Differential Gene Expression Analysis using Resampling

by

Yifan Yang

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Science

in

Biology

Waterloo, Ontario, Canada, 2024

© Yifan Yang 2024

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The primary objective in high-throughput sequencing is to identify differentially expressed genes, which provides substantial information of gene expression and regulation. A persistent challenge in this field is the bias caused by limited replicates. To address this, we have developed a novel Bootstrapping method. This approach enhances the power of DEG detection by augmenting the sample data points. New data points are generated through weighted geometric combinations of bootstrap samples and a pooled distribution. The pooled distribution consists of sample data from genes with similar expression levels.

Through simulation tests and evaluations on real-world data, our proposed Bootstrapping method exhibited competitive performance compared to common DEA tools (edgeR, DESeq2, Limma-Voom).

A key advantage of the Bootstrapping method is its independence from any assumptions about the sample distribution. This independence avoids the bias raised from inaccurate assumptions, offering the potential for broad application across various areas of genomic research.

Acknowledgements

I would like to thank the following people who have helped me in the completion of this research:

My supervisor Dr. Brendan McConkey, for his consistent support and patience during the running of the project. I am deeply appreciative of his encouragement guidance during challenging times when progress seemed elusive.

My committee members Dr. Liang and Dr. Doxey, for generously sharing knowledge and expertise during the research.

The Biology Department at the University of Waterloo for their invaluable input and financial support throughout my MSc program.

My friends, who supported me and endured my stresses over the past three years of study.

My parents, whose unconditional love and support sustained me throughout this research journey.

Finally, many thanks to lab members, librarians, peers, and others who provided suggestions and contributed to making this research possible.

Dedication

This is to my mom and dad.

Even overseas, you gave me the biggest support.

Table of Contents

Author's Declaration iii

Abstract iv

Acknowledgements v

Dedication vi

List of Figures ix

List of Tables xi

List of Abbreviations xii

Chapter 1 Introduction 1

1.1 RNA transcripts 1

1.2 History of transcriptomics technologies 1

1.2.1 DNA sequencing techniques 2

1.2.2 RNA sequencing techniques 4

1.3 Computational analysis of RNA sequencing data 6

1.4 The gap of current DEG analysis tools 10

1.4.1 The current methods 10

1.4.2 The proposed method 12

Chapter 2 Literature Review 13

2.1 The negative binomial-based methods 14

2.1.1 edgeR 15

2.1.2 DESeq2 25

2.2 Methods based on the normal distribution 34

2.2.1 Limma-Voom 35

Chapter 3 Methodology 42

3.1 Outline of new proposed method 42
3.2 Input section 43
3.2.1 Input of raw data 43
3.2.2 Unsupervised clustering 46
3.2.3 Filtering raw data 47
3.2.4 Normalization 51
3.2.5 Log-CPM transformation 54
3.3 Processing section 58
3.3.1 Bootstrapping 60
3.3.2 Mean estimator 65
3.3.3 Variance estimator 69
3.3.4 Variance stabilization 78
3.3.5 Pooled distribution 79
3.3.6 Simulation study 84
3.3.6.1 Weighted arithmetic mean 84
3.3.6.2 Weighted geometric mean 93
3.4 Output section 97
3.4.1 Statistical testing 97
Chapter 4 Results and Discussion 100
4.1 Simulation testing 100
4.2 Real-world testing 113
Chapter 5 Conclusion 129
References 132
Appendix 142

List of Figures

Figure 1 Scatterplot of BCV Distribution 19
Figure 2 Scatterplot of the QL Dispersion 25
Figure 3 Shrinkage Estimation of Dispersion in DESeq2 29
Figure 4 Shrinkage Estimation of Dispersion on Trastuzumab dataset 30
Figure 5 Shrinkage Estimation of LFC 33
Figure 6 Effect of Voom Transformation 40
Figure 7 Flowcharts of Common DGE Algorithms 41
Figure 8 MDS Analysis on Trastuzumab dataset 47
Figure 9 Density Plots Comparing log-CPM Data Before and After Filtering 50
Figure 10 Boxplots of Normalized vs. Unnormalized Data Distribution in Trastuzumab Dataset 53
Figure 11 Boxplots of Normalized vs. Unnormalized Data Distribution in Test Dataset 54
Figure 12 Mean-Variance Distribution Before and After Log-Transformation 58
Figure 13 Resampling Accuracy in Estimating Mean Across Sample Sizes 68
Figure 14 Resampling Variability in Estimating Mean Across Sample Sizes 69
Figure 15 Mean-Variance Distribution of Trastuzumab dataset 70
Figure 16 Resampling Accuracy in Estimating Variance Across Sample Sizes 72
Figure 17 Variability in Estimating Variance Across Sample Sizes: Resampling vs. Initial Sample .. 73
Figure 18 Resampling (After Adjustment) Accuracy in Estimating Variance Across Sample Sizes .. 75
Figure 19 Resampling (After Adjustment) Variability in Estimating Variance Across Sample Size .. 76
Figure 20 Density Plot of Estimated Variances After Adjustment Across Sample Size 77
Figure 21 Scatterplot Illustrating the Pooled Distribution Selection 80
Figure 22 Scatterplot Comparing Sample Variances to True Variances 90
Figure 23 Comparison Between Estimates of Variances 96
Figure 24 MA-plot Using Trastuzumab Dataset 101
Figure 25 Power and FPR of Algorithms Across Combination of Sample Size and Negative FC 103
Figure 26 Power and FPR of Algorithms Across Combination of Sample Size and Positive FC 104
Figure 27 Power of Algorithms in Simulation Studies with Negative FC 108
Figure 28 Power of Algorithms in Simulation Studies with Positive FC 109
Figure 29 FPR of Algorithms in Simulation Studies 110
Figure 30 Outlier handling: Power-FPR Curves of Algorithms Across Sample Size 112

Figure 31 Diagram of the Evaluation Scheme Using Real RNA-Seq Data 114
Figure 32 Estimated Power of Algorithms Using Trastuzumab Dataset..... 116
Figure 33 Estimated FPR of Algorithms Using Trastuzumab Dataset..... 117
Figure 34 Estimated Power of Algorithms Using Saccharomyces Cerevisiae Dataset..... 120
Figure 35 Estimated FPR of Algorithms Using Saccharomyces Cerevisiae Dataset..... 121
Figure 36 Estimated Power of Algorithms Using Mus Musculus Dataset..... 123
Figure 37 Estimated FPR of Algorithms Using Mus Musculus Dataset..... 124
Figure 38 Estimated Power of Algorithms Using Arabidopsis Dataset: 125
Figure 39 Detected DEGs using Arabidopsis Dataset:..... 126
Figure 40 Trend of Detected DEGs with Sample Size Variation in Arabidopsis Dataset: 127

List of Tables

Table 1 Comparison of Transcriptomics Methods 6
Table 2 Example Differential Gene Expression Analysis Methods 10
Table 3 Description of Four Public RNA-Seq Datasets 128

List of Abbreviations

cDNA	Complementary DNA
CML	Conditional Maximum Likelihood
CPM	Counts Per Million
CR	Cox-Reid profile-adjusted likelihood
DEA	Differential Expression Analysis
DEG	Differential Expressed Gene
EB	Empirical Bayesian
FDR	False Discovery Rate
FPR	False Positive Rate
GLM	Generalized Linear Model
LFC	Logarithmic (base 2) Fold Change
mRNA	Messenger RNA
NGS	Next Generation Sequencing
qCML	Quantile-adjusted Conditional Maximum Likelihood

Chapter 1

Introduction

1.1 RNA transcripts

A messenger RNA transcript (mRNA) is recognized as a cellular messenger, responsible for carrying genetic information from DNA to ribosomes and serving as a template for protein synthesis. The regulation of RNA transcription and processing has a direct impact on protein synthesis and plays a critical role in mediating cellular functions and determining the phenotype of the cells. Given these significant roles, modern RNA analysis has evolved to encompass more than just quantification. Current studies include the study of post-transcriptional modifications, RNA splicing, and RNA degradation. This broadened scope provides a more comprehensive understanding of RNA processing and the control of transcripts, further clarifying their roles in cellular phenotype regulation (Cook et al., 2015; Z. Lu & Chang, 2016; Wan et al., 2011).

Due to the essential role RNA plays as an intermediate between the genome and the proteome, the development of accurate transcriptomic measurement techniques is crucial. These sequencing techniques enable researchers to accurately quantify and characterize RNA molecules, thus offering deeper insights into the complex dynamics of gene expression and regulation.

1.2 History of transcriptomics technologies

Given the central role of RNA transcripts in RNA synthesis, processing, and cellular phenotype regulation, transcriptomics technologies have been pervasively spread and become standard protocols used in life science research community (Conesa et al., 2016).

The origin of transcriptomics technologies can be traced back to 1953, when Watson and Crick unveiled the double-helix structure of DNA. This discovery paved the way for many subsequent advancements and the creation of the discipline of molecular biology (reviewed in Hong et al., 2020). These subsequent discoveries, such as the concept of messenger RNA (mRNA), clarified the transmission of genetic information from DNA to proteins (reviewed in Cobb, 2015), highlighting the importance of understanding gene sequences.

The history of RNA sequencing technologies development is highly parallel to DNA sequencing development. In fact, before the invention of microarrays, direct sequencing of RNA was not the standard approach. Instead, the sequencing of RNA was typically reverse transcribed into complementary DNA (cDNA) and then analyzed using DNA sequencing methods.

1.2.1 **DNA sequencing techniques**

The Sanger sequencing method, developed by Frederick Sanger and colleagues in the 1970s, was one of the earliest DNA sequencing technologies. It revolutionized the field of molecular biology by providing a direct method to read the genetic code. The Sanger sequencing process starts with a short DNA primer sequence complementary to the interrogated template DNA. Using polymerase enzymes, this primer subsequently extends by incorporating complementary deoxynucleotide triphosphates (dNTPs). Falling in the category of a "chain termination method", this method relies on dideoxynucleotide triphosphates (ddNTPs) that lack a 3' hydroxyl group, halting DNA synthesis when incorporated. These ddNTPs are fluorescently labeled, which allows for the identification of the incorporated nucleotides. The precise DNA sequence is then ascertained by evaluating the lengths of the resultant fragments via capillary electrophoresis (Alekseyev et al., 2018; Hong et al., 2020; Jamuar et al., 2016). However, due to its capability to

analyze only one sequence per electrophoresis lane, the efficiency of Sanger sequencing is now limited and is classified as a low-throughput sequencing method (Alekseyev et al., 2018; Lowe et al., 2017).

The high-throughput sequencing revolution started in the early 2000s. These methodologies, often termed Next Generation Sequencing (NGS), benefits from fast speed, low sequencing cost and high accuracy (Hong et al., 2020). Pioneering platforms during this period included Illumina, which employs a process called “bridge amplification”; 454 (commercialized by 454 Life Sciences), utilizing a pyrosequencing method; and SOLiD, which utilizes a ligation-based sequencing approach (Giani et al., 2020). Unlike the fragment-by-fragment process of Sanger sequencing, these NGS technologies enabled parallel processing of millions of DNA fragments, significantly enhancing the depth and resolution of genomic studies and facilitating the detection of rare genetic variants (Qin, 2019).

Building on the successes of NGS, third-generation sequencing technologies emerged, capable of sequencing individual DNA molecules in real time. Pacific Biosciences (PacBio) introduced Single Molecule Real-Time (SMRT) sequencing in 2011, notable for producing exceptionally long reads. Typically, SMRT sequencing yields median read lengths of 10–20 kbp, with some reads even surpassing 50 kbp (Giani et al., 2020; Heather & Chain, 2016). A direct competitor in this space is Oxford Nanopore Technologies' (ONT) nanopore sequencing. In this method, DNA strands pass through bacterial nanopores inserted into an artificial lipid bilayer. As the DNA traverses the pore, it causes disruptions in an electric current. These disruptions, unique to each nucleotide, are recorded and translated into a DNA sequence. Typically, nanopore sequencing produces reads in the range of 10–20 kbp, with the longest sequenced molecule reaching an

impressive 2.3 Mbp (Giani et al., 2020; Heather & Chain, 2016). These technological advances not only refined DNA sequencing but also brought significant enhancements to RNA studies.

1.2.2 **RNA sequencing techniques**

The dominant contemporary techniques, microarray and RNA-Seq, were developed in the mid-1990s and 2000s, respectively. Microarray sequencing is based on the principle of complementary sequences hybridizing, allows for the simultaneous sequencing of thousands of transcripts with low cost and labor (Lowe et al., 2017; Yee & Ramaswamy, 2010). In the sequencing process, RNA fragments from an unknown sample are first tagged with fluorescent markers, then exposed to a 'chip' covered with an array of nucleotide oligomers. These fragments hybridize with complementary sequences on the chip. The positions and fluorescence intensities of these hybridized fragments can be quantified using image analysis techniques, which provides the valuable sequencing information (Govindarajan et al., 2012; Guigó, 2013; Lowe et al., 2017; Marroki & Bousmaha-Marroki, 2022).

Given the large arrays of nucleotide oligomers on the chip, microarray can measure many transcripts simultaneously. Celebrated for its speed and specificity, microarrays became a key tool in drug development and clinical research, dominating the field for over 16 years (Govindarajan et al., 2012; Law et al., 2014a). However, in recent years, microarrays have gradually been replaced by RNA-Seq due in part to the limitations of requiring pre-defined gene sets (Chang & Chen, 2009; Hong et al., 2020; TachibanaJul. 31 et al., 2015).

RNA-Seq, emerging in the 2000s, refers to the sequencing of transcript cDNAs (typically copied from mRNA) with abundance extracted from individual transcript counts. It also termed as next-

generation sequencing (NGS), brought several notable improvements over its microarray predecessors. First, RNA-Seq is not limited to detecting transcripts that correspond to existing genomic sequences. Secondly, different from microarrays whose sensitivity and dynamic range are constrained by fluorescence detection and saturation. RNA-Seq, on the other hand, boasts superior sensitivity with no set upper qualification limit as it is determined by the acquired sequence count (Hong et al., 2020; Hurd & Nelson, 2009; Wang et al., 2009).

Table 1 shows a comparative analysis of three primary transcriptomics techniques mentioned. RNA-Seq, in particular, shows remarkable improvements in efficiency, cost-effectiveness, sensitivity, and reduced background noise (Hong et al., 2020; Soneson & Delorenzi, 2013; Stark et al., 2019). Thus, RNA sequencing has evolved from a simple genomic data collection method to a pivotal instrument for detailed biological analysis (Kukurba & Montgomery, 2015; Stark et al., 2019; Thompson & Milos, 2011).

Table 1 Comparison of Transcriptomics Methods

Method	RNA-Seq	Microarray	Sanger Sequencing
Technology specifications			
Throughput	High	High	Low
Background noise	Low	High	High
Sensitivity	10^{-6} (limited by sequence coverage)	10^{-3} (limited by fluorescence detection)	10^{-1}
Application			
Dynamic range	$>10^{5}$ (limited by sequence coverage)	$10^{3} – 10^{4}$ (limited by fluorescence saturation)	Not practical
Cost for mapping transcriptomes of large genomes	Relatively low	High	High

1.3 Computational analysis of RNA sequencing data

RNA-Seq experiments generated a substantial volume of raw sequencing data, and while mostly accurate, may also contain biases. Thus, computational analysis becomes a critical subsequent step to mitigate these biases and identify differentially expressed genes (Lowe et al., 2017). It's important to note that, although growing numbers of computational analysis tools have been developed tailored for different specific research designs and analytical goals over the past decade, the fundamental steps for computational analysis have largely remained unchanged since their introduction. These steps includes: ***quality control, alignment and assembly, expression***

quantification*, and *differential expression analysis (Conesa et al., 2016; Hong et al., 2020; Lowe et al., 2017; Soneson & Delorenzi, 2013; Stark et al., 2019, 2019; Zhao et al., 2016).

Quality control

Sequence reads, while powerful, are not perfect representations of the target mRNA population. As the contaminated sequences could potentially lead to false positive and false negative errors, consistent application of quality control checks throughout all stages of analysis is essential to ensure the reproducibility and reliability of the results. A range of established ***quality control*** and preprocessing tools, such as **FASTQ**, **NGSQC**, **RNA-SeQC**, **Trimmomatic**, **PRINSEQ**, and **Soapnuke**, are available to researchers. These tools each utilize diverse strategies to optimize the integrity of the data (Givan et al., 2012; He et al., 2020; Hong et al., 2020; Zhao et al., 2016).

Alignment and assembly

Following ***quality control***, typically the next step involves aligning high-quality reads against a reference genome, transcriptome, or exome. However, in situations where a suitable reference dataset is not available, de novo alignment is employed. This method involves aligning reads to each other to construct transcripts. Tools such as **StringTie** and **SOAPdenovo -Trans** have been developed to facilitate the reconstruction of full-length transcript models from short-read sequences which use the identified gaps during alignment to determine exon boundaries and splicing sites. It is important to note, while powerful, de novo transcript assembly tends to be more computationally demanding than its reference-based counterpart, and often necessitates additional validation steps for identification of gene variants (Lowe et al., 2017; Zhao et al., 2016).

After establishing the transcript model, the aggregated counts of reads mapped to relevant genomic regions serve as indicators for feature abundance across different samples. The quality of these alignments, particularly in relation to a reference genome, is crucial as it directly impacts the accuracy of gene variant predictions in the subsequent ***quantification*** step. As such, in recent years, a variety of advanced mapping tools (**Tophat2**, **HISAT2**, and **Bowtie**, etc.) have emerged, enhancing both the sensitivity and specificity of aligners, furthermore improving the overall accuracy of RNA-Seq analyses (Poullet & Orlando, 2020; Stark et al., 2019; TachibanaJul. 31 et al., 2015; Zhao et al., 2016).

Quantification

Numerous computational tools have been developed to quantify transcript abundance from RNA-Seq data. A rigorous evaluation of their performance before choosing one for use is critical, as it affects the sensitivity and specificity of the results. The selection of appropriate tools should consider factors such as the nature of the sample, the sequencing platform used, and the intended downstream analysis. For gene-level quantification, established tools **FeatureCount** and **HTSeq-count** are frequently utilized. On the other hand, for transcript-level quantification, tools like **Cufflinks**, **eXpress**, and **RSEM** are often the preferred choices (Hong et al., 2020).

The outcome of the ***quantification*** process is usually compiled into an expression matrix where each row represents an expression feature (either gene or transcript), and each column corresponds to a sample. The values within the matrix signify either the estimated abundances or actual read counts (Hong et al., 2020; Stark et al., 2019).

Differential expression analysis

In the final stage of the computational analysis pipeline for RNA-Seq data, ***differential expression analysis*** is conducted. Using the expression matrix as a basis, this step assesses variations in gene transcript abundance across different groups within a transcriptome. Genes exhibiting significant differences in expression levels are identified as differentially expressed genes (DEGs). The object of Differential gene expression (DGE) tools, such as **DESeq2** and **Limma-Voom**, is identifying these DEGs, furthermore, unraveling the intricate genetic mechanisms that govern organisms (Conesa et al., 2016; Hong et al., 2020; McDermaid et al., 2018).

A common challenge in these differential expression analysis is the presence of variability across samples, which arises from differences in experimental design, biological variation, and biases associated with limited sample sizes (Soneson & Delorenzi, 2013). Each methodology adopts unique strategies to counteract these non-uniformities, each with its advantages and limitations (Conesa et al., 2016; Hong et al., 2020; Zhao et al., 2016). The heart of these methods is their capacity to distinguish between DE (differential expressed) and non-DE genes. Currently, **DESeq2** and **edgeR**, employing negative binomial distribution models, and **Limma-Voom**, which is based on mean-variance transformation, are the most prevalent tools in differential gene expression analysis. Models of these dedicated software are catalogued in Table 2. A comprehensive discussion of these tools, including their specific functionalities and applications, will be undertaken in Chapter 2.

Table 2 Example Differential Gene Expression Analysis Methods

Software	Environment	Model
edgeR	R/Bioconductor	Negative binomial distribution + GLM
DESeq2	R/Bioconductor	Negative binomial distribution + GLM
Limma-Voom	R/Bioconductor	Mean-variance transform + GLM

1.4 The gap of current DEG analysis tools

1.4.1 **The current methods**

Both DGE analysis methods of **edgeR** and **DESeq2** are based on the assumption that gene distributions follow a negative binomial distribution. Central to this assumption is adding the gene-wise dispersion parameter, which accounts for biological variability, to a simple Poisson model. Accurate estimation of this dispersion parameter is pivotal where underestimation leads to high false positive error and overestimation reduces the statistical power. However, given the typically small sample sizes in RNA-Seq experiments, obtaining reliable gene-wise dispersion estimates is challenging. Pooling information across neighboring genes is the common strategy used to enhance the accuracy of estimation. **edgeR** and **DESeq2** both utilize Empirical Bayesian approach to leverage this pooled information, but they differ in emphasis: **edgeR** focuses more on sample-wise dispersion, whereas **DESeq2** relies more on the fitted curve of the mean-variance relationship.

However, like **edgeR** and **DESeq2**, many tools based on the negative binomial (NB) distribution treat estimated dispersions as fixed parameters, neglecting the inherent uncertainty in their estimation. This oversight can result in statistical tests that are overly liberal in specific contexts, leading to a high false positive rate. Additionally, the benefits of parametric methods diminish when the observed mean-variance relationship doesn't align perfectly with the theoretical mean-variance relationship (Law et al., 2014a).

The **Limma-Voom** method employs a different strategy, combining a variance-stabilizing transformation with **limma**. Originally developed for microarray data, **limma** assumes a linear relationship between gene expression levels and experimental conditions. It employs empirical Bayes methods to estimate parameters in linear combinations and values of gene-wise variances, borrowing information from neighboring genes (Ritchie et al., 2015). Given that **limma** was initially developed for microarray data, it is built under the assumption of homoscedasticity or the transformed variables (e.g. the variance of residuals is constant across all expression levels). However, this assumption often does not hold for RNA-Seq data. To address this heteroskedasticity, the **voom** and **limma-trend** methods are employed (Ritchie et al., 2015). The details of this method will be elaborated in section 2.2.1. Although the DGE test of **Limma-Voom** have a lower false positive rate and is less sensitive to outliers compared to **edgeR** and **DESeq2**, it suffers from reduced sensitivity (Soneson & Delorenzi, 2013).

All of these methods are based on specific parametric assumptions, and deviations from these assumptions can lead to unexpected false discovery rates. This has been further demonstrated by simulation studies conducted by Li et al. (2022).

1.4.2 **The proposed method**

We propose a novel method to identify differentially expressed genes with the goal of minimizing detection errors and reducing the need for a specific distribution model. The core of our method is the bootstrapping approach which estimates population parameters non-parametrically. The advantage of using bootstrapping lies in its lack of reliance on specific assumptions about the underlying data distribution, thus minimizing potential biases inherent in such assumptions.

In our approach, we enhance the accuracy of variance estimation, especially with limited sample size, by integrating pooled information from neighboring genes and sample information for each gene. This integration effectively augments sample data points—such as generating 100 data points per gene per condition—allowing us to refine variance estimates for each gene across all conditions. Leveraging this augmented data distribution, we then apply a t-test to an estimated fold change distribution to determine significant differential gene expression. This improved estimate of the sample distribution allows for a more robust and reliable identification of differentially expressed genes. Moreover, it aids in accurately estimating key statistical metrics: the effect size of the gene expression differences, the statistical significance (alpha level), and the power (1-beta) of the experiment.

This bootstrapping-based method presents a promising route for RNA-Seq data analysis, offering an innovative solution to the common challenge of small sample sizes. By circumventing traditional distributional assumptions, our method aims to provide a more accurate and reliable approach for identifying differentially expressed genes.

Chapter 2

Literature Review

The comparison between DGE analysis tools

While it is claimed that the process of sequencing generates unbiased gene expression data for DGE analysis, it is not always the case in reality. Consequently, preprocessing steps like *filtering* and *normalization* are essential to minimize biases introduced by study design or operational errors. While numerous software packages have been developed to meet these preprocessing needs, there is no consensus on the optimal analysis approach. Each package has its own set of strengths and weaknesses. While selection, researchers must take into account several critical parameters of RNA-Seq technology, such as the number of replicates, sequencing depth, and balanced vs. unbalanced sequencing depth within and between groups (Zhang et al., 2014).

The crucial step in differential gene expression (DGE) analysis is variance stabilization, for which most existing tools rely on the assumptions of gene counts distribution. Initially, given the nature of sequencing techniques, it's logical to expect that a statistical model for gene counts should approximately follow a Poisson distribution. Indeed, this was confirmed during the early stages of RNA-Seq researches, where the distribution of feature counts among technical replicates fit well to the Poisson distribution, at least in the absence of biological replicates (Bullard et al., 2010; Robinson & Smyth, 2007). However, the introduction of biological replicates presented a new challenge in the form of overdispersion. This issue arises from the entanglement of biological variation, technical variation, and biases in the RNA-sequencing pipeline, leading to observed variances greater than what the Poisson distribution predicts.

Consequently, Poisson-based analyses are prone to a high false positive rate, necessitating more sophisticated models that can account for overdispersion (Robinson & Smyth, 2007; Zhang et al., 2014).

2.1 The negative binomial-based methods

To address the issue of overdispersion, the usage of beta-binomial (Baggerly et al., 2003), over-dispersed logistic (Baggerly et al., 2004) and over-dispersed log-linear models (J. Lu et al., 2005) have been explored. Among these models, the negative binomial distribution stands out as it incorporates a scale factor for variance, enabling more accurate isolation of genuine biological differences between groups. This leads to a more precise and reliable identification of differentially expressed genes (DEGs). Within this assumption, **edgeR** and **DESeq2** are two widely adopted representatives for differential expression analysis (DEA) (Robinson et al., 2010; Robinson & Smyth, 2007; Zhang et al., 2014).

In the sequencing processing pipeline, the steps prior to DEG analysis yield a matrix that contains the count values of each gene across all samples. DEA tools extract sample information from the matrix and apply a model-based approach to estimate the true gene expression levels. This estimation process involves assuming the counts of each gene g in each sample i follows negative binomial distribution, which can be mathematically represented as:

$$Y_{gi} \sim NB(M_i p_{gj}, \phi_g)$$

Where M_i is the library size of sample i, ϕ_g represents the dispersion factor for gene g and p_{gj} is the relative abundance of gene g in experimental group j to which sample i belongs. The NB

distribution has a mean of $\mu_{gi} = M_i p_{gj}$ and a variance of $\mathrm{var}(Y_{gi}) = \mu_{gi}(1 + \mu_{gi}\phi_g)$. Dividing both sides by ${\mu_{gi}}^2$ gives the relationship of coefficient of variance:

$$CV^2(Y_{gi}) = \frac{1}{\mu_{gi}} + \phi_g$$

As the term $\frac{1}{\mu_{gi}}$ equals the squared CV of the Poisson distribution, accounting solely for technical variance, the dispersion variable, ϕ_g captures the coefficient of variation of biological variation among samples (Robinson et al., 2010). This model allows effectively separation of biological variation from technical variation. Consequently, accurately estimating the dispersion parameters, ϕ_g, becomes a critical aspect of the inference procedure.

However, the often-small sample sizes in RNA-Seq studies makes reliable gene-wise dispersion parameter estimates challenging. To address this issue, both **DESeq2** and **edgeR** utilize the approaches that pools sample data across the entire dataset, serves as supplementary information for more stable estimates (Huang et al., 2015). Although both of these two methods achieve this using Empirical Bayes methods, the specific methodologies they are based on are different and are described below.

2.1.1 **edgeR**

edgeR (empirical analysis of DGE in R) is a software package based in R and available through the Bioconductor project (REF). Its methodology, developed by Robinson & Smyth, based on the assumption of gene expression levels follow negative binomial distribution. **edgeR** estimates the dispersion coefficients through an empirical Bayes procedure, which combines estimates of

variability from individual genes with a pooled estimate derived from all genes to achieve more stable and robust results (Robinson et al., 2010).

Tailored to different experimental designs, the differential gene expression analysis in **edgeR** developed two distinctive methods: the quantile-adjusted conditional maximum likelihood (qCML) tailored for single-factor experiments, and the more recently developed Cox-Reid profile-adjusted likelihood (CR) catering to multi-factor experiments (Chen et al., 2022).

Common dispersion

For experiments with single factor, the *common dispersion* model was initially proposed assuming all genes with same dispersion parameter ϕ. The parameter ϕ is calculated using the quantile-adjusted conditional maximum likelihood (qCML) method. It serves as a corrective measure for observed variations in count distributions that caused by the disparities in library sizes among samples. Specifically, qCML generate pseudo-data to adjust the observed counts based on whether the corresponding library sizes fall below or exceed the geometric mean of all samples (Huang et al., 2015; Robinson & Smyth, 2008). After qCML addresses the variability arising from differing sample size, the value of common dispersion estimator ϕ is estimated by maximizing the overall likelihood:

$$\mathrm{l_c}\,(\phi) = \sum_{g=i}^{G} l_g(\phi)$$

where G is the number of genes and $l_g(\phi)$ is the conditional log-likelihood for g^{th} gene which is determined by conditioning on the sum of counts for each condition. (Robinson & Smyth, 2007, 2008). This derived value of ϕ is the "*common dispersion*" shared by all the genes in dataset.

Trended dispersion

An extension to this "*common dispersion*" approach is to put a mean-dependent trend as a parameter in the dispersion function. As a result, genes with identical expected counts share the same dispersion, known as "*trended dispersion*". These approaches provide a more reliable estimate of dispersion, have proven particularly valuable when analyzing datasets with limited libraries (Robinson & Smyth, 2007, 2008). Nonetheless, while useful, it can sometimes oversimplify the diverse nature of gene expression, leading bias in DEA results. This recognition has prompted the development of subsequent methodologies that aim to estimate dispersions on a gene-by-gene basis.

Tag-wise dispersion (gene-wise dispersion)

Although the assumption of a single, common dispersion for every gene in the negative binomial distribution offers simplicity, it often falls short in accommodating the extensive variability inherent in gene expression data. This brings the attention to *gene-specific dispersions* and led to the consideration of a hybrid model that combines both common and gene-wise dispersions (Robinson & Smyth, 2007).

A pioneering method in this direction was introduced in **limma** package, which employs Empirical Bayesian (EB) hierarchical models to estimate dispersion in microarray data (Smyth, 2004). However, directly applying **limma**'s strategy to sequencing data presents computational challenges. This is primarily due to the negative binomial (NB) distribution—unlike the normal distribution—doesn't have a conjugate prior for ϕ_g (Robinson & Smyth, 2007). To solve this issue, **edgeR** introduced a polynomial approximation technique. This method employs the

weighted likelihood approach to emulate the EB solution. In this model, the weighted conditional log-likelihood (WL) for ϕ_g is considered as a combination of specific and common likelihoods:

$$WL(\phi_g) = I_g(\phi_g) + \alpha I_c(\phi_g)$$

Where α determines the amount of moderation given to the common likelihood. If $\alpha = 0$, the equation simplifies to gene-wise estimates. A larger α implies that the individual dispersion estimates are more robustly moderated or "pulled" towards the common estimate. As the proposed weighted combination is trying to simulation EB process, the appropriate value of α is selected to when $\widehat{\phi_g^{WL}}$ (the maximum estimator of $WL(\phi_g)$) coincides with EB solution $\widehat{\phi_g^B}$ (Robinson & Smyth, 2007).

Figure 1 offers a visualization of dispersion estimation results from Trastuzumab data (Rinas, 2020), which consists of three groups each from the Control and Herceptin cohorts. This graph shows the raw dispersion alongside the dispersion estimates derived from three strategies mentioned above. Evidently, the *tag-wise dispersion* estimates experience a "shrinking" effect, moving from the raw dispersion values towards the more general, common values. We also observed a discernible trend: with increasing abundance, the dispersion gradually decreases and asymptotic to a consistent value.

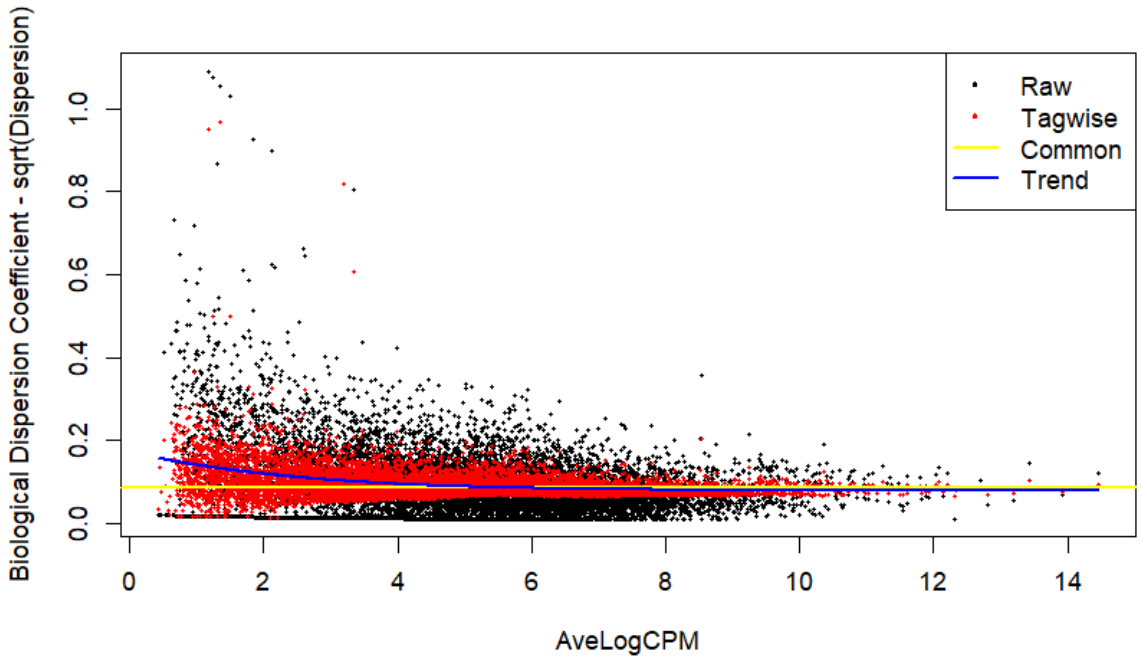

Figure 1 Scatterplot of BCV Distribution: *This scatterplot shows the distribution of the biological coefficient of variation (BCV) against the average abundance of each gene of raw, common, trend and tagwise dispersions. These estimations were the results of 6 randomly selected samples from Trastuzumab data (Rinas et al., 2020) with 3 samples each in the control group and Herceptin group. Black dots depict the raw dispersion, whereas the shrunk tag-wise dispersion is indicated by red dots. The common and trend dispersions are illustrated using yellow and blue lines, respectively. This visualization provides an insightful comparison of different dispersion estimates.*

The exact test

In the analysis of count data, once the dispersion has been estimated, the subsequent step involves assessing significant variations in gene expression. The exact test is used for DGE analysis in single factor, two-level experiments. Given typically small sample size in sequencing data, tests like the Fisher's exact test for 2-by-2 contingency tables is logical choice. Based on the properties of each gene, Robinson formulated the exact test which analogous to the Fisher's test

but based on the assumption of negative binomial distributions (Robinson et al., 2010; Robinson & Smyth, 2008). In the exact test, quantile adjustment is used to create pseudo-counts and adjust the counts to a common library size. Given the inherent property of the NB distribution that a summation of independent and identically distributed NB random variables also follows an NB distribution, the exact p-values are derived by conditioning on the total pseudo-sum. The value of p-value is calculated as the sum of all counts that have a lower probability under the null hypothesis compared to the observed sum of counts (Anders & Huber, 2010; Chen et al., 2022; Love et al., 2014; Robinson & Smyth, 2008).

The Cox-Reid profile-adjusted likelihood (CR) method

However, it's essential to note that the qCML technique is suitable only for datasets with a single-factor design and falls short in accommodating the complexities of multifactorial experiments. For experiments involving multiple factors or intricate designs, alternative approaches, the Cox-Reid profile-adjusted likelihood (CR) offers a more suitable approach in dispersion estimation (Love et al., 2014; Robinson & Smyth, 2007, 2008).

This method takes care of multiple factors by fitting generalized linear models (GLM) with a design matrix. The core of the CR method lies in its use of approximate conditional likelihood, which is grounded in the principles of Residual Maximum Likelihood (REML). REML is particularly adept at handling nuisance parameters - parameters that aren’t of primary interest but might skew results. By adjusting for these parameters, the CR method ensures an unbiased estimation of dispersion, thoroughly accounting for underlying systematic variations present in the experimental setup. This approach is vital in complex experiments where multiple factors and their interactions could potentially skew gene expression results. The ability of CR method in

providing accurate and unbiased dispersion estimates, especially under the intricate nature of experimental design, is a significant strength of **edgeR** in RNA-Seq analysis (Robinson et al., 2010).

Like qCML, the CR approach can be used for various dispersion estimations: *common* (for all genes), *trended* (mean-variance trend), and *tagwise* (individual dispersions for each gene). Mathematically, the Cox-Reid adjusted profile likelihood (APL) for each gene's dispersion ϕ_g is given as a penalized loglikelihood:

$$APL_g(\phi_g) = l(\phi_g; y_g, \widehat{\beta_g}) - \frac{1}{2}\log\det(I_g)$$

Where y_g is the measured counts for gene g, $\widehat{\beta_g}$ is the estimated coefficient of the target gene g, l is the log-likelihood function, I_g is the Fisher information of β_g evaluated at $\widehat{\beta_g}$ and ϕ_g. Maximization of $APL_g(\phi_g)$ leads to obtain an estimate for ϕ_g (Robinson et al., 2010).

Generalized linear models

In the analysis of experiments involving multiple factors, the use of a generalized linear model (GLM) incorporating log-transformed count data becomes a common approach for differential expression analysis. This method allows for the incorporation of multiple biological conditions, while also accounting for variations in the data through the dispersion estimates (Zhang et al., 2014).

The log-linear model elucidates how treatment conditions and library sizes impact the expected count sizes of any given gene. Mathematically, as the counts follow NB distribution $Y_{gi} \sim NB(\mu_{gi}, \phi_g)$, the mean expression in log-linear model for each gene g in sample i follows:

$$log\ \mu_{gi} = x_i^T \beta_g + log\ M_i$$

$$M_{gi} = M_i p_{gj}$$

Where x_i represents the experimental covariates (such as treatment condition, batch effects, etc.) for each sample g and β_g is a vector of regression coefficients by such covariate effects. The relative abundance of gene g in experimental group j (to which sample i belongs) is denoted as p_{gj}.

By fitting this model to the data using Fisher scoring matrix, GLM returns coefficients β_g, which is a vector of regression coefficients representing the log 2 fold change of covariates effects mediated for gene g (Anders & Huber, 2010; Love et al., 2014; Smyth, 2004). The maximum likelihood of β_g can be found using the Fisher scoring iteration, which has been further modified in **edgeR** using a Levenberg damping modification to ensure that the sequence of iterations converges for all genes and all datasets. Covariates linked with coefficients β that deviate significantly from zero suggest notable effects on gene expression (Robinson et al., 2010).

Adjustment of p-values

In the context of RNA-Seq data analysis using a GLM, the LFC for each gene is used in conjunction with p-values to discern differential expression. Given the nature of high-throughput sequencing data, the statistical tests are performed on thousands of genes simultaneously. This massive number of concurrent tests increases the risk of obtaining false positives if relying solely on unadjusted p-values. To this end, multiple methods have been proposed to adjust these p-values to account for the multiple testing scenario (Korthauer et al., 2019).

