15}$$

The natural parameter in a regression model is thus $\theta = \log\mu$. If we use take Equation (3.12) with $b(\theta) = \exp(\theta)$, $a(\phi) = 1$ and $c(y,\phi) = -\log(y!)$ we see that it is indeed part of the exponential family.

Example 2 (NB-Distribution as a Member of the Exponential Family)**.** *Recall the pmf of the NB distribution with parameterers μ and ϕ. For ease of notation we use $r = 1/\phi$:*

$$\begin{aligned}
\mathbb{P}(X = x) &= \frac{\Gamma(r+x)}{x!\Gamma(r)}\left(\frac{r}{r+\mu}\right)^r\left(\frac{\mu}{r+\mu}\right)^x \\
&= \frac{\Gamma(r+x)}{x!\Gamma(r)}\exp\left(\log\left[\left(\frac{r}{r+\mu}\right)^r\left(\frac{\mu}{r+\mu}\right)^x\right]\right) \\
&= \frac{\Gamma(r+x)}{x!\Gamma(r)}\exp\left(r\log\left(\frac{r}{r+\mu}\right) + x\log\left(\frac{\mu}{r+\mu}\right)\right) \\
&= \exp\left(x\log\left(\frac{\mu}{r+\mu}\right) + r\log\left(\frac{r}{r+\mu}\right) + \log\left(\frac{\Gamma(r+x)}{x!\Gamma(r)}\right)\right)
\end{aligned}$$

The term involving $\Gamma(r+x)$ does not allow the factoring required by the exponential family. The negative binomial is thus only a member of the exponential family when r is fixed.

If our dependent variables follow a distribution that is part of the exponential family, we can use a Generalized Linear Model to describe the data. For a GLM-model, the canonical link function is

$$\theta_i = \sum_{j=1}^{p} \beta_j x_{ij} \tag{3.16}$$

Taking the logarithm of our distribution we arive at our log-likelihood, defined as $L = \sum_{i=1}^{n} L_i$ where $L_i = \log(f(y_i; \beta, \phi))$. Taking the logarithm of Equation 3.12 yields the formula of the contribution to the log-likelihood of an observation from the exponential family

$$L_i = \frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \tag{3.17}$$

For Maximum-Likelihood-Estimation we need to differentiate w.r.t. β_j which we can do using the chain rule

$$\frac{L_i}{\beta_j} = \frac{\partial L_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} \tag{3.18}$$

$$= \frac{y_i - \mu_i}{a(\phi)} \frac{a(\phi)}{Var(y_i)} \frac{\partial \mu_i}{\partial \eta_i} x_{ij} = \frac{(y_i - \mu_i) x_{ij}}{Var(y_i)} \frac{\partial \mu_i}{\eta_i} \tag{3.19}$$

For a Poisson Loglinear Model, we have $\eta_i = \log \mu_i$. Thus, $\mu_i = \exp(\eta_i)$ and $\partial \mu_i / \partial \eta_i = \exp(\eta_i) = \exp(\log(\mu_i)) = \mu_i$ the *score contributions* are thus

$$v_i = (y_i - \mu_i) x_{ij} \tag{3.20}$$

Next, let us compute the score contributions for the Negative Binomial Model.

From Example 2 it follows that for fixed r we have the natural parameter

$$\theta_i = \log\left(\frac{\mu}{r + \mu}\right) \tag{3.21}$$

However, often the log-link is used. We then have $\eta = \log \mu$ and again $\partial \mu_i / \partial \eta_i = \mu_i$. Lastly, we again have $a(\phi) = 1$. Thus, the score contribution of our $i-$th observation is

$$v_i = \frac{L_i}{\beta_j} = \frac{(y_i - \mu_i) x_{ij}}{Var(y_i)} \frac{\partial \mu_i}{\eta_i} = \frac{(y_i - \mu_i) x_{ij} \mu_i}{\mu(1 + r\mu)} = \frac{(y_i - \mu_i) x_{ij}}{1 + r\mu_i} \tag{3.22}$$

We can thus see that compared to the poisson distribution that the influence of a large difference between y_i and μ_i on the sum of score-contributions can be controlled via r.

However, usually r is not known beforehand. In order to fit the model we use the iterative procedure implemented in `MASS::glm.nb()`. This process first fits a Poisson-GLM, then estimates $r|\mu$, then fits a NB-GLM given this estimate of r and repeats this until convergence.

The last topic of our preliminaries will cover how we can make use of the highly-parallel structure of RNA-Seq data i.e. how we can share information across genes to improve their properties.

3.5 Shrinkage and Empirical Bayes

The data we obtain from genetic experiments has a very unique shape. We often observe very few samples and want to estimate parameters for many genes. This is a classical scenario where borrowing strength between genes can be beneficial. Borrowing strenght is a specific kind of shrinkage. In this section, we will first motivate shrinkage using the James Stein Estimator. We will then relate James-Stein type shrinkage to Empirical Bayes Methods and finally how we can make use of this for genetic data.

Shrinkage was first theoretically motivated by James and Stein [17]. They gave a simple example how shrinkage can improve the performance of estimators.

Example 3. *Let $\theta_1, \theta_i, \dots \theta_n = \boldsymbol{\theta}$ be a vector of parameters that we want to estimate. Let $X \sim N(\boldsymbol{\theta}, \mathbf{I})$. Suppose we only have a single realization of X. We want to find an estimator of $\boldsymbol{\theta}$ that minimizes the Mean Squared Error, i.e.*

$$\mathbb{E}(||\hat{\theta} - \theta||) = \mathbb{E}(\sum_{i}^{n} (\hat{\theta}_i - \theta))^2 \tag{3.23}$$

James and Stein showed that the individual Least-Squares estimator $\hat{\theta}_{LS} = X$ is not the minimizer. But that it is outperformed by the James-Stein Estimator defined as follows:

$$\hat{\theta}_{JS} = \left(1 - \frac{(n-2)}{||X||^2}\right) X \tag{3.24}$$

Note that the first term is always smaller than 1. The James-Stein Estimator thus shrinks the individual estimates towards zero. Their findings show that when estimating many parameters simultaneously, minimizing the error of individual estimates does not imply minimizing the error of all estimates together. For minimizing the overall error, shrinkage can be beneficial.

Shrinkage has been generalized for shrinkage to other points than zero and for shrinkage of other estimators. Many types of shrinkage can also be motivated using Bayesian statistics. Often, Bayesian-Type shrinkage is employed in an otherwise completely frequentist procedure. A typical example are Empirical Bayes Estimators, which will be used through this thesis.

In order to explain what Empirical Bayes is, let us first recall Bayes' Theorem.

Theorem 3. *Suppose we observe data Y from a model with parameters θ. We are interested in obtaining the conditional probability $\mathbb{P}(\theta|Y)$. We can do this when specifying a prior $\mathbb{P}(\theta)$, as the data distribution $\mathbb{P}(Y|\theta)$ is given by the model. This is done using Bayes' rule:*

$$\mathbb{P}(\theta|Y) = \frac{\mathbb{P}(Y|\theta)\mathbb{P}(\theta)}{\mathbb{P}(Y)} \tag{3.25}$$

Where $\mathbb{P}(\theta)$ is called the prior, $\mathbb{P}(Y|\beta)$ is the likelihood and $\mathbb{P}(Y)$ is a normalizing constant.

Normally, the prior distribution is fixed before any data is observed. However, when we estimate many parameters θ in parallel, we have the possibility to estimate a prior from the data. This procedure is called *empirical Bayes*. The prior will then reduce the variance of our estimator by increasing it's bias. It is thus a form of shrinkage [11]. In this view, it is thus natural to approximate a fully Bayesian Hierarchical Model by estimating a prior for all genes, from which the parameters of the individual genes are sampled. We will see that this methodology is taken by all of the most popular parametric methods that we will discuss in the next section.

4 Parametric Approaches to Testing Differential Expression

In this section we are going to review three of the most popular methods of RNA-Seq Data. We will start with perhaps the simplest, that uses a normal approximation and then performs moderated $t-$tests. We then introduce the proposed Exact-Test for the NB-Distribution as implemented in edgeR. Lastly, we will introduce the shrinkage-based model underlying DESeq2. For all three methods, we focus on the statistical model and refer to the papers for more details about parameter estimation.

4.1 Limma's Moderated t-test

The easiest way to test our hypothesis is the use of a two-sample t-test. This was first motivated for Microarray data - a precursor of RNA-Seq. Microarray's measured colour intensities and it was thus commonplace to model the $\log_2$-transformed data as continuous. From the literature about the analysis of microarray data another important finding was taken - the use of shrinkage in the form of empirical bayes. This was first motivated in [24], where the posterior odds of differential expression for a two-colour microarray experiment are derived. It was extended to general microarray experiments in [35], and again extended to RNA-Seq data in [30]. We will first introduce the moderated $t-$test and show it's validity for normally distributed random variables.

Let $Y \in \mathbb{R}^{n\times g}$ be our matrix of observations and $X \in \{A, B\}^n$ our model matrix encoding the two groups. Write Y_j for the observations of the $j-$th gene where $j = \{1, 2, \ldots, G\}$. Write $Y_{ij}^k = Y_{ij}|X_i = k$ to index the two groups where $i = \{1, 2, \ldots, n\}$. If we assume that $Y_{ij}^k \sim N(\mu_j^k, \sigma_j^2)$ we can test the null hypothesis of $H_0 : \mu_j^0 = \mu_j^1$ using the standard $t-$test.

Let $\bar{X}_j^0$ and $\bar{X}_j^1$ to be the two sample means and s_j the pooled sample variance. For ease of notation suppose that the two groups have a equal number of samples i.e. $n^0 + n^1 = n$. In a linear model we then have $\hat{\beta}_j = \bar{X}_j^0 - \bar{X}_j^1$. Under the null hypothesis $H_0 : \beta_j = 0$ we have that

$$t_j = \frac{\hat{\beta}_j}{s_j\sqrt{n}} \sim t_{n-1} \tag{4.1}$$

where t_v denotes the t-distribution with v degrees of freedom. The distribution of our test-statistic follows by Cochrans theorem. Note that the $t-$test is exact when the normal assumption holds.

