STUDY AND COMPARISON OF NEXT GENERATION SEQUENCE ALGORITHMS AND TOOLS

by

HEATH LANDON YATES

B.S., University of Missouri - Kansas City, 2004

M.S., Kansas State University, 2011

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
Dr. Doina Caragea

Copyright

Abstract

This study is a comparison and exploration of next generation sequencing algorithms and tools. A simulation study was done to compare the performance of edgeR, DESeq, and baySeq in detecting differential gene expression. The methods were compared in context of a balanced pairwise design. The simulation results suggest that the methods are comparable under the conditions simulated. The study also explored real data comprised of one biological replicate between two treatments. Cufflinks and CummerRBund were used to detect differential gene expression. The visualization results from the real data suggest no differential expression is present.

Table of Contents

List of Figures

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Dr. Doina Caragea, you have been a tremendous mentor. Your advice on both my report and career has been very valuable. I would like to thank my committee members Dr. David Gustafson and Dr. Mitch Neilsen. I want to thank my committee for making my defense an enjoyable moment, and for your useful comments and suggestions. I would especially like to thank the researchers who helped me on this project. I thank Dr. Hardcaste at the University of Cambridge for assisting me in understanding his data simulation. Special thanks are extended to Dr. Charlotte Soneson at Lund University for providing assistance on compcodeR package. Special thanks are also extended to Ms. Jennifer Shelton for helping to teach me about NGS RNA-seq pipelines. I also wish to thank the NSF for support during my GK-12STEM fellowship.

A special thanks to my family. Special appreciation belongs to my wife, Chunfang who spent every moment encouraging me, I couldn't have done it without you. I am also grateful to my son and daughter. Special thanks to my mother and sister for helping proofread my report. I would also like to extend my thanks to all my friends who supported me during the entire process.

Dedication

I dedicate this to my wife, son, and daughter.

Chapter 1

Introduction

"I think the biggest innovations of the twenty-first century will be at the intersection of biology and technology." - Steve Jobs

1.1 Introduction

The discovery of deoxyribo-nucleic-acid(DNA) in 1953 has herald a new revolution in biology that has only accelerated in contemporary times. Advances in computer science has ushered an era of unparalleled progress in not only computation but all areas of science. Standing at the intersection of these two fields is a promising new technology known as Next Generation Sequencing (NGS). This is important technology due to its promise to dramatically accelerate biological and biomedical research by allowing scientists to analyze genomes and transcriptomes more easily and cheaply [Shendure and Hanlee, 2008]. The start of this revolutionary transformation began in 2004 with the publication of the complete sequencing of of the *Mycoplasma genitalium* and *Streptococcu pneumonia* genomes in Nature[Brown, 2013]. Since then, the pace has only accelerated as NGS experiments become more widespread and popular. At the same time, many biologists, computer scientists, and statisticians have developed new algorithms to help analyze the data that NGS experiments

produce such as baySeq, edgeR, and DESeq. The goal of this report is to explore differential expression algorithms baySeq, edgeR, and DESeq. We also examine tools found in the Galaxy Project to aid in the analysis of actual RNA-seq data.

In this introduction, we provide a high level overview of the NGS technology. Next, we discuss the history of the technology, how sequencing works, and the relevance of NGS as an important tool in modern science. The author focused on pairwise experimental design in comparing NGS differential expression algorithms in this report. Consequently, it is important to discuss the basics of experimental design as well as discuss other designs in order to put pairwise experiments in a proper framework. Finally, the author will provide a brief overview of this report.

1.2 Next Generation Sequencing

1.2.1 Motivation of the Term Next Generation Sequencing

In order to discuss NGS, it is important to begin our discussion with its origins in DNA sequencing. There was work on sequencing DNA since the early 1970's, but it only became more mainstream and popular after the the work of Frederick Sanger began in 1975 [Sanger and Coulson, 1975]. The method was rudimentary at first, but eventually went on to become the most popular sequencing technology for the next 25 years. In fact, Sanger sequencing as it became to be known was instrumental on the Human Genome project [Brown, 2013]. Thus, we begin our discussion with Sanger sequencing. The method we are familiar with was actually developed in 1977. We omit the technical details on how Sanger sequencing works in producing output as it is not important for the purposes of this report. Rather, it is important to consider the nature of the output and the abilities of the sequencing technology. Sanger technology has the ability to produce 900 or more high quality bp per read [Morozova and Marra, 2008]. Thus, it is an ideal technology in sequencing experiments due to the length and the quality of the read. Unfortunately, the associated high cost of

this technology has made it prohibitive of typical experiments. Thus, new technologies has developed and evolved to address this problem and are a part of the next generation sequencing revolution.

1.2.2 NGS Sequencing

We now turn our attention to NGS sequencing. What follows will be a brief introduction and overview. Please note more details of NGS are forthcoming in Chapter 2. The new sequencers produce shorter bp per read than Sanger, but they can do it more cheaply and in higher volume. We now briefly discuss the pipeline flow which produces RNA sequencing data (RNA-seq) which is used in differential expression analysis. The pipeline we propose is a general framework in which differential gene expression can be detected when there is already a reference genome available. The framework follows this basic pattern; first, a sequencing machine will generate raw sequence reads. Second, it is necessary to align these raw reads to a genome for comparison. Third, the aligned data will then be analyzed by a differential gene expression analysis algorithm or tool [Zvelebil and Baum, 2007].