Bonferroni method is one of popular method control family-wise error rate (FWER), the adjusted p-value is calculated by: p-value $/ m$ (m = total number of tests). While the Bonferroni method is rigorous in reducing the chance of any false positives, its stringent nature might lead to an increase in false negatives. This is especially true in high-dimensional settings like sequencing data, where its conservative nature can overshadow true discoveries (Freund et al., 2010).

An alternative, and often preferred approach in high-throughput biological experiments, is the control of the False Discovery Rate (FDR). The FDR represents the expected proportion of discoveries which are falsely rejected (Korthauer et al., 2019). Given that researchers in many biological settings are willing to tolerate a small proportion of false positives in favor of capturing more true discoveries, FDR control is particularly apt in biological contexts. The framework for FDR was introduced by Benjamini and Hochberg in 1995. In their procedure, genes are ranked by their raw p-values in a non-decreasing order. For each gene, an adjusted p-value is computed based on its rank and the total number of tests using the formula:

$$\text{adjusted p-value} = \left(\frac{\text{rank of the p-value}}{\text{total number of tests}}\right) \times \text{raw p-value}$$

Hypotheses are rejected if their adjusted p-values fall below a specified threshold (Benjamini & Hochberg, 1995). This elegant FDR controlling method is integral to multiple RNA-Seq data analysis tools, including **edgeR**, **DESeq2** and our proposed method **Bootstrapping**.

Quasi-likelihood (QL) method

While the NB model focuses on addressing biological variability which contributes a significant portion of observed variance. It can be extended with quasi-likelihood (QL) methods to account

for gene-specific variability from both biological and technical sources. The variance in QL model is in the form:

$$var(Y_{gi}) = \sigma_g^2(\mu_{gi} + \phi\mu_{gi}^2)$$

In this expression,

- ϕ represents the NB dispersion parameter. It represents a global parameter estimated by empirical mean-dependent relationships, accounting for overall biological variability across all genes.
- σ_g^2 represents the QL dispersion parameter. Its estimation aligns with methods employed by the **limma** package which utilizes the Empirical Bayes (EB) method to borrow information across genes. These values capture gene-specific variability that above or below the overall level (Chen et al., 2016).

The following Figure 2 represents the QL estimation results of Trastuzumab data (Rinas et al., 2020) which consisted of three groups each from the Control and Herceptin cohorts. The y-axis value represents the QL dispersion estimates for each gene. A higher y-value indicates that a particular gene has more specific variability than what the NB dispersion alone can capture.

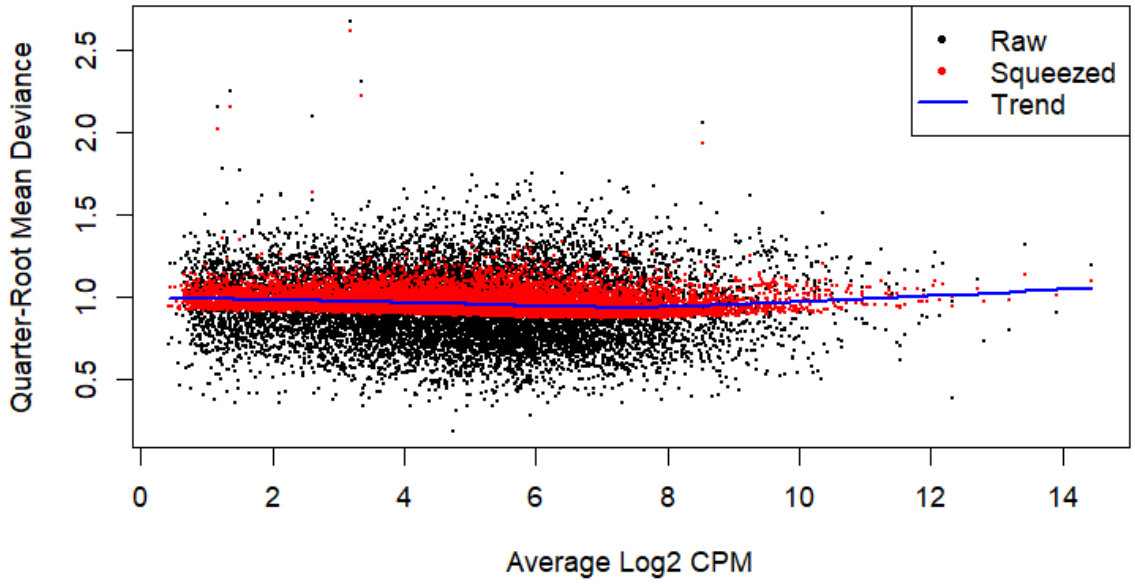

Figure 2 Scatterplot of the QL Dispersion: *This figure shows the scatterplot of the quarter-root quasi-likelihood (QL) dispersion against the average abundance of each gene. The data for this plot originates from a total of six samples, comprising three randomly selected samples from the control group and three from the Herceptin group in Trastuzumab dataset (Rinas et al., 2020). In this scatterplot, black dots depict the raw dispersions (before EB moderation), whereas the trended and squeezed (after EB moderation) dispersions are shown in blue line and red dots respectively.*

2.1.2 **DESeq2**

DESeq2 is another differential expression analysis method based on the assumption of a negative binomial distribution. As a successor to the **DESeq** method, **DESeq2** is an R package available through the Bioconductor project. It enhances the stability and interpretability of estimates by implementing shrinkage for both dispersion and fold changes (Love et al., 2014).

The starting point of DGE analysis is the count matrix with one row for each gene g in and one column for each sample i. With the assumption of negative binomial distribution, the raw count Y_{gi} can be mathematically represented by:

$$Y_{gi} \sim NB(s_{gi}q_{gi}, \phi_g)$$

The negative binomial distribution has a mean of $\mu_{gj} = s_{gi}q_{gi}$, where the mean is calculated as the normalized counts q_{gi} scaled by a normalization factor s_{gi}. Unlike the earlier model in **edgeR**, which applies a constant normalization factor for all genes in a sample to account for differences in sequencing depth, **DESeq2** adopts a more complex approach. It employs the "median-of-ratios" method, calculating a specific normalization factor for each gene in each sample. This method accommodates additional sources of technical bias, such as varying GC content and gene length, thereby providing a more accurate normalization (Love et al., 2014)

After normalization, the expression level of a given gene q_{gi} is modeled as a function of the design matrix element X_{gj} using a logarithmic generalized linear model:

$$\log_2 q_{gj} = \sum_j X_{gj}\beta_{gj}$$

By fitting this model to the data, the GLM returns coefficients β_{gj}, which offers the insights into two main aspects:

1. The overall expression level of the gene.
2. The log2-fold change between different biological conditions which quantifies the difference in expression levels of a particular gene across these conditions.

Thus, the genes with coefficients that differ significantly from zero are considered differentially expressed. The conventional method for identifying such genes is rooted in statistical hypothesis testing (Anders & Huber, 2010; Love et al., 2014). Specifically, for each gene, we are interested in testing the null hypothesis that the logarithmic fold change (LFC) between treatment and control groups is exactly zero:

$$\mathrm{H}_0 : \mathrm{LFC} = \log_2\left(\frac{\text{Expression in control group}}{\text{Expression in treatment group}}\right) = 0$$

Rejecting H_0 implies a notable difference in the gene expression level between the treatment and control group, leading to the classification of the gene as differentially expressed.

However, in well-designed experiments, a statistical test against the conventional null hypothesis of zero may not have biological significance, as it may identify genes with minimal expression changes (e.g., 10%). Such genes are often considered to be irrelevant or distracting (Love et al., 2014). The common procedure involves setting minimum threshold of LFC and selecting only those genes above the set threshold, $|\beta_{gj}| \geq \theta$. But in this thesis the test of LFC equal to zero is retained as a means to compare results across methods.

Variance stabilization

The primary objective of differential analysis is often to generate a list of genes that pass multiple-test adjustments, ranked by p-values. Given that this method relies on accurate dispersion estimation, challenges can arise, especially in the presence of variability for genes with low counts and limited sample size. To counter this, **DESeq2** used shared information across genes to enhance the stability of the dispersion, leading to more reliable detection of differentially expressed genes (Love et al., 2014). It's worth noting that while both **DESeq2** and

edgeR use EB techniques to borrow information from genes within the datasets, their methods diverge in the specific strategies (Robinson et al., 2010). **edgeR** estimates a common dispersion parameter based on all genes and then shrinks gene-specific dispersions toward this common value, effectively using it as a form of prior information. On the other hand, **DESeq2** assumes that dispersions conform to a specific prior distribution, often represented as a function of the mean expression level. It then fine-tunes the degree of shrinkage by estimating the breadth of this prior distribution based on the available data (Love et al., 2014).

The detailed algorithm of **DESeq2** is shown in Figure 3**.** Initially, each gene is evaluated separately, and its dispersion is estimated using a maximum likelihood approach that relies solely on data from that specific gene (represented by black dots). Subsequently, a smoothed curve, or prior distribution, is fitted to these gene-wise maximum likelihood estimates to encapsulate the overall trend of the dispersion-mean relationship (represented by the red line). This curve provides an expected dispersion value for genes with similar average expression levels. The final step involves obtaining the final dispersion values by shrinking the gene-specific dispersions towards the fitted curve using an empirical Bayesian approach (represented by the blue arrowheads). The degree of shrinkage for each specific gene is dependent on two factors: first, an estimate of how closely the true dispersion values align with the fitted curve, and second, the degrees of freedom, which is inversely related to the sample size (Love et al., 2014; Zhang et al., 2014).

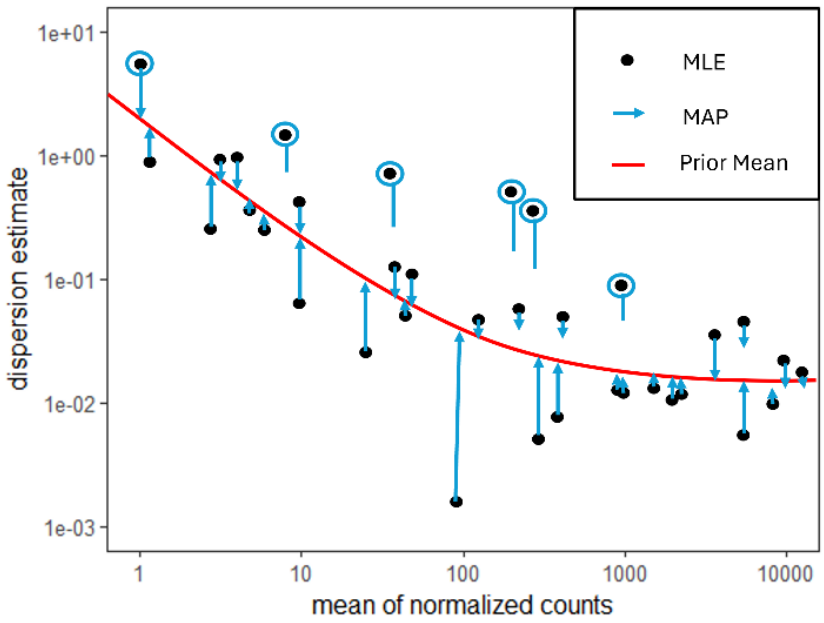

Figure 3 Shrinkage Estimation of Dispersion in DESeq2: *This plot, based on (Love et al., 2014), visualizes empirical Bayes approach on dispersion estimates over the average expression strength. The process begins with obtaining gene-wise maximum likelihood estimates (MLEs) using the data specific to each gene (black dots). Then, a curve (red) is then fitted to the MLEs to capture the overall trend of dispersion-mean relationship. This fitted line is used as a prior mean for a second estimation round, resulting in the final maximum a posteriori estimates of dispersion (arrow heads). This process can be thought of as shrinking the noisy gene-wise estimates towards the consensus represented by the red line (blue arrows). Some black points (circled in blue) are identified as dispersion outliers using Cook's distance. These outliers will keep their original value and not to be shrunk towards the prior (Love et al., 2014).*

In Figure 4, we visualized the dispersion estimation for Trastuzumab data (Rinas et al., 2020), specifically focusing on the Control and Herceptin groups. As illustrated in the graph, the final dispersion estimates undergo a "shrinking" process, moving from the gene-wise estimates towards the fitted values. Additionally, the graph highlights the presence of outliers, which are denoted by blue circles. These outliers are gene-wise estimates that, due to their distinct behavior or variance, are isolated and do not converge towards the fitted values.

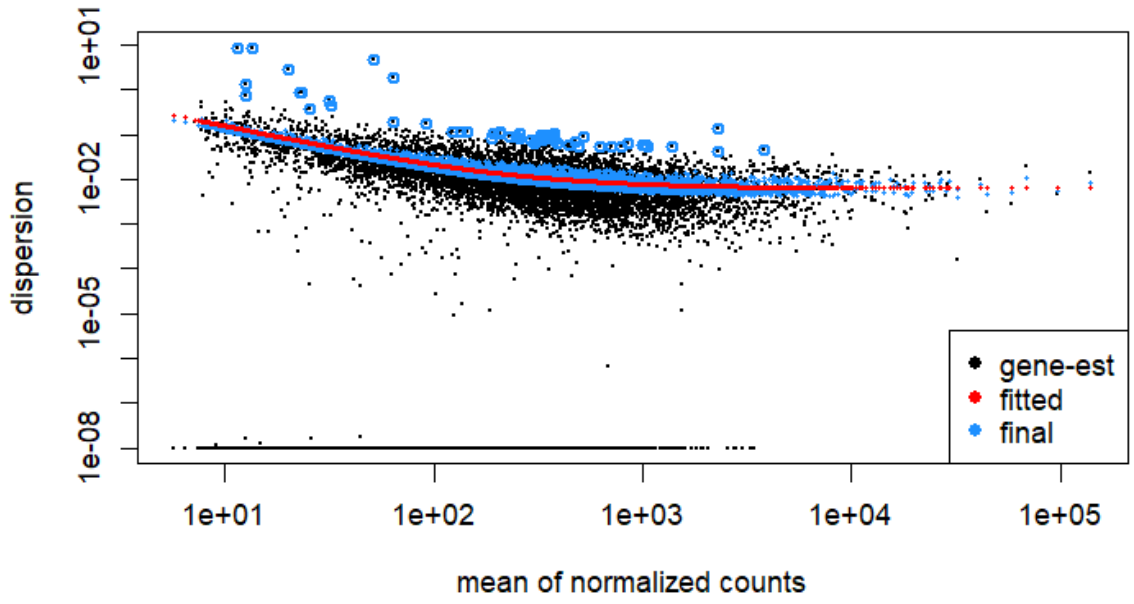

Figure 4 Shrinkage Estimation of Dispersion on Trastuzumab dataset: *This figure shows the results of shrinkage estimation of dispersion for Trastuzumab dataset (Rinas et al., 2020), comprising 6 samples divided into two groups: Control and Herceptin, with 3 samples in each. The plot effectively demonstrates how individual gene-wise estimates of dispersion are adjusted towards the fitted values, highlighting the shrinkage process. This process helps in reducing the noise inherent in individual estimates, thereby enhancing the reliability and accuracy of the dispersion estimation in the dataset.*

Shrinkage for fold-change estimation

Heteroskedasticity in genes refers to the phenomenon where genes with lower expression levels show larger dispersion between groups, making it challenging to obtain accurate estimates of log fold change (LFC) in downstream analysis. To address this issue, **DESeq2** again employs an empirical Bayesian method to shrink LFC estimates towards zero. This is particularly effective for genes that present insufficient information due to reasons such as low counts, high dispersion, or a limited degrees of freedom (Love et al., 2014).

The LFC shrinkage procedure comprises several steps: (1) the MLE for the LFCs is obtained using an ordinary GLM fit (GLM is a linear model with design matrix), then (2) a normal distribution with mean 0 is then fitted to the distribution of MLE, and (3) This fitted distribution from step (2) acts as a prior in second GLM fit, from which the maximum a posteriori (MAP) estimates are derived and retained as the final LFC estimates. Demonstrated by Love et al., (2014), Zhang et al., (2014), the impact of empirical Bayes shrinkage on LFCs can be described as a bias-variance trade-off:

1. Genes with lower expression levels originally displayed pronounced LFCs before shrinkage. While shrinkage pulls the estimates to a more homogeneous distribution proximate to zero and considerably reduces the variance, it introduces a certain degree of bias. This process results in more consistent estimates by reducing the overall mean squared error.
2. For genes that have high expression levels, their estimates are inherently stable. Thus, they maintain both low bias and low variance post-shrinkage.

The (2) step in the above LFC shrinkage method is based on assumption of the MLE of LFC follows a standard normal distribution, which serves as the prior distribution for a second round of GLM fit. This "**normal**" is the default method used in **DESeq2** (Love et al., 2014). However, the standard normal distribution may not be a perfect choice. A later technique, termed "**apeglm**," adopted a heavy-tailed Cauchy distribution for its prior. This choice was designed to reduce the bias while still leveraging the variance-reducing benefits of shrinkage, particularly for genes with lower expression (Zhu et al., 2019). Moreover, the "**ashr**" method was developed for scenarios with a multitude of effect size estimates, each accompanied by its own level of

uncertainty. In the context of RNA-Seq data, it postulates that the LFCs across genes follow a mix of normal distributions with varying variances but centered around zero. The amalgamation of these distributions results in a unimodal distribution with its mode at 0. Consequently, it accounts for genes with varied variances, thereby offering a high degree of adaptability and versatility (Stephens, 2017). Due to the differences between assumptions, the prior scaling of "**apeglm**" and "**ashr**" demonstrate higher sensitivity to extremely large observed LFCs, attributed to their wider tails. In contrast, the "**normal**" method might encounter issues of over-shrinking of largest LFCs given its thinner tails (Love et al., 2014). Figure 5 shows the results of Trastuzumab data (Rinas et al., 2020) with three different LFC shrinkage method, where the exaggerated LFCs for low genes were removed and shrinkage "**normal**" has smaller amounts of genes with significant LFCs.

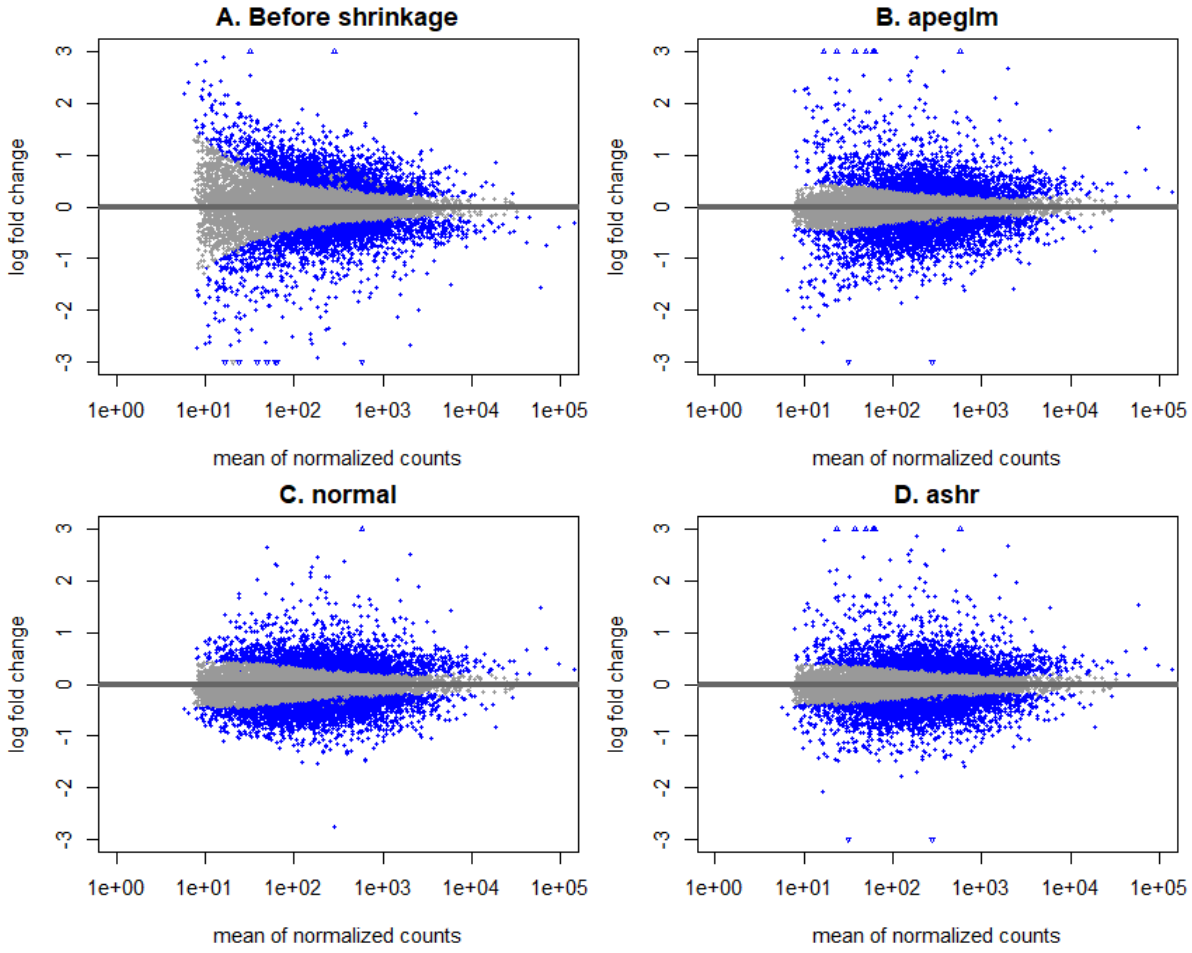

Figure 5 Shrinkage Estimation of LFC: *This scatterplot visualizes the effects of shrinkage on logarithmic fold change (LFC) estimates, using data from six samples – three from the control group and three from the Herceptin group in Trastuzumab dataset (Rinas et al., 2020). The plot is divided into four segments: (A) shows LFC estimates without shrinkage, (B) shrinkage with "apeglm" method, (C) shrinkage with "normal" method and (D) shrinkage with "ashr" method. In these plots, points are colored blue to denote instances where the adjusted p-value is less than 0.1, indicating statistical significance. Additionally, the points fall out of the window are depicted as open triangles, pointing either up or down, depending on whether they are above or below the window's limits. This visualization provides a clear comparative perspective on how different shrinkage methods affect LFC estimation.*

The Wald test

The LFC shrinkage procedure in **DESeq2** yields more reliable posterior estimates for both the LFC and its associated standard error. As discussed in previous section, differentially expressed genes (DEGs) are identified by testing for the significance of individual model coefficients. To accomplish this, **DESeq2** employs the Wald test. In this statistical test, the shrunken LFC estimate is divided by its corresponding standard error to generate a z-statistic. This z-statistic is subsequently compared against a standard normal distribution to determine significance (Love et al., 2014; Zhang et al., 2014). The Wald test p-values are adjusted for multiple testing using the Benjamini and Hochberg procedure, which controls the false discovery rate (Benjamini & Hochberg, 1995).

2.2 Methods based on the normal distribution

Prior to the advent of NGS technologies, microarrays have been used as standard for differential expression analysis for over a decade. **Limma** is one of the well-established methods. One would think adapting such a tried-and-tested method for RNA-Seq data analysis is a good idea. However, the transition of **limma** to the analysis of RNA-Seq data poses several challenges due to the fundamental differences in the data properties. One significant difference is that microarray data is continuous and commonly assumed to follow a log-normal distribution. In contrast, RNA-Seq data is discrete, consisting of non-negative integers, and follows a negative binomial distribution. As a result, many statistical methods developed for microarray data, which based on the normal distribution, are not applicable to read counts obtained from RNA-Seq (Law et al., 2014a; Soneson & Delorenzi, 2013).

Moreover, the methods based on the negative binomial distribution are, from a mathematical standpoint, more complex and less tractable. Compared to the normal distribution, it has less adoptability to different types of data with high or low dispersion heterogeneity, and has limited types of analysis that can be done (Law et al., 2014a). This has inspired scientists to revisit the idea of using normal-based statistical methods to analyze RNA sequencing data. However, the implementation of normal-based statistical methods on RNA sequencing data presents its own unique challenges. Specifically, the read counts from RNA-Seq display unequal variabilities, even after log-transformation (Law et al., 2014a).

2.2.1 **Limma-Voom**

Limma-Voom, introduced by Law et al. (Law et al., 2014a) aims to extend the application of **limma** method to RNA-Seq data. Their approach seeks to estimate the mean-variance relationship in a robust and non-parametric manner, which is then integrated into normal-based statistical method. One such strategy named "**voom**" (variance modeling at the observational level), integrating the mean-variance trend into the estimation of the weights for each individual normalized observation. Consequently, both the normalized log-counts and their corresponding precision weights can be feed into the **limma** analysis pipeline for DE genes detections (Law et al., 2014a).

The other approach is called **limma-trend**, incorporating the mean-variance trend in **limma**'s Empirical Bayes procedure. But **voom** methods outperform it with higher power and lower false positive rate. Thus, in the subsequent chapter, we will focus primarily on the **voom** method and its advantages in differential expression analysis (Law et al., 2014b).

Limma

The **limma** software package is a well-known Bioconductor software package designed to analyze data from microarray sequencing data. The name "**limma**" is an acronym for "linear models for microarray data," indicating that the package contains functionality for fitting linear models to analyze complex experiments with multiple treatment factors. Such setting benefits from making experiments as an integrated whole thus sharing information between samples. The response variable Y_g which measures the relative mRNA expression level of gene g can be formalized as

$$Y_g = \beta_g X + \epsilon_g$$

Where X represents the design matrix that describes the experimental design and treatment effects, while β_g quantifies the effect of the explanatory variable on the response variable. The random noise ϵ_g accounts for unexplained variation in the data (Ritchie et al., 2015).

The linear model possesses the ability to incorporate quantitative weights between observations and the strength borrowed between genes. These quantitative weights can be employed to manage various types of heterogeneity in the data, allowing for variations in data quality and, consequently, enhancing the power of differential gene expression (DGE) analysis. The gene-wise linear model, thus, has a mean $E(Y_g) = \beta_g X$ and variance $var(y_{gj}) = \sigma_g^2 / w_{gj}$, where w_{gj} represents the quantitative weights for gene g in sample j, allowing for variance in data quality. These weights delivered a balance between the gene-wise sample variances and the trended variance across all genes, forming posterior variance estimator for gene g (Ritchie et al., 2015; Smyth, 2004).

Due to its versatility and effectiveness, **limma** has become a widely used tool in differential gene expression analysis for microarray data. Moreover, the integration of **limma** with data transformation **voom** methods provides researchers with a robust and reliable approach for analyzing RNA sequencing data (Ritchie et al., 2015; Smyth, 2004).

Voom

For microarray data, **limma** was based on the assumption that the log-transformed expression values follow a normal distribution. It then employs linear modeling combined with empirical Bayes shrinkage to enhance the stability and power of the statistical tests. However, in context of RNA-Seq data, heteroskedasticity is present in the data. **Voom** and **limma-trend** were developed to address the observed dependence between the mean and variance using different strategies (You et al., 2023). Specifically, the **limma-trend** method modifies **limma**'s empirical Bayes process to incorporate a mean-variance trend. Meanwhile, **voom**, focused on incorporating the mean-variance trend into a precision weight for each individual normalized observation, thus more robust especially with unequal library size. Once the weights are calculated through **voom** method, they will serve as input of **limma** model to build a linear model for differential expression analysis. To estimate sample weights non-parametrically, the "**voom**" approach proceeds through the following steps:

Step one: log-CPM Transformation

Voom starts by transforming the count data to log2-counts per million (log-CPM), making the data more homoscedastic and symmetric than the original count data. The reasons why log-CPM is a better choice than raw counts data will be elaborated in section 3.2.5.

Step two: Fitting Gene-wise Linear Models

For each gene, a linear model, consisting of factors like treatment conditions and batch effects, is fitted to the normalized log-cpm values. Mathematically, the equation is defined as

$$E(y_{gi}) = \mu_{gi} = x_i^T \beta_g$$

Where μ_{gi} is the mean expression of gene g in sample i, x_i represents the matrix of experimental covariates for each sample i and β_g is a vector of regression coefficients by such covariate effects.

Step three: Computing Residuals

Fitting the linear model gives estimations of regression coefficient $\widehat{\beta_g}$. Residuals are then computed by subtraction between the observed log-cpm values and the values predicted by the linear models. These residuals essentially capture the variability in the gene's expression that isn't explained by the factors in the model, thus is crucial for the **voom** method to adjust the data appropriately. Square-root residual standard deviations, denoted as s_g, are used as criteria because of its symmetrically distributed property (Law et al., 2014a).

Step four: Trend Fitting to Residuals

The **voom** method approximates the non-parametric mean-variance trend by deploying a robust Locally Weighted Scatterplot Smoothing (LOWESS) fit, which selectively utilizes data points close to the point of interest. Specifically, when fitting the curve, it employs genes with similar expression levels (Law et al., 2014a). This fitted line illustrates how the standard deviation of these residuals varies as a function of the mean log-cpm value.

Step five: Calculating weights

Based on the trend from step four, the appropriate weights for each observation can be obtained as the inverse variances $w_{gi} = lo(\widehat{\lambda_{gi}})^{-4}$, where $lo(\widehat{\lambda_{gi}})^{-4}$ is the predicted square root standard deviation of y_{gi}. These weights are used to account for the heteroskedastic nature of the data, making the variances more consistent across genes and conditions (Law et al., 2014a).

By implementing this methodology, **voom** successfully accommodates the inherent heteroskedasticity encapsulated in RNA-Seq count data, thereby enabling the application of **limma's** flexible and robust statistical framework to this type of data (Law et al., 2014b). Figure 6 depicts the mean-variance relationship before and after "**voom**" adjustment implanted in our dataset which shows the power of **voom**.

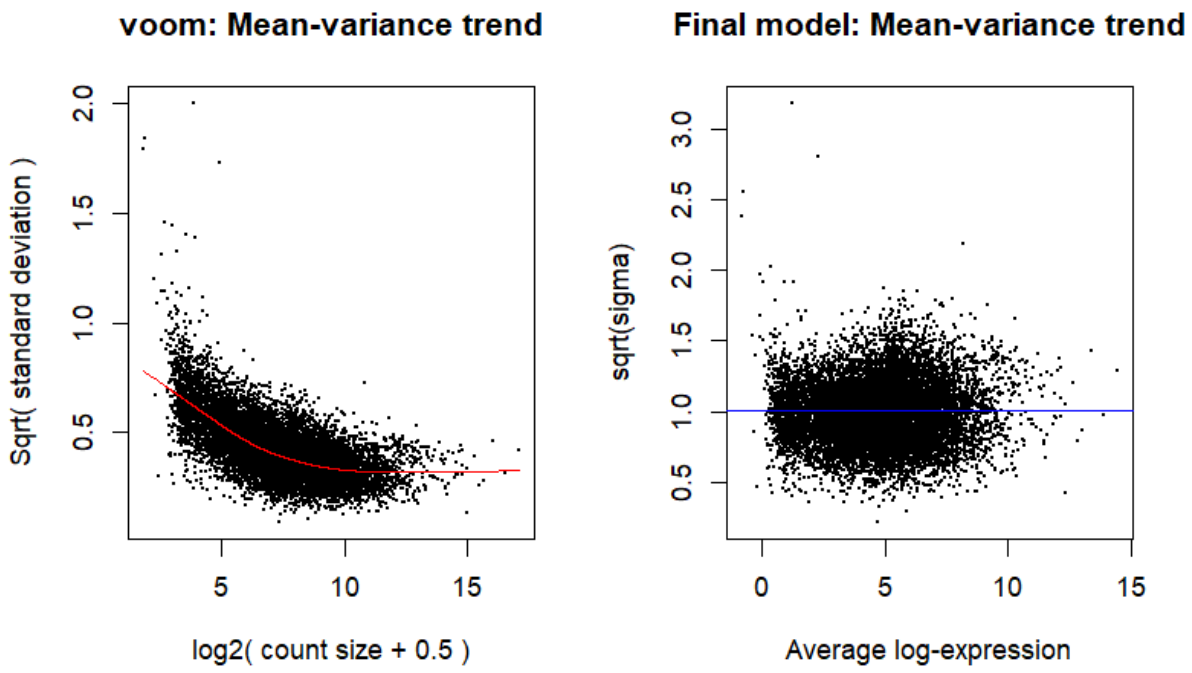

Figure 6 Effect of Voom Transformation: *The figure presents the comparative Analysis of Residual Variance Before and After "Voom" Transformation based on Trastuzumab dataset (Rinas et al., 2020) which consisted with three samples from control group and three samples from Herceptin group. Left plot depicts the relationship between log-CPM (x-axis) and quarter-root of residual variances (y-axis) for each gene prior to "**voom**" application; a red line represents the fitted linear model generated by **voom** model. Right plot shows the data post-"**voom**," plotting log2 residual standard deviations against mean log-CPM values. The elimination of the original trend is evident, with a horizontal blue line indicating the average log2 residual standard deviation.*

Upon the application of the **limma** method, gene-wise variances are adjusted towards the trended variance, effectively mitigating the false positive rate associated with genes of small variance and enhancing the power to detect differentially expressed genes (DEGs) with large variance. The estimated coefficient of the linear model, denoted as $\widehat{\beta_g}$, represents the log-fold change in expression level for each individual gene. The significance of this change can be evaluated using empirical Bayes moderated t-statistics and the associated p-values. Moderated F-statistics is

employed to combine the t-statistics for all contrasts into a comprehensive significance test for each gene (Ritchie et al., 2015). The **limma** package also provides options for adjusting for multiple testing, inclusive of the family-wise error rate and the false discovery rate (Benjamini & Hochberg, 1995), thereby enabling a more accurate and reliable statistical analysis.

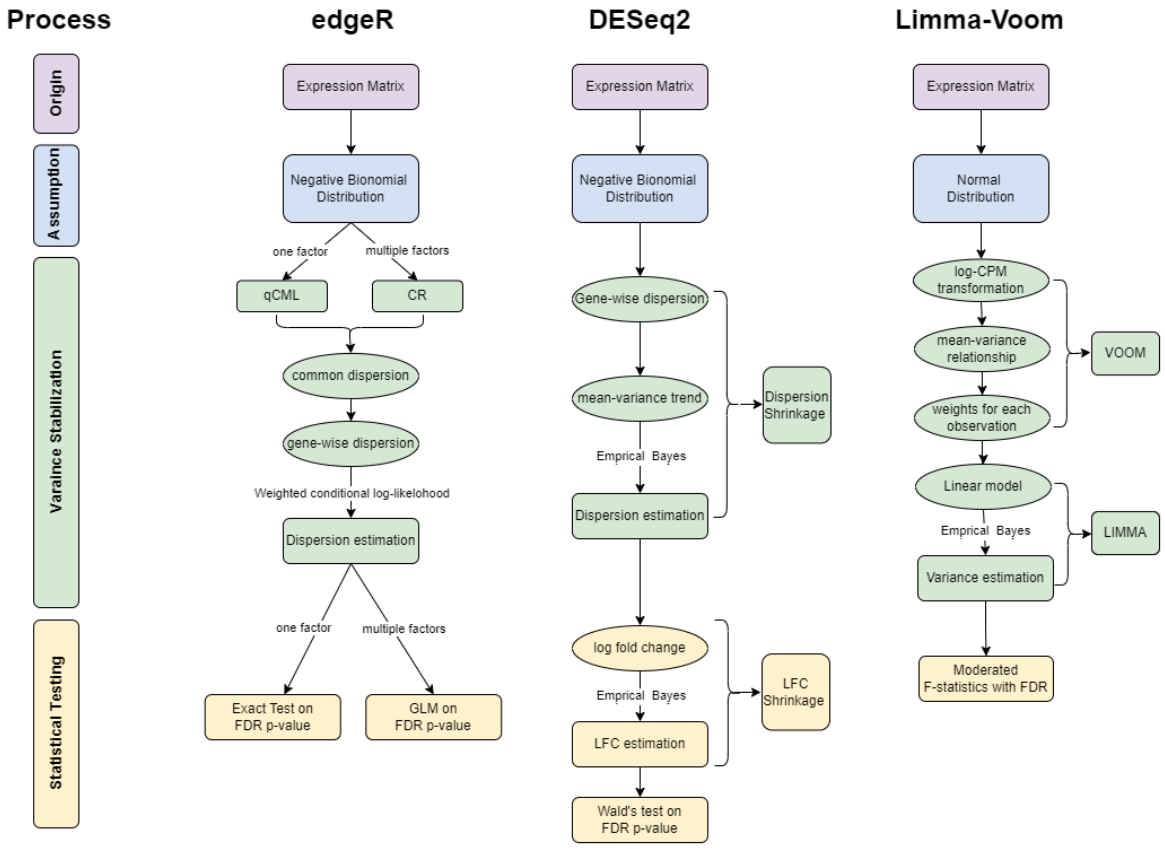

Figure 7 Flowcharts of Common DGE Algorithms: *Flowcharts summarizing the algorithms of previously discussed differential gene expression (DGE) analysis techniques, namely* ***edgeR****,* ***DESeq2****, and* ***Limma-Voom****. These diagrams elucidate both the specialized tools and statistical concepts deployed at each step, from the initial input of the expression matrix through to the final statistical testing phase.*

Chapter 3

Methodology

The previous chapter provided a comprehensive overview of established methods in differential gene expression analysis. This chapter will elaborate the details about development of the **bootstrapping** method, which includes the explanations on statistics, experimental designs, and resultant outcomes. The **bootstrapping** method is designed to address the core challenges common to differential gene expression analysis, with a particular emphasis on enhancing analytical power in the context of limited replications in experimental design. As the term "**bootstrapping**" suggests, the statistical technique leverages the advantages of bootstrap procedures to improve the precision of differential gene analysis. This method is encapsulated within scripts written in the R programming language. It's noteworthy that the presented **bootstrapping** method is self-contained, without the reliance on external functions, greatly reducing dependencies on other bioinformatics-centric R packages.

3.1 Outline of new proposed method

The proposed resampling method is structured into three primary segments: ***Input***, ***Processing*** and ***Output***.

- ***Input*** acts as the first component of differential gene expression analysis, including reading in, organizing and pre-processing of the raw counts data generated from the *quantification* steps within the sequencing pipeline.

- ***Processing*** is the core component of the proposed method, involving the application of kernel functions to achieve variance stabilization.
- ***Output*** is the concluding stage which identifies the differentially expressed genes.

3.2 Input section

The ***Input*** component consists of ***filtering raw data, normalization, and log-transformation***.

3.2.1 **Input of raw data**

The ***quantification*** step, preceding our analysis in the sequencing pipeline, produces a file storing the raw count read information. This file is typically organized as an expression matrix which effectively captures the sample-level details outlined in the experimental design. Within the matrix, each row represents an expression feature (either a gene or transcript), while each column pertains to a distinct sample. The alignment of sample-level information—derived from the experimental design—to the respective columns of this expression matrix is important in subsequent DGE analysis. They contain the information to distinguish the variability between the samples.

When assessing variability within the dataset, it is expected that the intergroup variability – depicting the differences between experimental conditions and control conditions, is greater than the intragroup variability which represents the technical and biological variability:

- ***Biological Variability***: Refers to the inherent differences observed among genes. In other words, certain genes naturally exhibit more variability than others

- ***Technical Variability***: Refers to the variations introduced during the process of experimentation, data collection, or analysis which are not due to the underlying biology. Although these technical variations often exhibit a subtler magnitude compared to biological effects, they can not be neglected and need to be addressed (Law et al., 2018b; McIntyre et al., 2011).