However, when doing multiple thousand tests with small sample sizes we expect some of the test statistics to be very large caused by very small, possibly underestimated sample variances. Thus shrinking the test-statistics was proposed. Perhaps the easiest shrinkage is to add a small constant to the denominator, effectively shrinking the $t-$statistic towards zero [11]. The test statistic then becomes

$$t_j = \frac{\hat{\beta}_j}{a_o + s_j\sqrt{n}} \sim t_{n-1} \tag{4.2}$$

While this type of shrinkage can lead to good empirical results, it is not well motivated by a model. Motivation can be given using Empirical Bayes, as proposed in [35]. Here, a hierarchical model is employed. Note that the conjugate prior on the variance of a normal distribution with known mean is the inverse-χ^2 distribution. Thus, it is proposed that $1/\sigma^2$ is distributed as follows:

$$1/\sigma_j^2 \sim \frac{1}{d_0 s_0^2}\chi^2_{d_0} \tag{4.3}$$

Here, d_0 and s_0^2 are thus parameters of the prior that need to be chosen or estimated. Using this, a shrunken estimator $\tilde{s}_j^2$ of σ_j^2 is then defined. Given the fact that $\tilde{s}_j^2$ has a scaled inverse-chi-squared distribution, we can compute it's posterior mean

$$\tilde{s}_j^2 = \mathbb{E}(\sigma_g^2|s_j^2) = \frac{d_0 s_0^2 + d_j s_j^2}{d_0 + d_j} \tag{4.4}$$

This directly leads to a moderated $t-$statistic

$$\tilde{t}_j = \frac{\hat{\beta}_j}{\tilde{s}_j^2\sqrt{\frac{2}{n}}} \tag{4.5}$$

Under H_0, the moderated t-statistic follows a t-distribution with $d_g + d_0$ degrees of freedom. The added degrees of freedom reflect the information borrowed between genes. The validity of the choice of priors for the distribution of the test-statistic is shown in [35].

However, this model now requires us to make choices about the hyperparameters d_0 and s_0^2. This estimation is done by equating the empirical and expected values of the first two moments of $\log(s_j^2)$. The mathematical details can be found in [35]. For our purpose it suffices to note that shrinkage is reduced when the variance of s_j^2 is large. This method is implemented in the `Limma` package [30].

However, as described so far the method is not valid for RNA-Seq Data as our observed are counts while our assumption specifies a normal-distribution. To take this into account, Limma transforms the data by calculating log-cpm-values of all observations. The log-cpm-value of an observation is obtained by summing over all genes, then dividing by 1 million and finally taking the log. However, we might still be left with heterscedasticity. To deal with this,

we can estimate a function $Var(\mu_{ij}) = f(\mu_{ij})$ that specifies the relationship between the mean and the variance of a function. Two ways of incorporating this function into the model are proposed and implemented in Limma. The first is called `Limma-Trend`, which uses $f(\mu_i)$ to specify the parameters of the prior as a function of μ_i. The second approach is call `Limma-Voom` which incorporates f using weighted least-squares. We will focus on latter, as it is the more popular one. However, the functioning of the former should be clear after we discussed the shrinkage-procedure of EdgeR and DESeq2. We will now Limma-Voom it in more detail.

4.1.1 Borrowing Strength between Genes using Quasi-Likelihood Methods

Suppose we have n independent observations $X_i, \ldots, X_n$. In a GLM with $\eta_i = g(\mu_i) = \sum_j \beta_j x_{ij}$ we have likelihood equations

$$\sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{V(\mu_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right) = 0 \tag{4.6}$$

Note that in the linear model we simply have $g(\mu_i) = \mu_i$ and thus $\frac{\partial \mu_i}{\partial \eta_i} = 1$. The likelihood equations - and consequently the parameter estimates - only depend on the variance function $V(\mu_i) = Var(y_i)$ and the structure of the model as specified by η. Quasi-likelihood methods specify the link function g, but instead of assuming a particular distribution for y_i they only assume a mean-variance trend [1]. Note that the variance function is essentially a weighting function, specifying how much weight is given to the denominator. Model fitting is done by incorporating these weights into the fitting procedure.

In our application, we can estimate this mean-variance trend using all of our genes. This is done as follows:

Let y^*_{ij} be the normalized log-cpm transformed value of y_{ij}. We first estimate a linear model and obtain $\hat{y}_{ij} = X\hat{\beta}$. Using $\hat{y}_{ij}$ we can thus get parameter estimates for the standard deviation σ_j. We can then fit a non-parametric regression model such a Locally Estimated Scatterplot Smoothing (LOESS) to obtain an estimate for V. Using V, we can obtain a prediction for the standard-deviation of each observation i. These predictions are then incorporated into our moderated $t-$tests as *weights*.

We have plotted six estimated functions V from the TCGA-LIHC data in Figure 5. The functions have different parameters for span. Three of them are based on the unfiltered dataset and three are based on a filtered dataset. We can see that the choice of span has a large influence on the estimated trend and consequently on the weights incorporated into our model. Without filtering, the default span of 0.5 underfits, which could potentially lead to Type-I errors. However, a badly fitting trend can equally cause Type-II errors. Furthermore, we can see that for the fits without filtering we can see a drop for very low counts. This is caused by the discreteness of the counts which leads to an

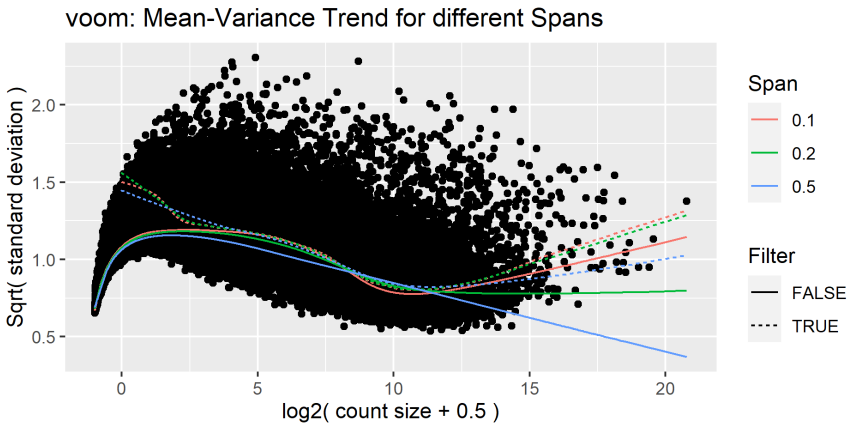


Figure 5: Mean Variance Trend as computed by voom for different span-parameters of the smoothing line with and without filtering.

underestimation of the variance. In fact, we will later see that Limma is reliant on filtering to control Type-I error rates, as otherwise the underestimation of the variance will lead to many false rejections of lowly expressed genes. Let us now finish the discussion of the method by analyzing it's advantages and disadvantages.

First of all, a big advantage of Limma is it's simplicity which also leads to very fast model fitting and evaluation. For genes with high counts, for which the normal approximation is appropriate, we thus have a simple but flexible model that allows the inclusion of covariates and other more complex experimental designs and whose test is exact and will known. The main disadvantage is the reliance on filtering as the model has inflated Type-I error rates for lowly expressed genes. However, we argue that also the reliance on the tuning parameter *span* is a disadvantage, as it influences the model and which hypotheses are rejected. We argue that the presence of such a tuning parameter is not ideal as it can be chosen to confirm certain beliefs about a given dataset. Thus, findings might not be reproducible when other parameters are chosen. Furthermore, modelling the count-data using a more appropriate distribution might lead to a more powerful testing procedure. We will now discuss a model that uses the negative-binomial distribution.

4.2 EdgeR's Exact Test

EdgeR models the data using a Negative-Binomial distribution. However, it thus relies on the esitmation of the overdispersion parameter. A first step is obtaining estimates for the overdisperision parameter ϕ. As the employed test is highly sensitive to misestimation of ϕ some attention should be given to the various estimators.

4.2.1 Estimators of ϕ

EdgeR uses a Conditional Maximum Likelihood Estimator of ϕ, as it has favourable properties even for very small samples [33]. The CML is defined as ϕ that maximizes

$$l_{Y|Z=z} = \left[\sum_{i=1}^{n} \log\Gamma(y_i + \phi^{-1})\right] + \log\Gamma(n\phi^{-1}) - \log\Gamma(z + n\phi^{-1}) - n\log\Gamma(\phi^{-1}) \tag{4.7}$$

Here, z is the total number of counts observed for a given gene. Note that z is a sufficient statistic for the NB-model on which we condition.[1] However, the CML only exists when $y_1, \ldots, y_n$ are identically distributed. In the setting with different library sizes this is not the case. Thus, we need to normalize our data so that it is approximately i.i.d.. Let s_i be the library size of sample i. Compute the geometric mean of the library sizes as $s^* = (\prod_{i=1}^{n} s_i)^{1/n}$. We can then use the following algorithm for normalization:

Algorithm 1 Quantile Adjustment Method

Require: $\epsilon > 0$
Require: $\hat{\phi}_0 > 0$
Require: $j = 1$
while $|\hat{\phi}_j - \hat{\phi}_{j-1}| > \epsilon$ **do**

1. Fix ϕ_{j-1} and estimate λ_j
2. For each y_i, compute the percentiles using the NB-distribution with mean $m_i\lambda_j$ as

$$p_i = \mathbb{P}(Y \leq y_i; s_i\lambda_j, \phi_j) + \frac{1}{2}\mathbb{P}(Y = y_i; s_i\lambda_j, \phi_j)$$

3. Compute pseudodata from an NB distribution with mean $s^*\lambda$ and dispersion ϕ having quantiles p_i using a linear interpolation
4. Calculate ϕ_j using the CML on the pseudodata

end while

Note that the pseudodata is continuous and can be as negative as -0.5. However, this is not a problem for the evaluation of equation (8) or its derivatives. After obtaining estimates for ϕ and normalization, we can now do hypothesis-testing, which we are going to discuss now:

[1] Furthermore, note that there is a closed form for $f(x) = \log(\Gamma(x))$ and its derivative - the digamma function. This allows the maximization without numerical problems caused by evaluating $\Gamma(x)$ for large x.