A comprehensive view of the data aids in discerning variations among replicates and in determining if predefined experimental groups exhibit discernible differences. Visualization techniques like MDS, further detailed in section 3.2.2, is useful tool for depicting such variations (Koch et al., 2018).

Trastuzumab test data

The sequencing data in this study was sourced from experiments conducted in the McConkey Lab, contrasting the effects of the antibody-based drug Trastuzumab (brand name Herceptin®) with a biosimilar version which is anticipated to produce similar cellular impacts.

The HER2 receptor, short for human epidermal growth factor receptor 2, is a transmembrane tyrosine kinase (also called erbB-2) that is a major classifier of invasive breast cancer. The breast cancer involving HER2 receptor is an aggressive subtype, accounting for approximately 20% of all invasive breast cancers and target of therapy for the disease. Before the introduction of HER2-targeted therapies, patients with this cancer often treated by chemotherapy and faced higher relapse and mortality rates (Ross et al., 2009; Wilson et al., 2018). In 1998, Herceptin® was introduced as a groundbreaking treatment for HER2-positive breast cancer (HER2+ BC), targeting the HER2 receptor. It functions as a monoclonal antibody, targeting and binding to the extracellular domain IV of HER2. This binding suppresses cell signaling pathways vital for cell

growth and survival. In parallel, Herceptin flags these tumor cells for elimination by the immune system (Wilson et al., 2018).

Recently, in order to obtain drugs with similar biological function, the prospective biosimilar products were proposed. They shared the same protein sequence as Herceptin, but with variations in the distribution of post-translational modifications (PTMs) primarily as a result of minor variations in growth conditions. PTMs are typically covalent adjustments such as glycosylation made to proteins after mRNA translation. PTMs play a pivotal role in modulating a protein's attributes, including its physical or chemical properties, stability, and localization. These distinct PTMs effectively differentiate the biosimilar product from the original Herceptin drug, although the core amino acid sequence remains the same. (Marín-García, 2014; R. Farley & J. Link, 2009)

To study the contrasting effects of Herceptin and its biosimilar counterpart, the expression matrix we focused on contains three groups of sequencing data:

1. Control group (denoted as CONT in subsequent analyses)
2. Drug treatment with Herceptin (referred to as HER)
3. Drug treatment with one of the proposed biosimilar products (labeled as BIOS)

The dataset will be referred to as the *Trastuzumab dataset* in subsequent sections. It contains the counts data from 11910 distinct genes and 9 to 10 replicates for each distinct treatment. We aim to perform differential expression analysis among these groups, identify potential variances in gene expression between the original drug and its biosimilar counterpart.

3.2.2 **Unsupervised clustering**

The multi-dimensional scaling (MDS) plot serves as an exploratory tool examining the overall differences between samples in an unsupervised manner before conducting a formal differential expression analysis. Through MDS, RNA samples can be represented in a two-dimensional space where the distance between samples represents the “leading fold change”. This is defined as the root-mean-square average of the log-fold-changes for the most discriminating genes between each sample pair (Chen et al., 2016; Ritchie et al., 2015).

Ideally, the technical replicates shown in MDS should lie close to one another, indicating minimal variation among them. While all samples cluster by groups, the largest transcriptional difference is observed between groups CONT and HER, and CONT and BIOS over dimension 1. Given this observation, it’s expected that pairwise comparisons between these cell groups will yield a higher number of DE genes for contrasts involving CONT samples. Conversely, fewer DE genes are expected when comparing BIOS to HER samples.

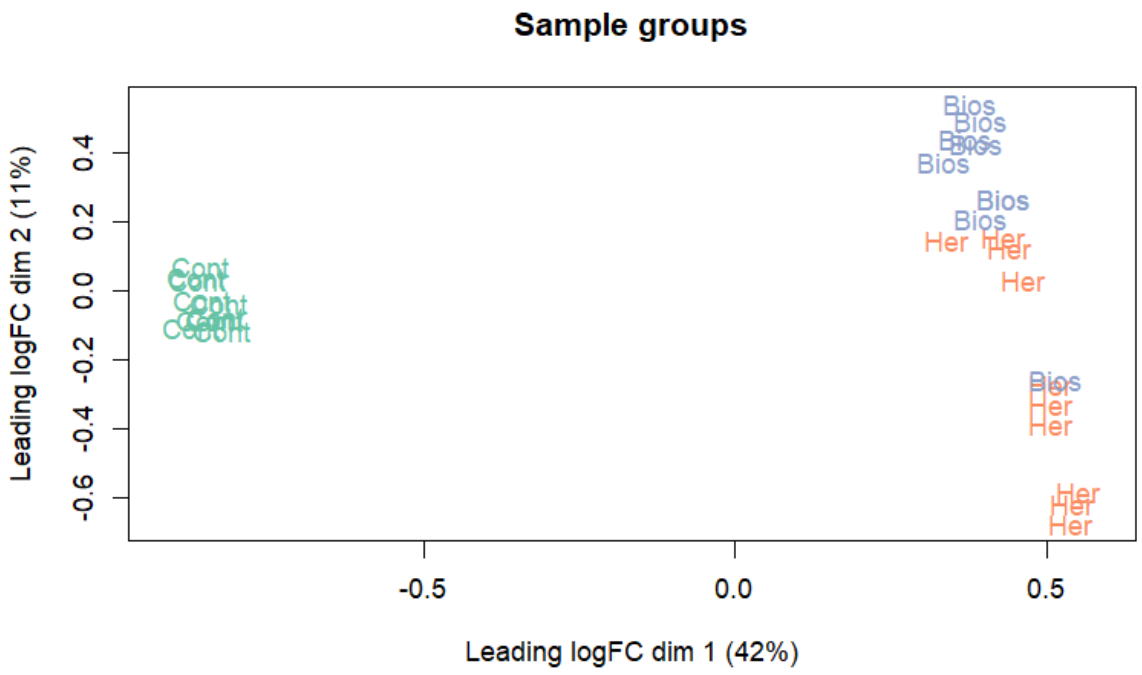

Figure 8 MDS Analysis on Trastuzumab dataset: *This figure displays a raw multi-dimensional scaling (MDS) plot of log-CPM values from high-throughput sequencing data. The plot is generated using the `plotMDS` function in* ***Limma*** *package and visualizes the results of Trastuzumab dataset (Rinas et al., 2020***)***. Each group is uniquely color-coded for clear differentiation. Distances on the plot correspond to the leading log2-fold changes between the samples, providing a visual aid to the relative similarities and differences in gene expression profiles across the groups.*

3.2.3 **Filtering raw data**

Over the past twenty years, RNA sequencing technology has developed significantly. Yet, separating sample noise from biological differences in expression is not feasible for low-expression genes, compromising the sensitivity of differential gene expression (DEG) detection. Implementing filtering steps can greatly enhance the power of the analysis by limiting the impact of noise at low expression levels.

From both biological and statistical perspectives, genes with consistently low expression are of less interest. Biologically, a gene must pass a threshold of expression so that it can be translated into protein and considered as biologically relevant. Statistically, the genes with consistently low expression levels across different groups are unlikely to be identified as significantly differentially expressed since low count data has relatively high noise in addition to similar expression levels across samples. Together these factors result in limited statistical evidence for reliable analysis (Chen et al., 2016). In addition, filtering low-expression genes provides several advantages for downstream analyses. First, it enhances the reliability of the mean-variance relationship estimations within the data. Second, it reduces the total number of hypothesis tests required in statistical testing. This dual benefit effectively enriches the set of true DEGs while decreasing the overall number of hypothesis tests. As a result, the procedure for correcting for multiple hypothesis testing becomes less stringent, subsequently boosting the statistical power of the analysis (Bourgon, 2010; Law et al., 2018b).

While effective filtering techniques offer clear advantages, the validity of filtering methods remains a subject of debate. Using inappropriate filtering can create pitfalls, such as poor control of type I error (Sha et al., 2015). Over the past couple decades, various filtering methods have been proposed, each relying on statistics calculated from gene expression levels but based on different criteria. The selection of an optimal filtering threshold involves two critical considerations: 1) choosing the most suitable filtering criteria, and 2) selecting the optimal threshold value (Chen et al., 2016).

While variance-based filters are frequently used in microarray data analyses, their application to RNA-Seq data remains limited. The typically limited number of replicates in RNA-Seq

experiments poses challenges for accurate per-gene variance estimation. Consequently, filtering based on expression levels is more prevalent in high-throughput datasets.

Raw counts are seldom used directly due to possible sequencing depth bias and counts per million (CPM) has been advanced as a solution to this problem. It measures how many counts would be obtained for a gene if the sample has a library size of 1 million, thus providing normalized data that accounts for variations in sequencing depth across samples (Chen et al., 2016; Sha et al., 2015). The CPM value for gene g in sample i is calculated as:

$$CPM_{gi} = \frac{\text{Number of reads mapped to gene } g \text{ in sample } i \times 10^6}{\text{Total number of mapped reads in sample } i}$$

The choice of threshold statistics and value are worthwhile to discuss as well. A threshold for gene expression that has been used previously in (Chen et al., 2016) is if a gene has a count of at least 10 in some libraries. In our dataset, with the library size ranges from 3 to 9 million, the CPM cutoff value roughly equates to $10/\mathrm{L}$, or 10/5 = 2, with the median library size (L) of 5 million. Precision in the cutoff value is not crucial, as downstream differential expression analyses are generally not sensitive to minor variations in this parameter (Chen et al., 2016).

We are interested in filtering out the genes that have low expression levels across all libraries. It's intuitive that the *minimum (min)* can be considered as filtering statistic criteria. Genes with minimum counts less than or equal to a pre-specified threshold, are filtered from the subsequent analysis. However, this strict application of the minimum criterion may inadvertently exclude genes that are differentially expressed only under specific conditions (Law et al., 2018b; Sha et al., 2015). To mitigate this issue, we adapted from the *filterByExpr* function in **edgeR**, which only keeps genes that have a CPM value exceeding the cutoff in at minimum of n samples (n is the

number of replicates). In Trastuzumab dataset (Rinas et al., 2020), we kept genes where a minimum of 10 samples exhibited a CPM value of 2 or higher. This approach ensures that all the genes will be included if they expressed in at least one complete set of replicates for any of the three groups under study (Chen et al., 2016; Sha et al., 2015). After filtering, $\frac{503}{11910} = 4.22\%$ of genes were removed from the dataset. Figure 9 illustrates the plot of the log-CPM distribution, measuring the proportion of genes within each sample before and after filtering method. It's noticeable part of genes below CPM threshold (dash line) were filtered out.

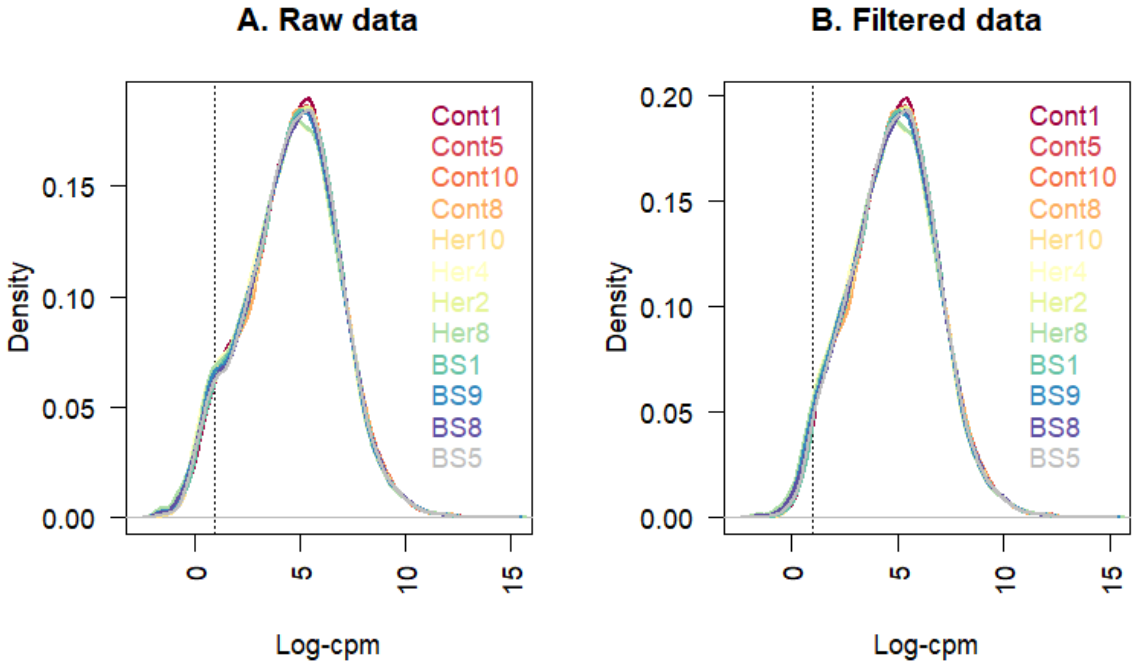

Figure 9 Density Plots Comparing log-CPM Data Before and After Filtering: *This figure presents density plots of log2-counts per million (log-CPM) values from Trastuzumab dataset (Rinas et al., 2020), showcasing both raw (A) and filtered (B) data. In this data, four randomly selected samples from each group were included. The black dotted vertical line marks the log-CPM threshold (approximately 1) set in the data filtering step. The comparison between two density plots illustrates the impact of filtering process on the dataset's overall expression profile.*

3.2.4 **Normalization**

During the sample preparation and sequencing stages, various non-biological factors, such as experimental conditions, can impact gene expression levels. These influences, while not of primary interest, can introduce biases into the final differential gene expression (DGE) analysis. The normalization method is implanted in the data processing workflow to mitigate these ***technical variabilities*** (Law et al., 2018b).

Most normalization methods primarily focus on addressing the technical variabilities arisen from sequencing depth differences. They scale the read counts in each library to a common value across all libraries in the experiment, thereby enabling comparisons across samples (Robinson et al., 2010). However, these methods often overlook the impacts of RNA composition: the presence of highly expressed genes in one experimental condition can appear as a broad decrease in expression of other genes, thereby potentially increasing false positives and reducing detection power (Robinson & Oshlack, 2010). To counteract these challenges, the Trimmed Mean of M-values (TMM) method becomes a preferred method for normalization, which is specifically designed to mitigate the influence of library composition by scaling samples based on the mean expression of a subset of non-differentially expressed (DE) genes. The term "trimmed Mean" signifies that the genes selected for determining the scaling factor undergo a process of double trimming, ensuring their categorization as non-DE genes. Specifically, the double trimming process excludes genes with extreme log-fold changes M_{gk}^{r} (sample k relative to sample r for gene g) where:

$$M_{gk}^{r} = \log_2 \frac{Y_{gk}/N_k}{Y_{gr}/N_r}$$

and the genes with extreme absolute expression levels, given by:

$$A_{gk}^{r} = \frac{1}{2}\log_2\left(\frac{Y_{gk}}{N_k} \times \frac{Y_{gr}}{N_r}\right)$$

where Y_{gk} represents the observed count for gene g in library k and N_k is the corresponding library size. By default, the system trims genes with either 30% of the extreme M values or 5% of the extreme A values before computing a weighted average, but the criteria can be adjusted based on given experimental conditions. The remaining genes after trimming step are considered as non-DE, and subsequently used to determine the scaling factor. Specifically, the normalization factor for sample k using reference sample r is calculated as:

$$\log_2 TMM_k = \frac{\sum_{g \in G} w_{gk}^{r} M_{gk}^{r}}{\sum_{g \in G} w_{gk}^{r}}$$

Where

$$w_{gk}^{r} = \frac{N_k - Y_{gk}}{N_k Y_{gk}} + \frac{N_k - Y_{gr}}{N_r Y_{gr}}$$

In here, G is subset of non-DE genes. Due to its design, TMM is a robust method against outliers, providing reliable results in subsequent differential expression analysis (Robinson et al., 2010). The normalization can be executed by *calcNormFactors* function in *edgeR* package. However, we have replicated and implemented the same functionality within our own code. This means that users of our method do not need to install the additional *edgeR* package, as the necessary normalization functionality is already integrated.

A boxplot is useful in determining the consistency among samples expression distribution. Figure 10 presents the boxplot of log-CPM values both before and after the application of TMM

normalization. The dataset utilized is the Trastuzumab dataset (Rinas et al., 2020) with 8 samples in each group. These 8 samples were randomly selected from the total of 9-10 samples in each group to ensure a balanced dataset for testing. The left panel of the figure shows the data prior to normalization and the right panel displays the distribution after normalization. The comparison between two graphs provides insights into the effectiveness of the TMM method in harmonizing the datasets.

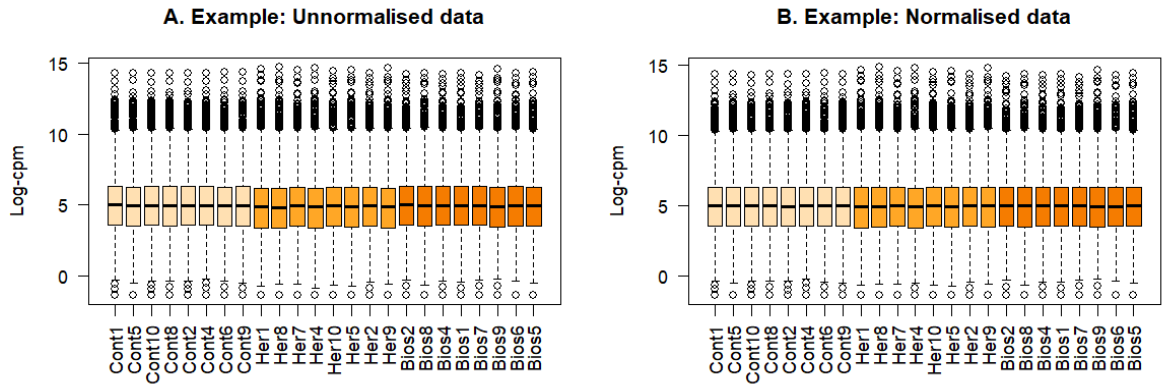

Figure 10 Boxplots of Normalized vs. Unnormalized Data Distribution in Trastuzumab Dataset: *This figure shows the normalization results of log2-counts per million (log-CPM) values from Trastuzumab dataset (Rinas et al., 2020), showcasing both unnormalized data (A) and normalized data (B) data. In this process, eight randomly selected samples from each group within the dataset were included. The groups are ordered from left to right as follows: the Control group, the Herceptin group, and the Biosimilar group. This visualization aims to demonstrate the impact of normalization on the dataset.*

To better illustrate the efficacy of TMM normalization, we deliberately manipulated the counts in the dataset: counts were halved for the first sample (Cont1), and an additional sample (Cont5) had counts increased fivefold. The impact of these adjustments, both before and after applying

TMM normalization, is depicted in Figure 11. The figure effectively provides an example of the robustness of the TMM method in normalization.

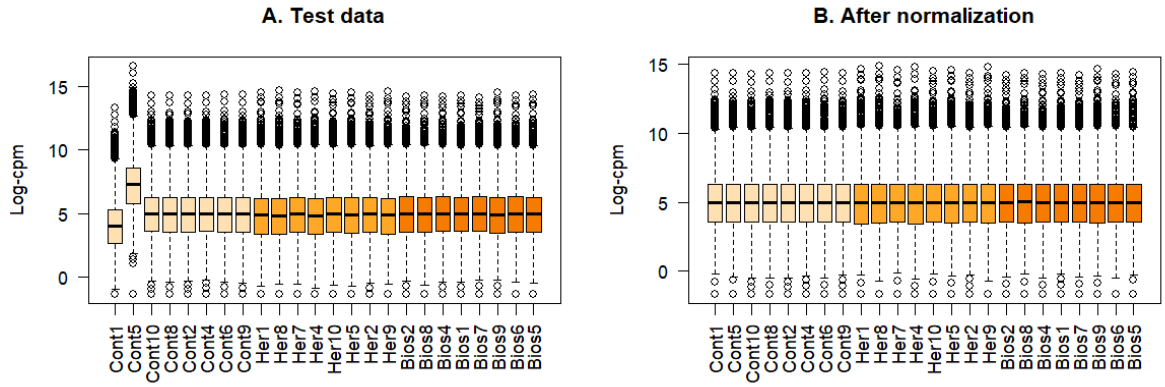

Figure 11 Boxplots of Normalized vs. Unnormalized Data Distribution in Test Dataset: *This figure displays the normalization results for log2-counts per million (log-CPM) values from sample dataset in Figure 10, but with specific manipulations were applied. In this altered dataset, 'Cont1' was adjusted to half of its original value, and 'Cont5' was increased to five times its original size. The figure shows the comparsion between test data (A) and normalized data (B) data. With such manipulation, it's able to provid a clearer demonstration of the normalization process's impact on the dataset.*

3.2.5 **Log-CPM transformation**

In RNA-Seq data analysis, log-CPM values are often preferred over raw counts. The use of CPM allows for comparisons between libraries with varying sequencing depths, while the logarithmic transformation helps mitigate the heteroskedasticity inherent in sequencing data, thus enhancing the DGE analysis. Notably, most established DGE analysis tools predominantly work with transformed log-CPM values instead of raw count data (see details in Chapter 2).

Counts Per Million

The number of reads observed for a specific gene is proportional not only to the gene's expression level but also to its transcript length and the sequencing depth of the library. To enable the comparison across libraries of different sizes, counts per million (CPM) is generated by dividing each read count by the corresponding library size (in millions) (Law et al., 2014b). Another common metric is RPKM (reads per kilobase per million), which accounts for gene length. However, given the interest of study lies in the relative changes in expression between conditions rather than in absolute expression levels, CPM becomes a more suitable and convenient choice for subsequent analysis, particularly as it doesn't require gene length data (Law et al., 2014a).

Log-transformation

After the normalization of the gene data, log-transformation is performed on the count data. In biological research, the log-transformation is widely used to deal with skewed data and it offers several benefits. (Feng et al., 2014). One benefit is that once log-transformed values are used, differences will be based on proportional change rather than additive changes, which is typically seen as more biologically relevant. The differences between log-cpm values can be interpreted as the log-fold-change between samples. Log-fold-change, i.e. log2-fold-change, describing the ratio of values in two conditions A and B, calculated as

$$\text{logFC} = \log_2\left(\frac{A}{B}\right) = \log_2(A) - \log_2(B)$$

Where $\log_2(A)$ is calculated mean of log2 values of samples in condition A, equivalent to taking geometric mean of the original data. Thus, log fold change can be expressed as

$$\text{logFC} = mean(\log_2(x_A)) - mean(\log_2(x_B))$$

where x_i represents raw counts in condition i. The use of logarithm base 2 is preferred for its ease of interpretation; a doubling or halving of expression levels, often considered biologically significant, is quantified as a unit change of +1 or -1 on the log2 scale, which is more intuitive than the +/- 0.3 units on a log10 basis.

Moreover, the primary advantage of certain transformations in RNA-Seq data analysis lies in mitigating the inherent heteroskedastic nature of the sequencing data. Heteroskedasticity, derived from the Greek words “hetero” ('different') and “skedasis” (‘dispersion'), represents a situation where variability is not consistent across different values of a dataset. This stands in contrast to homoskedastic distributions where dispersion remains uniform. In RNA-Seq datasets, the patterns of heteroskedasticity are quite evident. High mean expression genes tend to show variances exceeding their mean. In contrast, genes with lower expression have reduced variance, but they demonstrate considerable variability. One potential reason for this is that genes expressed at a low level, often close to sequencing background noise, are more influenced by technical variability. This characteristic behavior often gives rise to a funnel-shaped distribution when one visualizes variance versus mean expression in scatter plots (shown in Figure 12A). Interestingly, as expression levels rise, the influence of technical variability wanes but biological variability doesn’t follow the same declining trend. In fact, for genes with high counts, biological variability may be the main source of uncertainty. This assertion becomes clearer when plotting the trend of squared coefficient of variation (CV^2) against count values. Exemplified in the negative binomial model used in **edgeR** and **DESeq2**, the trend can be described by the equation: $CV^2 = \frac{1}{\lambda} + \phi$ where λ is the mean and ϕ is the dispersion parameter. Specifically, the CV

decreases for small to moderate counts and asymptotically approaches a constant value for higher counts. The value ϕ is associated primarily with the biological variability (Law et al., 2014b).

The linearization calculations have shown that the variance of log-cpm values aligns closely with the squared CV of the counts (Law et al., 2014b). Consequently, log-cpm data showed the aforementioned trend, presenting an asymptotic trend with genes symmetrically distributed on both sides (Figure 12B). Visually, the transformation appears to mitigate the heteroskedasticity, thereby enhancing the power to detect DEGs.

To avoid taking logarithm of zero, a small positive value (usually 0.5) is added to counts, i.e., $\log(\mathrm{CPM}+\epsilon)$ when implementing the log-transformation. Suppose our experiments generate a set of n RNA samples and each RNA sample has been sequenced. The generated RNA-Seq data is then consisted of a matrix of read counts r^{gi}, where sample $i = 1, \ldots, n$, and genes $g = 1, \ldots, G$. For each sample, R_i is the total number of mapped reads for sample i.

$$R_i = \sum_{g=1}^{G} r^{gi}$$

Then the log-counts per million values for each count as

$$Y_{gi} = \log_2\left(\frac{r_{gi} + 0.5}{R_i + 1.0} \times 10^6\right)$$

The counts are offset from zero by 0.5 to avoid taking the log of zero, and the library size is offset by 1.0 is to ensure that the fraction inside log2 transformation is strictly less than 1 and greater than zero (Law et al., 2014a).

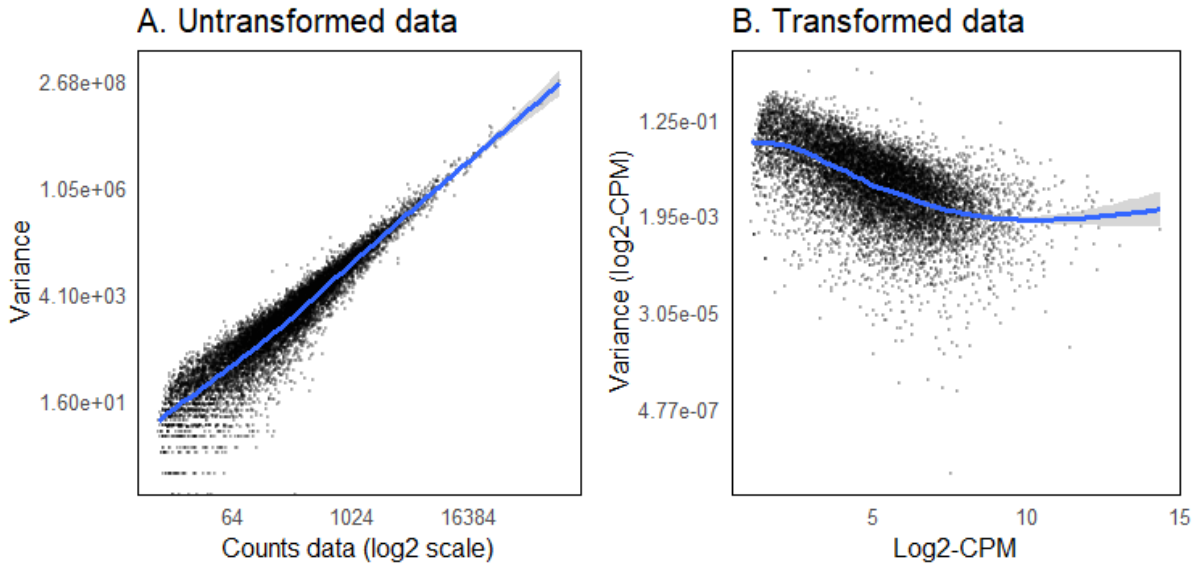

Figure 12 Mean-Variance Distribution Before and After Log-Transformation: *This scatterplot presents the effects of log-transformation on Trastuzumab dataset (Rinas et al., 2020) pre and post log-transformation, which includes three groups from the Control group. The limited sample selection mirrors the challenge we aim to address: constraints in sample size. The figure comprises two parts: (A) Scatterplot showing the variance plotted against the mean value of raw counts data, before log-transformation. The axes are displayed in log2 scale for clearer comparison. (B) Post log-transformation scatterplot with the variance of log-CPM plotted against the mean value of log2-CPM. The blue lines represent the fitted trend using a generalized additive model (GAM) with shaded area indicating the confidence interval. The comparative between these scatterplots demonstrates that the data distribution exhibits more homoskedasticity (i.e., more uniform variance) following the log-transformation. This transformation is particularly effective in stabilizing the variance across different levels of gene expression.*

3.3 Processing section

The "processing" section is pivotal in differential gene expression analysis, particularly when identifying differentially expressed genes between "control" and "treated" groups. It starts with sample observations for "treated" groups $(T_1, T_2, \cdots, T_n)$ and for the "control" group $(C_1, C_2, \cdots, C_n)$, assuming equal sample sizes (n) for both groups. Traditional approaches, begin

by calculating means (μ_T and μ_C) and estimating variances (σ_T^2 and σ_C^2) by applying variance stabilization techniques to the observed data for both groups. Subsequently, t-test statistics ($t = \frac{Z}{s}$) are computed, where Z represents the difference between μ_T and μ_C, and s is the standard error of the difference between means ($s = \sqrt{\frac{\sigma_T^2+\sigma_C^2}{n}}$). The obtained t-statistics are compared against the t-distribution with degrees of freedom of $2n - 2$, and p-values, along with False Discovery Rate (FDR) statistics, are used to determine significance.

In our methodology, significant changes have been made in two key areas. Firstly, instead of relying on the assumption that count data follows a specific distribution, we use a smoothed bootstrap procedure to estimate the mean and variance for control and treated groups directly. Generally, the mean μ_T and μ_C are unbiased estimation, but the variance estimate requires stabilization. Our method diverges from traditional Empirical Bayesian approaches and instead utilizes a smoothed resampling technique. This method involves combining residuals from both sample-specific and pooled distributions for each gene, utilizing weights determined through a simulation study. With the chosen weights, a resampled dataset with 100 or more resampled data points in each "treatment" and "control" group is generated, effectively defining the expected distribution. The resampled data points are then considered as representative populations of "treatment" and "control" groups.

The second improvement lies on the resampling method for statistical analysis. Instead of directly applying a t-test on the augmented dataset, our method involves selecting n (the initial sample size) samples from each population and calculating the differences in means. By repeatedly performing this selection and calculation, we form a distribution of these mean

differences. Consequently, the average of this generated distribution closely approximates the original calculation $Z = \mu_T - \mu_C$, while the standard error of this distribution serves as an estimate for s in t-statistics. Subsequently, t-statistics are computed as $t = \frac{Z}{s}$ and compared to the t-distribution with degrees of freedom of $2n - 1$. This methodology enhances the robustness of the statistical analysis by leveraging resampling techniques to address limitations associated with small sample sizes.

Throughout this procedure, it is crucial to emphasize the core of our approach relies on the accurate estimations of mean and variance. Given the limited replicates in typical RNA-Seq datasets, it's improbable that these sample data aptly represent population data. While the mean estimations are reasonably unbiased, the variance estimations of individual samples have considerable noise and necessitate adjustments. The subsequent sections employ simulation studies to shed light on the estimation accuracy. Unlike real-world data, which is restricted to observed sample data without access to expected outcomes, simulation studies have the advantage of known true population parameters. This distinction allows simulation studies to offer valuable insights into real-world datasets. By optimizing weighting parameter values, simulation studies can provide insight on real-world datasets as they can optimize weighting parameter values to effectively minimize the difference between a 'true' or known population variance and the observed variance of a dataset with a low number of replicates.

3.3.1 **Bootstrapping**

In this section a standard methodology for bootstrap analyses is described, and aspects of this analysis are incorporated into our developed resampling procedure. The resampling method was

initially introduced because, in the estimation of certain parameters in a statistical model, obtaining a single set of estimates is insufficient. There is a need for an assessment of the uncertainty associated with these estimates (Ruppert & Matteson, 2015). The resampling method achieves this by facilitating the calculation of standard errors, construction of confidence intervals, and execution of hypothesis tests for a variety of sample statistics, such as the mean or variance. Specifically, we employ ***bootstrapping*** as a form of resampling technique and introduce adaptations and modifications to refine and customize this method to better suit the specific requirements of our study.

The choice of bootstrapping lies on its non-parametric resampling properties, often referred to as model-free resampling. As implied by its name, model-free resampling involves estimating the parameter of interest, θ, solely based on the original dataset without making any assumptions about data distributions. The bootstrapping method operates directly on the dataset, by repeatedly drawing samples (with replacement) from our initial data pool and subsequently calculating statistics on these resampled datasets, it's able to estimate the distribution of statistics from that very data, allowing the data to metaphorically "pull itself up by its own bootstrap" (Efron, 1979).

Within the bootstrapping process, we took a further step to mitigate bias arising from limited sample size. This involved an innovative approach where we incorporated residuals from genes exhibiting similar expression levels when estimating the variance for individual genes. Our innovation is grounded in the expectation that genes with comparable expression levels are likely to exhibit similar variances. A detailed exploration of this procedure will be provided in section 3.3.5.

First, we want to talk about the general bootstrap procedure, which encompasses the following stages:

1. ***Origin***: suppose the gene g's expression level following the unknown population P_g. The bootstrapping begins with a sample of size n drawn from population $X_{g,1}, X_{g,2}, \dots, X_{g,n} \sim P_g$. In our context, n represents the number of replicates for a gene g under a given condition. Let $\widehat{\theta_g}$ be the raw estimate derived from the initial sample.

2. ***Bootstrap Sampling***: Draw a sample of size m with replacement from the initial sample data, referred as "bootstrap Sample", denoted as $X^*_{g,1}, X^*_{g,2}, \dots, X^*_{g,m}$. Here, m is the resampling sample size. The desired statistics (such as mean, variance) are computed from the resampled data, i.e. $\widehat{\theta^*_g} = g\left(X^*_{g,1}, X^*_{g,2}, \dots, X^*_{g,m}\right)$

3. ***Iteration***: Repeat the bootstrap sampling step B times to produce B sets of bootstrap samples. Calculate the desired statistics for each set of resampled data, yielding B estimates $\widehat{\theta^*_{g,1}}, \dots, \widehat{\theta^*_{g,B}}$.

4. ***Distribution Formation***: The computed B estimates $\widehat{\theta^*_{g,1}}, \dots, \widehat{\theta^*_{g,B}}$ form a "bootstrap distribution" of the targeted statistic. This distribution is presumed to be the best estimate of the population's parameter θ_g, thus providing insight into the statistic's variability.

5. ***Inference***: With a sufficient number of repeats, the bootstrap distribution can be utilized for inferential statistics. This involves deriving confidence intervals or probabilistic assertions about the population parameter θ_g. A possible use of bootstrap statistics is to

construct a 95% confidence interval as $\widehat{\theta_g} \pm Z_{\frac{\alpha}{2}} \times s_{boot}(\widehat{\theta_g})$ where $s_{boot}(\widehat{\theta_g})$ is the estimated standard error given by

$$s_{boot}(\widehat{\theta_g}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\theta^*_{g,b}} - \widehat{\overline{\theta^*_g}}\right)^2}$$

The above equation assumes a quasi-symmetrical distribution. For non-symmetrical distributions bootstrap estimates may be calculated separately for the upper and lower confidence limits, and symmetry is not a required property for the underlying distribution.

Understanding the intricacies of the bootstrapping method becomes more intuitive when one recognizes that, within the context of bootstrapping, the original sample effectively represents the broader population. Thus, bootstrap samples drawn from the initial sample can be seen as if they were directly sampled from this population. Some choices made during the above procedures are:

Bootstrap sample size:

Building upon the notion that our original sample adequately mirrors the broader population, it's imperative that each bootstrap sample maintains the same sample size, as the original $(m = n)$. This ensures bootstrap sampling closely resembles the original sampling process, striving for consistency and accuracy in the resampling method (Ruppert & Matteson, 2015).

Sample with replacement:

Sampling "with replacement" is a choice rooted in the desire for independent observations. Such an approach ensures that the resamples are independent and identically distributed (i.i.d.) just like

the original sample. If we opted for sampling without replacement, a set of *n* resamples from the original dataset would duplicate the original dataset, thereby eliminating variability between repeated resampling data (Ruppert & Matteson, 2015).

The number of resamples (B):

In most cases where the initial sample size exceeds 10, choosing an appropriate value for B becomes a critical consideration. B, which can extend into thousands or more depending on the application, is selected because resampling for a sample size of n entails a vast number of potential combinations—specifically, n^n combinations when picking n items from a set of n distinct items. Considering each unique set yields a distinct value of $\widehat{\theta_g^*}$, an exhaustive exploration often isn't feasible. Instead, a large number (B) of "bootstrap samples" is employed to approximate the parameter distribution. For standard applications aiming to construct a 95% two-tailed confidence interval, it is recommended to have 25 points or more below or above the given interval for a reliable approximation of the parameter distribution. Consequently, typical recommendations for the resample count B range from 1000 to 5000, striking a balance between computational efficiency and achieving a sufficiently accurate representation of the parameter distribution.

However, in our experiment with a very limited sample size n (usually smaller than 8), the potential combinations are inherently limited. Additionally, our focus isn't on estimating the confidence interval but rather on the parameters themselves. As a result, there isn't a need for an extensive number of resamples. We found that resampling 100 times suffices for our purposes, while in the meantime significantly reduced the computational complexity.

Although the bootstrap method generally yields more robust estimates for the parameters of interest, it is unable to mitigate selection bias and remains significantly influenced by the size of the initial sample. Larger samples tend to offer a comprehensive reflection of the entire population, capturing a broader range of characteristics. In contrast, smaller samples, while providing some insights, may miss the nuances of the population, potentially leading to an underestimation of variability within both the original and bootstrap samples. Such limitations pave the way for introducing ***pooled distribution*** in subsequent analyses.

A pooled distribution consists of sample data from other genes within a highly dimensional dataset (e.g. RNA-seq). The introduction of pooled distribution not only mitigates biases from limited sample sizes but also addresses fundamental challenge raised by the nature of real-world data. While bootstrap methods often presuppose continuity in data, actual datasets frequently exhibit discreteness. This mismatch is especially pronounced with smaller sample sizes. Incorporating a pooled distribution introduces a level of "jitter" or minor variability during resampling, effectively smoothing out resampled values. Further details on the integration and advantages of pooled distribution will be elaborated upon in section 3.3.5.

3.3.2 **Mean estimator**

In this section, we explore whether the resampling mean qualifies as an unbiased estimator. An estimator is considered unbiased if its expected value converges toward the actual parameter value it seeks to estimate. Suppose $X_1, X_2, \ldots, X_n$ are random samples selected from a population with mean μ, the expected value of the bootstrap estimator $\overline{\widehat{\theta^*}}$ is unbiased estimator of μ. Mathematically,

$$\mathrm{E}\left(\overline{\widehat{\theta^*}}\right) = \mathrm{E}\left[\frac{1}{\mathrm{B}}\sum_{\mathrm{i}=1}^{\mathrm{B}} \overline{\widehat{\theta^*}}_{\mathrm{i}}\right] = g(P_n)$$

In here, $\overline{\widehat{\theta^*}}_{\mathrm{i}}$ represent the resampled mean calculated from i-th iteration during the bootstrap resampling process. In our simulation experiments, the value of i is set to 100, indicating that the resampling process is repeated 100 times. The function $g()$ represent the statistic of interest (in this case, mean) and P_n represents the population parameters. When the interested statistic $g()$ is mean, the expected value of the estimator $\overline{\widehat{\theta}^*}$ is equal to the true mean μ. The proof of this statement is shown below:

$$\mathrm{E}\left(\overline{\widehat{\theta^*}}\right) = \mathrm{E}\left[\frac{1}{\mathrm{B}}\sum_{\mathrm{i}=1}^{\mathrm{B}} \widehat{\theta_i^*}\right]$$

$$= \frac{1}{B}\sum_{i=1}^{B} E\left(\widehat{\theta_i^*}\right)$$

$\widehat{\theta_i^*}$ is mean estimate of resampled data $X_i^*, \ldots X_m^*$

$$= \frac{1}{B}\sum_{i=1}^{B} E\left[\frac{1}{m}\sum_{i=1}^{m} X_i^*\right]$$

resampled data $X_i^*, \ldots X_m^*$ is mimicking sample data $X_i, \ldots X_n$

$$= \frac{1}{B}\sum_{i=1}^{B} E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]$$

$$= \frac{1}{B}\sum_{i=1}^{B} \mu$$

$$= \frac{1}{B} \times B\ \mu$$

$$= \mu$$

The approximation presupposes that $\overline{\overline{\hat{\theta}^*}}$ furnishes a precise estimate of $\hat{\theta}$ for a particular gene. Thus, the overall resampled mean is unbiased. This can be further proved through meticulous simulation studies, setting our initial focus on a ***normally*** distributed population. Our hypothetical population possesses a mean μ of 10 and a standard deviation σ of 5. From this population, we selected a sample of size n (where n ranges from 2 to 10, and additionally, includes a size of 100 for a broader perspective) to mimic experimental gene count data. Repeated selection 300 times to simulate a dataset involving 300 equally behaved genes. Subsequently, each "gene" underwent a resampling procedure, where n samples were selected with replacement for 100 iterations and calculated the resampled mean values.

Our results, derived from initial sample sizes ranging from 2 to 10 across 300 genes, aligned closely with the aforementioned mathematical propositions. In Figure 13, the near-overlapping trajectories of the three lines suggested that the average of resampled means were consistent with the average of raw sample data. Both of these averages values closely aligned with the true mean value of 10. This observation underscored the efficacy of the resampled mean served as an unbiased estimator in this particular setting. Concurrently, Figure 14 visually mapped the dispersion of resampled means, showcasing boxplots that become increasingly narrow as the initial sample size expands. The shape of these boxplots also hinted at a symmetric distribution of the estimated mean values around the expected value, with increasingly precise as the initial sample size enlarges.

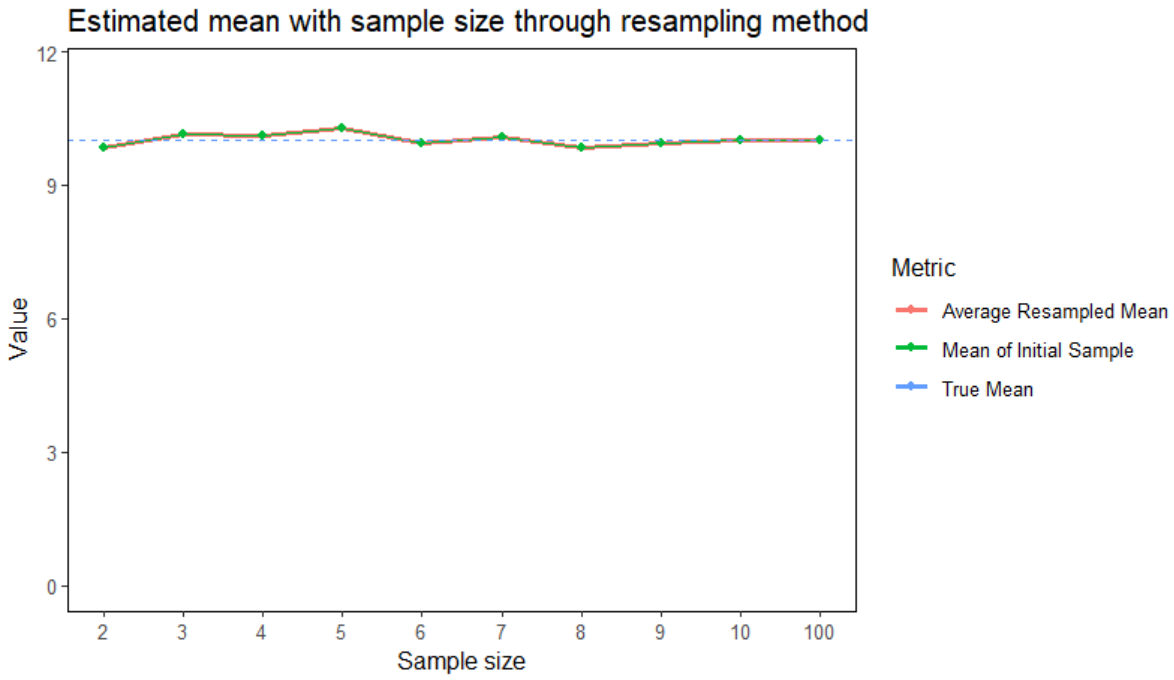

Figure 13 Resampling Accuracy in Estimating Mean Across Sample Sizes: *This line graph depicts the estimated means across varying sample sizes, ranging from 2 to 10, with 100 serving as a reference point. In the graph, the orange line represents the average resampled mean. These estimates of means are calculated from 300 resampling iterations. The green line depicts the average of the initial sample, and the blue dashed line denotes the true mean, which serves as a benchmark for comparison. The close alignment of the orange line (representing the average resampled mean) with the blue dashed line (indicating the true mean) in the graph demonstrates that the resampling method offers an overall unbiased estimation of the mean across various sample sizes.*

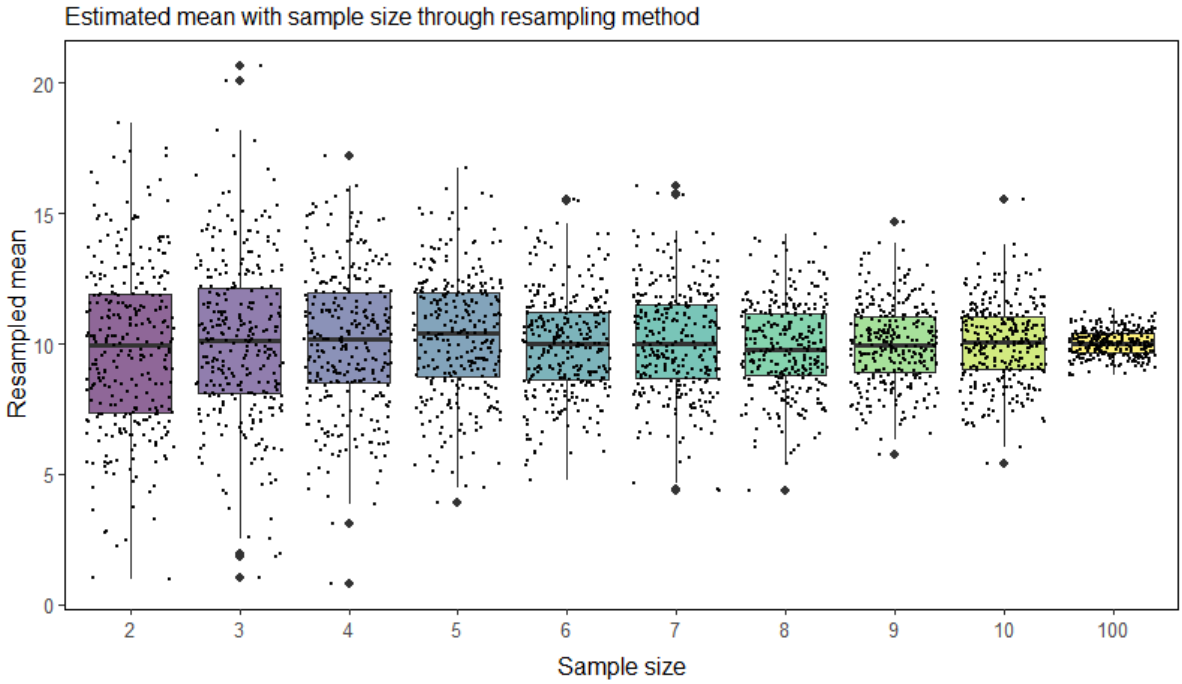

Figure 14 Resampling Variability in Estimating Mean Across Sample Sizes: *Boxplot overlaid with jitter points depicting the distribution of resampled means across different initial sample sizes. For each sample size, 300 estimations of the mean were conducted. The median line of each boxplot, observed to consistently lie near the center of each box and close to the expected value of 10, indicates an unbiased distribution of the estimated means across different sample sizes. Besides, the dispersion of the estimated resampled mean visibly decreases as the sample size increases. This narrowing spread signifies improved precision in mean estimation with increasing sample size.*

3.3.3 **Variance estimator**

The accurate estimation of variance is pivotal when identifying differentially expressed genes. However, unlike the estimator of means, which exhibits a confined range around the expected value (Figure 14), variance estimation can be notably noisy, especially with limited sample size. This is further supported by the mean-variance plot shown in Figure 15, where samples with only three replicates exhibited a broader distribution of variance compared to with 10 replicates. The primary reason for this noisy variance estimation is the inadequate representation of biological

variability due to limited sampling. With fewer replicates, the data obtained is limited, providing a narrow view of the true variations in gene expression. As a result, variance estimates from such small samples tend to be less stable and more prone to fluctuations. Random variations and outliers, which would typically be mitigated in larger sampling, have a disproportionate impact on the results in smaller samples. This leads to difficulty in distinguishing true biological signals from random noise, adding to the noisiness of these variance estimates. To mitigate the noisiness caused by limited sample size, resampling is introduced.

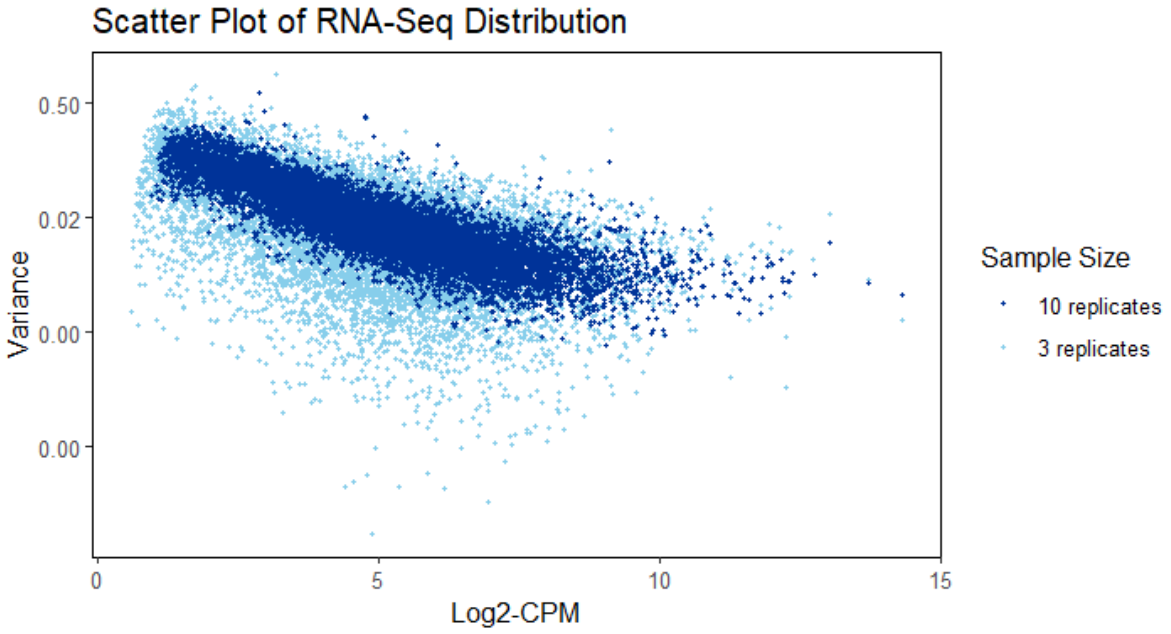

Figure 15 Mean-Variance Distribution of Trastuzumab dataset: *This figure presents a mean-variance plot focusing on the Control group of Trastuzumab dataset (Rinas et al., 2020). Both mean and variance values are calculated from the log-CPM transformed counts data. This graph compares the variance estimates obtained from different numbers of replicates: sample variance from three replicates (light blue) and sample variance from ten replicates (dark blue). The figure shows the variance estimates from the smaller sample size are noticeably noisier compared to those from the larger sample size. Additionally, we observed when sample size decreases, the points tend to spread downward, indicating a tendency toward underestimation of variances.*

To illustrate, our simulation experiment involved drawing 300 genes from a population characterized by a mean (μ) of 10 and a standard deviation (σ) of 5, with initial sample sizes ranging from 2 to 10. Additionally, one sample with an initial size of 100 was included for a broader comparative context. The results, depicted in Figure 16 and Figure 17, provided intriguing insights. Firstly, the blue line represented the average raw variance estimate over 300 repetitions, aligning closely with the green dashed line indicating the true variance. This alignment suggested that, despite displaying significant noise, the raw variance estimates were unbiased. However, the trend and distribution of the raw variance in Figure 17 showed the existence of outliers with values up to 10 times larger than expected when sample sizes were limited. With the median close to the first quantile in the boxplot, it underscored the left-skewed nature of the raw variance data distribution. Overall, this revealed a noisy pattern where most variances are underestimated, yet some are substantially overestimated.

Secondly, the introduction of the resampling method substantially reduced the number of outliers, and the range of estimations converged towards the expected values (as seen in the boxplot in Figure 17). However, the unadjusted resampling method also introduced biases. In Figure 16, a noticeable gap emerged when comparing the orange line (representing the average of resampled variances) and the green line (representing the true variances). This divergence highlighted the underestimations occurring during an unweighted resampling process, particularly pronounced with smaller sample sizes.

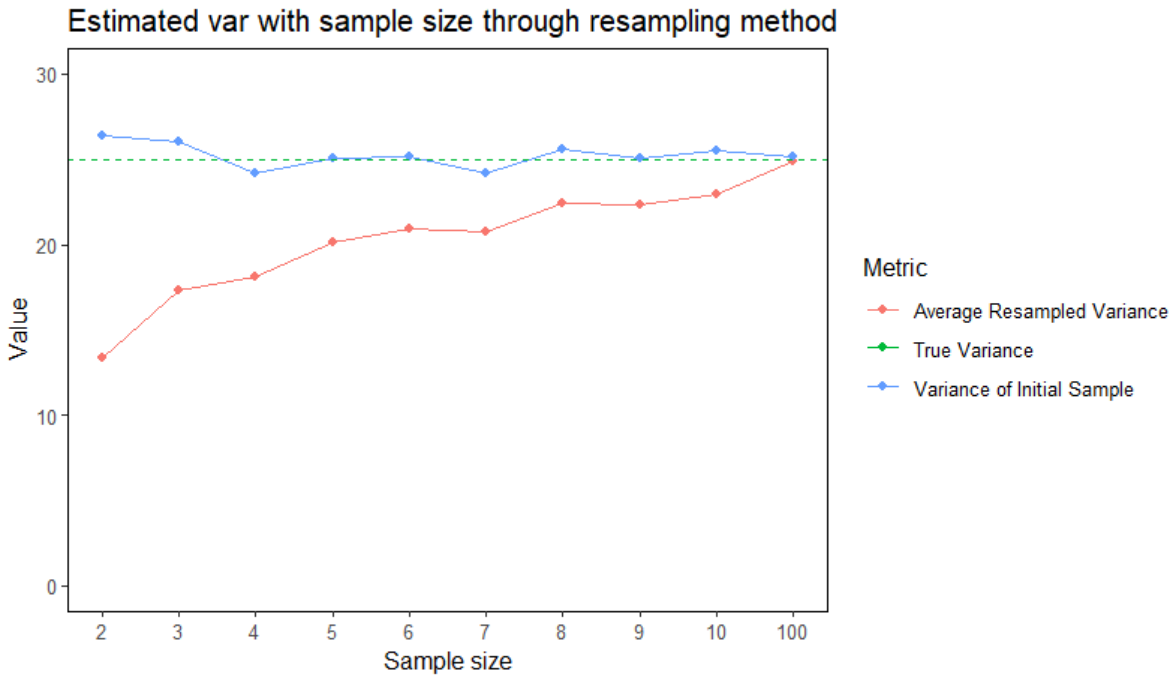

Figure 16 Resampling Accuracy in Estimating Variance Across Sample Sizes: *This line graph illustrates the estimated variances for a range of sample sizes, from 2 to 10, with the sample size of 100 used as a reference. The blue line depicts the variances of the raw samples, and the green dashed line denotes the true variances, serving as a benchmark. The orange line, representing the average resampled variances. These estimates of variances are calculated from the average of 300 iterations, showing that as the sample size decreases, the estimated variances deviate significantly from the expected values. This deviation suggests the introduction of bias in the variance estimation due to the resampling method related to sample size.*

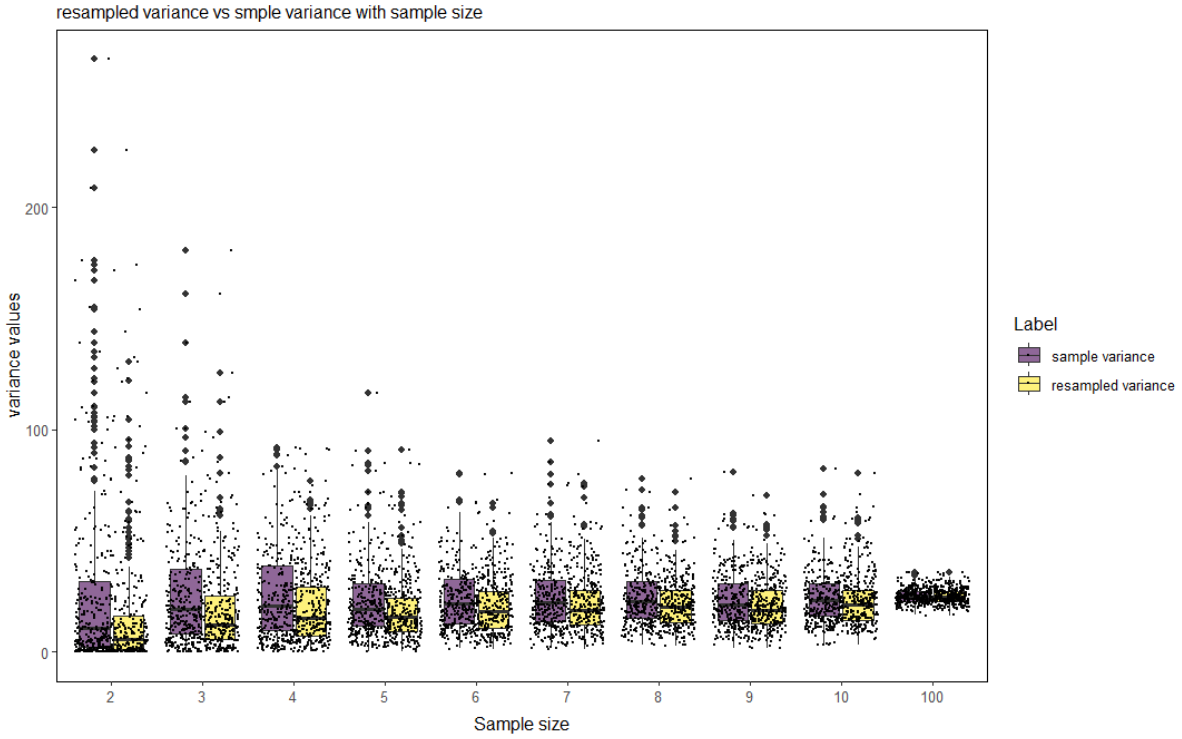

Figure 17 Variability in Estimating Variance Across Sample Sizes: Resampling vs. Initial Sample: *Boxplot overlaid with jitter points depicting the distribution of resampled variances (yellow) compared to the initial sample variances (purple) across different initial sample sizes. For each sample size, 300 estimations of the mean were conducted. The dispersion of the estimated resampled mean visibly decreases as the sample size increases. This narrowing spread signifies improved precision in mean estimation with increasing sample size. Notably, when the sample size is small, the median line of the boxplot is situated closer to the lower quartile, showing a left-skewed distribution of resampled variance. Additionally, there's a presence of jitter points indicating exceptionally high variance values particularly at smaller sample sizes.*

To address the observed variance underestimation, an adjustment factor was implemented, ensuring the aggregate variance across all genes remains consistent before and after resampling. The logic behind this adjustment is rooted in resampling procedure. Consider a basic example of an initial sample containing just two values, X_1 and X_2. In the scenario of resampling with replacement, there's a 50% likelihood for the bootstrap sample to consist of X_1 and X_2.

Concurrently, there's an equal probability for the bootstrap sample to be purely X_1 or X_2. Assuming the original sample has a variance, σ^2, the bootstrap's expected variance would be reduced to $0.5 \times \sigma^2 + 0.5 \times 0 = 0.5\sigma^2$ — a mere half of the anticipated value. This brings us to the mathematical underpinning: the relationship between the adjustment factor for variance estimation and the sample size, succinctly captured by a factor of $\frac{n-1}{n}$. To refine our estimates during the resampling process, consider $X_1, X_2, \dots, X_n$ as the original sample and its resampled counterpart $X_1^*, X_2^*, \dots, X_n^*$ (obtained by drawing with replacement), the adjusted data becomes:

$$X_i' = \overline{X^*} + r_i^* \times \sqrt{\frac{n}{n-1}} \qquad (for\ i = 1,2,3, \dots, n)$$

Where $\overline{X^*} = \frac{1}{n} \sum_{i=1}^{n} X_i^*$ represents the average of the resampled data and $r_i^* = X_i^* - \overline{X^*}$ is the residual of i-th sample. We opt for residuals instead of raw data to uphold the precision of the estimated mean (notably mirroring the true mean) while adjusting the variance. Once this adjustment is applied, we derive a new dataset $X_i', X_2', \dots, X_n'$. The variance, denoted as $\widehat{\theta'}$, is calculated as $\widehat{\theta'} = var(X_i', X_2', \dots, X_n')$.

Figure 18 and Figure 19 graphically portrayed the distribution of variances post adjustment. As anticipated, the adjusted resampled variance offers an unbiased estimate of the population variance. As expected, the variance range became narrower with increasing sample size. Yet, the boxplot for the variance distribution revealed a nuance: while the average delivered unbiased estimates, the distribution exhibited a left skew. This indicated both under and overestimations in the variance estimates. Though most genes showed underestimated variances, outliers with substantially high variances pushed the average towards unbiased values, as seen in Figure 19. One could argue that this scenario mainly raised by the limited replication inherent to RNA-seq,

yielded noisy variance estimates for individual genes. Figure 20 further supported this claim, revealing how the distribution of resampled variance evolved with sample sizes of 2, 4, 6, and 10. As the sample size increased, the variance distribution became more concentrated, its skewness diminished, and it progressively aligned with a normal distribution. To counteract these anomalies, especially prevalent in smaller sample sizes, **variance stabilization** was introduced. This step now becomes the pivotal phase in the "Processing" section.

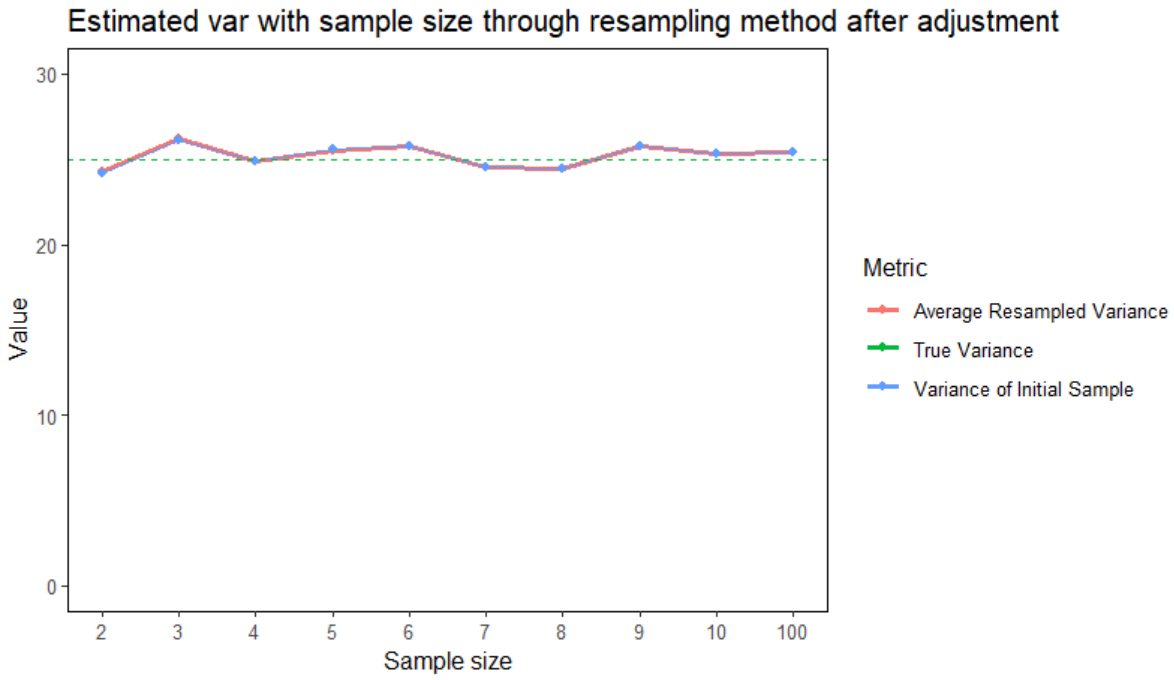

Figure 18 Resampling (After Adjustment) Accuracy in Estimating Variance Across Sample Sizes: *This line graph illustrates the estimated variances after adjustment for a range of sample sizes, from 2 to 10, with the sample size of 100 used as a reference. The blue line indicates the variance of the raw sample. The orange line denotes the adjusted average resampled variance, closely aligning with the green dash line which represents the true variance. These estimates of variances are calculated from the average of 300 iterations. The post-adjustment convergence of the average resampled variance and true variance lines suggests an improved estimation process and leads to an unbiased estimation of resampled variance, thereby enhancing the reliability of variance estimations across varying sample sizes.*

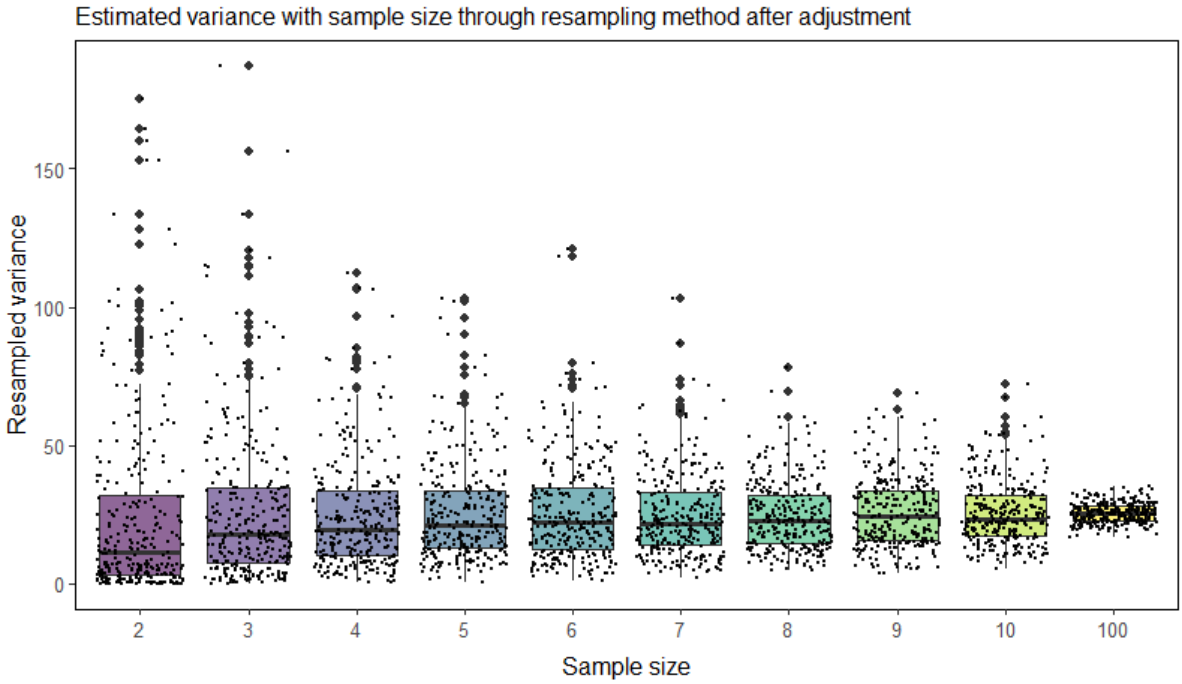

Figure 19 Resampling (After Adjustment) Variability in Estimating Variance Across Sample Size: *Boxplot overlaid with jitter points depicting the distribution of resampled variances after adjustment across different initial sample sizes. For each sample size, 300 estimations of the mean were conducted. Post-adjustment, the boxplots for smaller sample sizes depict elevated variance values, with an overall upward shift in the plots. As sample size grows, the spread of variance estimators around the population variance narrows, highlighting the reduced dispersion of the 300 resampled variance estimators. With smaller sample sizes, the median line of the boxplot approaches the lower quartile, indicative of a left-skewed resampled variance distribution. Moreover, jitter points signal exceptionally high variance, especially prominent for smaller sample sizes.*

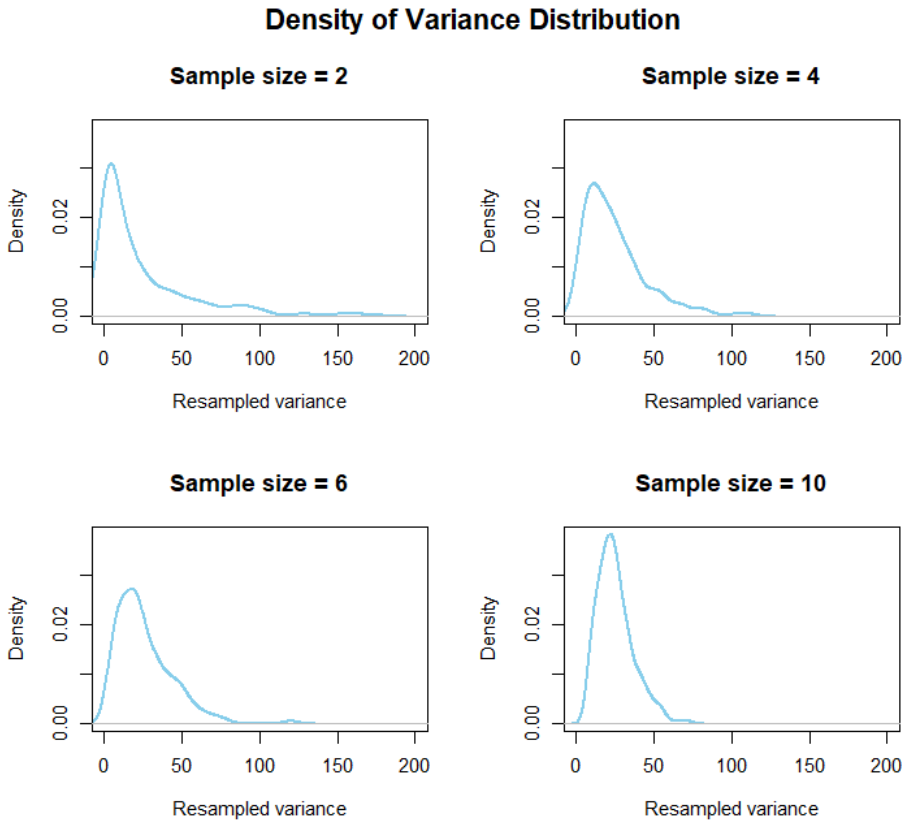

Figure 20 Density Plot of Estimated Variances After Adjustment Across Sample Size: *This figure presents density plots that illustrate the distribution of resampled variances across different sample sizes: 2, 4, 6, and 10. Each sample size includes 300 data points. A pronounced left skew in the distribution of estimated variances is observed when the sample size is small. This skew gradually diminishes as the sample size increases. By the time the sample size reaches 10, the distribution of estimated variances approaches normality, indicating a more balanced and accurate representation of variance as the sample size grows.*

The aforementioned results were obtained through simulations based on the initial assumption of genes following normal distributions. However, similar results were observed when applying the same approach to other initial assumptions, such as negative binomial distributions (the most prevalent assumption for RNA-Seq data) and uniform distributions (a significantly different

distribution). In all cases, the resampled mean estimates provided unbiased results, with the average of estimated means closely approximating the true mean value. The distribution of estimated means exhibited symmetry around the true mean value. On the other hand, the resampled variance estimates yielded average correct values. However, when the sample sizes were small, the distribution of estimated variances skewed to the left. These results are detailed in the Appendix .

3.3.4 **Variance stabilization**

Variance stabilization plays a crucial role in enhancing the accuracy of gene variance estimation in RNA-Seq studies. It helps to mitigate the inherent noisy variance estimates associated with limited sample size, significantly improves the effectiveness of differential expression analysis. Another key benefit of variance stabilization is its ability to mitigate the heteroskedastic nature of RNA-Seq data. In the context of RNA-seq, heteroskedasticity refers to the condition where the variability of a variance is unequal across the range of expression levels. Variance stabilization addresses this issue, making RNA-Seq data more compatible with common statistical methods for exploratory analysis of multidimensional data. Techniques such as principal component analysis (PCA) and its variants, which are commonly employed for clustering and ordination, predominantly rely on the assumption of homoskedasticity. In simpler terms, these techniques are most effective when the variability of variance is independent of the expression level.

Therefore, variance stabilization becomes pivotal in RNA-Seq data analysis. Our proposed method does not assume an initial distribution for the data. Instead, it uses bootstrapping to integrate the sample data and corresponding pooled distributions to achieve accurate estimation of variance, addressing the challenges posed by the limited sample size.

3.3.5 **Pooled distribution**

Traditional methods for variance stabilization in RNA-Seq data typically rely on borrowing information across the entire dataset. Prior methodologies, like **edgeR** and **DESeq2**, fit mean-variance trends across all genes and use it as a *prior* in their EB approach. In contrast, our proposed method opts for a more targeted strategy by constructing a ***pooled distribution*** for each target gene, using genes with similar expression levels. The pooled distribution is analogous to the prior distribution in Bayesian statistics, providing an initial estimate of the distribution and an initial estimate of the variance. In constructing pooled distribution, the residual values of these neighboring genes are used, rather than the raw counts data. The advantage of using residuals is their abilities to capture variance information while maintaining a mean of zero. Consequently, when these residual values are added to individual gene data in the subsequent steps, they subtly adjust the data without influencing the mean gene estimation.

The process of selecting neighboring genes plays a critical role in influencing the sample variance estimation. Typically, we start by gathering residuals from a group of 100 genes that share similar expression levels. However, as these points may include outliers with large variances, we filter out genes with extreme variance, retaining only the genes within the second and third quantile as references. To illustrate, when the sample size is 4, the pooled distribution comprises $100 \times 4 \times 50\% = 200$ residuals. We refer to the group of 100 genes as the "slide window" due to its dynamic nature; it shifts in accordance with the target gene, as depicted in Figure 21. The size of this "slide window" and the extent of filtering could potentially impact the efficacy of DGE analysis, a topic we'll delve into in the results section. But currently, window size of 100 was used in Trastuzumab dataset with 11,910 genes.

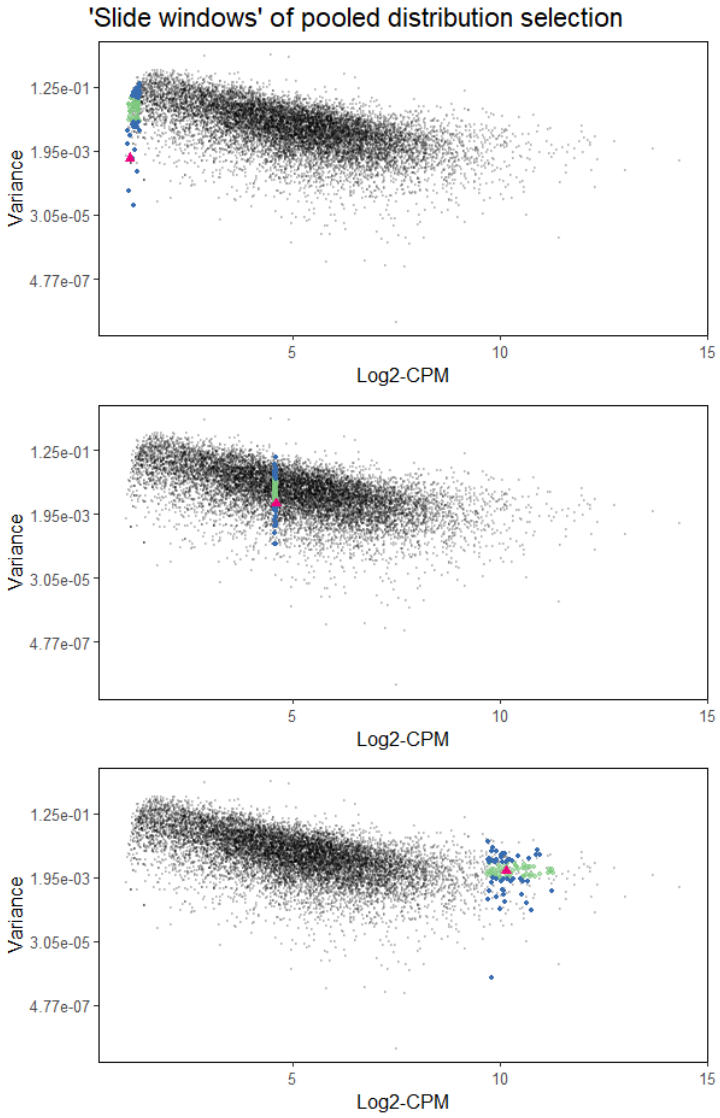

Figure 21 Scatterplot Illustrating the Pooled Distribution Selection: *This figure features a scatterplot that illustrates the sequencing distribution in Trastuzumab dataset (Rinas et al., 2020). In order to show the distribution in scenario with limited sample, three randomly selected samples from Control group are used. In the figure, the variance is plotted against the mean value of log2-CPM. Red triangles represent the target genes, while the other colored dots represent 100 neighboring genes for the target gene. Notably, only the dots highlighted in green signify the 50 selected neighboring genes that were used to construct the pooled distribution for each target gene. This visualization provides a clear depiction of the selection process for the pooled distribution, offering insight into the spatial relationships among the genes in variance stabilization.*

Suppose we are interested in estimating the variance of gene g. I_g represents the index of gene g if the dataset is ordered by expression levels. The selection of neighboring genes falls within the range $[b, e]$, where "b" signifies the "beginning" and "e" denotes the "end". The criteria for selecting these neighboring genes are:

$$[b, e] = \begin{cases} [1, 100] & I_g \leq 50 \\ [I_g - 50, I_g + 50] & 50 < I_g < t - 50 \\ [t - 100, t] & I_g \geq t - 50 \end{cases}$$

Where t represents the total number of genes in the expression matrix. To construct the pooled distribution, we use the design matrix that includes detailed gene information within the specified interval $[b, e]$. This matrix, denoted as M_g, is a subset of the original sample dataset and is pivotal in the pooling process. It with the dimensions of $(e - b) \times n$, where each row represents an individual gene, while the columns correspond to the count data of each sample. The count data for the i^{th} gene in the j^{th} sample is represented as $X_{i,j}$ within the matrix. Thus, the design matrix can be represented by:

$$M_g = \begin{bmatrix} X_{b,1} & \cdots & X_{b,n} \\ \vdots & \ddots & \vdots \\ X_{e,1} & \cdots & X_{e,n} \end{bmatrix}$$

By encompassing this range of gene information and count data, the matrix M_g provides all the necessary information to effectively construct the pooled distribution. We determine the residual matrix by subtracting the gene-wise mean value $\overline{M_g}$ from the design matrix M_g, where $\overline{M_g}$ is a vector representing the average of each row:

$$\overline{M_g} = \begin{bmatrix} \overline{X_b} \\ \overline{X_{b+1}} \\ \vdots \\ \overline{X_e} \end{bmatrix}$$

And the residual matrix r_g can be calculated by

$$r_g = M_g - \overline{M_g} = \begin{bmatrix} r_{b,1} & \cdots & r_{b,n} \\ \vdots & \ddots & \vdots \\ r_{e,1} & \cdots & r_{e,n} \end{bmatrix}$$

The residual matrix for gene g, denoted as r_g, forms the basis of our pooled distribution. But only part of data in this matrix will be used. In the selection process, we rank genes by their sample variances, and only take residuals from central 50% genes in order to mitigate the impact of outliers. The residuals obtained from this filtered subset form the pooled distribution, denoted as P_g, signifying a pooled distribution corresponding to gene g.