4.2.2 Exact Test of NB-Random Variables

Let $Y_{ik} \sim NB(s^*\lambda_k, \phi)$, for $k \in \{A, B\}$ our group-indicator and s^* our adjusted library size for all samples. Suppose ϕ is known. We want to test $H_0 : \lambda_A = \lambda_B$. By conditioning on the sufficient-statistic, an exact test is available [33]:[2]

Let $\bar{Y}_k$ denote the observed sample means. We can rewrite the null hypothesis as $H_0 : |\log(\bar{Y}_A/\bar{Y}_B)| = 0$. Thus, we can use the test statistic $T = |\log(\bar{Y}_A/\bar{Y}_B)|$. After conditioning on the sample sums we can find the distribution of this test statistic. Let $Z_k = \sum_i^n Y_{ik}$ let $Z = Z_A + Z_B$. Note that by the additivity of NB random variables, these are also NB random variables whose parameters directly follow from the parameters of their parts. We can thus evaluate $\mathbb{P}(Z_A, Z_B) = \mathbb{P}(Z_A)\mathbb{P}(Z_B)$. We can calculate $p-$values as follows:

Define

$$\mathcal{O} = \{(z_A, z_B) : z_A + z_B = z\} \tag{4.8}$$

$$\mathcal{I} = \{(z_A, z_B) : \mathbb{P}(z_A, z_B) \leq \mathbb{P}(Z_A, Z_B) \text{ and } (z_A, z_B) \in \mathcal{O}\} \tag{4.9}$$

By conditioning on the sufficient variable Z, we can then compute the $p-$value as follows:

$$p_i = \mathbb{P}(T > t \Big| Z = z) = \mathbb{P}(T > |\log(\bar{Y}_A/\bar{Y}_B)| \Big| Z = z) \tag{4.10}$$

$$= \frac{\sum\limits_{(z_a, z_b) \in \mathcal{I}} \mathbb{P}(z_a, z_b)}{\sum\limits_{(z_a, z_b) \in \mathcal{O}} \mathbb{P}(z_a, z_b)} \tag{4.11}$$

The discreteness of the test makes it conservative, especially for low values of Z. To see this, note that $\mathbb{P}(T = t) > 0$. As the set $\mathcal{I}$ does not include ties, we have a conservative test.

However, the test so far depends on knowledge of ϕ, for which we can only use an estimate. Even an unbiased estimator is not sufficient to control the Type-I error rate, as the sampling distribution of the estimator is skewed. This means that our p-values will not follow a uniform distribution. This can be seen in Figure 6. We generated 50000 samples with sample size 4, where all samples are i.i.d. distributed as $Y_{ij} \sim NB(100, 0.25)$. We then used the exact-test and plotted the obtained $p-$values. For the histogram shown in (a) we used the true-value of ϕ. We can see that here the test is indeed conservative and that the $p-$values are approximately uniformly distributed. For the histogram shown in (c) we used the (unshrunken) estimate of ϕ. We can see that now we have a strong peak of small $p-$values. In graph (b), we plotted the 2D-histogram of $p-$values against their estimate of ϕ. We can see that indeed the small $p-$values are caused by an underestimation of the overdispersion parameter.

The problems caused by underestimation of the overdispersion, and by the estimation error of the overdispersion parameter more generally can be some-

[2]Note that the test is in practice not exact, as we will need to estimate ϕ, we still decided to refer to it as exact as this is done in the publications where the test is introduced.

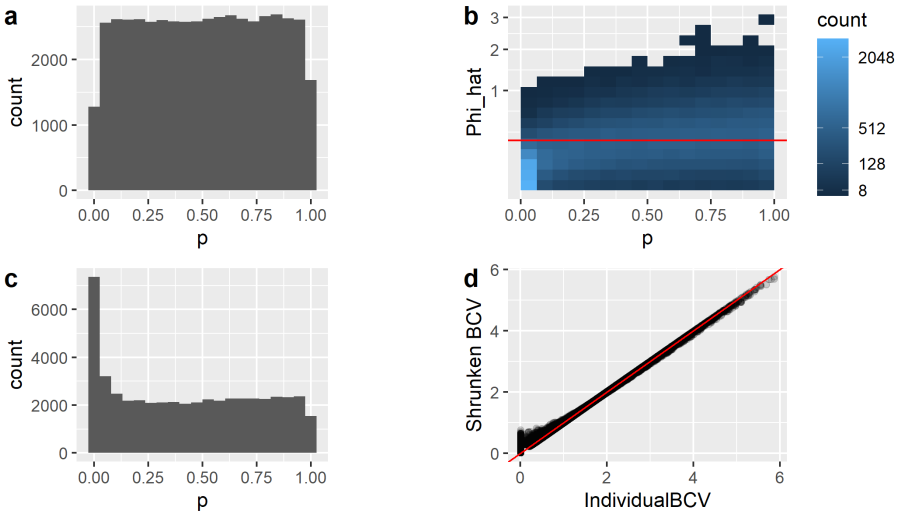


Figure 6: Analysis of the NB-Exact-Test. Plot (a)- (c) are based on 50,000 simluations from a $NB(100, 0.25)$ distribution, each with $n = 4$. In (a) we plotted the histogram of $p-$values obtained from the exact test when the true overdispersion parameter is used. In (c) we used the unshrunken estimates of the over-dispersion parameter and observe and inflation of low $p-$values. In (b) we plot the $p-$values against their dispersion estimate, and observe that an underestimation of the dispersion parameter causes the small $p-$values. In plot (d) we plot the Individual BCV against the shrunken BCV from our illustratory data-set. We observe that shrinkage has the largest effect on very low estimates of the dispersion, and can thus avoid Type-I errors due to underestimation of the overdispersion.

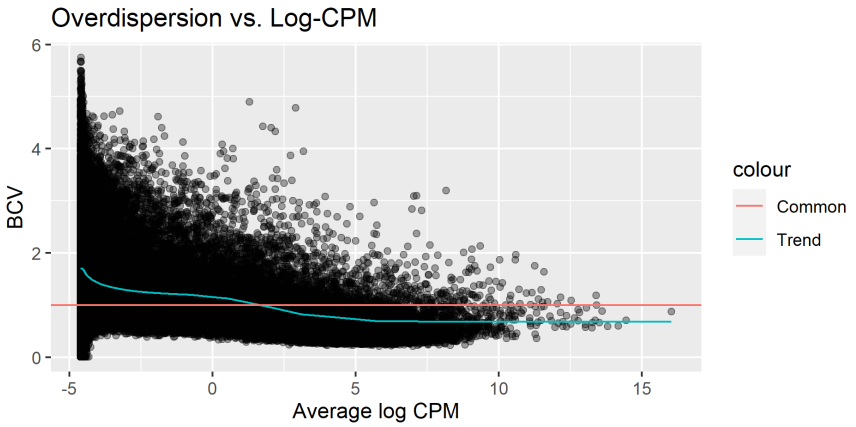


Figure 7: Plot of the Estimated Genewise Biological Coefficient of Variation, defined as $\sqrt{\hat{\phi}}$ against average gene abundance. Furthermore, a constant trend is shown and a trend as estimated by LOESS which is used for the shrinkage by the final model.

what reduced using shrinkage. However, empirical Bayes cannot easily be implemented, as the NB-distribution is not a member of the exponential family. Thus, no conjugate prior exists and MCMC-methods would need to be employed. However, shrinkage can instead be done using a weighted likelihood:

$$WL(\phi_j) = l_j(\phi_j) + \alpha l_C(\phi_j) \tag{4.12}$$

Where l_j is the gene-wise likelihood, l_C is the common likelihood and α is a weight with which the amount of shrinkage is controlled. The estimation of the weight is discussed in [32].

Similar to Limma-Trend, shrinkage is done by estimating a smooth-function $f(\mu_i) = \phi(\mu_i)$. Our dispersion parameters are then shrunken towards this trend. The estimated trend of our illustratory dataset is shown in Figure 7. We can see that especially for very low counts, the estimated overdispersion parameters vary a lot between genes. In general, larger overdispersion is observed for low-counts than for higher-counts.

We additionally visualized the effect of shrinkage in Figure 6 (d). Here, we again used the TCGA-LIHC dataset. We plotted the individual estimates against the shrunken estimates using the default shrinkage. We can see that the shrinkage has a strong effect on very small $p-$values, which it increases. We can thus assume that shrinkage reduces the probability of Type-I errors caused by underestimating the variance. However, at the same time, the bias introduced into genes with large-overdispersion can cause false-positives. We will evaluate these hypothesis later in our simulation study.

However, there is another problem. Let $Z_A \sim NB(\mu_A, \phi)$ and $Z_B \sim NB(\mu_B, \phi)$ be two random variables. Suppose we condition on $Z = Z_A + Z_B$. Our con-

ditional distributions $Z_A|Z$ and $Z_B|Z$ are unimodal for small ϕ, i.e. under the null hypothesis we expect Z_A and Z_B to be relative close together. However, for large ϕ this no longer holds. We now expect the sum to be dominated by only one of the observed values. This means that if we observe two test statistics and $T_1 < T_2$, this does not imply $P(t = T_1) > P(t = T_2)$. The denominator in Equation 10 is then no longer monotonously decreasing in T, which is necessary for our test statistic to be valid. The authors acknowledged these problems, and adjusted the computation of their $p-$values by doubling the smaller tail probability.[3]

We have now discussed two models: The first used a normal-approximation of transformed data and quasi-likelihood - which ran into problems with lowly expressed genes. The second one used an exact-test of the Negative Binomial distribution, however the test was reliant on shrinkage to reduce the effect of estimation-error of the over-dispersion parameter. The next model will in some sense be a combination of the two ideas, as it combines the use of quasi-likelihood methods with the Negative-Binomial distribution.

4.3 Quasi-Likelihood NB-GLM

In order to take the uncertainty of the estimation of the dispersion parameter into account, we can formulate a Quasi-Likelihood model using a NB-GLM. This will also have the advantage that we can model covariates. By now it should be clear that the methods share a lot of similarities. The Quasi-NB method will thus merely be outlined. We might wonder why using a Quasi-Poisson model is not sufficient to account for the overdispersion. This approach was proposed in [2]. However, previous studies have shown that this will only correct when the overdispersion is relatively large, and can thus lead to a liberal test [27][20]. Using the NB-GLM, each gene then gets an additional dispersion parameter ψ_k - the quasi-likelihood dispersion. The variance of our observations then becomes

$$Var(Y_{ij}) = \psi_j(\mu_{ij} + \phi_j\mu_{ij}^2) \tag{4.13}$$

Note that the additional parameter does not generally increase the flexibility of our modelled variance. However, we can now incorporate the uncertainty of the modelled variances into our testing procedure. Testing is done using a quasi-likelihood ratio test statistic

$$LRT_j = 2(\ell_j(\hat{\mu}_j|\mathbf{y}_j) - \ell_j(\hat{\mu}_0|\mathbf{y}_j)) \tag{4.14}$$

where $\hat{\mu}_j$ and $\hat{\mu}_0$ are the maximum quasi-likelihood estimates for μ under the null and alternative hypotheses respectively. It can be shown that this statistic follows approximately a χ^2-distribution, i.e.

$$LRT_k \sim \psi_j\chi_q^2 + O_p(n^{-1/2}) \tag{4.15}$$

[3]To our knowledge, neither the problems with the test, nor it's adjustment were extensively documented in publications. However, the authors of the package explain it in the documentation of the R-package and in forums online. See https://rdrr.io/bioc/edgeR/man/exactTest.html for the documentation and https://support.bioconductor.org/p/51302/ for an answer to a support request.

where q is the difference between the dimensions of the full and constrained parameter spaces. For parameter estimation see [27].

Shrinkage is done by assuming that $\hat{\psi}_k$ follows a scaled-inverse χ^2-distribution. This is not theoretically justified, but leads to a nice motivation of shrinkage through empirical bayes by imposing a conjugate prior. Again a smooth-trend is fitted to the mean-dispersion trend and the estimates are shrunken towards this trend. We can see that now our test-statistic depends on the dispersion through the parameter ψ_j. However, our test again fails to be exact because of the error term in equation 4.15. The last parametric approach we discuss using again the NB-distribution and shrinks the estimatated parameters.

4.4 Testing of shrunken estimators using DESeq2

DESeq2 uses a relative simple model as it's basis. It uses a GLM with negative binomial likelihood and a $\log_2$ link function [25]. However, shrinkage is employed for both coefficient as well as dispersion parameters. We first again outline parameter estimation using Profile Likelihood, then the employed shrinkage and lastly the testing-procedure.

First, a Negative-Binomial GLM is fitted. However, the obtained estimate of our dispersion is biased. Assume our models parameter θ can be divided into parameters of interest ϕ and nuisance parameters β. Our likelihood function thus becomes $L(\theta) = L(\beta, \phi)$. If β would be known, we could get an unbiased estimate of ϕ by using the conditional likelihood. The profile likelihood attempts to approximate this estimate. It has the form

$$L_p(\phi) = \max_{\beta} L(\beta, \phi) \tag{4.16}$$

Cox and Reid [9] showed that if ϕ and β are orthogonal, it can be simplified. This is the case for our NB-distribution. The Cox-Reid adjusted profile log-likelihood (CR) as used in DESeq2 is then

$$\ell_{CR}(\phi) = \ell(\phi) - \frac{1}{2}\log(\det(\mathcal{I})) \tag{4.17}$$

Where $\ell(\phi)$ is our negative binomial likelihood and $\mathcal{I}$ is our fisher information of the regression parameters which will be estimated using the standard estimator for the GLM: $\hat{\mathcal{I}} = X^T \hat{W} X$. For more details on the Cox-Reid adjusted likelihood see [9].

Using our CR-estimates, a trend is fit by regressing ϕ_j on the normalized counts. The gene-wise dispersion estimates are then shrunken towards this trend. For this, a log-normal prior π is estimated. The final estimate for ϕ is then obtained by maximizing the sum of the profile likelihood ℓ_{CR} and the prior π similar to the weighted likelihood of EdgeR (Equation 4.12). However, estimates that are more than two standard deviations away from their predicted value are not shrunken to avoid overt bias of the estimates. Furthermore, the estimates of β_j are shrunken by imposing a zero-centered normal prior on them. It's dispersion parameter is again estimated from the data. Note that the shrinkage

Name	Model	Test
Limma	Normal after log-transformation	Moderated $t-$test
EdgeR Exact	NB	Exact for fixed ϕ
EdgeR QL	QL-NB	QL-LRT-Test
DESeq2	NB	Wald-Test using shrunken estimators

Table 1: Summary of Parametric Methods

of the regression parameters is thus similar to ridge-regression. Testing is done using a Wald-Test, i.e.

$$\frac{\hat{\beta}_j}{\mathrm{SE}(\beta_j)} \sim N(0,1) \tag{4.18}$$

Where $\mathrm{SE}(\beta_j)$ is the estimated standard error of the regression coefficient, defined as the square root of the diagonal elements of the estimated covariance matrix Σ_j. As we are using an asymptotically consistent estimator of the standard error, we get a asymptotically exact test by the Law of Large Numbers (LLN).

We've discussed four of the most popular methods for analysing RNA-Seq Data. We've seen that the employed models are very different from each other. In general, the consensus seems to use a negative binomial distribution to model the data. However, there is no exact test known to test the regression-coefficients. Most of the models thus rely on asymptotic distributions of the test-statistics. As our sample-sizes are usually very small, the use of asymptotic properties can only be reasonably justified by sharing information across genes i.e. through shrinkage. A summary of the methods is given in Table 1. Let us now move on to the discussion of permutation methods.

5 Permutation Methods

5.1 Introduction to Permutation Methods

Let us first introduce how basic permutation tests work. Permutations tests construct the distribution of a test statistic by applying a set of permutatios G to the labels of the observations. Recall that the computation of a $p-$value involves the computation of a tail probability. For a one sided test, this can be formalized as

$$\mathbb{P}(T \geq t|H_0) = \int_t^\infty f_0(x)dx \tag{5.1}$$

Where t is our observed test statistic and f_0 is the distribution of our test-statistic under the null hypothesis.

Suppose we observe the data

$$(Z_1, \ldots, Z_n) = [(X_1, Y_1), \ldots, (X_n, Y_n)] \tag{5.2}$$

where $Y_i \in \{A, B\}$ is our group label.

Suppose now, that we want to test the null hypothesis $H_0 : f_A(x) = f_B(x)$. Then - under the null hypothesis - $\mathbb{P}(X_i|Y_i) = \mathbb{P}(X_i)$ and the group labels are uniformly distributed over the $N!$ permutations that form G.

For each of the permutations $g \in G$, we can thus obtain a test statistic which we'll denote $T(gX)$. Let $T^{(1)} \leq \cdots \leq T^{(|G|)}$ be our ordered test-statistics with $\#G$ denoting the number of elements in G. Define $k = \lceil (1-\alpha)\#G \rceil$ with $\alpha \in [0, 1)$. We then have an exact test, as the following theorem shows.

Theorem 4. *Under* $H_0 : \mathbb{P}(T(X) > T^{(k)}(X)) \leq \alpha$

Proof. See [13]. □

We can thus compute the probability of Equation 5.1 even for an unknown distribution f_0, as we condition on the *orbit* of our observed data as given by the G. Note that the inequality is caused by the discreteness. If we have a non-zero probability of multiple permutation test statistics being exactly $\alpha|G|$, then $\mathbb{P}(T(X) = T^{(k)}X) > 0$ and $\mathbb{P}(T(X) > T^{(k)}X) < \alpha$.

There can be various choices for the group operation of G, as long as it is an (algebraic) group [13]. This means that G needs to contain the identity transformation. This avoids obtaining p-values of 0, which is known to cause problems [29]. Furthermore, every element of G has an inverse in G and for all $g_1, g_2 \in G$, $g_1 \circ g_2 \in G$. The condition of the group ensures that $Gg = G$ for all g. This means, if we apply any transformation from the set G to all elements of G, the resulting set is still G. Other transformations such as sign-flipping or rotations are also allowed as long as under H_0 the condition of *exchangeability* $gX \overset{d}{=} X$ is fulfilled. We can even relax this condition slightly to the following condition:

$$T(gX) \overset{d}{=} T(X) \tag{5.3}$$

where $\stackrel{d}{=}$ denotes equality in distribution.

Permutation tests can control the type-I error rate for all sample sizes. However, statistical analysis of their power is still an active area of research. In general, we can choose the test-statistics in such a way that the permutation distribution approximates the null distribution for large sample sizes. The permutation tests can thus be asymptotically as powerful as the test based on the asymptotic null distribution when all assumptions are met [19].

It is clear that permutation-tests are a large group of tests. However, evaluation of all all transformations might be computationally impossible, when the data set is very large. Thus, often only a random subset of permutations is used. We will justify this formally in the next section.

5.2 Exact Testing using Random Permutations

The number of possible transformations can be very large. For example, when we use permutations we have $n!$ test-statistics to evaluate. Even when n is only 10 we have more than 3 million test-statistics to compute, which might be too time-consuming to do. Instead, we can use a random subset of all permutations. In the literature, this was often seen as a method to approximate $p-$values. $\mathbb{1}\{T \geq T(X)\}$ is a Bernoulli random variable, an approximate $p-$value p^*, based on a random subset of the permutation group, would thus asymptotically be normally distributed with $\mathbb{E}(p^*)$ the exact $p-$value. However, by selecting the sampling strategy from G, it can be shown that also a test based on random permutations can be exact.

Theorem 5. *Let $\mathcal{G}$ be a group of transformations. Let $\mathcal{G} \supseteq G = (g_1, g_2, \ldots, g_w)$ be a vector with g_1 the identity transformation and $g_2, \ldots, g_w$ being random elements from $\mathcal{G}$ beeing drawn with replacement. Let $T^{(1)}(X, G) \leq T^{(w)}(X, G)$ be the ordered test statistics. Let T(X, G) be the observed test statistic and define $k = \lceil (1-\alpha)w \rceil$ Then*

$$\mathbb{P}\left[T(X, G) > T^k(X, G)\right] \leq \alpha \tag{5.4}$$

Proof. See [13] □

The $p-$value based on random permutations is *stochastically larger* than the $p-$value based on all permutations. This means, that we maintain exactness as our probability of rejection is smaller, but also reduce the power of the test.

However, even before computational power was readily available, permutation were widely employed. For the two-group problem this was done using ranks in the form of the Mann-Whitney U test (MWU). We are introducing the test now, showing it's relationship with permutation tests.