Upon constructing the pooled distribution, it is incorporated with the previously calculated resampled variance in the variance stabilization phase. This integration combines these two components using optimal weights to optimize the estimation precision, especially for datasets that have limited replicates. The following steps elucidate the initial methodology used to estimate the resampled distribution and associated parameters. This methodology is further refined in later thesis chapters based on the results here. Using gene g as an example, assuming the resampled weights for gene g is $w_{g,sample}$ and the contribution of pooled distribution is $w_{g,pooled}$, the final variance is calculated by:

1. ***Initialization***: For a given gene g, consider $X_{g,i}$ with $i \in [1, n]$ as its original samples with sample size of n. $\overline{X_g}$ is sample mean.

2. ***Sample Residuals***: Compute the sample residuals by subtracting the average of the sample data $\overline{X_g}$. This provides the residual dataset $r_{g,i}$ with $i \in [1, n]$.

3. ***Pooled Distribution Creation***: Construct the pooled distribution P_g of gene g, which is consisted of n_j residuals from neighboring genes.

4. ***Sampling from Pooled Distribution***: Randomly select n samples with replacement from the pooled distribution P_g, labeled as $p_{g,j}$ with $j \in [1, n_j]$.

5. ***Resampled Residuals***: Randomly select n sample residuals with replacement to obtain resampled residuals data $r^*_{g,i}$ with $i \in [1, n]$.

6. ***Generating New Data***: The pooled distributions are combined with sample data during resampling process to stabilize the variance estimates. The stabilized data for gene g, represented by X'_{g,i^*} with $i^* \in [1, n]$ is determined using the equation:

$$X'_{g,i^*} = \overline{X_g} + w_{g,sample} \times r^*_{g,i} \times \sqrt{\frac{n}{n-1}} + w_{g,pooled} \times p_{g,j} \times \sqrt{\frac{n_j}{n_j - 1}} \quad i \in [1, n] \; j \in [1, n_j]$$

Where $w_{g,pooled}$ and $w_{g,sample}$ represents the pooled weight and sample weight respectively. $\sqrt{\frac{n_j}{n_j - 1}}$ is corresponding adjusted factor for pooled data where n_j is the number of residuals in pooled distribution.

7. ***Parameters Calculation*** Based on the new data set, parameters of interest (such as mean and variance $\widehat{\theta'_g}$) can be determined as:

$$\widehat{\theta'_g} = var(X'_{g,1}, X'_{g,2}, X'_{g,3}, \dots, X'_{g,n})$$

8. ***Repetition***: Steps 4-7 were repeated for B times, resulting in B measures of $\widehat{\theta_g'}$. The average of these estimates is the estimated variance for gene g, symbolized as $\overline{\widehat{\theta_g'}}$.

The $\overline{\widehat{\theta_g'}}$ values represent the estimated variance for gene g. The same approach can be used to estimate the variances for the other genes in the dataset.

3.3.6 **Simulation study**

In the resampling procedure, the values of $w_{g,sample}$ and $w_{g,pooled}$ determines the balance between observed data and pooled distribution so the overall distribution reasonably approximates the underlying 'true' distribution. While the contribution of the pooled distribution reduces the overall noise in the data, the observed sample data is also important in the preservation of essential variance characteristics of each gene. To decide the optimal weights that better captured the true population variance, a simulation study is proposed. It possesses the ability to compare expected and calculated variances, while the real-world data provides only observed samples without any reference to expected data.

Two main categories of approaches for combining observed and pooled distributions are currently under consideration:

1. Weighted arithmetic mean of observed and pooled distributions
2. Weighted geometric mean of observed and pooled distributions

3.3.6.1 Weighted arithmetic mean

The intuitive thought of combining resampled and pooled residuals is the "weighted arithmetic mean". It can be expressed as a linear combination function of these two components:

$$X'_{g,i^*} = \overline{X_g} + w_{g,sample} \times r^*_{g,i} \times \sqrt{\frac{n}{n-1}} + w_{g,pooled} \times p_{g,j} \times \sqrt{\frac{n_j}{n_j - 1}} \quad i \in [1, n] \; j \in [1, n_j]$$

As demonstrated before, we anticipate that both the resampled variance and pooled variance provide unbiased estimations of the true variance. Consequently, the expected relationship between the weights for gene g is represented by the equation:

$$w^2_{g,sample} + w^2_{g,pooled} = 1$$

With such relationship, we would expect the sum of the estimated variances across all genes to stay relatively constant across all choices of $w_{g,pooled}$ and $w_{g,sample}$ before and after variance stabilization. However, the integration of pooled residuals can introduce unpredictability which can be traced back to the creation of pooled distributions. The frequency of genes used in constructing the pooled distribution depends on their relative expression levels within the dataset. For instance, with a sliding window of 100, the count data of gene x_1 (i.e., the gene with the smallest sample mean value) only features in the pooled distribution when measuring x_1 to x_{50}, appearing 50 times. However, the counts data for gene x_{50} are present in the pooled distribution of x_1 to x_{100} being used 100 times when building the pooled distribution for all genes. This discrepancy means that different genes vary in their contribution to the sum of the variance. Furthermore, biases can be intensified by the filtering process undertaken after range selection during the pooled distribution construction. Only variances in middle 50% were selected to construct pooled distribution, significantly reducing the influence of genes with extreme high variances.

While it's computationally taxing to track each gene's contribution during pooled distribution, a ***global adjustment*** factor simplifies this task. This factor ensures variance summation remains consistent. Given $\widehat{\overline{\theta_1'}}, \widehat{\overline{\theta_2'}}, \widehat{\overline{\theta_3'}}, \cdots, \widehat{\overline{\theta_t'}}$ are the estimated variance for dataset contains t genes, the adjusted factor is given by:

$$\text{adjusted factor} = \frac{\sum_{g=1}^{t} \theta_g}{\sum_{g=1}^{t} \widehat{\overline{\theta_g'}}}$$

Here, θ_g presents the variance of observed sample data of gene g. This adjusted factor then applied to generate the final dataset:

$$X''_{g,i^*} = \overline{X_g} + \left(w_{g,sample} \times r^*_{g,i} \times \sqrt{\frac{n}{n-1}} + w_{g,pooled} \times p_{g,j} \times \sqrt{\frac{n_j}{n_j - 1}} \right) \times \sqrt{\frac{\sum_{g=1}^{t} \theta_g}{\sum_{g=1}^{t} \widehat{\overline{\theta_g'}}}}$$

$$i \in [1, n] \;\; j \in \left[1, n_j\right]$$

Then the estimated variance for gene g is variance of new dataset:

$$\widehat{\Theta_g''} = \text{var}\left(\text{X}''_{g,1}, \text{X}''_{g,2}, \ldots, \text{X}''_{g,\text{n}}\right)$$

With constant weights for all sample

The investigation of selecting optimal weights begins by assuming equal weighting for every gene in the dataset, denoted as $w_{1,sample} = w_{2,sample} = \cdots = w_{t,sample} = w_{sample}$. The objective is to find the optimal weights, w_{sample} and w_{pooled}, that most accurately estimate the true variances for all genes.

The simulation studies initiated with dataset consisted of two log-normal/negative binomial distributions with same mean but distinct variances. These distributions were specifically

selected as they are two common hypotheses about the nature of gene expression distributions in RNA-Seq data analysis. This design aims to test our method's ability to mitigate the overestimation and underestimation in gene variance measurement while in the meantime, keep its unique features. The second set of simulation data was designed containing characteristics similar to actual RNA-Seq data. The Trastuzumab dataset (Rinas et al., 2020) contains 10 samples under the same experimental setting. Given this comparatively large sample size, the calculated sample means and variances among these samples are considered to represent 'true' values. In the second set of simulation data, the performance of resampling methods is evaluated by assessing the accuracy of estimated means and variances derived from a randomly selected subset of n samples within the same dataset. This approach ensures that our method is not only theoretically sound but also practical and applicable to real-world data scenarios.

Besides the design of datasets, the other critical aspect in simulation study is the selection criteria for these optimal weights. Initially, the optimal weight is identified as the one that minimizes the absolute difference between the true population variance and the estimated variance. However, under this criterion, we have observed a tendency for the optimal pooled weight, w_{pooled}, tended towards 1, particularly in experiments with limited sample sizes. This trend resulted from the existence of extremely high overestimation in sample variances, leading to a greater reliance on pooled variance to mitigate this effect. Furthermore, we noted that the optimal weights shifted correspondence to changes in sample size. In smaller samples, due to inherent noise in the RNA-Seq process, the estimated variance tended to stray from the sample variance. In contrast, as the sample size increased, the accuracy of sample variance measurements improved, reducing the dependency on pooled distribution and thus lowering the value of w_{pooled}.

However, a limitation of this approach became evident: using common weights for all genes fails to account for the unique variability inherent in individual samples. This realization highlights the need for a more refined method, one that can accurately estimate true variance while recognizing the distinct characteristics of each gene in the dataset.

With sample-variant weights

In the method of common weights, the close-to-one w_{pooled} value made the estimated variances rely heavily on the pooled variances, thus causing the mean expression level the sole determinant of estimated variance. The limitation of such method lies in its tendency to over-correct samples, ignoring the factor of sample variances and introducing significant bias especially in genes exhibiting similar expression levels but differing in sample variance. Figure 21 demonstrates that genes with similar expression levels can have very diverse variance values. While part of this variance discrepancy may arise from overestimation or underestimation in experiments with limited sample sizes – an issue our method aims to address – it should also differentiate between a gene's inherent biological variability and the variability introduced by experimental constraints. This distinction is vital for accurate gene expression analysis.

To enhance the accuracy of variance estimation, our second-generation model incorporates sample variance into the weight determination process. This model assigns varies weights to each sample based on their respective sample variances. Intuitively, samples with higher variance, which tend to exhibit a wider spread, are considered more representative of the population expression levels and are thus assigned greater weight. Conversely, samples with lower variance tend to cluster closely, and are prone to introduce biases in the estimation of expression levels. However, a critical aspect to consider in this variance model approach is its influence on the

subsequent t-tests used for identifying differentially expressed genes (DEGs). Higher variance values generally result in lower expected t-values, potentially increasing false negative rates and reducing the statistical power of the tests. Thus, during the variance stabilization step, we implemented adjustments for samples with variances that are markedly overestimated or underestimated. Specifically, these samples were adjusted to have a less impact on the overall variance estimation compared to the pooled distribution. Conversely, for samples with moderate variances, we would expect the overall estimate leans forwards the sample variance. This theory was further supported by the patterns observed in Figure 22, which depicts a comparison of raw sample variances exhibiting extreme values against the true variance data, highlighting the need for variance stabilization.

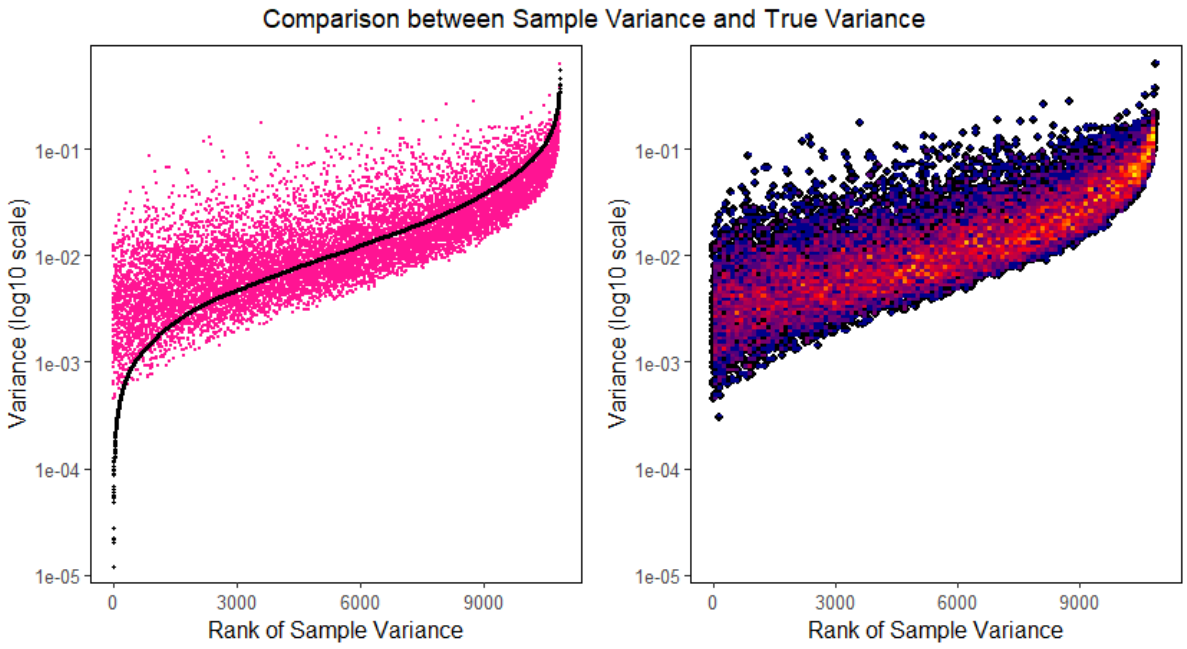

Figure 22 Scatterplot Comparing Sample Variances to True Variances: *This figure presents a scatterplot based on a control sample from Trastuzumab dataset (Rinas et al., 2020), with a sample size of 4. On the left, the scatterplot displays the distribution of sample variances (black dots) against their ranks on the x-axis and their corresponding log10-scaled variance values on the y-axis. The pink dots depict the 'true' variance values based on n=10 for each gene corresponding to the x-axis positions. The right graph enhances this visualization of true variances density, with blue, red, and yellow indicating regions of increasing density of points. This graph reveals that most sample variances approximate the true variances, particularly evident when sample variance is large. It can be observed a narrower range and higher density regions around positions of large sample variances. However, there still existence outliers with extreme variance values. Combining graphs demonstrate the necessity for shrinkage of sample variance with extreme values to better align with the true variances.*

In the application of arithmetic means for combining variances, a key consideration is the significant impact of the magnitude of sample variance values on the overall outcome. For example, consider a scenario where the pooled variance is set at 10 and the sample weight is 0.5, sample variances of 1 and 100 contribute markedly differently to the overall variance, despite

having the same weight. This discrepancy necessitates a method to effectively handle such variations. Informed by trends observed in Figure 22, we noted that extreme variances, when plotted on a logarithmic scale, tend to distribute around the expected values evenly. Consequently, the determination of weights in our method should focus on the proportional relationship between sample variances and pooled variances, rather than a mere additive relationship. This approach acknowledged the varying influence of sample variances and ensured a more balanced and accurate integration of sample and pooled variance information.

Considering all these factors, the weights kernel function we have developed for gene g was defined as:

$$w_{g,pooled}^2 = \mathrm{K} \times \left(1 - \frac{\sigma_{g,pooled}^2}{\sigma_{g,sample}^2 + \sigma_{g,pooled}^2}\right) \qquad \mathrm{K} \in [0,1]$$

Here, $\sigma_{g,sample}^2$ represents the raw sample variance measured of gene g, and the $\sigma_{g,pooled}^2$ is the variance estimated from the pooled distribution of gene g, calculated as the median of sample variances from 50 filtered neighboring genes. The median is preferred over the average due to its lower sensitivity to skewness in the pooled distribution.

Consequently, once the pooled weights are determined, the corresponding sample variances weight can be calculated:

$$w_{g,sample}^2 = 1 - w_{g,pooled}^2$$

With the restriction of K within the range of 0 and 1, $w_{g,pooled}$ within the range of $\left[0, \sqrt{K}\right]$, and $w_{g,sample}$ within the range of $\left[\sqrt{1-K},\ 1\right]$.

Subsequently, the next step involves determining the optimal value of K to ensure the most accurate weighting functions. Through simulation studies using the dataset mentioned, we found that minimizing the sum of differences between estimated and true variances wasn't an optimal criterion. Instead, we used a complementary criterion: minimizing the number of genes with variances over or underestimated by more than 50% compared to 'true' values. This approach indicated that the optimal value of K around 0.9.

Additionally, we initially included an adjustment factor to maintain constant total variance before and after the variance stabilization process (mentioned in previous chapter). However, upon further experimental evaluation, we observed that this factor often led to an inflated overall estimated variance. This inflation was likely due to the presence of extreme outliers. This phenomenon is further observed in Figure 23 scatterplots, where a majority of the genes were positioned above the diagonal line, indicating an overestimation of sample variances. Subsequent differential gene expression (DGE) analysis using this method also encountered similar issues. Removing the adjustment factor from our process resulted in more accurate variance stabilization (Figure 23) and, consequently, enhanced the power of the statistical testing.

Despite the complexity of this linear combination method, an alternative approach using the geometric mean for combining pooled and sample variances emerged as a more natural and intuitive method. This method, avoiding the need to consider the magnitude of sample variance, yielded better outcomes than the linear combination approach.

3.3.6.2 Weighted geometric mean

The fundamental principle of our linear combination approach is to determine optimal weights for "weighted arithmetic mean" of sample and pooled data. Inspired by this, a statistic of central tendency, "weighted geometric mean" is proposed, possessing substantial robustness against sample with "large-value noise". In general, the geometric mean of variances from pooled distribution and sample data is calculated by

$$\sigma_g'^2 = \sqrt{\sigma_{g,sample}^2 \times \sigma_{g,pooled}^2}$$

Where $\sigma_g'^2$ represents the stabilized sample variance for gene g, $\sigma_{g,sample}^2$ is the raw sample variance and $\sigma_{g,pooled}^2$ is the median value of its corresponding pooled genes, which is selected as the best estimator for the expected pooled variance due to its robustness. This method ensures that pooled distributions are effectively integrated with the sample data during resampling.

Further refining this method, we employ a weighted geometric mean that considers sample size as a critical factor. This approach allows sample variance to contribute more significantly when the sample size is larger. The weighted geometric mean, considering a sample replication of n and a pooled balance parameter k, is calculated as:

$$\sigma'^2_g = (\sigma_{g,sample}^{n-1} \times \sigma_{g,pooled}^{k})^{\frac{2}{n+k-1}}$$

The pooled balance parameter k can be interpreted as the equivalent number of samples in the pooled distribution contributing to the geometric mean. The $n-1$ term accounts for degrees of freedom in sample variance estimation and avoids undefined cases for $n = 1$. A minimum value

of $\sigma_{g,sample}^{n-1}$ is set as $\delta \cong 10^{-5}$ to prevent the undefined values when $\sigma_{g,sample} = 0$, a situation common in limited sample sizes.

The residuals for both sample and pooled distributions are then adjusted to match this target variance σ'^{2}_{g}. The adjusted sample residuals $r'_{g,i}$ and pooled residuals $p'_{g,j}$ are calculated as follows:

$$r'_{g,i} = r^{*}_{g,i} \times \sqrt{\frac{n}{n-1}} \times \frac{\sigma'_g}{\sigma_{g,sample}}$$

$$p'_{g,j} = p_{g,j} \times \sqrt{\frac{n_j}{n_j - 1}} \times \frac{\sigma'_g}{\sigma_{g,pooled}}$$

The terms $\sqrt{\frac{n}{n-1}}$ and $\sqrt{\frac{n_j}{n_j-1}}$ correct for variance underestimation due to resampling with replacement. The adjusted sample and pooled residuals, $r'_{g,i}$ and $p'_{g,j}$, are then linearly combined to generate stabilized results:

$$X'_{g,i^*} = \overline{X_g} + w_{g,sample} \times r'_{g,i} + w_{g,pooled} \times p'_{g,j} \quad i^*, i \in [1,n] \;\; j \in \left[1, n_j\right]$$

Here, $i^* \in [1,n]$ represents the generation of n resampled data points. The adjusted sample residuals weights $w_{g,sample}$ and adjusted reference residuals $w_{g,pooled}$ are derived directly from the number of sample observations n and the reference balance parameter k:

$$w_{g,sample} = \sqrt{\frac{n-1}{n+k-1}}$$

$$w_{g,pooled} = \sqrt{\frac{k}{n+k-1}}$$

These weighting equations are consistent with rule

$$w_{g,sample}^2 + w_{g,pooled}^2 = 1$$

No global variance adjustment is required with this approach.

As the parameter k is equivalent to number of samples in the pooled distribution contributing to the overall variances, our next step is identifying the optimal value of k for the most accurate estimation of true variance values. For this purpose, we conducted simulation studies using the criteria mentioned above. Overall, k with the value of 4 was selected as the optimal value for our weight equations.

Figure 23 illustrates the comparative results of various variance estimates used above, against the true variances in the logarithmic scale, using real-world data with initial sample size of 4. In this figure, “raw” labeled the directly measured sample variances, while “common” represents the results obtained using constant weights. “Arithmetic mean (adj)” and “arithmetic mean” represent the estimated variances derived from weighted arithmetic mean with and without the global adjustment. Lastly, “geometric mean” displays the outcomes from the weighted geometric mean, which is deemed the most optimal method. It's important to note that the preference of the “geometric mean” method isn't solely based on its highest correlation value, as correlation doesn't always equate to accuracy. Rather, it is chosen because its estimated variances lie closest to the diagonal, indicating a close match to the true variances.

This comparison outcome demonstrates the accuracy of the weighted geometric mean method in achieving precise variance estimations, thereby significantly enhancing the overall reliability of RNA-Seq data analysis.

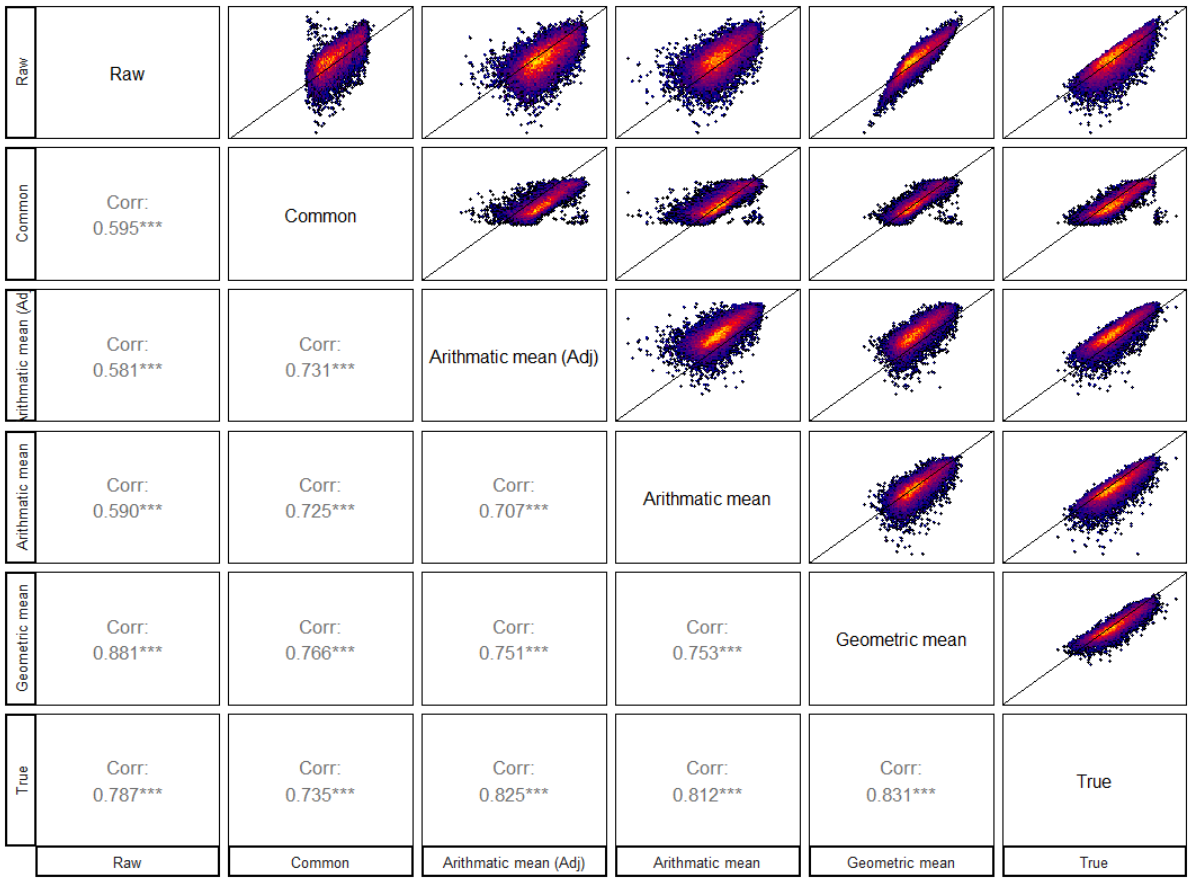

Figure 23 Comparison Between Estimates of Variances: *This plot compares the various estimation of variance against the true variance in the logarithmic scale (base 10) from simulated experiments. Variances were estimated from 4 randomly selected groups within the Control group of Trastuzumab dataset (Rinas et al., 2020). The categories of estimates shown are: measured sample variance, the common weights; weighted arithmetic mean with global adjustment; arithmetic mean without adjustment; and weighted geometric mean. The upper panels of figures show the scatterplot of distributions where the blue, red, and yellow colors indicate regions of increasing density of points. In the bottom panels of the figure, correlations were presented, which quantitatively compare the relationship between any two of these estimation methods.*

3.4 Output section

3.4.1 **Statistical testing**

Our method enhances the traditional approach for estimating significance and p-values in RNA-Seq analysis by introducing a resampling technique for more accurate variance estimation:

Traditional approach

1. Obtain sample observations for treatments $X_{g(t),i}$ with $i \in [1, n]$ and for controls $X_{g(c),i}$ with $i \in [1, n]$.
2. Calculate sample mean for treatment group $\overline{X_{g(t)}}$ and control group $\overline{X_{g(c)}}$.
3. Generate stabilized variance values for treatment group $\sigma'^2_{g(t)}$ and the control group $\sigma'^2_{g(c)}$.
4. Calculate the t-test statistic using $t = \frac{Z}{s}$, where Z is the difference of means $\left(\overline{X_{g(t)}} - \overline{X_{g(c)}}\right)$ and s is the standard error of the two sample, $s = \sqrt{\frac{\sigma'^2_{g(t)} + \sigma'^2_{g(c)}}{n}}$ for equal sample size of n.
5. The t-test statistic is compared to the t-distribution with $df = 2n - 1$ and calculate the corresponding p-values after FDR adjustment.

Our methodology builds on the traditional approach for estimating significance and p-values in RNA-Seq data analysis by incorporating optimal weights to create a data representation that more accurately reflects the population from which the samples were drawn. The improvement method creates a distribution for each sample with resample number $100 \gg n$ through resampling, providing a robust estimator for variances.

The t-statistics that are directly measured in the traditional approach outlined above require accurate estimates of sample means and variances. In contrast, our **bootstrapping** approach estimates s by measuring the standard error of the mean from the resampled distribution.

Improved method with resampling:

1. Obtain sample observations for treatments $X_{g(t),i}$ with $i \in [1, n]$ and for controls $X_{g(c),i}$ with $i \in [1, n]$.
2. Calculate sample mean for treatment group $\overline{X_{g(t)}}$ and control group $\overline{X_{g(c)}}$.
3. Build pooled distribution for both treatment and control groups. The treatment pooled distribution $P_{g(t)}$ for gene g consists of hundreds of residuals from genes with similar expressions. These residuals are denoted as $p_{g(t),j}$ with $j \in [1, n_j]$ where n_j is the number of residuals in pooled distribution. Perform a similar construction for the control group, generating a pooled distribution $P_{g(c)}$ with residuals $p_{g(c),j}$ with $j \in [1, n_j]$.
4. Find the optimal weights for treatment group: $w_{g(t),sample}$ and $w_{g(t),pooled}$. Compute stabilized sample residuals $r'_{g(t),i}$ and pooled residuals $p'_{g(t),j}$. Same parameters are calculated for control group: $w_{g(c),sample}$, $w_{g(c),pooled}$, $r'_{g(c),i}$ and $p'_{g(c),j}$
5. Generate stabilized data $X'_{g(t),i^*}$ and $X'_{g(c),i^*}$ through linear combination of resampled residuals and pooled distribution, where i^* ranges from 1 to 100 represents the 100 new generated sample. i is randomly selected from 1 to original sample size n, and j is randomly selected from 0 to n_j.

$$X'_{g(t),i^*} = \overline{X_{g(t)}} + w_{g(t),sample} \times r'_{g(t),i} + w_{g(t),pooled} \times p'_{g(t),j}$$

$$X'_{g(c),i^*} = \overline{X_{g(c)}} + w_{g(c),sample} \times r'_{g(c),i} + w_{g(c),pooled} \times p'_{g(c),j}$$

6. Repeat step 5 for 100 iterations to generate 100 stabilized data points for control and treatment group separately. These 100 data form the variance-stabilized sample distribution, denoted as L_t and L_c.
7. Randomly draw n (matching the original sample size) samples with replacement from L_t and L_c respectively. Calculated the difference of means between these two new samples, denoted as l_{diff}.
8. Repeat step 7 for 100 times and producing a distribution for difference of means P_{diff}.
9. Calculate the mean of P_{diff}, denoted as M_{diff}. This value should be relatively unchanged from the original estimate $\overline{X_{g(t)}} - \overline{X_{g(c)}}$.
10. Calculate the standard error of P_{diff}, denoted as s_{diff}.
11. The corresponding t-test statistic can be calculated $t = \frac{Z}{s} = \frac{M_{diff}}{s_{diff}}$.
12. The t-test statistic is compared to the t-distribution with $df = 2n - 1$ and calculate the corresponding p-values after FDR adjustment.

The improved method creates the variance-stabilized populations with a greater number of data points compared to the initial sample size. It ensures a more robust estimation of gene variances, thereby enhancing the accuracy of statistical testing.

Chapter 4

Results and Discussion

4.1 Simulation testing

Simulation experiments 1

Our simulation was modeled on real-world Trastuzumab dataset (Rinas et al., 2020), involving 11,910 genes per sample. Given our focus on addressing issues arising from limited sample sizes, we tested on different total sample sizes for each group, specifically $n \in \{2,3,4\}$, which are also a common choice in experimental settings for RNA-seq. Both the control and treatment groups were of equal sample size. In the initial simulation, true fold changes were independently assessed, implying that the dataset contained only one type of differentially expressed gene. The selection of fold changes for the simulation was informed by the range observed in the MA plot (Figure 24), derived from a comparison between control and treatment groups, each consisting of 10 samples. Given that the sample size of 10 is robust and sufficient, we considered the results obtained from these samples as representative of the 'true' fold changes in Trastuzumab dataset. This assumption is crucial as it allows us to use these fold changes as a benchmark for assessing the accuracy and effectiveness of the proposed **bootstrapping** models and other common methods. Given that the log2 fold changes observed predominantly ranged between -1.5 and 1.5, the simulated fold changes were set between 1.5 to 3 in both directions. Specifically, we set fold changes ranging from 3 to -3 with intervals of 0.5, with each category encompassing 2% of the total genes. These differentially expressed genes were randomly selected across the mean count

range. Furthermore, to create realistic simulation data without assuming specific distributions, we directly multiplied or divided the counts of selected genes by the expected fold change. For example, genes with a two-fold change had their counts multiplied or divided by 2.

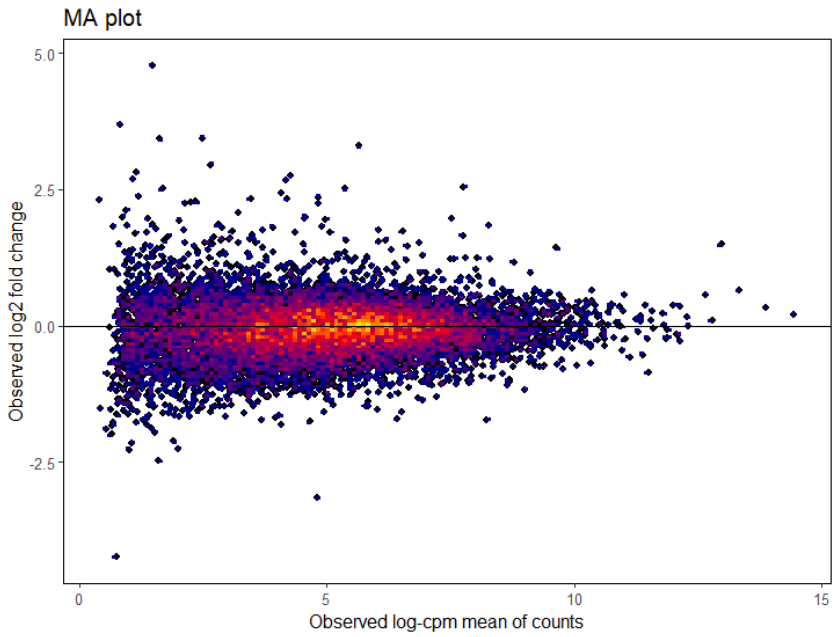

Figure 24 MA-plot Using Trastuzumab Dataset: *This figure presents an MA-plot generated from a comparative analysis of 10 samples each from the control and treatment groups within Trastuzumab dataset (Rinas et al., 2020). The values estimated from 10 samples are considered as the 'true' level of gene expression. In the MA-plot, the y-axis depicts the log2 fold changes between the control and treatment groups, while the x-axis represents the average expression level across these groups. This plot provides a visual representation of differential expression, highlighting the relationship between the magnitude of change (fold change) and the intensity of gene expression (mean expression level).*

Sensitivity and precision

The performance of algorithms in the simulation data was evaluated based on their power and false positive rate. The power of the test was calculated as the proportion of genes with a Benjamini-Hochberg (BH) adjusted p-value < 0.05 among the genes with a true difference

between the two groups. The false positive rate (FPR) measured the likelihood of incorrectly identifying a gene as differentially expressed when it is not. Figure 25 and Figure 26 plotted the power versus false positive rate graph, showing the results of comparative analysis of our proposed bootstrapping method against commonly utilized methods in identifying differentially expressed genes. From the analysis presented in the figure, it was observed that the results for fold changes at the same magnitude were consistent, regardless of whether they were positive or negative. This finding aligned with our expectations, as in a two-group test scenario, positive and negative fold changes ultimately represent the same relationship. In the scenario where sample size was extremely limited ($n = 2$) or fold changes were closed to one, the **bootstrapping** method demonstrated less optimal power compared to other methods. However, in all other scenarios, the **bootstrapping** method exhibited comparative power among the various methods tested. Additionally, the **bootstrapping** method stood out for its high ability to control the false positive rate, underscores its conservative nature in identifying differentially expressed genes (DEGs).