5.3 The Mann-Whitney U Test as a Permutation Test

The probably most commonly used permutation test is the Mann-Whitney U-test. Suppose we observe two vectors $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$ whose elements

are independent. Define $N = n + m$. For simplicity assume that they their distribution is continuous so that the probability of observing ties is 0. We want to test the null hypothesis $H_0 : f_x(x) = f_y(x)$ for all x. The Mann-Whitney-U-Test allows testing this null-hypothesis without making any assumptions about the distributions of our random variables. It does so using the $U-$statistic:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}_{(X_i > Y_j)} \tag{5.5}$$

For each X_i, we thus compute the number of Y_j that are larger, and sum this number over all X_i. To see why this is a permutation test, note the alternative way of computing the test statistic U:

1. Combine X and Y into a vector i.e. $(X, Y) = (X_1, \ldots, X_n, Y_1, \ldots, Y_m)$.
2. Rank the vector (X, Y) and replace the values of (X, Y) with the ranks of the observation e.g. $Z = (1^X, 2^X, 3^Y, \ldots, N)$, where the superscript indicates from which sample the rank was taken. Note that the ranks (N natural numbers) sum to $\frac{N(N+1)}{2}$.
3. We can now compute 5.5 as

$$U_X = R_1 - \frac{n(n+1)}{2} \tag{5.6}$$

$$U_Y = R_2 - \frac{m(m+1)}{2} \tag{5.7}$$

$$U = \max(U_X, U_Y) \tag{5.8}$$

   Where R_k is the sum of ranks from group k.

We can compute the null distribution of this test-statistic using permutations. For this, compute the test statistic U for all possible orderings of the ranks of the groups in Z. Write g for such a permutation and gZ for a reordering of Z. Denote $U(Z)$ the observed test statistic and $U(gZ)$ the permuted one. Finally, write $|Z|$ for the size of the set Z. Then, under the null hypothesis.

$$\mathbb{P}(X < Y) = \frac{1}{|Z|} \sum_{k}^{|Z|} \mathbb{1}_{U(Z) > U(g_k(Z))} \tag{5.9}$$

In practice, this test statistic is easily tabulated for small N and a normal approximation is used for large samples. It is straightforward to make this a two tailed-test by also using the minimum of the two test statistics and by considering both sides of the the distribution of the test-statistic. This test has been proposed to use with RNA-Seq Data [21]. It is combined with a resampling-based normalization approach. We will use another normalization approach to ease comparison with other discussed methods, which we will discuss in Section 5.5.

It is clear that permutation tests provide an extremely flexible class of tests. However, from our discussion so far we will use two permutation tests: One, where the test statistic is simply $\mu_j^A - \mu_j^B$, and the MWU. However, they do not allow the inclusion of covariates and might not be powerful enough. We will thus introduce a third permutation-based test that is based on sign-flipping score contributions.

5.4 Semi-Parametric Permutation Tests based on sign-flipping Score Contributions (!!)

In this section, we will introduce the Permutation Test based on Sign-Flipping Score contributions. There are multiple reasons why we might prefer such a test over a simple test based on permuting group labels. Firstly, using a parametric distribution in the modelling process might improve the power. Secondly, we will be able to include covariates in the same model. We will discuss this in Section 5.4.2. However, let us first introduce the basic test.

Using the score contributions v_i we can now define a test-statistic for our permutation test. However, let us first note two properties of the score function. First, it is important to note that under the null hypothesis, $\mathbb{E}(v_i) = 0$. Secondly, note that Lindeberg's condition of the fullfilled and the sum of scores is thus asymptotically normal under the null hypothesis by the Central Limit Theorem. We can then define the test statistic

$$T_j^n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} g_{ji} v_i \tag{5.10}$$

where v_i is the score-contribution of our $i-$th observation and g_j is a random variable uniformly distributed on $\{-1, 1\}^n$.

Theorem 6. *Suppose that the Lindeberg condition holds and consider the test that rejects H_0 when $|T_1^n| > |T_{(1-\alpha)}^n|$. Then, as $n \to \infty$, the probability of rejection of this test converges to $\lfloor aw \rfloor / w \leq \alpha$. Furthermore, the statistics are asymptotically normal with mean 0 and variance s_n^2 under H_0. I.e.*

$$T_i^n \xrightarrow{d} N(0, s^2) \tag{5.11}$$

where $\xrightarrow{d}$ denotes convergence in distribution.

Proof. The proof consists of showing that the Lindeberg condition holds for our score-contributions. The distribution of the vector of test statistics $T^n = (T_1^n, \ldots, T_w^n)$ converges in distribution to a multivariate normal. As $T_{[1-\alpha]}^n$ converges to the true 0.95-quantile we have $\mathbb{P}(T_1^n > T_{[1-\alpha]}^n) \to \lfloor aw \rfloor / w \leq \alpha$. For more details see [13]. Here, we gave the special case when the rejection levels of the two-sided test are equal, i.e. $\alpha_1 = \alpha_2$. Then, we can use the absolute value of T and use rejection level $\alpha = 2\alpha_1$. □

The test has an important property that distinguishes it from it's parametric counterpart. Recall the general form of the score-contributions from an exponential family:

$$v_i = \frac{L_i}{\beta} = \frac{(y_i - \mu_i)x_i}{Var(y_i)} \frac{\partial \mu_i}{\partial \eta_i} \tag{5.12}$$

The proof of 6 only relies on the Lindeberg condition. Thus, our test is asymptotically exact even when we misspecify the variance of our model. For example, suppose that we want to test $H_0 : \beta = 0$ and we simply compare two groups. This means that μ_i is the same for all observations under the null hypothesis. Even when $Var(y_i) = V(\mu_i)$ for some arbitrary function V, the test is equivalent as our test-statistics will simply all be scaled by the same constant.

So far, we relied on the asymptotic properties of our test-statistic for the construction of our test. However, we are naturally most interested in the finite-sample performance of the method. The following proposition will be crucial for this:

Proposition 1. *Suppose that $v_1, \dots, v_n$ are independent and without ties. Furthermore, suppose that the distribution of all v_i is symmetric, i.e. that $v_i \overset{d}{=} -v_i$. Furthermore, suppose that $g_2, \dots g_2$ are uniformly drawn from $\{-1, 1\}^n$ without replacement. Then the size of the test is at most $\lfloor \alpha w \rfloor / w$.*

Note that we do not know the finite-sample distribution of our score-contributions, the above lemma is thus of limited use. Especially when our data is discrete, we can expect ties and consequentially also tied score-contributions. The applicability of the test will thus need to be confirmed using simulation studies. Another issue with our test is that our scores are dependent on estimated nuisance parameters. We will adress this issue in the next section.

5.4.1 Accounting for Estimation of Nuisance Parameters

The estimation of nuisance parameters (this includes the intercept) introduces dependence into our score-contributions, thus the Lindeberg condition is not fulfilled. Thus, the use of the *effective score* was proposed [14]. The effective score will require the use of the Fisher information, which we will review now.

Definition 6 (Fisher Information)**.** *The fisher information is the variance of the score*

$$\mathcal{I}(\boldsymbol{\beta}) = \mathbb{E}\left[\left(\frac{\partial}{\partial \boldsymbol{\beta}} \log f(X; \boldsymbol{\beta}) \right)^2 \middle| \boldsymbol{\beta} \right] = \mathbb{E}\left(\frac{-\partial^2 L(\boldsymbol{\beta})}{\partial \beta_h \partial \beta_j} \right) = \sum_{i=1}^{n} \frac{x_{ih} x_{ij}}{Var(y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \tag{5.13}$$

Where the first equality follows by regularity conditions on the density which hold in GLMs. Note that the fisher information is thus a $k \times k$ matrix where k is the number of covariates including the intercept.

Note that in GLMs $\hat{\mathcal{I}} = n^{-1}\mathbf{X}'\widehat{\mathbf{W}}\mathbf{X}$ where X is the design matrix and $\hat{\mathbf{W}}$ is the estimated weight matrix. The fisher information is directly available from the GLM functions in R.

Next, note that for each of our parameters we get a score. Denote these scores as $\mathbf{S} = (S^1, \ldots, S^k)$. Here, we are always interested in in testing hypotheses about the $k-$th component. We then define the effective score of the $j-$th permutation as

$$T^j_{\text{eff}} = T^j - \hat{\mathcal{I}}_{12}\hat{\mathcal{I}}_{22}^{-1}\mathbf{T}^{(k-1),j} \tag{5.14}$$

Where $\mathbf{T}^{(k-1),j}$ denotes the vector of scores without the $k-$th column. The matrix $\hat{\mathcal{I}}_{12}$ is the covariance of [7].

Let us illustrate the computation of the effective scores for our two distributions.

Illustration

5.4.2 Permutation Tests with Covariates

Suppose we observe data from the model

$$\mathbf{Y} = \alpha + \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \boldsymbol{\epsilon} = \mathbf{X}\beta + \boldsymbol{\epsilon} \tag{5.15}$$

Where $\mathbf{Y}, \mathbf{x}_1, \boldsymbol{\epsilon} \in \mathbb{R}^n$, $\mathbf{x}_2 \in \{0,1\}^n$, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2 I)$. Suppose we want to test $H_0 : \beta_2 = 0$. Using the distributional assumption of the error ϵ this is straightforward using standard tests, such as the Likelihood-Ratio-Test (LRT), the Wald-Test, or the Score-test. However, because we need to make these distributional assumptions, none of the tests is exact. Now suppose we do not the distribution of $\boldsymbol{\epsilon}$, we are tempted to simply permute the group labels $\mathbf{x_2}$ and compute a test statistic from the resulting fitted model, e.g. $|\beta_2|$. However, such an approach may violate the assumptions of the permutation test. As soon as $Cov(x_1, x_2) \neq \mathbf{0}$, we have

$$P(x_1, x_2) \neq P(x_1)P(x_2) = P(x_1)P(gx_2) \tag{5.16}$$

5.5 Applying Permutation Methods to RNA-Seq Data. (!!!)

6 Simulation Study 10p

Our simulation study is divided into three parts. First, we are going to run all discussed models on our TCGA-LIHC dataset. Some of the results have already been presented throughout the thesis but we will compare the results of the methods in this chapter. Secondly, from this dataset we obtain a set of DE genes and non-DE genes. We are then going to permute the group-labels of the non-DE genes and take random samples to test Type-I error control and performance of the methods on smaller sample sizes. Lastly, we are going to perform parametric simulations from a known distribution. Here, our goal is to test specific settings for which we can evaluate the performance of our methods.

6.1 Analysis of RNA-Seq Data

In this section, we are going to compare the $p-$values obtained by the various methods discussed throughout the thesis. First, we computed the sets of genes that were differentially expressed at $\alpha = 0.01$ without additional multiple testing correction. The contingency table can be seen in Table 2.

We can see that overall there is considerable agreement between the methods, as for 9330 out of 60660 genes H_0 was rejected by all methods, for 21246 genes by none but limma and for 5357 genes by none. The large number of rejections by limma are due to the lack of filtering. As already mentioned before, Limma is reliant on filtering as otherwise the normal approximation fails for lowly expressed genes. Furthermore noteworthy are the 3826 genes for which H_0 was rejected by no method but the Direct Permutations. Because of the exactness of the method, Type-I errors are unexpected. However, errors could be caused by the normalization. Thus, further investigation is necessary. Lastly, there is also a surprising number of genes that were rejected by all methods but limma (3012) and of genes that were rejected by all methods but the Direct Permutations (2975).

In order to make the results more tractable, we computed the Jaccard Distance on the set of rejections between all methods. The results can be seen in Table 3. Similar to the results in Table 2, we can see that Limma is the furthest removed from all other methods. The four methods based on distirbutions of count-data all have relatively similar sets of rejections. Next, we are doing a permutation based simulation, that allows us to do more detailed analyses.

Limma	EdgeR ET	EdgeR QL	DESeq2	MWU	DP	FlipScores	
						False	True
False	False	False	False	False	False	5357	134
					True	3826	1645
				True	False	295	18
					True	93	35
			True	False	False	41	8
					True	9	86
				True	False	2	2
					True	3	1
		True	False	False	False	225	187
					True	22	507
				True	False	1	0
					True	0	3
			True	False	False	2	1
					True	0	12
				True	False	0	0
					True	0	1
	True	False	False	False	False	11	1
					True	3	44
				True	False	1	1
					True	1	4
			True	False	False	15	7
					True	12	138
				True	False	1	5
					True	1	19
		True	False	False	False	346	203
					True	94	1562
				True	False	10	6
					True	0	9
			True	False	False	75	106
					True	64	3012
				True	False	15	18
					True	14	1363

Limma	EdgeR ET	EdgeR QL	DESeq2	MWU	DP	FlipScores	
						False	True
True	False	False	False	False	False	21246	2116
					True	44	72
				True	False	1960	446
					True	33	53
			True	False	False	7	1
					True	1	0
				True	False	135	84
					True	12	33
		True	False	False	False	589	435
					True	2	83
				True	False	55	2
					True	0	0
			True	False	False	2	0
					True	0	0
				True	False	5	0
					True	0	1
	True	False	False	False	False	3	0
					True	3	2
				True	False	12	42
					True	0	0
			True	False	False	3	3
					True	2	1
				True	False	33	75
					True	0	2
		True	False	False	False	247	56
					True	1	43
				True	False	113	86
					True	0	4
			True	False	False	27	18
					True	5	295
				True	False	150	2975
					True	35	9330

Table 2: (a) Contingency table of rejections of all methods. We can see that in general the methods agree for most genes, as 5829 genes are rejected by all methods. Other cells with high counts are the one counting the genes rejected by all but Limma and Wilcoxon (976), the genes only rejected by FlipScores (711) and the genes rejected only by Limma and Wilcoxon. The table thus indicates that there is relatively large agreement overall, that Limma and Wilcoxon will form a cluster, and that EdgeR, DESeq2 and FlipScores will form a cluster.

	Limma	EdgeR.et	EdgeR.ql	DESeq2	MWU	DP	FS
Limma	0.00	0.72	0.70	0.71	0.63	0.81	0.68
EdgeR.et	0.72	0.00	0.11	0.16	0.40	0.41	0.27
EdgeR.ql	0.70	0.11	0.00	0.24	0.45	0.42	0.26
DESeq2	0.71	0.16	0.24	0.00	0.34	0.45	0.32
MWU	0.63	0.40	0.45	0.34	0.00	0.62	0.48
DP	0.81	0.41	0.42	0.45	0.62	0.00	0.38
FS	0.68	0.27	0.26	0.32	0.48	0.38	0.00

Table 3: Jaccard Distance between methods of Sets of Rejected Genes for $\alpha = 0.01$

6.2 Permutation based Simulation (!)

Next, we performed a permutation based simulation test. This allowed us to keep as much realism as possible from RNA-Seq data while allowing us to evaluate Type-I and Type-II error rates of the different methods. To achieve this, we randomly selected 500 genes for which the null hypothesis was rejected at $\alpha = 0.01$ by all of the 7 methods and 2500 for which at least 5 methods fail to reject the null hypothesis at $\alpha = 0.1$. We chose these values for the following reasons. First, we wanted to maintain the ratio between Differentially Expressed (DE) genes and Non-Differentially expressed (NDE) genes from our original dataset. Secondly, we wanted to avoid to have DE genes in our set of NDE genes, as this would create additional variance after the permutation of our group labels. We then proceeded as follows for each iteration:

1. Normalize the data by multiplying it by the normalization factors computed by TMM.
2. Then randomly shuffle the samples of the NDE genes.
3. Undo the normalization by dividing by the normalization factors again.
4. Choose a random subset of samples from both groups to obtain a dataset with equal group sizes.
5. Run all algorithms on the obtained dataset.

We used 200 iterations and simulated sample sizes $n = 10$ and $n = 40$. We did not use any filtering.

First, we check the distribution of $p-$values under H_0. We plot qq-plots using the uniform as our reference distribution. The results can be seen in Figure 8. In the top we see the raw qq-plots of the $p-$values We plotted the line $y = x$ for reference. We see that Limma has vastly inflated Type-I error rates for both sample sizes. In the bottom we divided the observed Type-I error rate by it's nomial α-level and plotted the factors of inflation. We can see that Limma, DESeq2 and EdgeR's Exact test fails to control the Type-I error rate

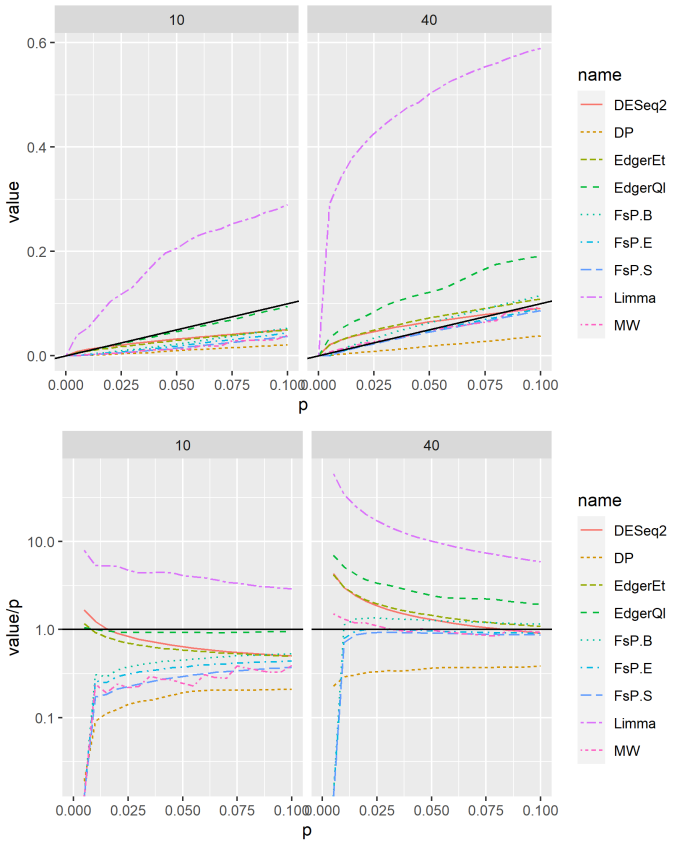


Figure 8: QQ-Plots of $p-$values of NDE-genes.

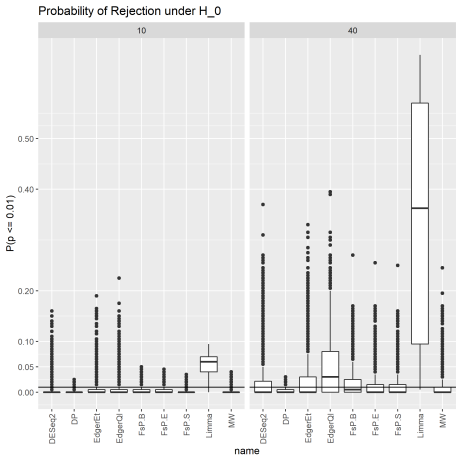


Figure 9: Probability of rejection at $\alpha = 0.01$ under H_0

for small α-levels for $n = 10$. For the larger sample size, the only methods that control the Type-I error rate are the Effective and Standardized FlipScores Test and the Direct Permutation test. Even the Mann-Whitney-U Test fails. This is likely to be caused by the normalization.

Next, for each NDE-gene we computed the proportion of replications for which the $p-$values were less than or equal to 0.01. The results can be seen in **??**. Note that ideally we would want to have rejection rates of 0.05 for all genes. However, due to the random nature and the limited number of replications we can not expect this. However, again the rejection rates are higher that we would expect. Especially Limma has highly inflated Type-1 error rates for many genes for both sample sizes. However, also the Quasi-Likelihood-NB Model as implemented in EdgeR has inflated Type-I error rates for more than 50% of the NDE genes for $n = 40$. For the smaller sample size, only the permutation methods achieved control over Type-I error rates. It is clear that the parametric methods control the Type-I error rate for some of the genes, but fail to so for others. It is thus an obvious question to ask for what type of genes they fail. We will look into this later. However, let us first get an overview about how the power of the different methods compare.

Next, we looked at the power of the methods. However, comparing power of methods of which some do not control the Type-I error rate is not straightforward. In order to allow fair comparisons, we computed the rejection regions so that the Type-I error rate was on average 0.01 for all genes. We then computed the probability of DE-genes falling into these rejection regions. This of course cannot be done in practice, when it is not known which genes are NDE. The aim of this adjustment is thus not the analysis of absolute Type-II error rates, but of a comparison between the methods. The results can be seen in Figure 10.