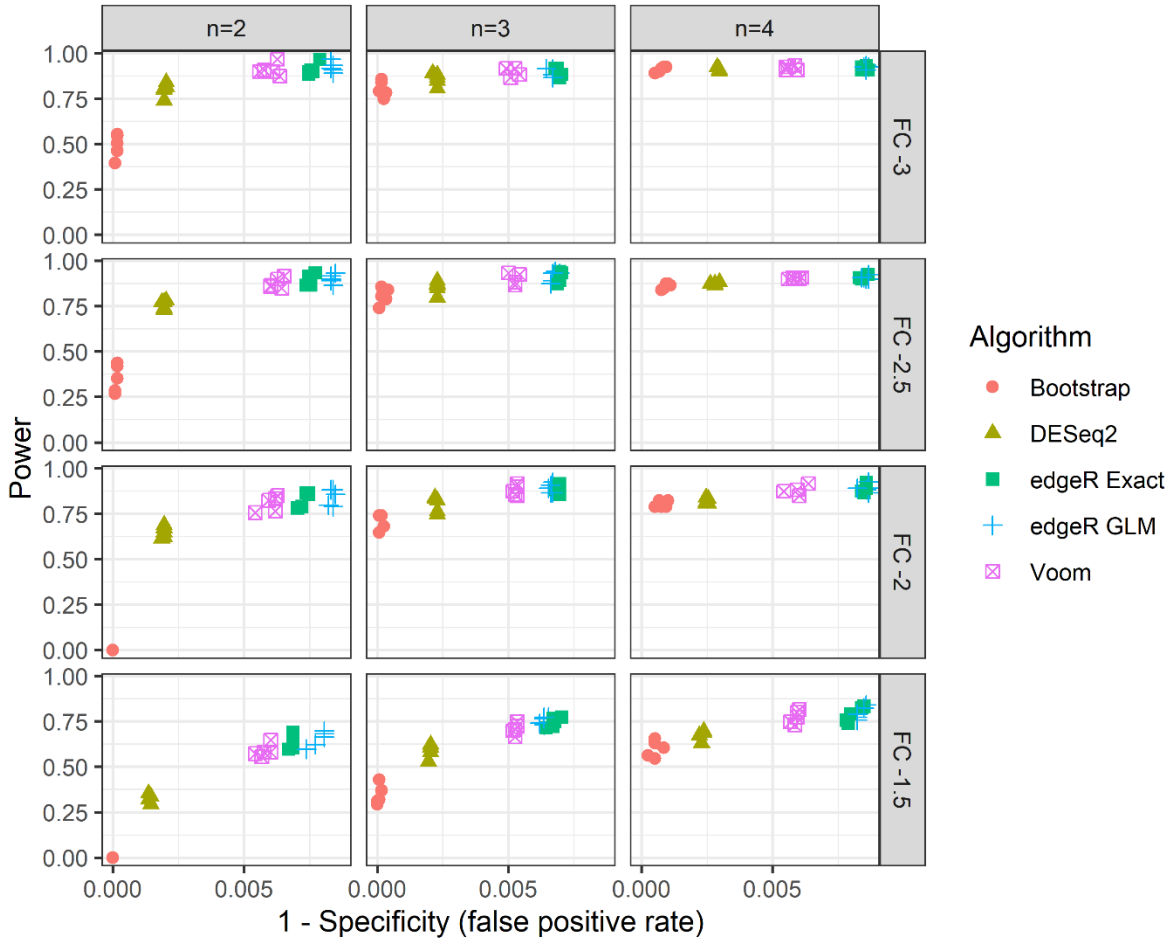

Figure 25 Power and FPR of Algorithms Across Combination of Sample Size and Negative FC: *This figure illustrates the results of proposed simulation study designed to assess the power and false positive rate of various algorithms under the scenario when DEGs are low expressed in treatment group (FC smaller than 0). Each combination of FC and initial sample size was run 5 times. The study specifically evaluates the performance of our proposed bootstrapping method in comparison to other algorithms. Power is calculated as the proportion of genes with an adjusted p-value less than 0.05, among those genes that truly exhibit differences between group means. The false positive rate is determined as the proportion of genes incorrectly identified as differentially expressed among those genes that do not have true differences. The adjustment of p-values is calculated using the Benjamini-Hochberg (BH) method, with 0.05 being a commonly accepted threshold for significance. The results show that our proposed bootstrapping method consistently achieves the lowest false positive rate among the algorithms tested, while maintaining comparable power in the detection of differentially expressed genes (DEGs).*

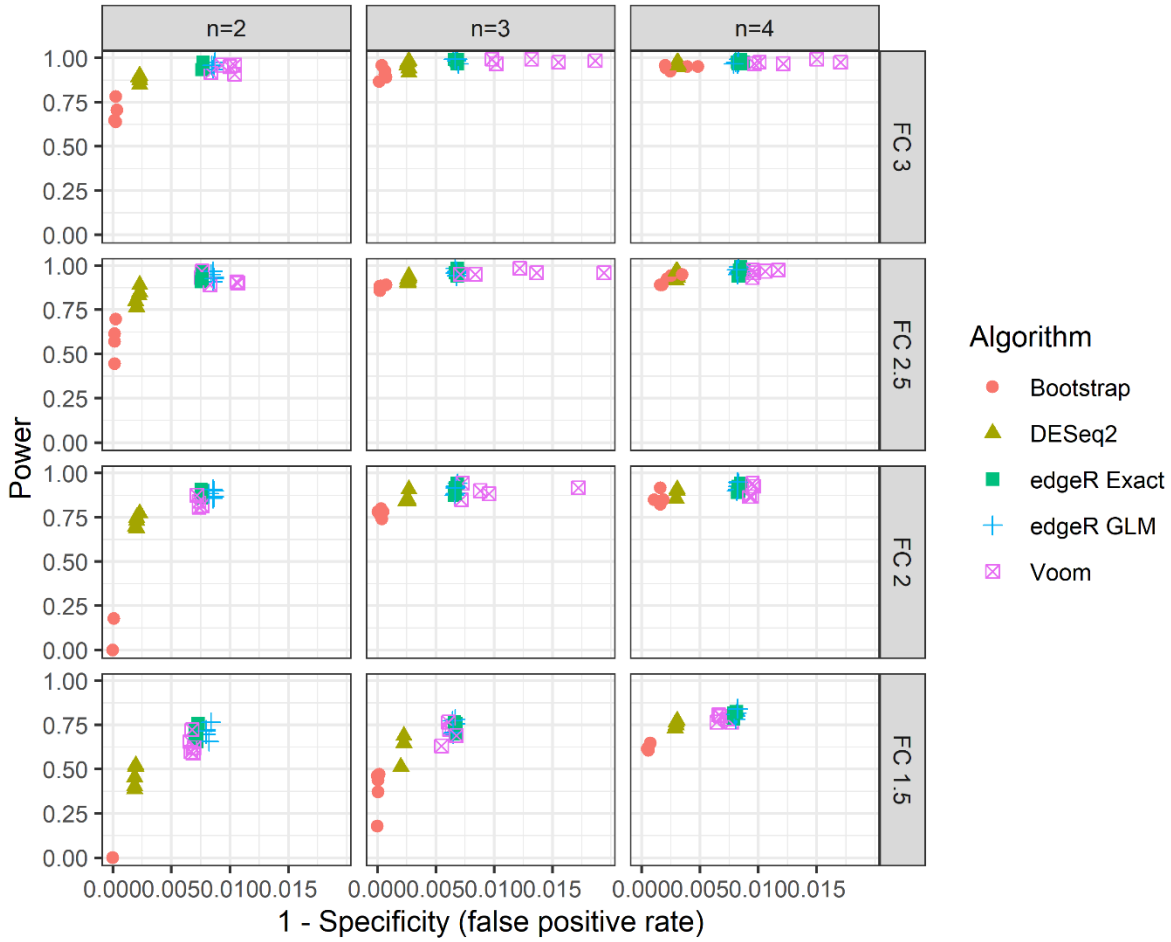

Figure 26 Power and FPR of Algorithms Across Combination of Sample Size and Positive FC : *This figure illustrates the results of proposed simulation study designed to assess the power and false positive rate of various algorithms under the scenario when DEGs are high expressed in treatment group (FC larger than 0). Each combination of FC and initial sample size was run 5 times. The study specifically evaluates the performance of our proposed bootstrapping method in comparison to other algorithms. Power is calculated as the proportion of genes with an adjusted p-value less than 0.05, among those genes that truly exhibit differences between group means. The false positive rate is determined as the proportion of genes incorrectly identified as differentially expressed among those genes that do not have true differences. The adjustment of p-values is calculated using the Benjamini-Hochberg (BH) method, with 0.05 being a commonly accepted threshold for significance. The results show that our proposed bootstrapping method consistently achieves the lowest false positive rate among the algorithms tested, while maintaining comparable power in the detection of differentially expressed genes (DEGs).*

Simulation experiments 2

Multiple kinds of fold changes coexisted in the dataset for second simulation experiments. This dataset incorporated genes with all the previously mentioned fold changes, with each type of fold change accounting for 2% of the genes. This setup resulted in 16% of the genes in the dataset experiencing a diverse range of fold changes. This approach aimed to closely simulate the complex nature of gene expression variations typically observed in actual RNA-Seq data. We evaluated the power and false positive rate for initial sample sizes of 2, 3, and 4 in each group under such setting. To ensure the robustness of our findings, we conducted the simulation 50 times, assessing the consistency of differential gene expression (DGE) analysis. The outcomes of such setting were shown in Figure 27, Figure 28 and Figure 29. When integrating findings from both figures, we observed a general trend: all methods demonstrated lower power in detecting genes with low fold changes. This was expected, as minor fold changes can often be mistaken for measurement biases and consequently addressed during the variance stabilization step, resulting in lower detection power. Similarly, the detection power of tools was decreased in cases with fewer replicates, as most methods tended to treat sample data as less reliable and leaned more heavily on overall trends. This approach can lead to a conservative bias, potentially resulting in the under-detection of true differences in gene expression. However, this conservative approach also brings benefits. The conservative properties ensure a generally lower false positive rate. This means that while they may occasionally miss detecting true differences, they are less likely to falsely identify differences where none exist. This balance between power and specificity (1-FPR) is crucial in ensuring the reliability and validity of results, especially in studies with constrained sample sizes.

In Figure 27 and Figure 28, genes were categorized by different fold changes (positive and negative) to specifically assess the effectiveness in detecting these diverse changes. This categorization facilitates a detailed analysis of each method's performance in identifying genes across a range of differential expression levels. Within these categories, it was observed that the **bootstrapping** method exhibited slightly lower power compared to **edgeR** and **Limma-Voom** when dealing with limited sample sizes and fold changes close to one. This observation aligned with the trends noted in the first simulation study. However, it is noteworthy that, compared to its performance in the first simulation experiment, the **bootstrapping** method demonstrated a considerable improvement in overall power of detection under these extreme conditions, reaching a comparable level with **DESeq2**. Overall, the power of the **bootstrapping** method in detecting differentially expressed genes becomes more pronounced and reliable in more realistic experimental conditions.

In the comparison results displayed in Figure 29, both our **bootstrapping** method and **DESeq2** exhibited more conservative behavior than other methods when sample size is limited. This conservative approach implies that despite exhibiting lower power, these methods were more adept at controlling false positives. Essentially, they prioritize specificity over power, reducing the likelihood of false detections. Unlike **DESeq2**, where the changes in power and false positive rate are more gradual, the **bootstrapping** method exhibits a rapid increase in both power and control over false positive rates as the sample size becomes larger. When working with sufficient sample sizes, the performance of the **bootstrapping** method aligns closely with that of **edgeR** and **Limma-Voom**. This indicates that while the **bootstrapping** method maintains a conservative approach in limited sample scenarios, it quickly adapts and improves in efficacy with the

availability of more data. This adaptability is a key strength of the **bootstrapping** method, allowing it to provide reliable results across a range of experimental conditions.

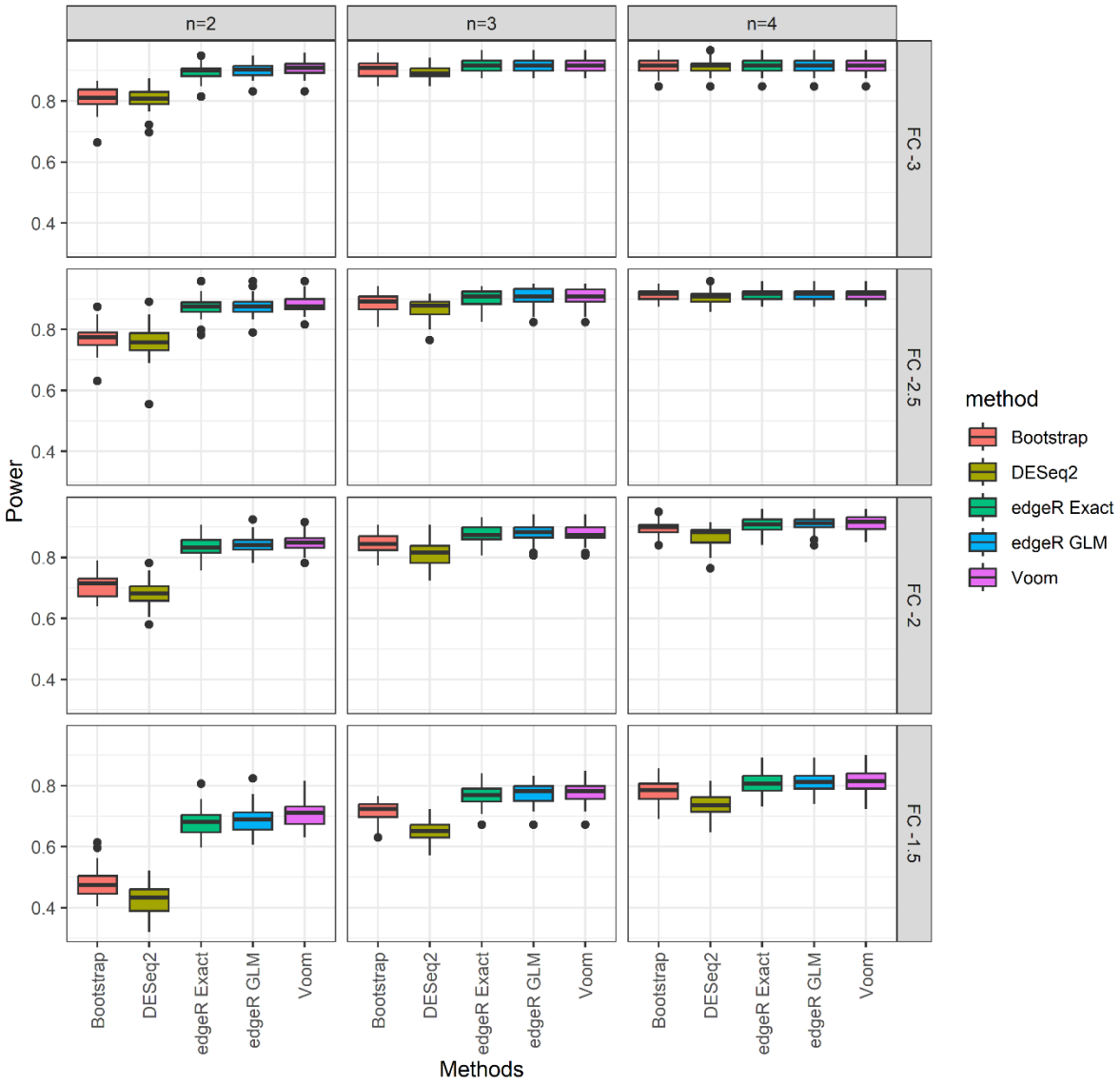

Figure 27 Power of Algorithms in Simulation Studies with Negative FC: *This figure illustrates the results of a simulation study evaluating the efficacy of different algorithms in scenarios where all fold changes are present. It focuses on showcasing the algorithms' ability to detect differentially expressed genes (DEGs) with low expression levels in the treatment group (FC < 0). Central to this evaluation is the comparison of our proposed bootstrapping method against other existing algorithms. The boxplot in the figure depicts the power results obtained from 50 simulation runs. Here, power is defined as the proportion of genes correctly identified as differentially expressed (with an Benjamini-Hochberg (BH) adjusted p-value less than 0.05) among the genes that truly have differences between the group means.*

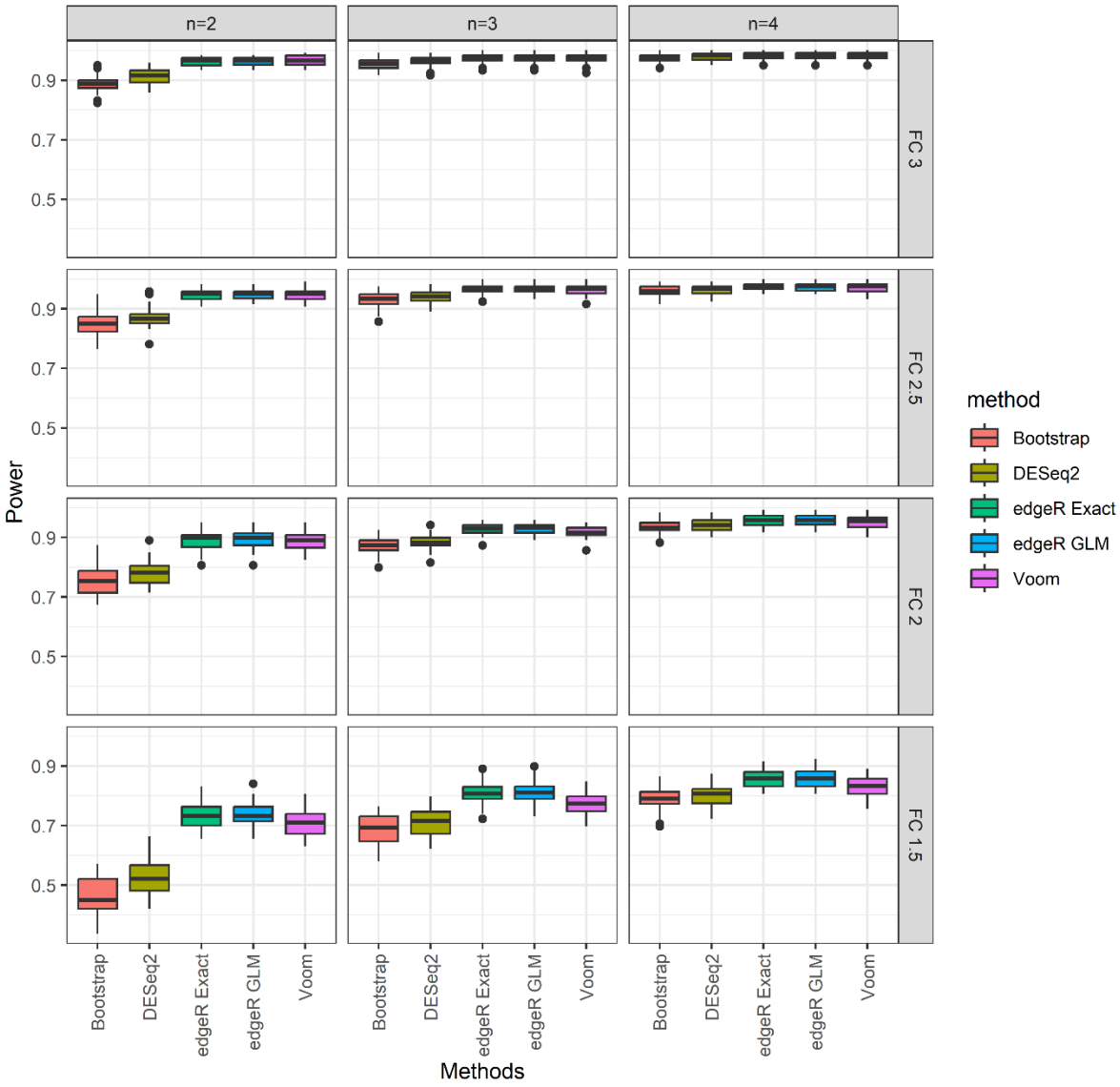

Figure 28 Power of Algorithms in Simulation Studies with Positive FC: *This figure illustrates the results of a simulation study evaluating the efficacy of different algorithms in scenarios where all fold changes are present. It focuses on showcasing the algorithms' ability to detect differentially expressed genes (DEGs) with high expression levels in the treatment group (FC > 0). Central to this evaluation is the comparison of our proposed bootstrapping method against other existing algorithms. The boxplot in the figure depicts the power results obtained from 50 simulation runs. Here, power is defined as the proportion of genes correctly identified as differentially expressed (with an Benjamini-Hochberg (BH) adjusted p-value less than 0.05) among the genes that truly have differences between the group means.*

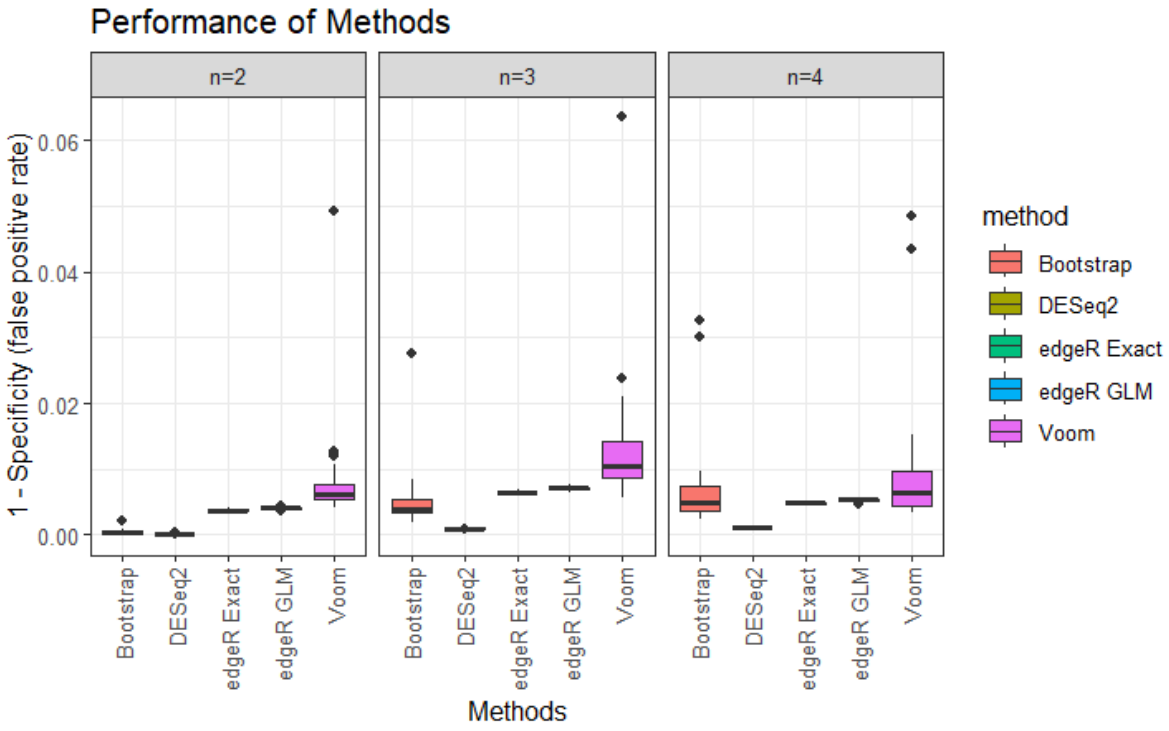

Figure 29 FPR of Algorithms in Simulation Studies: *This figure presents the outcomes of a simulation study designed to assess the false positive rate of various algorithms within a dataset that all fold changes are present. Central to this evaluation is the comparison of our proposed bootstrapping method against other existing algorithms. The boxplot in the figure depicts the power results obtained from 50 simulation runs. In this context, the false positive rate is defined as the proportion of genes incorrectly identified as differentially expressed (with an Benjamini-Hochberg (BH) adjusted p-value less than 0.05) among the genes that truly have differences between the group means. The results demonstrate that, in the vast majority of cases, our bootstrapping method consistently maintains a lower false positive rate compared to* ***edgeR*** *and* ***Limma-Voom****. This overall trend underscores the reliability and effectiveness of our bootstrapping approach in controlling false positives in most situations.*

Outlier sensitivity

The presence of outliers, which are often introduced by errors in the sequencing process, affects the overall DGE analysis. To address this issue, we conducted an additional test. Building upon the dataset created in the second simulation experiment, we intentionally introduced outliers.

These outliers were generated by randomly selecting single counts in genes and amplifying them 100 times. We then evaluated the performance of our **bootstrapping** method, as well as commonly used methods, under varying conditions of outlier presence: no outliers, 50, 100, and 150 outliers. These tests were conducted for sample sizes of 2, 3, and 4. To enhance the robustness of our results, we averaged the outcomes of 10 tests for each setting. Both power and false positive rate were calculated based on the thresholding of p-values. Figure 30 displays the power-false positive rate graph derived from this analysis, with triangle points marking where adjusted p-values reach the threshold of 0.05.

From the figure, we found the **bootstrapping** method, behaved similar to **Limma-Voom**, had lower control of false positive rate in scenarios with a significant number of outliers and limited replicates. However, as the sample size increased, the bootstrapping method's capability to identify and mitigate the influence of outliers improved. This enhancement led to stronger performance in DGE analysis, balancing detection power with a well-managed false positive rate.

Moreover, the triangle points on the graph confirmed that an adjusted p-value threshold of 0.05 was an effective choice. While increasing the p-value threshold slightly enhances the power in identifying DEGs, it significantly raises the false positive rate in the meantime. Such tendency was observed in both circumstances of with and without outliers, reinforcing a threshold of 0.05 strikes a good balance between sufficient detection power and good false positive control in most real-world situations.

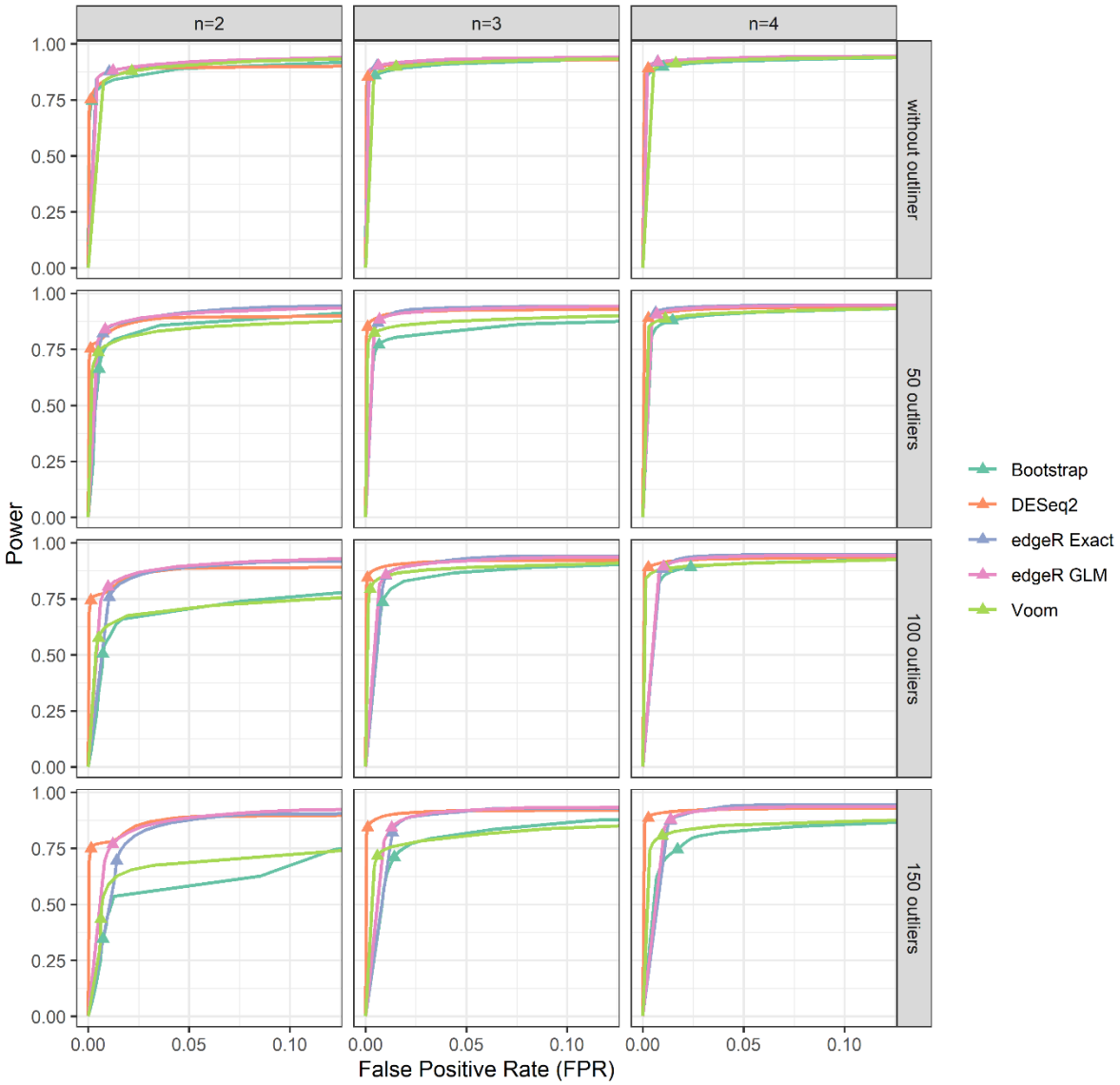

Figure 30 Outlier handling: Power-FPR Curves of Algorithms Across Sample Size: *This figure illustrates the trade-off between power and false positive rate (FPR) in DGE analysis in the context of outlier presence. The dataset was created based on the dataset generated in simulation experiment 2, with adding outliers with the number of 0, 50, 100 and 150. The power and false positive rate were calculated using the threshold of Benjamini-Hochberg (BH) adjusted p-values. Triangle points on the graph indicate the position where adjusted p-value of 0.05 was used as the threshold. The analysis was conducted with initial sample sizes of 2, 3, and 4, and each combination of sample size and number of outliers was tested ten times to ensure the robustness and reliability of the results.*

4.2 Real-world testing

Although previous simulation experiments with real-world cell-line data are useful in assessment of algorithm behaviour, the fixed fold changes we used might not always align with the complexities of other real-world RNA-Seq datasets, and hence cannot fully inform us about how accurately the algorithms reflect real-world scenarios. However, we the lack of ability to know the underlying truth of RNA-Seq data in real world. Under this circumstance, Love et al. proposed an indirect method that circumvents this limitation. This method under the proxy of experimental reproducibility on independent samples a condition that generally holds true in our data. Thus, we adopted a similar strategy, but customized the method to suit our data, with the specific workflow being illustrated in Figure 31 (Love et al., 2014).

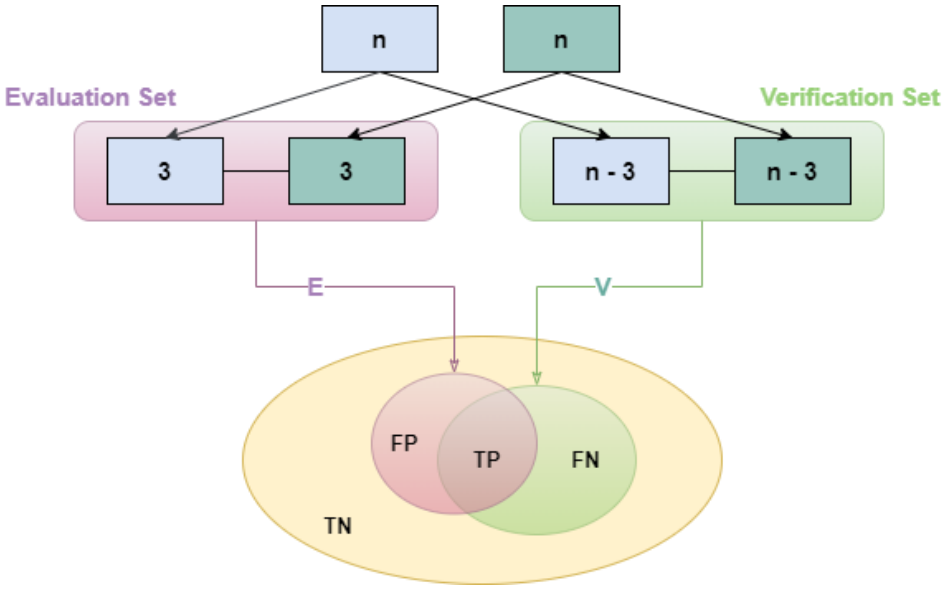

Figure 31 Diagram of the Evaluation Scheme Using Real RNA-Seq Data : *This figure illustrates the evaluation protocol implemented on the real RNA-Seq dataset (Rinas et al., 2020). The dataset was divided into two sets: an "evaluation set" with 3 replicates per group and a "verification set" with (n-3) replicates per group. The genes identified as differentially expressed in the verification set, denoted as set V, were considered the 'true' differentially expressed genes as (n-3) is usually large enough to give near-to-truth results. Subsequently, the performance of various algorithms was assessed based on their ability to identify differentially expressed genes in the evaluation set (set E), in comparison to the 'gold-standard' differential expression calls from set V. The power of each algorithm was computed using the formula TP/(TP+FN) and the false positive rate was calculated as FP/(FP+TN). The evaluation involved comparing each algorithm's differential expression results in the evaluation set against their results in the verification set, thereby providing a thorough assessment of each algorithm's accuracy and reliability in real-world RNA-Seq data analysis.*

Trastuzumab dataset

In the experiments setting of Trastuzumab dataset (Rinas et al., 2020), both control and Herceptin groups have sample size of 10. We repeatedly divided each group into two sets: an evaluation set consisting of three samples and a verification set containing the remaining seven samples for

each group. This division ensures that the verification set is sufficiently large to mitigate stochastic errors in the experiment, as the results from this set are considered the 'truth' for this analysis. The performance of different algorithms was then evaluated by comparing the results from the evaluation set against this 'truth' derived from the verification set.

In our study, we tested our proposed **bootstrapping** method alongside common algorithms—**edgeR**, **DESeq2**, and **Limma-Voom**—using a specific evaluation setup. Each algorithm took turns in the DGE analysis of the verification set to identify the "true" DEGs. For each algorithm's verification set, the evaluation set was then tested to assess performance. Compared to standard methods, this approach is advantageous as it does not inherently favor or disfavor any particular algorithm. We repeated this process of initial splitting, rotation, and evaluation 30 times to ensure robustness. The results of these assessments, focusing on two key criteria—power and false positive rates—were presented in Figure 32 and Figure 33.

The power of each test was calculated by the proportion of genes identified as differentially expressed (with an adjusted p-value less than 0.05) in the evaluation set among those identified as differentially expressed in the verification set. In Figure 32, we observed that the ranking of algorithms was generally consistent, regardless of which algorithm was used to determine the "true" DEGs in the verification set. Notably, our **bootstrapping** method showed high consistency with the **Limma-Voom** method and had comparable power to the **edgeR** method. Except for **DESeq2**, all other methods demonstrated a power range between 0.6 to 0.8, which is considered high, regarding the limited sample size tested on ($\boldsymbol{n = 3}$).

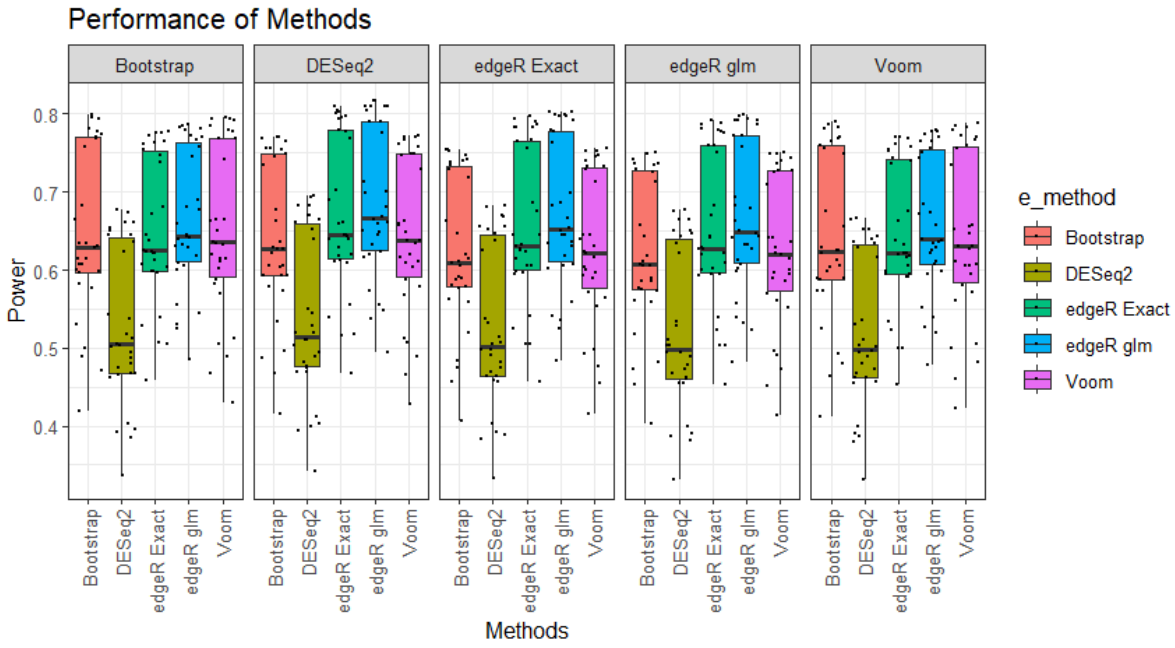

Figure 32 Estimated Power of Algorithms Using Trastuzumab Dataset: *This figure presents the outcomes of an evaluation study designed to assess the power of various algorithms within the Trastuzumab dataset (Rinas et al., 2020). Central to this evaluation is the comparison of our proposed bootstrapping method against other existing algorithms. The boxplot in the figure depicts the power results obtained from 30 evaluation runs, where algorithm's power in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (grey panels). The findings reveal that both our* ***bootstrap*** *method exhibit comparable power to both* ***edgeR*** *and* ***Limma-Voom****, highlighting its efficacy in the context of RNA-Seq data analysis.*

The second crucial criterion in our assessment is the false positive rate. In this context, the false positive rate was defined as the proportion of genes incorrectly identified as differentially expressed (with an adjusted p-value less than 0.05) among those genes that do not exhibit "true" differences. Again, the true" differential expression was defined as the DGE analysis results from the verification set. The false positive rate estimates were presented in Figure 33. This figure also proved the consistent performance of our **bootstrapping** method and the **Limma-Voom** method. While **DESeq2** displays the lowest false positive rate among the algorithms, the false positive

rates for the bootstrapping method were predominantly below 0.125. Given the limited sample size of 3 in this study, such a false positive rate is considered low and indicative of the method's accuracy in avoiding false detections of differential expression.

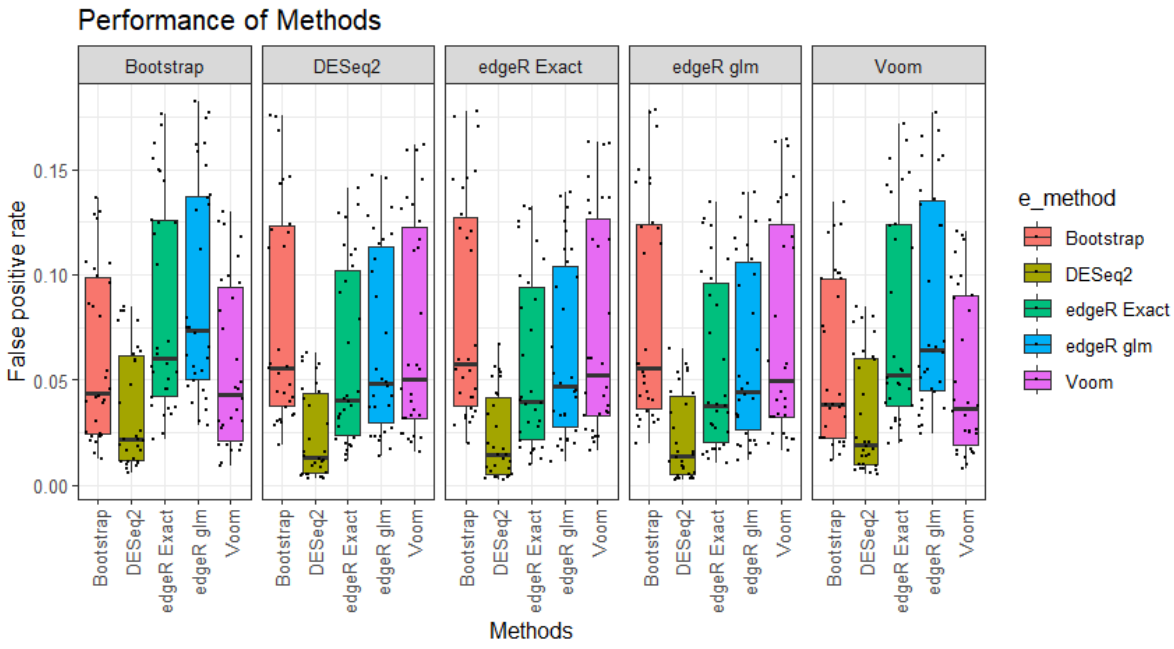

Figure 33 Estimated FPR of Algorithms Using Trastuzumab Dataset : *This figure presents the outcomes of an evaluation study designed to assess the false positive rate of various algorithms within Trastuzumab dataset (Rinas et al., 2020). Central to this evaluation is the comparison of our proposed bootstrapping method against other existing algorithms. The boxplot in the figure depicts the false positive rate obtained from 30 evaluation runs, where algorithm's false positive rate in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (panels with grey label). The findings reveal our bootstrapping method generally maintains a false positive rate lower than 0.125, showcasing its effectiveness in controlling false positive rate.*

Saccharomyces cerevisiae

Although the proposed **bootstrapping** method showed comparable or even superior results to other common methods in differential gene expression of human breast cancer, it is also

important to validate its robustness under other circumstances. To this end, we selected three additional datasets, each with distinct characteristics compared to the Trastuzumab data. These datasets, summarized in Table 3, were derived from mRNA sequencing of different species. The genetic differences and evolutionary divergence among these species could potentially influence gene expression and regulation, thereby impacting the performance of differential expression analysis.

One of the dataset, collected by (Robson& Zambrano, 2018) was designed to identify the differential expressed genes of H4T80A mutants vs wild-type (WT) histone protein H4 in Saccharomyces cerevisiae. In histones, specific amino acid residues can be post-translationally modified, affecting its chromatin structure and gene expression. The "T80A" notation suggests that the threonine (T) at position 80 in the histone H4 protein was mutated to alanine (A). Investigating such mutants is pivotal for understanding the impact of this specific modification on chromatin dynamics and gene regulation.