We can see that Limma has very low power compared to the other methods

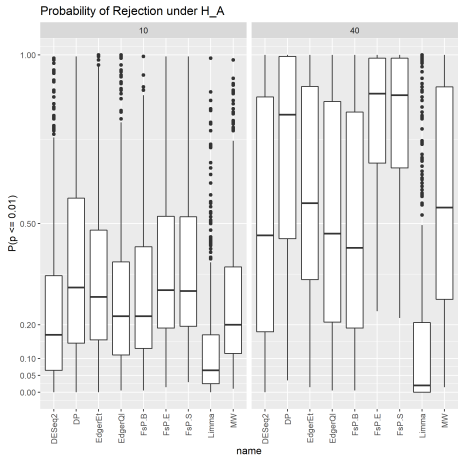


Figure 10: Probability of rejection at $\alpha = 0.01$ under H_A

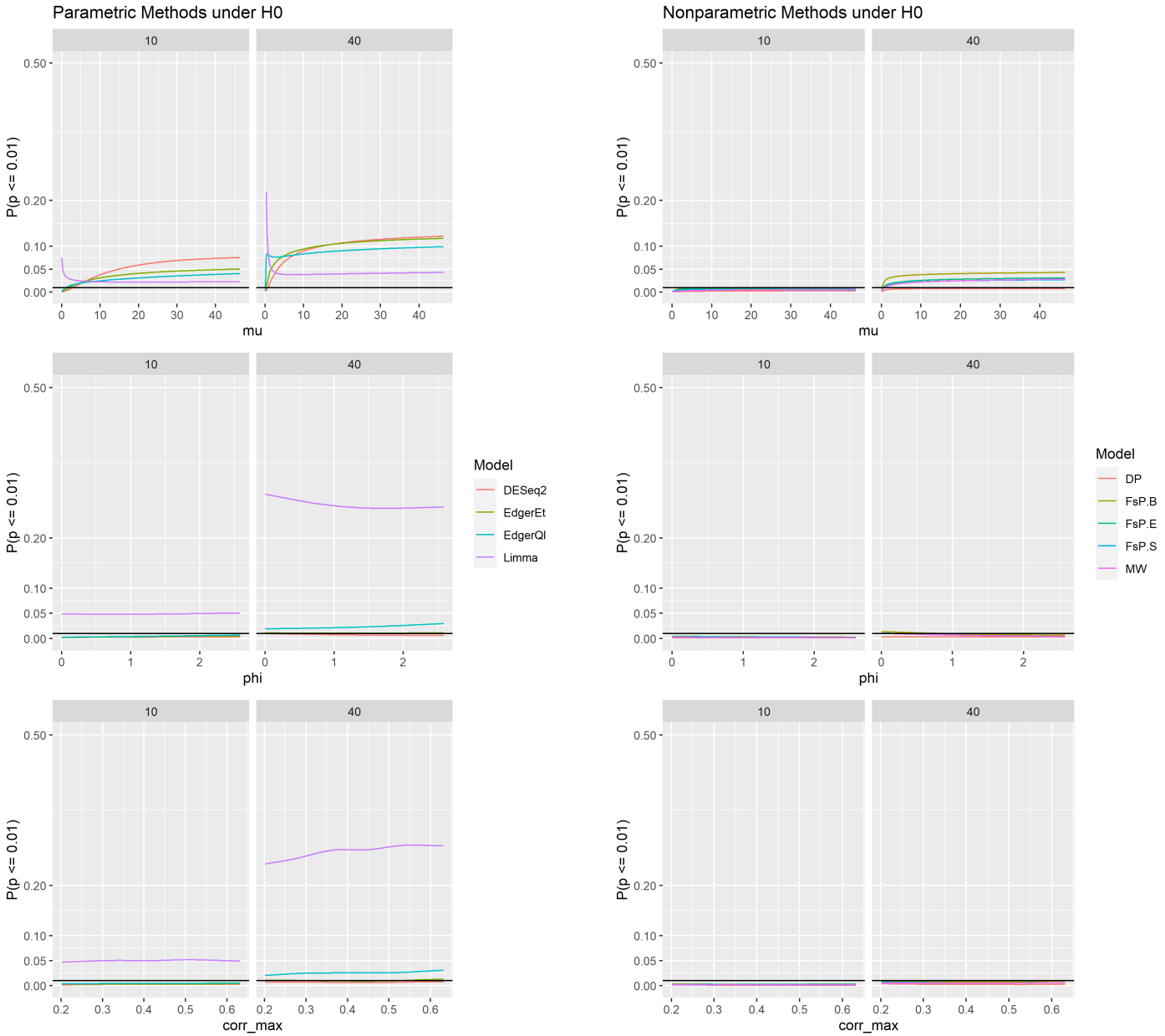
for both sample sizes. This is because we needed to shrink the rejection region quite drastically so that it's nominal α-level and it's empirical α-level are equivalent. Furthermore, we can see that our test based on Direct Permutations, the Effective and Standardized Flipscore Test and the Exact Test of EdgeR perform relatively similar for the small sample size. However, for the large sample size the Direct Permutation Method, FSP-E and FSP-S outperform all other methods quite significantly.

We already hinted at the non-uniformity of both error rates. We can analyze these in more detail. From the simulations we get Type-I errors for each of the genes. For each gene we estimated three parameters. The mean μ, the overdispersion ϕ, and the maximum covariance with other genes $\max |\rho|$. However, we assume that these parameters are not independent, but would like to estimate their marginal effects. We thus estimated a Generalized additive model for each of the methods of the following form:

$$g(\mathbb{E}(Y)) = \beta_0 + f_\mu(\mu) + f_\phi(\phi) + f_\rho(\max |\rho|) \tag{6.1}$$

Where $y_i \sim Beta(\alpha, \beta)$ and g is the logistic link. All $f_\cdot$ are estimated using smoothing splines. Estimation is done using `mgcv::gam`. The results of the Type-I error rate can be seen in Figure **??** and Figure **??**.

We can see that there are indeed systematic trends mostly for the mean. Most notable are Limma's vastly inflated Type-I error rates for very lowly expressed genes. Interestingly, DESeq2 and EdgerEt control the Type-I error rates for low expressed genes but fail to do so for higher expressed genes. This can be problematic, as control over the Type-I error rate then depends on the distribution of expression levels in the data set and the sequencing depth. However, we see no especially remarkable trends for the overdispersion and the correlation. Especially the latter might be surprising, as we expected highly correlated genes



(a) Parametric Methods

(b) Non- and Semi-Parametric Methods

Figure 11: Probability of Type-I Error of Parametric Methods based on the non-parametric simulation

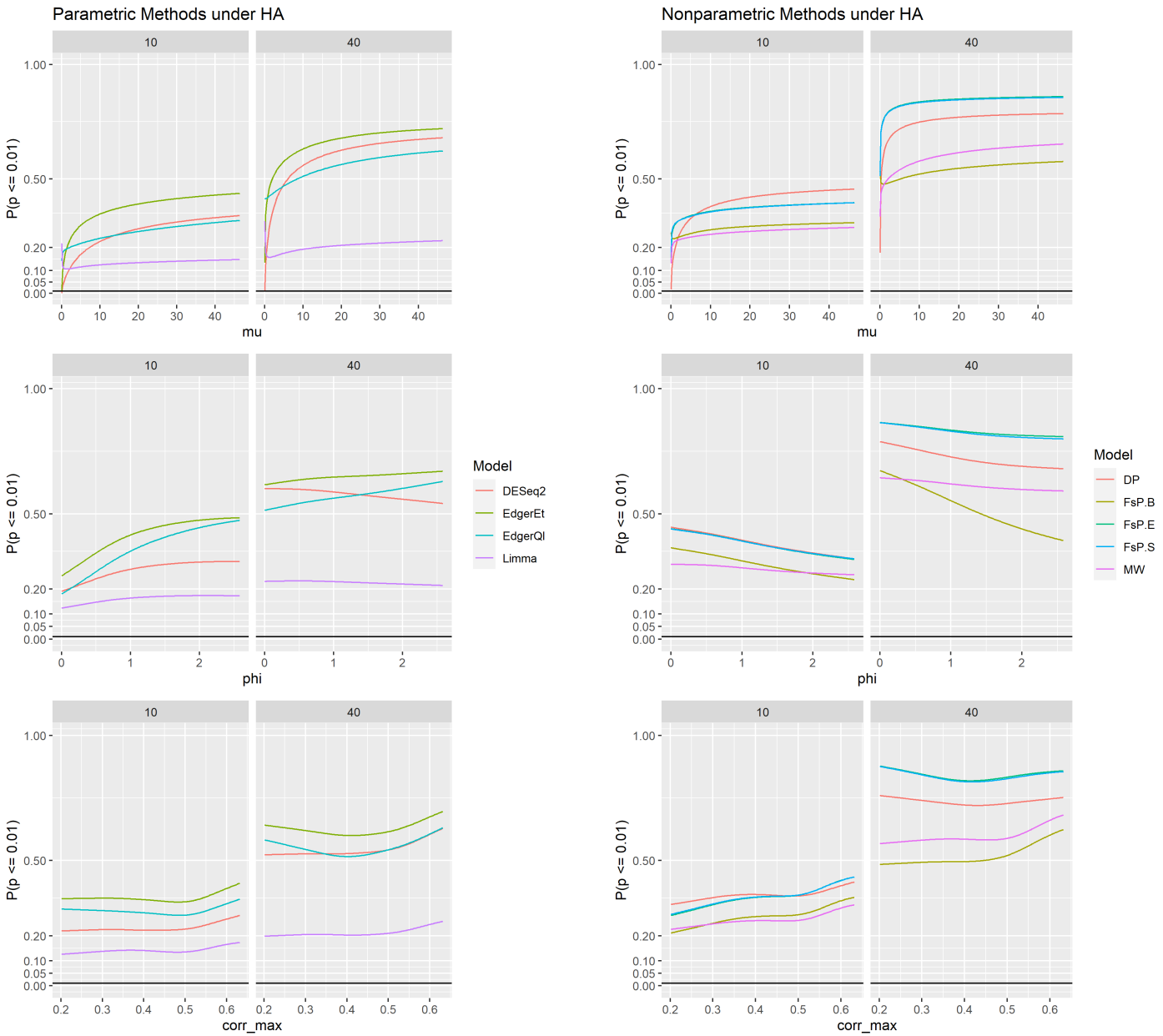
to be shrunken together. This shrinkage could've thus affected their Type-I error rates.

Our nonparametric methods are all conservative for all parameters for the small sample-size. However, for the larger sample size we see that the Type-I error rate depends on the expression level of genes. Again the tests are conservative for lowly expressed genes but are anti-conservative for higher levels of expression. The only method immune to this effect is DP. We see no noteworthy effect of the overdispersion or of the correlation on the Type-I error.

Next, we looked at trends between the discussed parameters and the probability of rejection. Again, we adjusted our rejection regions such that the nominal alpha level is equivalent to the empirical alpha level to ease comparison between methods. The results can be seen in Figure 12.

We can see similar trends then the ones observed for the Type-I error. The parametric methods based on the Negative-Binomial distribution all have lower power for lowly expressed genes. This is to be expected, as it is caused by the discreteness of the counts. For these methods, low over-dispersion seems to negatively affect the power for the small sample sizes but does not seem to have a strong effect for the larger sample size. Lastly, we see no strong trends for the correlation between genes. Our non- and semi-parametric methods seem to be less affected by low expression, as their curves level out faster and higher. Interestingly, our basic scores are negatively affected by high overdispersion. This might be caused by the dependence of the score-contributions caused by the estimation of the nuisance parameter.