The dataset contains three independent colonies in each condition. With each of these biological replicates repeatedly processed four times, each condition ends up with 12 samples. The five methodologies were tested on the dataset including such information. The power and analysis of this dataset followed a similar protocol to that outlined in **Error! Reference source not found.**, but with a customized group division: a 3 vs 3 group in the evaluation set and a 9 vs 9 group in the verification set.

Figure 34 and Figure 35 shows the comparison results between these five methods. The ranking of five methodologies in power and false positive rate were consistent despite the choices of verification sets. Among them, the **bootstrapping** consistently exhibited the highest and most stable power in detecting DGEs. In terms of false positive rate (FPR), a pattern similar to what

was observed with the Herceptin data emerged. The **bootstrapping** and **Limma-voom** methods formed one group, while the other three methods formed another. Each group showed the lowest FPR when its own methods were used as the verification sets but resulted in more false positives when verified against methods from the other group. However, it was also noted that the FPRs were generally high across all methods. The highest FPR observed, which occurred only once, reached 0.4. This could be attributed to the smaller number of genes in the Saccharomyces dataset, around 3k, compared to over 10k genes in the Trastuzumab dataset (Rinas et al., 2020). The reduced number of genes in Saccharomyces made the mean-variance trend less predictable, leading to a higher FPR.

Another critical factor was the adjustment of the "window" size for selecting pooled distributions, as discussed in section 3.3.5. The results from this study used pooled distributions generated from 50 genes with similar expression levels, which yielded better results compared to a window size of 100. The selected pooled window sizes that **bootstrapping** used in these tested datasets were detailed in Table 3. Generally, the selection was set around 1% of the total number of genes in each group, with a minimum value of 50. This selection criteria aims to ensure the selected genes to be better representatives of the expected variance at this given expression level. However, it's important to note that this selection process acts as a general guideline and requires further refinement through experiments. Future research could provide more insights into the optimal window size for pooled distributions, potentially leading to even more accurate and reliable results in differential gene expression analysis using the **bootstrapping** method.

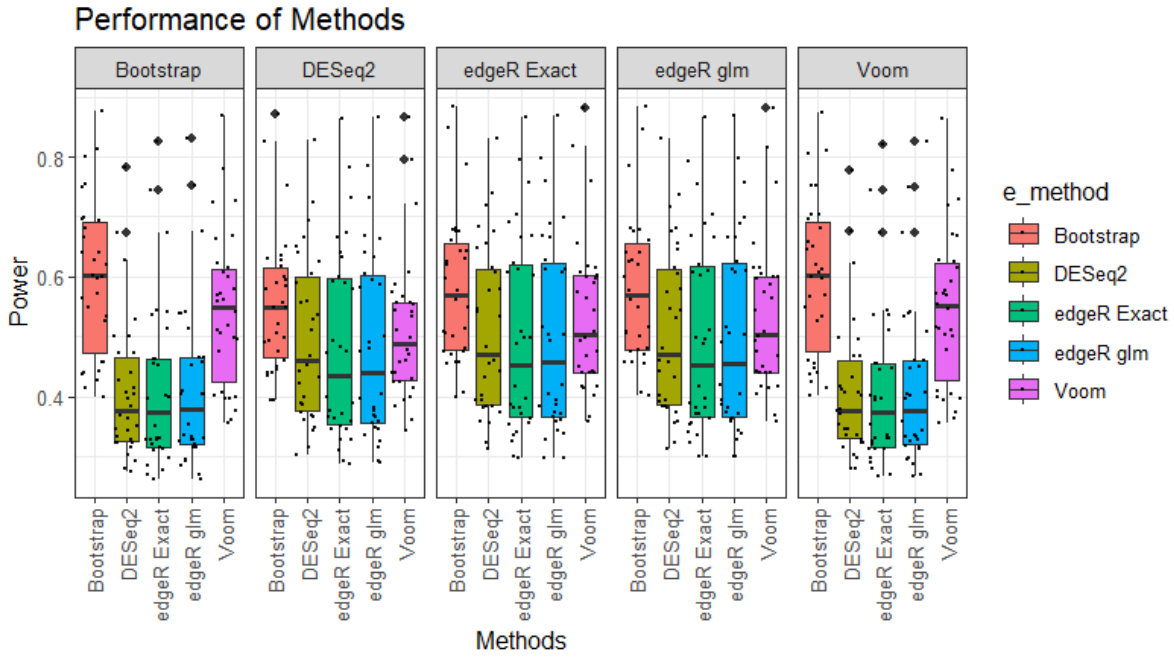

Figure 34 Estimated Power of Algorithms Using Saccharomyces Cerevisiae Dataset: *This figure presents the outcomes of an evaluation study designed to assess the power of various algorithms within Saccharomyces Cerevisiae Data (Roboson& Zambrano, 2018). The boxplot in the figure depicts the power results obtained from 30 evaluation runs, where algorithm's power in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (grey panels). The bootstrapping reached the highest power despite the selection of verification set.*

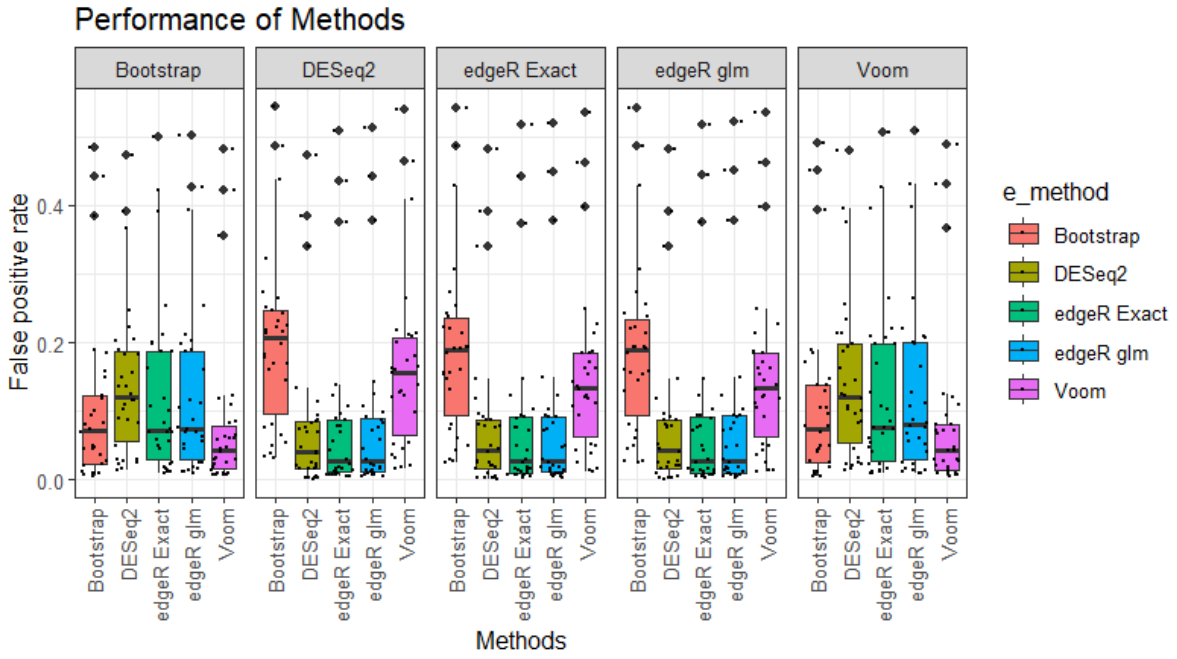

Figure 35 Estimated FPR of Algorithms Using Saccharomyces Cerevisiae Dataset: *This figure presents the outcomes of an evaluation study designed to assess the false positive rate of various algorithms within Saccharomyces Cerevisiae Data (Roboson& Zambrano, 2018). The boxplot in the figure depicts the false positive rate obtained from 30 evaluation runs, where algorithm's false positive rate in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (panels with grey label). These false positive rates were generally higher than expected.*

Mus musculus

The dataset created by Joerg Betschinger, 2021, aimed to explore the transcriptional changes underlying the defective phenotype observed in Zbtb2 mutant mouse embryonic stem cells, particularly focusing on their inability to exit the pluripotent state. Among multiple conditions exists in the study, we selected to compare the sequencing changes of wild-type mouse embryonic stem cells after they have exposed to epiblast-like cell (EpiLC) differentiation

conditions for 24 hours and 48 hours (Olivieri et al., 2021). The evaluation of DGE tools were conducted with 3 samples in evaluation set and 12 samples in verification set for each condition. The results shown in Figure 36 and Figure 37 demonstrated consistent superior performance of **bootstrapping** method in terms of power. While the false positive rate of the **bootstrapping** method was slightly higher compared to others when **DESeq2** and **edgeR** were used as verification sets, its highest value was less than 0.02 and generally stabilized around 0.013. Given that the p-value threshold was set to 0.05, this level of false positive rate is considered optimal. An interesting aspect of this dataset, which differs from the previous two datasets, is the content of differentially expressed (DE) genes is much lower – only 16% compared to around 60% in the last two datasets. As we believe, the proportion of DE genes has influence on the power of DGE analysis. A higher proportion of DE genes typically indicates stronger biological effects and therefore leads to higher detection power. Conversely, a lower proportion suggests more subtle differences, which may require more sensitive analytical methods. Despite these variations, the proposed **bootstrapping** method successfully maintained high power even when the percentage of DE genes was as low as 16%.

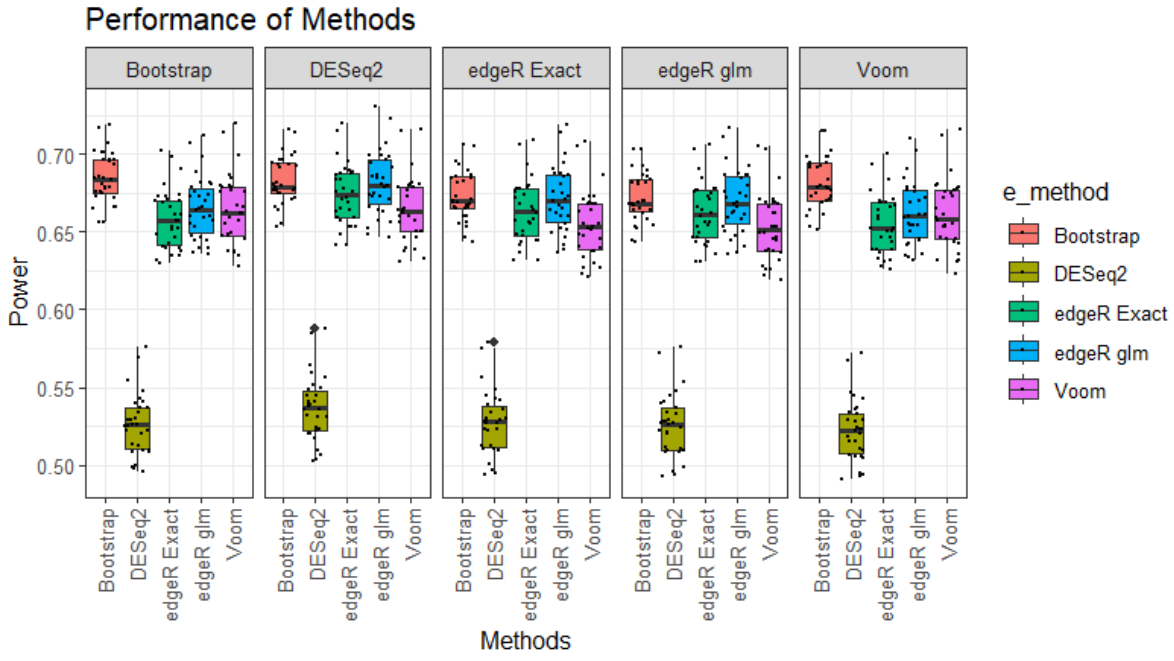

Figure 36 Estimated Power of Algorithms Using Mus Musculus Dataset: *This figure presents the outcomes of an evaluation study designed to assess the power of various algorithms within the Mus musculus dataset (Betschinger et.al, 2021). The boxplot in the figure depicts the power results obtained from 30 evaluation runs, where algorithm's power in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (grey panels). The bootstrapping performance has higher power despite the selection of verification methods.*

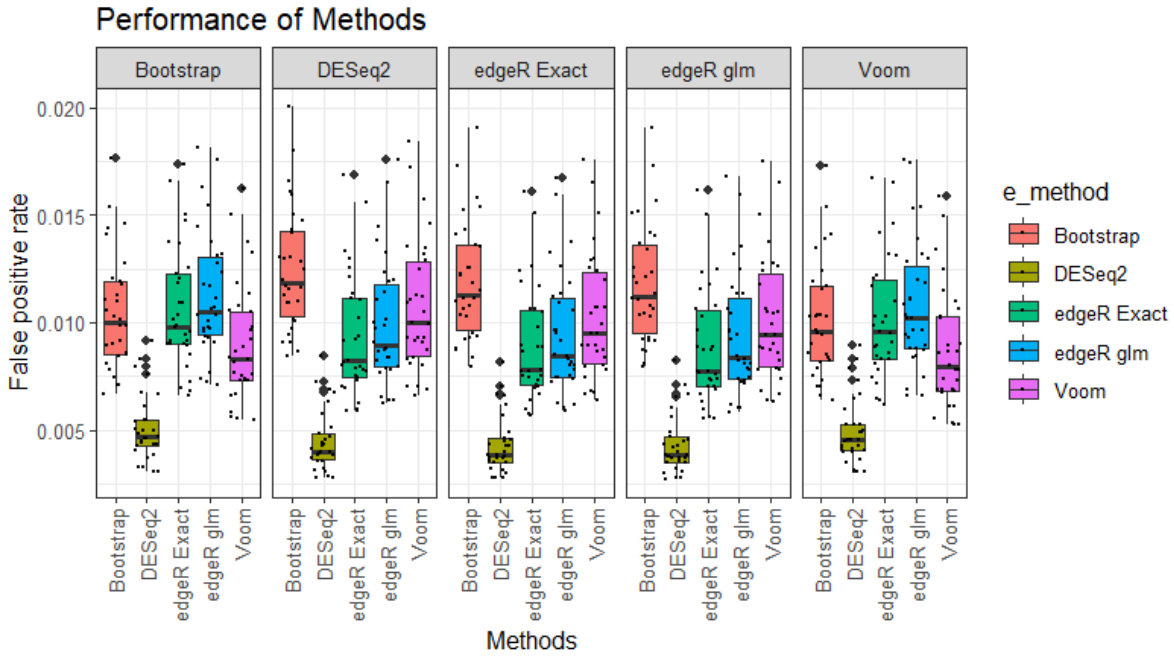

Figure 37 Estimated FPR of Algorithms Using Mus Musculus Dataset: *This figure presents the outcomes of an evaluation study designed to assess the false positive rate of various algorithms on the Mus musculus dataset (Betschinger et.al, 2021). The boxplot in the figure depicts the false positive rate obtained from 30 evaluation runs, where algorithm's false positive rate in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (panels with grey label).*

Arabidopsis lyrata

The final dataset in our comprised plant gene sequencing data from Arabidopsis lyrata, a white cress. Wos & Willi (2018) raised Arabidopsis lyrata under three types of pre-exposure treatments: bouts of frost, bouts of heat, or constant temperature in order to understand the ecological and evolutionary implication of acclimatized and non-acclimatized plants. In our analysis, we focused on the differential expression of genes resulting from a 3-week period of heat stress treatment. Despite the presence of 16 biological replicates in both the heat stress and control groups, which we initially expected would provide a robust and substantial verification

set, the results in this dataset indicated otherwise. When employing an evaluation set with a sample size of 3 and a verification set comprising the remaining 13 samples, both the **bootstrapping** and **Limma-Voom** methods showed a power close to zero (Figure 38). This outcome could be attributed to the non-uniformity in DGE analysis tools. As demonstrated in Figure 39, even with a sample size of 16, the common DGE genes only accounted for 10% of the DEGs detected in each group. Both the **DESeq2** and the proposed **bootstrapping** method exhibited a significant proportion of unique detections. This suggests a considerable variation in the genes identified as differentially expressed by each method.

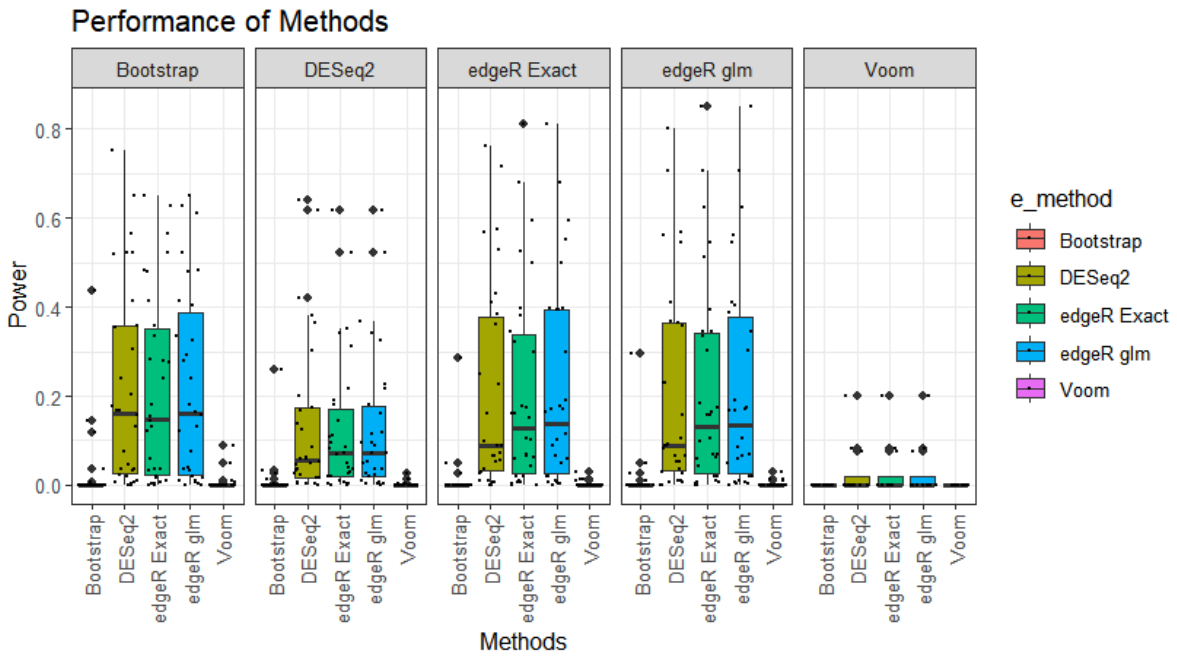

Figure 38 Estimated Power of Algorithms Using Arabidopsis Dataset: *This figure presents the outcomes of an evaluation study designed to assess the power of various algorithms within the Arabidopsis dataset (Wos & Willi, 2018). The boxplot in the figure depicts the power results obtained from 30 evaluation runs, where algorithm's power in the evaluation set (box plots) is determined by comparing its performance with the results of each other algorithm in the verification set (grey panels). The bootstrapping performance has higher power despite the selection of verification methods.*

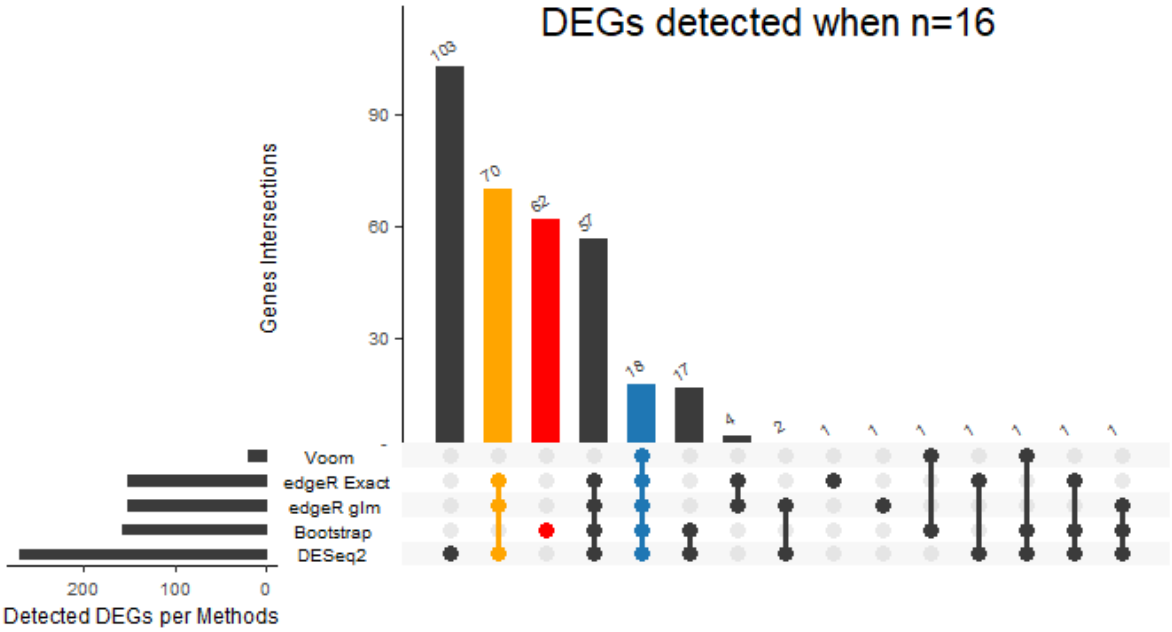

Figure 39 Detected DEGs using Arabidopsis Dataset: *This figure presents the outcomes of an evaluation study designed to assess the power of various algorithms within the Arabidopsis dataset (Wos & Willi, 2018). The Upset diagram illustrates the overlap and unique differentially expressed genes (DEGs) identified by the methodologies, using a sample size of 16. The Upset diagram uses bar graphs to show the number of detected DEGs falls into certain intersecting sections (Jake et.al, 2017). In this diagram, blue labels the commonly detected DEGs, orange labels the genes detected by* ***edgeR*** *and* ***DESeq2****, and red indicates the genes identified solely by the* ***Bootstrap*** *method.*

To further investigate this problem, we explored how the number of detected DEGs changes as the sample size increases from 3 to 16 (Figure 40). The **bootstrapping** method mostly aligned with expectations, showing an increase in detected genes corresponding to the increment in sample size. On the other hand, **edgeR** and **DESeq2** displayed an initial decrease followed by an increase in detected genes, potentially signaling an initial reduction in false positives and a subsequent rise in true positives. **Limma-Voom**, meanwhile, presented consistently conservative behaviors across the entire range of sample sizes - slowly increases but detecting only around 20 DEGs when sample size reached 16.

The observed patterns in the performance of these methodologies may contribute to the low proportion of DE genes in the Arabidopsis dataset, which is less than 1%. This outcome highlights a common challenge: lower detection power when the percentage of DE genes in the dataset is very small. This finding is a valuable aspect to investigate in the future.

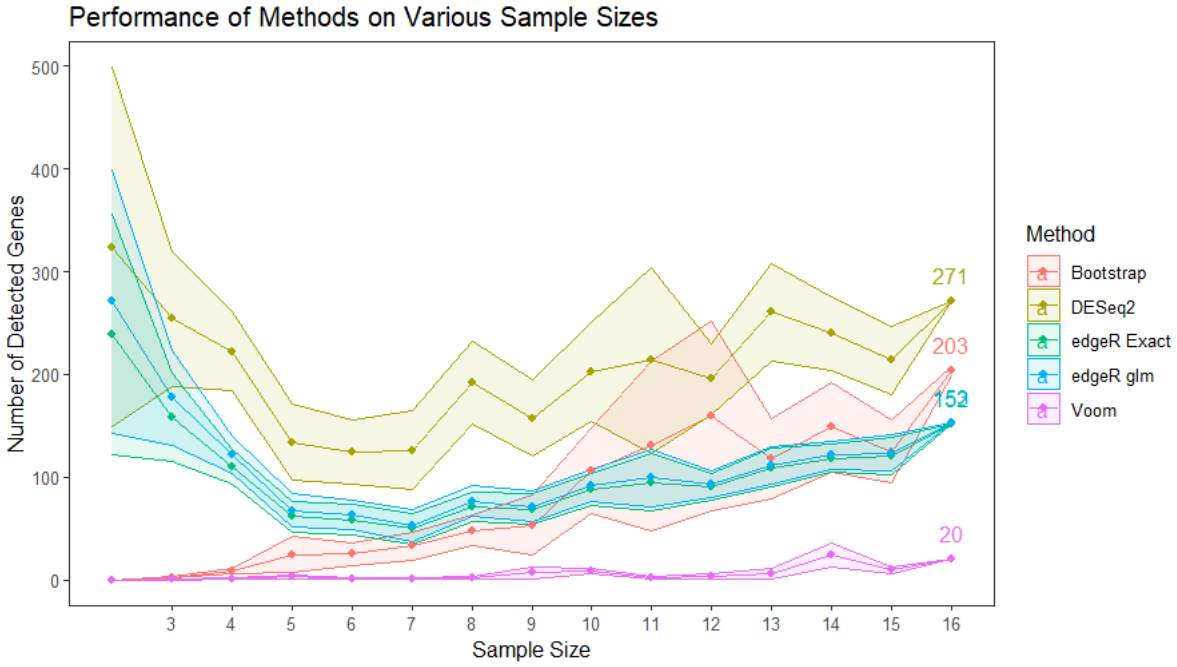

Figure 40 Trend of Detected DEGs with Sample Size Variation in Arabidopsis Dataset: *This figure presents a comprehensive evaluation of various algorithms within the Arabidopsis dataset (Wos & Willi, 2018). This line graph displays the trend of how the number of detected genes changes as the initial sample sizes vary from 3 to 16. The shaded areas in this part of the figure indicate the standard error, calculated across 20 runs, providing a visual representation of the reliability and consistency of gene detection across different sample sizes.*

Table 3 Description of Four Public RNA-Seq Datasets

Dataset	Organism	Number of samples	Median of library size (log_{10} scale)	Number of genes in each sample	"Window" size for pooled distribution	Percentage of DE genes
Rinas et al 2020	Human breast cancer cell	20	6.771832	11910	100	(61.94%-63.51%)
Robson& Zambrano 2018	Saccharomyces cerevisiae	24	6.759547	3564	50	(54.00%-60.35%)
Betschinger et al 2021	Mus musculus	30	6.549636	55573	500	(16.20%-16.54%)
Wos&Willi 2018	Arabidopsis	32	7.208839	32667	300	(0.06%-0.83%)

Chapter 5

Conclusion

The proposed **bootstrapping** method provides a comprehensive and general solution for RNA sequencing analysis, particularly suitable in situations with limited replicates. The weighted combination of bootstrapping techniques with selected pooled distribution effectively merges sample-specific information with broader environmental data, thereby stabilizing the variance estimation for genes.

One of the major strengths of using **bootstrapping** is that it does not rely on predetermined assumptions about the initial sample distribution. This approach helps avoid the potential biases that arise from discrepancies between theoretical assumptions and actual real-world data. Moreover, the flexibility of the **bootstrapping** method extends its applicability beyond RNA-Seq data; it can also be theoretically applied to microarray data. This versatility enhances the method's utility in various genomic research contexts, allowing the reintegration of tools initially developed for microarray data into contemporary genomic research.

Another significant benefit of the **bootstrapping** method is the variances were estimated separately for each gene and conditions. As stated by (Anjum et al., 2016) “A gene is declared differentially expressed if an observed difference or change in read counts or expression levels between two experimental conditions is statistically significant”. Differentially expressed genes exhibit distinct expression levels across conditions while maintaining consistent biological variability. With the earlier filtering and normalization steps reducing technical variations and measurement errors, the estimated variance would be a great representative reflecting the gene's

unique biological variability. This aspect opens up the exciting possibility: the potential use of estimated gene variances across different datasets. If feasible, this approach could utilize the vast amount of data available across numerous datasets for a single gene. Even if a gene has limited replicates in one dataset, the aggregation of data from hundreds of datasets could dramatically increase available data for that gene, leading to a more precise estimation of its variance. This cross-dataset application could significantly enhance the power and accuracy of gene expression studies.

The test results of the proposed **bootstrapping** method, along with five commonly used methodologies in both simulation and real-world databases, affirm the robustness of our approach. However, to comprehensively evaluate our method, further testing is essential in future experiments. The power to detect differential expression is influenced by various factors, such as the fraction of differentially expressed (DE) genes, the distribution of the magnitude of DE genes, the distribution of gene expression levels, sequencing coverage, and the choice of type I error control (Wu et al., 2015). Therefore, testing our method on datasets with diverse properties is essential.

One possible direction refining proposed **bootstrapping** method is the construction of the pooled distribution. Firstly, the "window" size used for creating pooled distributions could be optimized. A more effective window size might enhance the accuracy of the pooled distribution, leading to better results. Secondly, there's potential in utilizing pooled distributions leveraging the sample information of the same gene from different datasets. This approach can be considered as indirectly incorporating data from various databases, potentially augmenting the sample size.

In summary, the proposed **bootstrapping** method demonstrates performance comparable to existing methods, while also offering significant potential for further development and enhancement.

References

Alekseyev, Y. O., Fazeli, R., Yang, S., Basran, R., Maher, T., Miller, N. S., & Remick, D. (2018). A Next-Generation Sequencing Primer—How Does It Work and What Can It Do? *Academic Pathology*, *5*, 2374289518766521. https://doi.org/10.1177/2374289518766521

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*(10), R106. https://doi.org/10.1186/gb-2010-11-10-r106

Anjum, A., Jaggi, S., Varghese, E., Lall, S., Bhowmik, A., & Rai, A. (2016). Identification of Differentially Expressed Genes in RNA-seq Data of Arabidopsis thaliana: A Compound Distribution Approach. *Journal of Computational Biology*, *23*(4), 239–247. https://doi.org/10.1089/cmb.2015.0205

Baggerly, K. A., Deng, L., Morris, J. S., & Aldaz, C. M. (2003). Differential expression in SAGE: Accounting for normal between-library variation. *Bioinformatics*, *19*(12), 1477–1483. https://doi.org/10.1093/bioinformatics/btg173

Baggerly, K. A., Deng, L., Morris, J. S., & Aldaz, C. M. (2004). Overdispersed logistic regression for SAGE: Modelling multiple groups and covariates. *BMC Bioinformatics*, *5*(1), 144. https://doi.org/10.1186/1471-2105-5-144

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Bourgon, R. (2010, March 22). *Independent filtering increases detection power for high-throughput experiments*. https://doi.org/10.1073/pnas.0914005107

Bullard, J. H., Purdom, E., Hansen, K. D., & Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, *11*(1), 94. https://doi.org/10.1186/1471-2105-11-94

Chang, H. Y., & Chen, X. (2009). Chapter 51—Microarray Analysis of Stem Cells and Differentiation. In R. Lanza, J. Gearhart, B. Hogan, D. Melton, R. Pedersen, E. D. Thomas, J. Thomson, & I. Wilmut (Eds.), *Essentials of Stem Cell Biology (Second Edition)* (pp. 449–457). Academic Press. https://doi.org/10.1016/B978-0-12-374729-7.00051-2

Chen, Y., Lun, A. T. L., & Smyth, G. K. (2016). *From reads to genes to pathways: Differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline* (5:1438). F1000Research. https://doi.org/10.12688/f1000research.8987.2

Chen, Y., McCarthy, D., Ritchie, M., Robinson, M., & Smyth, G. (2022). *edgeR: differential analysis of sequence read count data User's Guide*.

Cobb, M. (2015). Who discovered messenger RNA? *Current Biology*, *25*(13), R526–R532. https://doi.org/10.1016/j.cub.2015.05.032

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., & Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology*, *17*(1), 13. https://doi.org/10.1186/s13059-016-0881-8

Cook, K. B., Hughes, T. R., & Morris, Q. D. (2015). High-throughput characterization of protein–RNA interactions. *Briefings in Functional Genomics*, *14*(1), 74–89. https://doi.org/10.1093/bfgp/elu047

Efron, B. (1979). *Bootstrap Methods: Another Look at the Jackknife*.

Feng, C., Hongyue, W., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. (2014). Log-transformation and its implications for data analysis. *Shanghai Archives of Psychiatry*, *26*, 105–109. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009

Freund, R. J., Wilson, W. J., & Mohr, D. L. (2010). CHAPTER 6—Inferences for Two or More Means. In R. J. Freund, W. J. Wilson, & D. L. Mohr (Eds.), *Statistical Methods (Third Edition)* (pp. 245–320). Academic Press. https://doi.org/10.1016/B978-0-12-374970-3.00006-8

Giani, A. M., Gallo, G. R., Gianfranceschi, L., & Formenti, G. (2020). Long walk to genomics: History and current approaches to genome sequencing and assembly. *Computational and Structural Biotechnology Journal*, *18*, 9–19. https://doi.org/10.1016/j.csbj.2019.11.002

Givan, S. A., Bottoms, C. A., & Spollen, W. G. (2012). Computational Analysis of RNA-seq. In H. Jin & W. Gassmann (Eds.), *RNA Abundance Analysis: Methods and Protocols* (pp. 201–219). Humana Press. https://doi.org/10.1007/978-1-61779-839-9_16

Govindarajan, R., Duraiyan, J., Kaliyappan, K., & Palanisamy, M. (2012). Microarray and its applications. *Journal of Pharmacy & Bioallied Sciences*, *4*(Suppl 2), S310–S312. https://doi.org/10.4103/0975-7406.100283

Guigó, R. (2013). Chapter 2—The Coding and the Non-coding Transcriptome. In A. J. M. Walhout, M. Vidal, & J. Dekker (Eds.), *Handbook of Systems Biology* (pp. 27–41). Academic Press. https://doi.org/10.1016/B978-0-12-385944-0.00002-2

He, B., Zhu, R., Yang, H., Lu, Q., Wang, W., Song, L., Sun, X., Zhang, G., Li, S., Yang, J., Tian, G., Bing, P., & Lang, J. (2020). Assessing the Impact of Data Preprocessing on Analyzing Next Generation Sequencing Data. *Frontiers in Bioengineering and Biotechnology*, *8*. https://www.frontiersin.org/articles/10.3389/fbioe.2020.00817

Heather, J. M., & Chain, B. (2016). The sequence of sequencers: The history of sequencing DNA. *Genomics*, *107*(1), 1–8. https://doi.org/10.1016/j.ygeno.2015.11.003

Hong, M., Tao, S., Zhang, L., Diao, L.-T., Huang, X., Huang, S., Xie, S.-J., Xiao, Z.-D., & Zhang, H. (2020). RNA sequencing: New technologies and applications in cancer research. *Journal of Hematology & Oncology*, *13*(1), 166. https://doi.org/10.1186/s13045-020-01005-x

Huang, H.-C., Niu, Y., & Qin, L.-X. (2015). Differential Expression Analysis for RNA-Seq: An Overview of Statistical Methods and Computational Software. *Cancer Informatics*, *14*(Suppl 1), 57–67. https://doi.org/10.4137/CIN.S21631

Hurd, P. J., & Nelson, C. J. (2009). Advantages of next-generation sequencing versus the microarray in epigenetic research. *Briefings in Functional Genomics*, *8*(3), 174–183. https://doi.org/10.1093/bfgp/elp013

Jamuar, S. S., D'Gama, A. M., & Walsh, C. A. (2016). Chapter 12—Somatic Mosaicism and Neurological Diseases. In T. Lehner, B. L. Miller, & M. W. State (Eds.), *Genomics, Circuits, and Pathways in Clinical Neuropsychiatry* (pp. 179–199). Academic Press. https://doi.org/10.1016/B978-0-12-800105-9.00012-3

Joerg Betschinger Joerg Betschinger Email: joerg.betschinger@fmi.ch Role: data analyst;investigator Affiliation: Friedrich Miescher Institute for Biomedical Research, P. P. P. P. E. panagiotis. papasaikas@fmi. ch R. submitter A. F. M. I. for B. R., Joerg Betschinger, Daniel Olivieri Daniel Olivieri Email: daniel. olivieri@fmi. ch Role: experiment performer;investigator Affiliation: Friedrich Miescher Institute for Biomedical Research, Daniel Olivieri, & Papasaikas, P. (2021). *RNA-seq for wild-type and Zbtb2 knock-out clones of mESCs in 2iLIF, EpiLC and SerumLIF conditions*. https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-9797

Koch, C. M., Chiu, S. F., Akbarpour, M., Bharat, A., Ridge, K. M., Bartom, E. T., & Winter, D. R. (2018). A Beginner's Guide to Analysis of RNA Sequencing Data. *American Journal of Respiratory Cell and Molecular Biology*, *59*(2), 145–157. https://doi.org/10.1165/rcmb.2017-0430TR

Korthauer, K., Kimes, P. K., Duvallet, C., Reyes, A., Subramanian, A., Teng, M., Shukla, C., Alm, E. J., & Hicks, S. C. (2019). A practical guide to methods controlling false discoveries in computational biology. *Genome Biology*, *20*(1), 118. https://doi.org/10.1186/s13059-019-1716-1

Kukurba, K. R., & Montgomery, S. B. (2015). RNA Sequencing and Analysis. *Cold Spring Harbor Protocols*, *2015*(11), 951–969. https://doi.org/10.1101/pdb.top084970

Law, C. W., Alhamdoosh, M., Su, S., Dong, X., Tian, L., Smyth, G. K., & Ritchie, M. E. (2018). RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR. *F1000Research*, *5*, ISCB Comm J-1408. https://doi.org/10.12688/f1000research.9005.3

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, *15*(2), R29. https://doi.org/10.1186/gb-2014-15-2-r29

Li, Y., Ge, X., Peng, F., Li, W., & Li, J. J. (2022). Exaggerated false positives by popular differential expression methods when analyzing human population samples. *Genome Biology*, *23*(1), 79. https://doi.org/10.1186/s13059-022-02648-4

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. https://doi.org/10.1186/s13059-014-0550-8

Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS Computational Biology*, *13*(5), e1005457. https://doi.org/10.1371/journal.pcbi.1005457

Lu, J., Tomfohr, J. K., & Kepler, T. B. (2005). Identifying differential expression in multiple SAGE libraries: An overdispersed log-linear model approach. *BMC Bioinformatics*, *6*(1), 165. https://doi.org/10.1186/1471-2105-6-165

Lu, Z., & Chang, H. Y. (2016). Decoding the RNA structurome. *Current Opinion in Structural Biology*, *36*, 142–148. https://doi.org/10.1016/j.sbi.2016.01.007

Marín-García, J. (2014). Chapter 2—Methodologies in the Era of Cardiovascular "Omics". In J. Marín-García (Ed.), *Post-Genomic Cardiology (Second Edition)* (pp. 15–55). Academic Press. https://doi.org/10.1016/B978-0-12-404599-6.00002-0

Marroki, A., & Bousmaha-Marroki, L. (2022). Antibiotic Resistance Diagnostic Methods for Pathogenic Bacteria. In N. Rezaei (Ed.), *Encyclopedia of Infection and Immunity* (pp. 320–341). Elsevier. https://doi.org/10.1016/B978-0-12-818731-9.00133-6

McDermaid, A., Monier, B., Zhao, J., Liu, B., & Ma, Q. (2018). Interpretation of differential gene expression results of RNA-seq data: Review and integration. *Briefings in Bioinformatics*, *20*(6), 2044–2054. https://doi.org/10.1093/bib/bby067

McIntyre, L. M., Lopiano, K. K., Morse, A. M., Amin, V., Oberg, A. L., Young, L. J., & Nuzhdin, S. V. (2011). RNA-seq: Technical variability and sampling. *BMC Genomics*, *12*(1), 293. https://doi.org/10.1186/1471-2164-12-293

Olivieri, D., Paramanathan, S., Bardet, A. F., Hess, D., Smallwood, S. A., Elling, U., & Betschinger, J. (2021). The BTB-domain transcription factor ZBTB2 recruits chromatin remodelers and a histone chaperone during the exit from pluripotency. *Journal of Biological Chemistry*, *297*(2), 100947. https://doi.org/10.1016/j.jbc.2021.100947

Poullet, M., & Orlando, L. (2020). Assessing DNA Sequence Alignment Methods for Characterizing Ancient Genomes and Methylomes. *Frontiers in Ecology and Evolution*, *8*. https://www.frontiersin.org/articles/10.3389/fevo.2020.00105

Qin, D. (2019). Next-generation sequencing and its clinical application. *Cancer Biology & Medicine*, *16*(1), 4–10. https://doi.org/10.20892/j.issn.2095-3941.2018.0055

R. Farley, A., & J. Link, A. (2009, November 3). *Chapter 40 Identification and Quantification of Protein Posttranslational Modifications—ScienceDirect*. https://www.sciencedirect.com/science/article/abs/pii/S0076687909630408

Rinas, K. (2020). *Assessing biosimilarity of Trastuzumab monoclonal antibody therapeutics using RNA sequencing* [dataset]. University of Waterloo, Dr. McConkey's Lab. Unpublished.