6.3 Parametric Simulation with Covariates (!!!)



(a) Parametric Methods

(b) Non- and Semi-Parametric Methods

Figure 12: Probability of Rejection based on the non-parametric simulation

7 Discussion (!!!)

In experiments that aim to discover differential expression, ranking the genes can be seen as equally important as the absolute value of assigned $p-$values. This is because follow-up experiments, in which Family-Wise Error Rate might be used, are restricted in the number of genes to test. A statistical method that reliably ranks the genes by evidence of differential expression is thus often sufficient [35].

References

[1] Alan Agresti. *Foundations Linear Generalized Linear Models*. 2015, p. 444. ISBN: 978-1-118-73003-4.

[2] Paul L. Auer and Rebecca W. Doerge. "A two-stage poisson model for testing RNA-Seq data". In: *Statistical Applications in Genetics and Molecular Biology* 10.1 (2011). ISSN: 15446115. DOI: 10.2202/1544-6115.1627.

[3] Sam Benidt and Dan Nettleton. "SimSeq: A nonparametric approach to simulation of RNA-sequence datasets". In: *Bioinformatics* 31.13 (2015), pp. 2131–2140. ISSN: 14602059. DOI: 10.1093/bioinformatics/btv124.

[4] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing Author (s): Yoav Benjamini and Yosef Hochberg Source : Journal of the Royal Statistical Society . Series B (Methodological), Vol . 57 , No . 1 (1995), Publi". In: *Journal of the Royal Statistical Society* 57.1 (1995), pp. 289–300.

[5] Conrad J. Burden, Sumaira E. Qureshi, and Susan R. Wilson. "Error estimates for the analysis of differential expression fromRNA-seq count data". In: *PeerJ* 2014.1 (2014), pp. 1–26. ISSN: 21678359. DOI: 10.7717/peerj.576.

[6] Travers Ching, Sijia Huang, and Lana X. Garmire. "Power analysis and sample size estimation for RNA-Seq differential expression". In: *Rna* 20.11 (2014), pp. 1684–1696. ISSN: 14699001. DOI: 10.1261/rna.046011.114.

[7] Sungsub Choi, W. J. Hall, and Anton Schick. "Asymptotically uniformly most powerful tests in parametric and semiparametric models". In: *Annals of Statistics* 24.2 (1996), pp. 841–861. ISSN: 00905364. DOI: 10.1214/aos/1032894469.

[8] Ana Conesa et al. "A survey of best practices for RNA-seq data analysis". In: *Genome Biology* 17.1 (2016), pp. 1–19. ISSN: 1474760X. DOI: 10.1186/s13059-016-0881-8.

[9] D. R. Cox and N. Reid. "Parameter Orthogonality and Approximate Conditional Inference". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 49.1 (1987), pp. 1–18. DOI: 10.1111/j.2517-6161.1987.tb01422.x.

[10] Alexander Dobin et al. "STAR: Ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21. ISSN: 13674803. DOI: 10.1093/bioinformatics/bts635.

[11] Bradley Efron and Carl Morris. "Stein's estimation rule and its competitors—an empirical bayes approach". In: *Journal of the American Statistical Association* 68.341 (1973), pp. 117–130. ISSN: 1537274X. DOI: 10.1080/01621459.1973.10481350.

[12] Jelle J Goeman and Aldo Solari. "Multiple hypothesis testing in genomics". In: *Statistics in Medicine* 33.11 (2014), pp. 1946–1978. ISSN: 10970258. DOI: 10.1002/sim.6082.

[13] Jesse Hemerik and Jelle Goeman. "Exact testing with random permutations". In: *Test* 27.4 (2018), pp. 811–825. ISSN: 11330686. DOI: 10.1007/s11749-017-0571-1. arXiv: 1411.7565. URL: https://doi.org/10.1007/s11749-017-0571-1.

[14] Jesse Hemerik, Jelle J Goeman, and Livio Finos. "Robust testing in generalized linear models by sign flipping score contributions". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 82.3 (2020), pp. 841–864. ISSN: 14679868. DOI: 10.1111/rssb.12369. arXiv: 1909.03796.

[15] Daniel Horspool. *Extended Central Dogma with Enzymes*. 2008. URL: https://commons.wikimedia.org/wiki/File:Extended_Central_Dogma_with_Enzymes.jpg.

[16] Qichao Huang et al. "RNA-seq analyses generate comprehensive transcriptomic landscape and reveal complex transcript patterns in hepatocellular carcinoma". In: *PLoS ONE* 6.10 (2011). ISSN: 19326203. DOI: 10.1371/journal.pone.0026168.

[17] W. James and C M Stein. "Estimation with Quadratic Loss". In: *Breakthrough in Statistics: Volume I*. Ed. by Samuel Kotz and L Norman Johnson. 1992, pp. 443–461. ISBN: 978-1-4612-0919-5. URL: https://doi.org/10.1007.

[18] Wentao Jiang et al. "Identification of the Pathogenic Biomarkers for Hepatocellular Carcinoma Based on RNA-seq Analyses". In: *Pathology and Oncology Research* 25.3 (2019), pp. 1207–1213. ISSN: 15322807. DOI: 10.1007/s12253-019-00596-2.

[19] Ilmun Kim, Sivaraman Balakrishnan, and Larry Wasserman. "Minimax optimality of permutation tests". In: *The Annals of Statistics* 50.1 (2022). ISSN: 21688966. DOI: 10.1214/21-aos2103. arXiv: 2003.13208.

[20] Charity W Law et al. "Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts". In: *Genome Biology* 15.2 (2014), pp. 1–17. ISSN: 1474760X. DOI: 10.1186/gb-2014-15-2-r29.

[21] Jun Li and Robert Tibshirani. "Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data". In: *Statistical Methods in Medical Research* 22.5 (2013), pp. 519–536. ISSN: 09622802. DOI: 10.1177/0962280211428386.

[22] Zhuolin Li et al. "Identification and Analysis of Potential Key Genes Associated With Hepatocellular Carcinoma Based on Integrated Bioinformatics Methods". In: *Frontiers in Genetics* 12.March (2021), pp. 1–14. ISSN: 16648021. DOI: 10.3389/fgene.2021.571231.

[23] Gerald Litwack. *Human Biochemistry*. 2nd ed. Elsevier Inc., 2020.

[24] Ingrid Lönnstedt and Terry Speed. "Replicated Microarray Data". In: *Statistica Sinica* 12.1 (2002), pp. 31–46.

[25] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome Biology* 15.12 (2014), pp. 1–21. ISSN: 1474760X. DOI: 10.1186/s13059-014-0550-8.

[26] Aaron T.L. Lun, Yunshun Chen, and Gordon K. Smyth. "It's DE-licious: A recipe for differential expression analyses of RNA-seq experiments using quasi-likelihood methods in edgeR". In: *Methods in Molecular Biology* 1418 (2016), pp. 391–416. ISSN: 10643745. DOI: 10.1007/978-1-4939-3578-9_19.

[27] Steven P. Lund et al. "Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken dispersion estimates". In: *Statistical Applications in Genetics and Molecular Biology* 11.5 (2012). ISSN: 15446115. DOI: 10.1515/1544-6115.1826.

[28] Elie Maza. "In papyro comparison of TMM (edgeR), RLE (DESeq2), and MRN normalization methods for a simple two-conditions-without-replicates RNA-seq experimental design". In: *Frontiers in Genetics* 7.SEP (2016), pp. 1–8. ISSN: 16648021. DOI: 10.3389/fgene.2016.00164.

[29] Belinda Phipson and Gordon K Smyth. "Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn". In: *Statistical Applications in Genetics and Molecular Biology* 9.1 (2010), pp. 1–12. ISSN: 15446115. DOI: 10.2202/1544-6115.1585. arXiv: 1603.05766.

[30] Matthew E. Ritchie et al. "Limma powers differential expression analyses for RNA-sequencing and microarray studies". In: *Nucleic Acids Research* 43.7 (2015), e47. ISSN: 13624962. DOI: 10.1093/nar/gkv007.

[31] Mark D. Robinson and Alicia Oshlack. "A Scaling normalization method for differential expression analysis of RNA-Seq Data". In: *Genome Biology* 11.3 (2010), pp. 1–9. ISSN: 14747596. URL: http://genomebiology.com/2010/11/3/R25.

[32] Mark D. Robinson and Gordon K. Smyth. "Moderated statistical tests for assessing differences in tag abundance". In: *Bioinformatics* 23.21 (2007), pp. 2881–2887. ISSN: 13674803. DOI: 10.1093/bioinformatics/btm453.

[33] Mark D. Robinson and Gordon K. Smyth. "Small-sample estimation of negative binomial dispersion, with applications to SAGE data". In: *Biostatistics* 9.2 (2008), pp. 321–332. ISSN: 14654644. DOI: 10.1093/biostatistics/kxm030.

[34] Nicholas J Schurch et al. "How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use?" In: *Rna* 22.6 (2016), pp. 839–851. ISSN: 14699001. DOI: 10.1261/rna.053959.115.

[35] Gordon K Smyth. "Linear models and empirical bayes methods for assessing differential expression in microarray experiments". In: *Statistical Applications in Genetics and Molecular Biology* 3.1 (2004). ISSN: 15446115. DOI: 10.2202/1544-6115.1027.

[36] Charlotte Soneson and Mauro Delorenzi. "A comparison of methods for differential expression analysis of RNA-seq data". In: *BMC Bioinformatics* 14 (2013). ISSN: 14712105. DOI: 10.1186/1471-2105-14-91.

[37] Rory Stark, Marta Grzelak, and James Hadfield. "RNA sequencing: the teenage years". In: *Nature Reviews Genetics* 20.11 (2019), pp. 631–656. ISSN: 14710064. DOI: 10.1038/s41576-019-0150-2. URL: http://dx.doi.org/10.1038/s41576-019-0150-2.

[38] *TCGA*. URL: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga.

[39] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. "The Cancer Genome Atlas (TCGA): An immeasurable source of knowledge". In: *Wspolczesna Onkologia* 1A (2015), A68–A77. ISSN: 14282526. DOI: 10.5114/wo.2014.47136.

8 Appendix