Rinas, M., Scrocchi, L., & McConkey, B. (2020). *Assessing biosimilarity of Trastuzumab monoclonal antibody therapeutics 3 using RNA sequencing*. 32.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, *43*(7), e47. https://doi.org/10.1093/nar/gkv007

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, *11*(3), R25. https://doi.org/10.1186/gb-2010-11-3-r25

Robinson, M. D., & Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, *23*(21), 2881–2887. https://doi.org/10.1093/bioinformatics/btm453

Robinson, M. D., & Smyth, G. K. (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, *9*(2), 321–332. https://doi.org/10.1093/biostatistics/kxm030

Ross, J. S., Slodkowska, E. A., Symmans, W. F., Pusztai, L., Ravdin, P. M., & Hortobagyi, G. N. (2009). The HER-2 Receptor and Breast Cancer: Ten Years of Targeted Anti–HER-2 Therapy and Personalized Medicine. *The Oncologist*, *14*(4), 320–368. https://doi.org/10.1634/theoncologist.2008-0230

Ruppert, D., & Matteson, D. S. (2015). Resampling. In D. Ruppert & D. S. Matteson (Eds.), *Statistics and Data Analysis for Financial Engineering: With R examples* (pp. 137–156). Springer. https://doi.org/10.1007/978-1-4939-2614-5_6

Sam Robson Sam Robson Email: samuel.robson@port.ac.uk Role: submitter Affiliation: University of Portsmouth, G. M. Z. G. M. Z. E. gm520@cam. ac. uk R. investigator A. T. W. R. U. G. I., Sam Robson, & Zambrano, G. M. (2018). *Global gene expression analysis of H4T80A mutants vs WT H4 in Saccharomyces cerevisiae*. https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-7090

Sha, Y., Phan, J. H., & Wang, M. D. (2015). Effect of low-expression gene filtering on detection of differentially expressed genes in RNA-seq data. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2015*, 6461–6464. https://doi.org/10.1109/EMBC.2015.7319872

Smyth, G. K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*(1), 1–25. https://doi.org/10.2202/1544-6115.1027

Soneson, C., & Delorenzi, M. (2013). A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, *14*(1), 91. https://doi.org/10.1186/1471-2105-14-91

Stark, R., Grzelak, M., & Hadfield, J. (2019). RNA sequencing: The teenage years. *Nature Reviews Genetics*, *20*(11), 631–656. https://doi.org/10.1038/s41576-019-0150-2

Stephens, M. (2017). False discovery rates: A new deal. *Biostatistics*, *18*(2), 275–294. https://doi.org/10.1093/biostatistics/kxw041

TachibanaJul. 31, C., 2015, & Pm, 1:15. (2015, July 31). *Transcriptomics today: Microarrays, RNA-seq, and more*. Science | AAAS. https://www.sciencemag.org/features/2015/07/transcriptomics-today-microarrays-rna-seq-and-more

Thompson, J. F., & Milos, P. M. (2011). The properties and applications of single-molecule DNA sequencing. *Genome Biology*, *12*(2), 217. https://doi.org/10.1186/gb-2011-12-2-217

Wan, Y., Kertesz, M., Spitale, R. C., Segal, E., & Chang, H. Y. (2011). Understanding the transcriptome through RNA structure. *Nature Reviews Genetics*, *12*(9), Article 9. https://doi.org/10.1038/nrg3049

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews. Genetics*, *10*(1), 57–63. https://doi.org/10.1038/nrg2484

Wilson, F. R., Coombes, M. E., Brezden-Masley, C., Yurchenko, M., Wylie, Q., Douma, R., Varu, A., Hutton, B., Skidmore, B., & Cameron, C. (2018). Herceptin® (trastuzumab) in HER2-positive early breast cancer: A systematic review and cumulative network meta-analysis. *Systematic Reviews*, *7*(1), 191. https://doi.org/10.1186/s13643-018-0854-y

Wos, G., & Willi, Y. (2018). Thermal acclimation in Arabidopsis lyrata: Genotypic costs and transcriptional changes. *Journal of Evolutionary Biology*, *31*(1), 123–135. https://doi.org/10.1111/jeb.13208

Wu, H., Wang, C., & Wu, Z. (2015). PROPER: Comprehensive power evaluation for differential expression using RNA-seq. *Bioinformatics*, *31*(2), 233–241. https://doi.org/10.1093/bioinformatics/btu640

Yee, A. J., & Ramaswamy, S. (2010). Chapter 7—DNA Microarrays in Biological Discovery and Patient Care. In G. S. Ginsburg & H. F. Willard (Eds.), *Essentials of Genomic and Personalized Medicine* (pp. 73–88). Academic Press. https://doi.org/10.1016/B978-0-12-374934-5.00007-6

You, Y., Dong, X., Wee, Y. K., Maxwell, M. J., Alhamdoosh, M., Smyth, G. K., Hickey, P. F., Ritchie, M. E., & Law, C. W. (2023). Modeling group heteroscedasticity in single-cell RNA-seq pseudo-bulk data. *Genome Biology*, *24*, 107. https://doi.org/10.1186/s13059-023-02949-2

Zhang, Z. H., Jhaveri, D. J., Marshall, V. M., Bauer, D. C., Edson, J., Narayanan, R. K., Robinson, G. J., Lundberg, A. E., Bartlett, P. F., Wray, N. R., & Zhao, Q.-Y. (2014). A Comparative Study of Techniques for Differential Expression Analysis on RNA-Seq Data. *PLOS ONE*, *9*(8), e103207. https://doi.org/10.1371/journal.pone.0103207

Zhao, S., Zhang, B., Zhang, Y., Gordon, W., Du, S., Paradis, T., Vincent, M., & Schack, D. von. (2016). Bioinformatics for RNA-Seq Data Analysis. In *Bioinformatics—Updated Features and Applications*. IntechOpen. https://doi.org/10.5772/63267

Zhu, A., Ibrahim, J. G., & Love, M. I. (2019). Heavy-tailed prior distributions for sequence count data: Removing the noise and preserving large differences. *Bioinformatics*, *35*(12), 2084–2092. https://doi.org/10.1093/bioinformatics/bty895

Appendix

Mean and variance estimates on Negative Binomial distribution

In our hypothetical population, we assume a negative binomial distribution with mean μ of 10 and a dispersion parameter size of 50. The dispersion parameter represents the shape parameter of the gamma mixing distribution. Consequently, the population has a variance of $\mu + \frac{\mu^2}{r}$, which equals 12.

From this population, we draw samples of varying sizes (n ranging from 2 to 10) to emulate experimental gene count data. We also include a sample size of 100 for a broader perspective. This sampling process is repeated 300 times to simulate a dataset containing 300 genes with similar behavior. Each "gene" within this dataset undergoes a resampling procedure in which we select n samples with replacement for 100 iterations, and then calculate the mean values from these resampled datasets. The results of this simulation study are presented below.

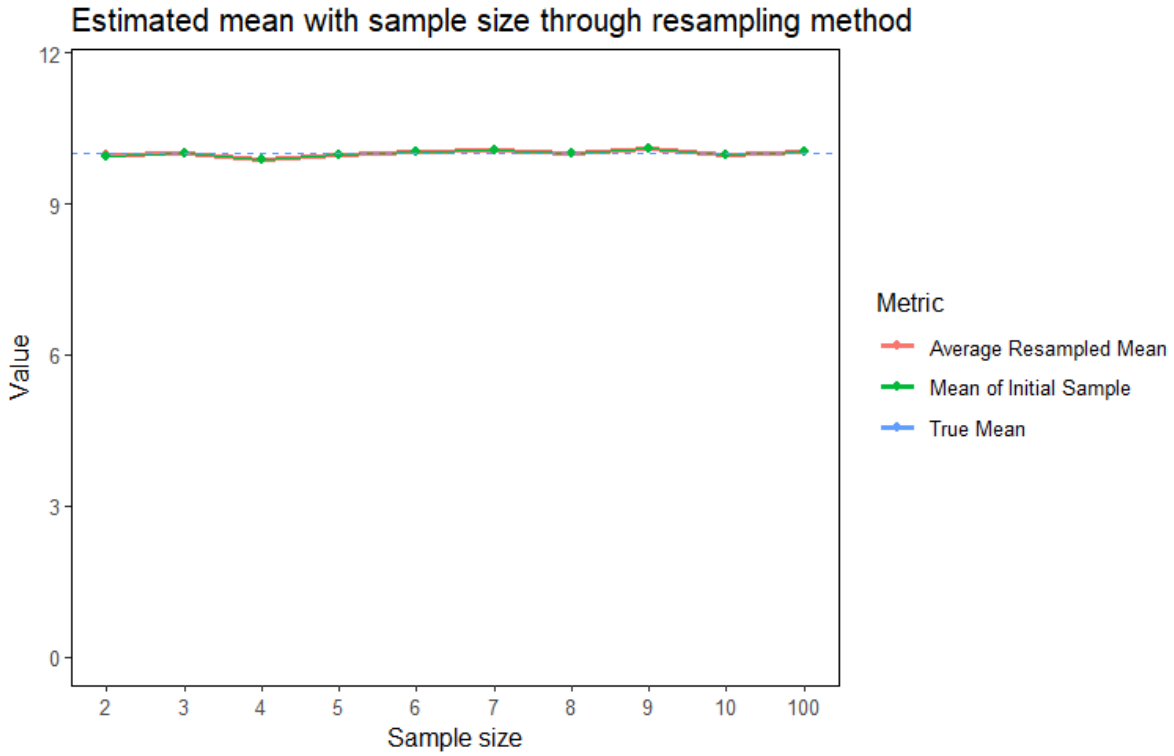

Figure S 1: *This line graph displays the estimated means across different sample sizes using the resampling method based on the negative binomial assumption.*

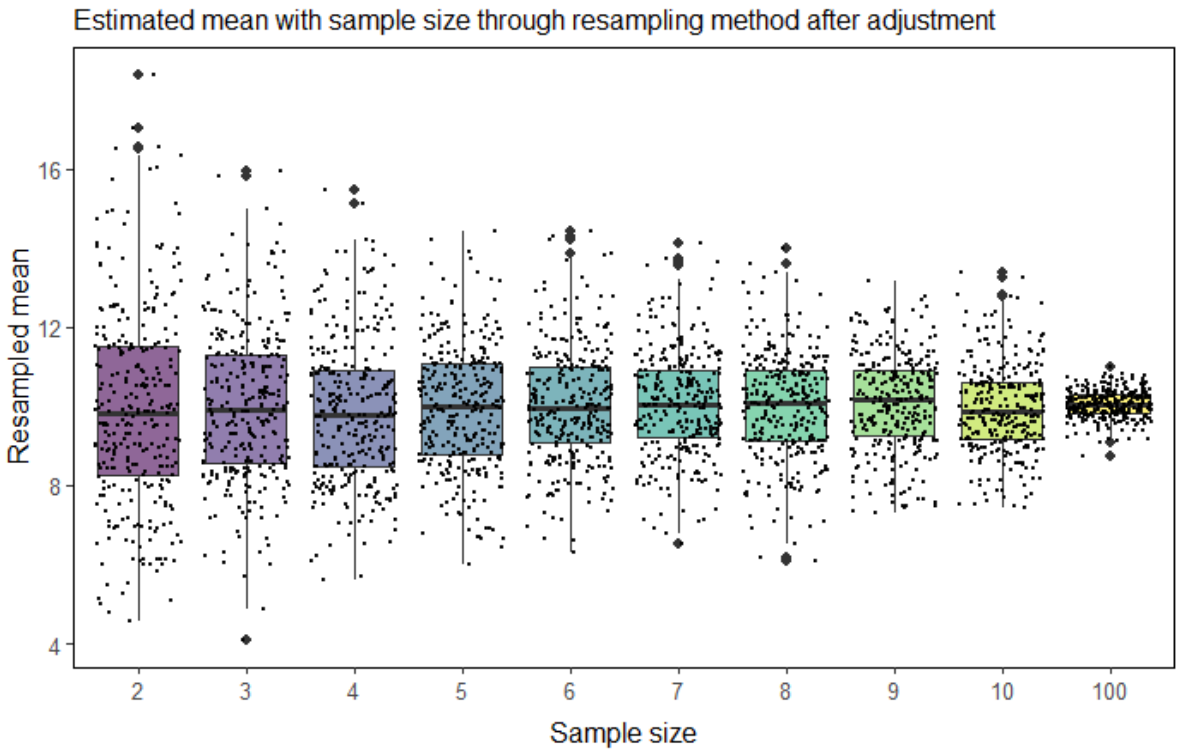

Figure S 2: *This boxplot is overlaid with jitter points, providing a visual representation of the distribution of resampled means across various initial sample sizes based on the negative binomial distribution.*

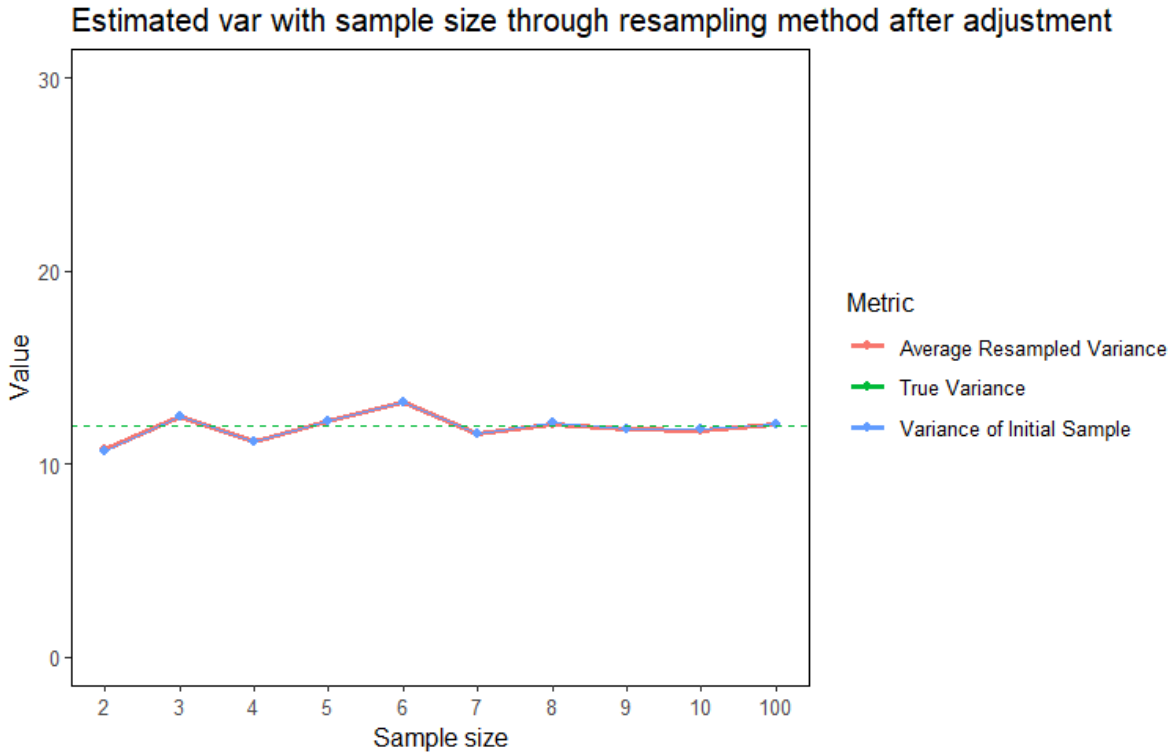

Figure S 3: *This line graph illustrates the estimated variance across different sample sizes using the resampling method after adjusting. This data was created based on the assumption of negative binomial distribution.*

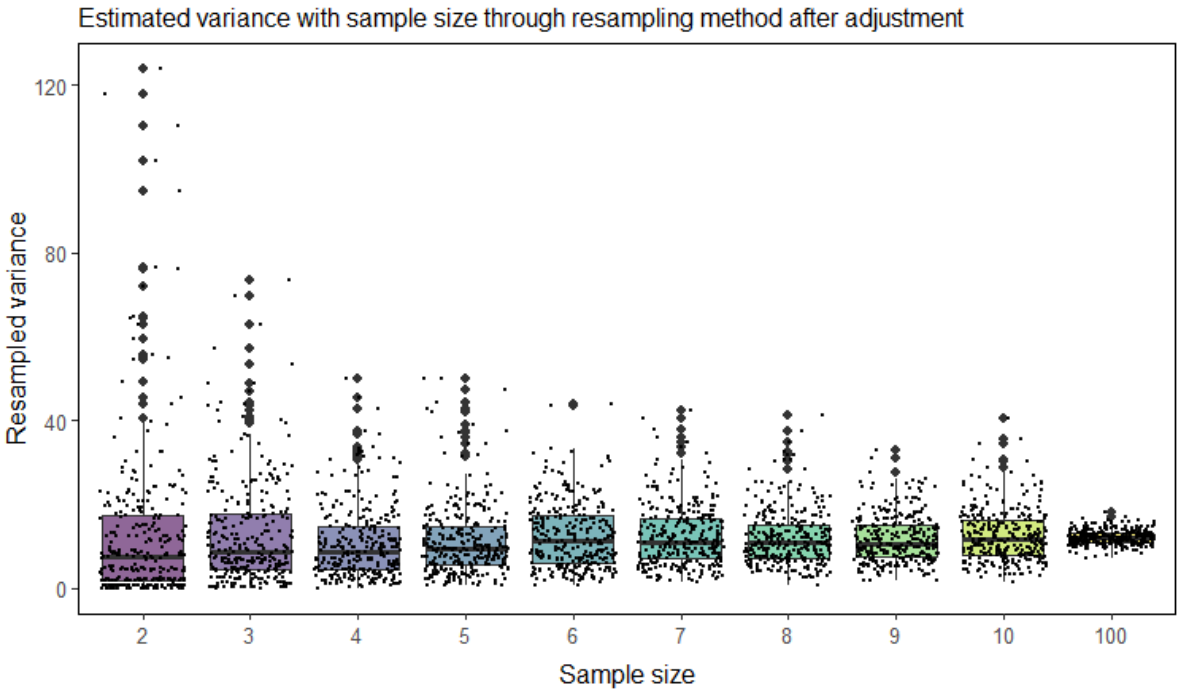

Figure S 4: *This boxplot, overlaid with jitter points, illustrates the adjusted distribution of resampled variances across varying initial sample sizes. This data was created based on the assumption of negative binomial distribution.*

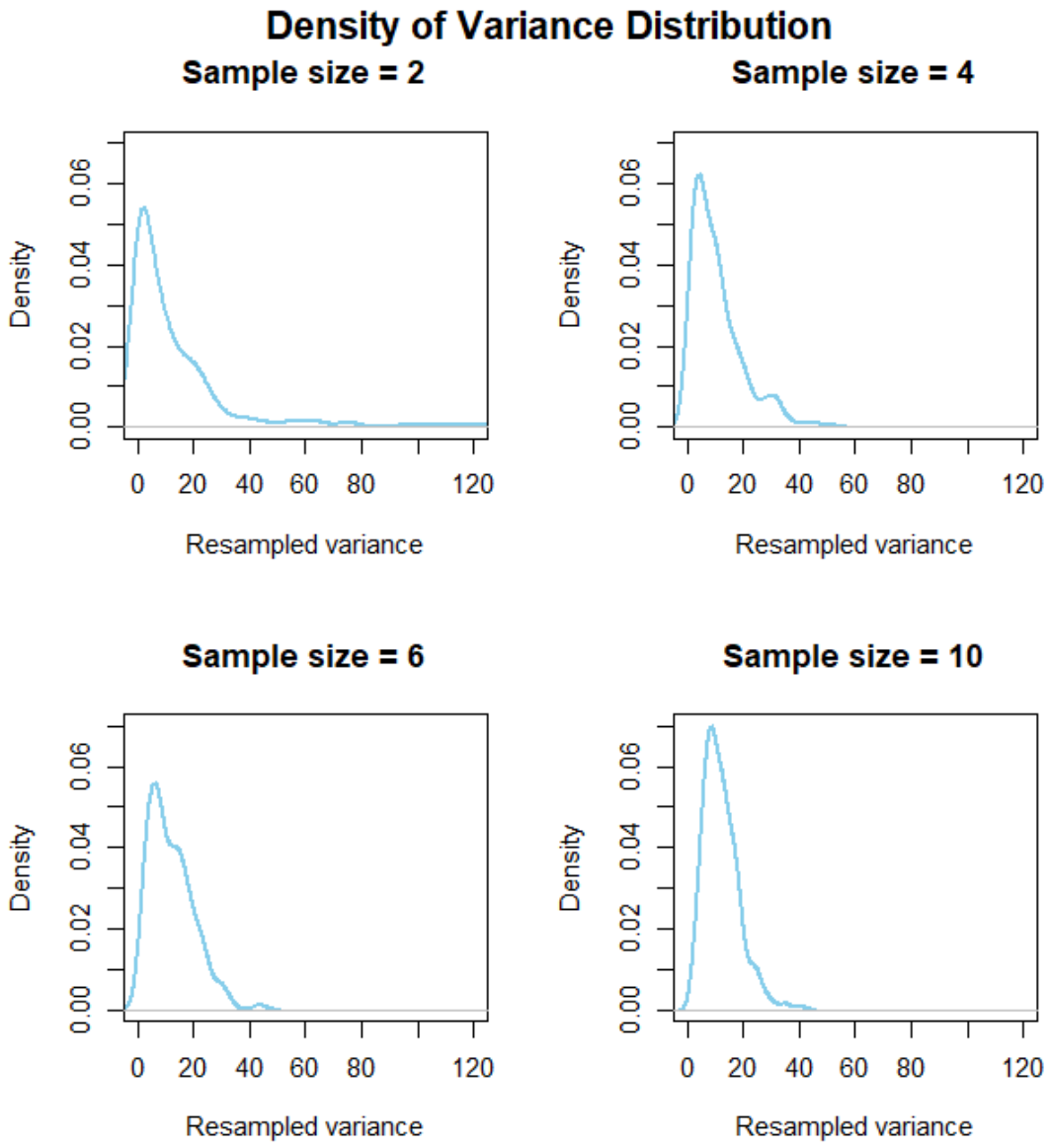

Figure S 5: *Density plots illustrate the distribution of resampled variances across various sample sizes. This data was created based on the assumption of negative binomial distribution.*

Mean and variance estimates on Uniform distribution

In our hypothetical population, we assume a uniform distribution with min of 1 and max of 25. Thus, the population has a mean $\mu = \frac{\min + \max}{2} = 13$ and variance of $\sigma^2 = \frac{(\max - \min)^2}{12} = 48$. From this population, we draw samples of varying sizes (n ranging from 2 to 10) to emulate experimental gene count data. We also include a sample size of 100 for a broader perspective. This sampling process is repeated 300 times to simulate a dataset containing 300 genes with similar behavior. Each "gene" within this dataset undergoes a resampling procedure in which we select n samples with replacement for 100 iterations, and then calculate the mean values from these resampled datasets. The results of this simulation study are presented below.

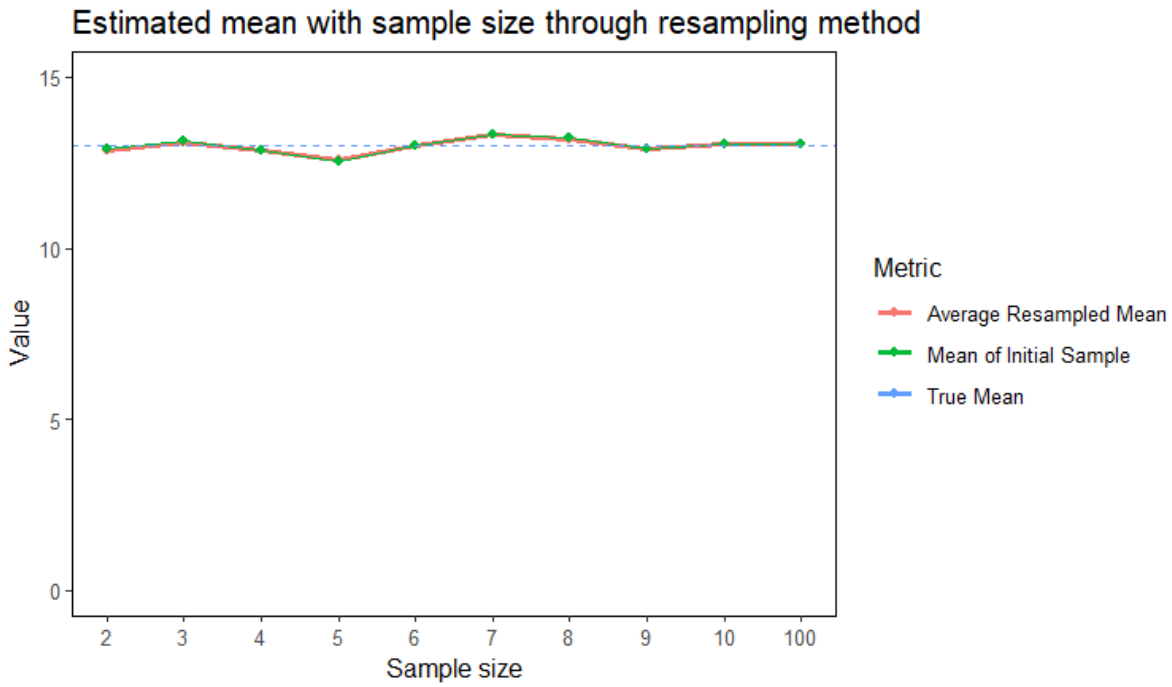

Figure S 6: *This line graph displays the estimated means across different sample sizes using the resampling method based on the assumption of normal distribution.*

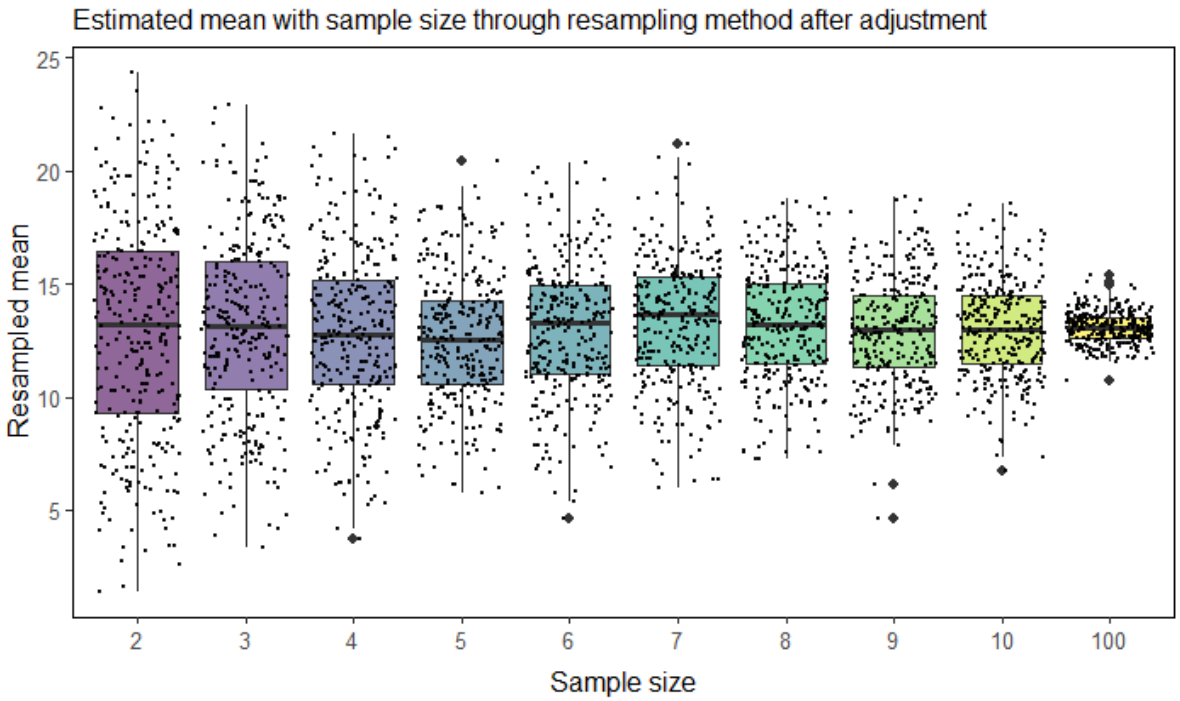

Figure S 7: *This boxplot is overlaid with jitter points, providing a visual representation of the distribution of resampled means across various initial sample sizes based on the assumption of uniform distribution.*

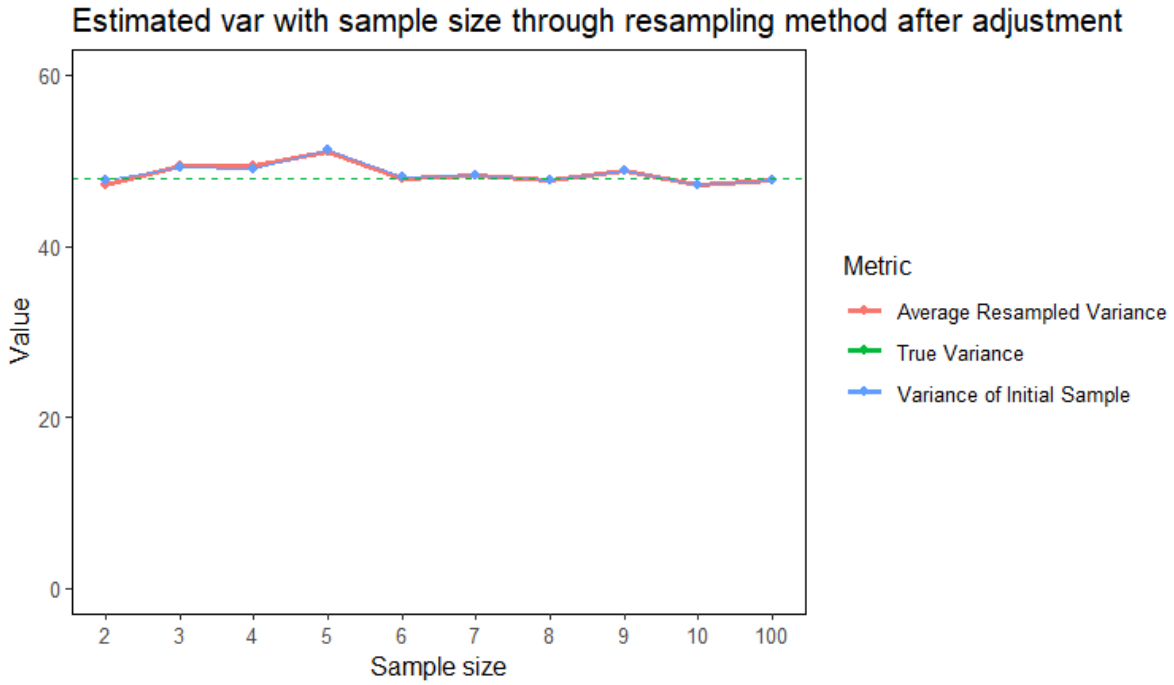

Figure S 8: *This line graph illustrates the estimated variance across different sample sizes using the resampling method after adjusting. This data was created based on the assumption of uniform distribution.*

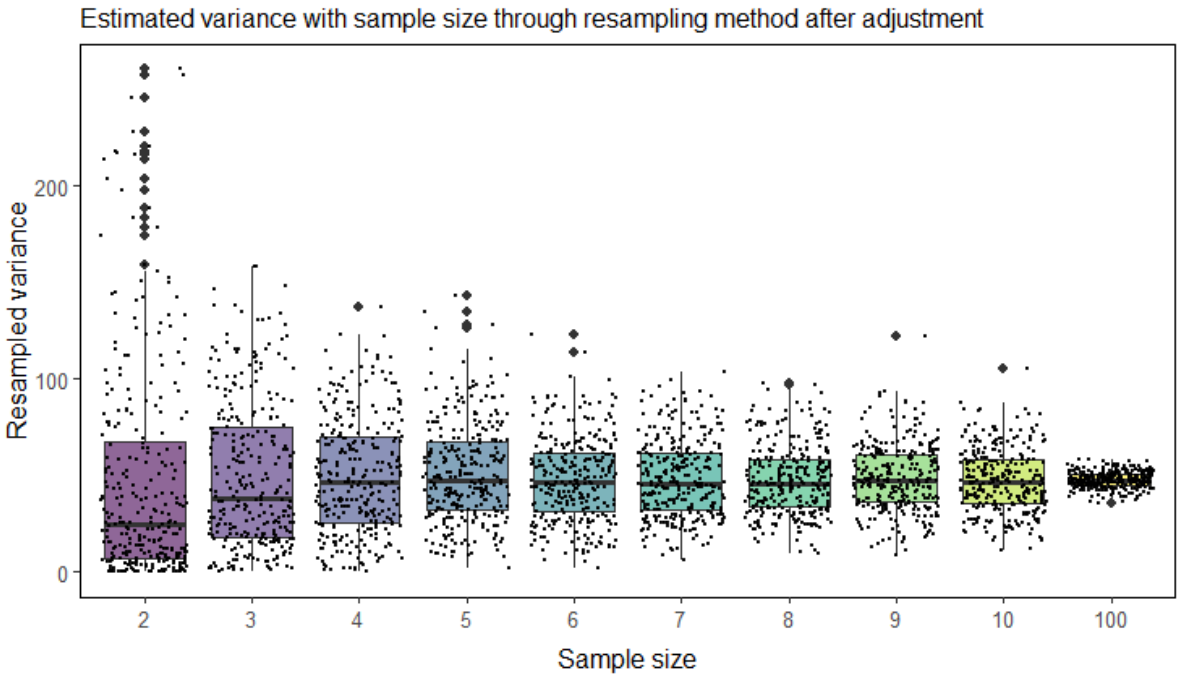

Figure S 9: *This boxplot, overlaid with jitter points, illustrates the adjusted distribution of resampled variances across varying initial sample sizes. This data was created based on the assumption of uniform distribution.*

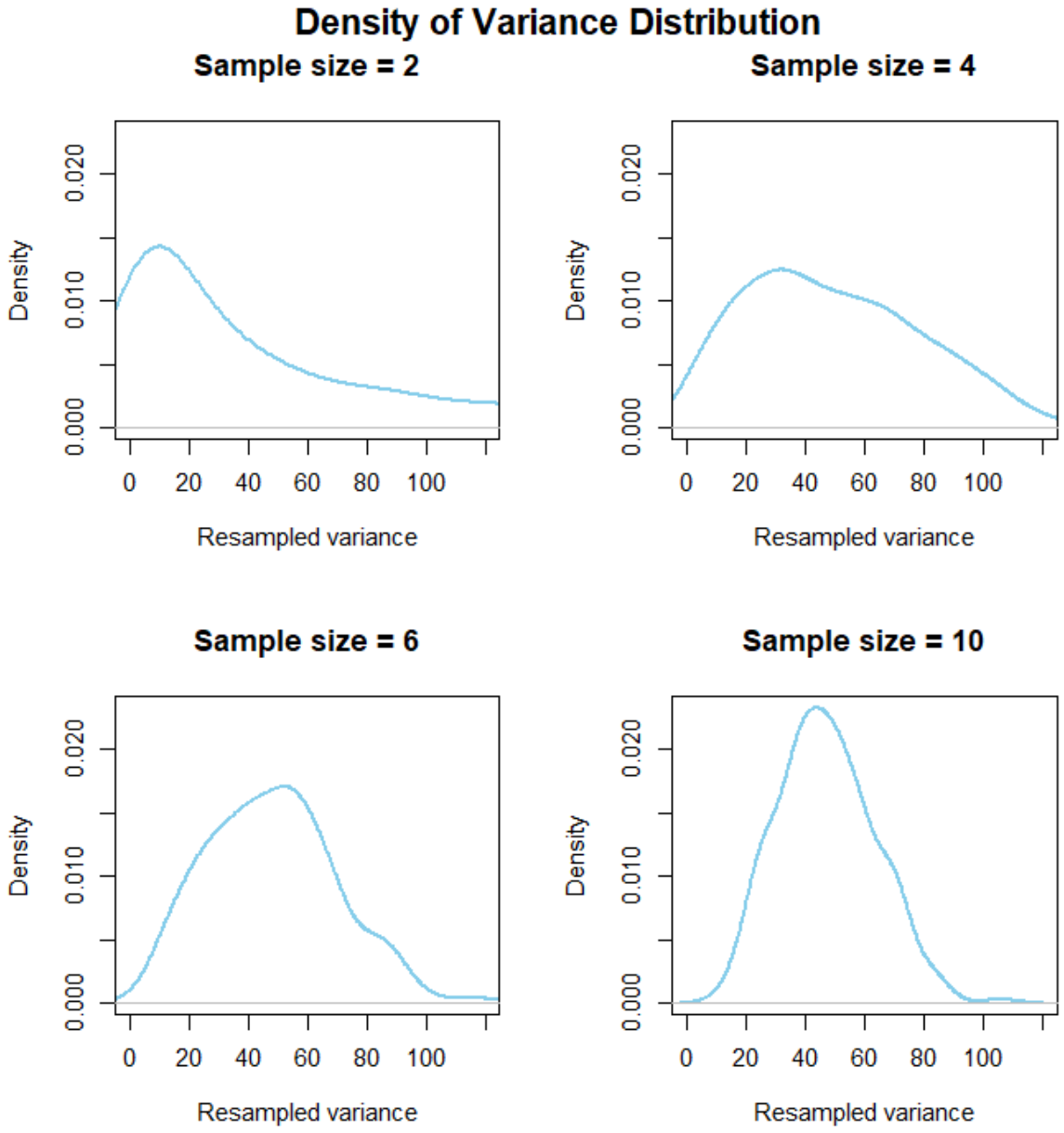

Figure S 10: *Density plots illustrate the distribution of resampled variances across various sample sizes. This data was created based on the assumption of uniform distribution. A pronounced left skew is observed for size 2, which gradually diminishes as sample size grows. The distribution approaches symmetry when the sample size exceeds 4, with the skewness dropping below 0.5.*