Scientists often wish to study transcriptomes in cells. For instance, seeking to obtain information regarding protein synthesis at the time they collect data from an organism. They will use an NGS sequencer to obtain RNA-seq data which relies on the sequencing of mRNA fragments [Brown, 2013]. Once the raw reads have been generated by a machine, it is necessary to align these reads to a reference genome. Since the reads are RNA, then it follows that this information will only align with the exons in the genome and not the introns [Brown, 2013]. This is not a problem because in practice, this information is often of high interest to a biologist. For example, suppose a scientist is studying tissues obtained from a tumor. The transcribed mRNA can originate from genes formed by the transposition of DNA between two chromosomes [Brown, 2013]. In this example, RNA-seq data is powerful in this instance because these transpositions can be detected because one end of an RNA read could map to a location on one chromosome and the other end of the RNA read could

map to another chromosome. Since RNA-seq analysis can be extremely tenuous, then this is further motivation for the importance of NGS methods and tools.

1.2.3 Relevance of NGS

The relevance of NGS technology cannot be understated. In the decade since the human genome has been sequenced, we are currently undergoing a revolution. Brown [2013] argues that when accumulated knowledge and new technologies reach sufficient maturity, then scientific problems are often solved concurrently by many different scientists making simultaneous discoveries. In other words, it is argued that the first revolution in DNA sequencing technology was ushered in by Sanger and Coulson [1975] and now we are undergoing a similar revolution in NGS technologies. The technology has doubled output capabilities every year since 2004. In Chapter 2, readers will learn more about the characteristics of NGS. Its revolutionary characteristics can be described as high data throughput, short read lengths, and less accuracy as compared to Sanger sequencing, at extremely low economic cost. The ability to analyze this information and to develop new techniques as the technology evolves will play an important role in both biology and computer science for years to come. Brown [2013] also indicates that most laboratories do not have the computational infrastructure and skills that are required to perform the analysis on the complex data sets that these technologies produce. In short, we can safely claim that demand is high for these skills while the supply is low. Thus, NGS technology is extremely relevant to current and future endeavors of science.

1.3 Types of Experimental Designs

Experimental design is a critical component of modern science and thus deserves a brief discussion in relation to NGS experiments. The basic motivation behind the proper design of an experiment is to reduce variability across samples while enhancing measurement of the

effect one wishes to observe. In other words, one will often have a control group where all the effects are well understood and then compare it to another sample which is experimental group and the exact effect not known. Replication is used in both the control and treatment groups in order to average over individual variation, thus enhancing measurements of the effects we wish to detect.

In RNA-seq, regardless of the design, there are three levels of sampling behind a statistically valid experimental design. First, the organism must be selected randomly from a larger population. In this way, the results of the study could then be inferred to the larger population at large. Second, RNA sampling must occur when the scientist is isolating the RNA of interest from the cell. Finally, it is important to note that only fragments of the RNA taken from the second step are retained. We now discuss experimental design as related to NGS by considering both pairwise balanced design and multifactor design.

1.3.1 Balanced Pairwise Design

In NGS experiments, scientists are often interested in detecting differential gene expression between treatment groups in the form of a pairwise comparison. The basic idea is to obtain genes from two organisms under two different conditions. We call the first condition treatment 1 and the second condition treatment 2. Consider figure 1.1 below.

	Treatment1	Treatment2
Gene	n_{11}	n_{21}
Other Genes	n_{21}	n_{22}

Figure 1.1: *Balanced pairwise design to compare one gene across treatment 1 and treatment 2. The cell counts n_{ij} represent DGE count for Gene or Other Genes.*

The cell count n_{ij} represents differential gene expression count for the gene ($j = 1$) or other remaining genes ($j = 2$) for treatment i. The design can apply statistics to the above contingency table to determine the probability that the classification of this gene has affected the gene expression. In other words, we can determine if the treatment can

help explain whether the gene above is expressed or not. The design is considered balanced because there is a balanced quantity of treatments and replications. It is pairwise because there are two treatments under consideration. Thus, it is a balanced pairwise design. Given the widespread application of this experimental design in NGS studies, we focused on this experimental design in our data simulations.

1.3.2 Multifactor Pairwise Design

Multifactor design simply refers to more than two treatments under consideration. Hence, it is possible to have three, four, five, or more treatments where gene counts are collected. As before, we wish to observe the gene expression between treatments. We simply enumerate our treatments as follows, the first condition treatment 1, second condition is treatment 2, and so on. We extend our example given above to a three way experimental design. Please see figure 1.2.

	Treatment1	Treatment2	Treatment3
Gene	n_{11}	n_{21}	n_{13}
Other Genes	n_{21}	n_{22}	n_{23}

Figure 1.2: *Balanced multifactor design to compare one gene across multiple treatments. The cell count n_{ij} represents DGE count for Gene or Other Genes.*

The idea is very similar to the pairwise design, the cell count n_{ij} represents differential expression count for gene($j = 1$) or other remaining genes ($j = 1$). However, the statistics applied in this situation would be different than for the pairwise design because the experiment is different. While the measurement goal is the same, how it is achieved will rely on different statistics because the experimental design is different. Depending on the design of the experiment, some methods might be more appropriate than others. There are other kinds of experimental designs, but this project's focus is limited to balanced pairwise experimental design. This is because they are common in contemporary experiments and data simulations.

1.4 Report Outline

The report advances the discussion on the biological foundation that motivates NGS studies in Chapter 2. In Chapter 3, Chapter 4, and Chapter 5 methods are explored that analyze RNA-seq count data. These are baySeq, edgeR, and DESeq. In Chapter 6, the report explores the pipeline for analyzing Red Flour Beetle data using Bowtie, Tophat, and Cufflinks. cummeRbund is used to analyze and visualize the results. In chapter 7, we perform our data simulation and compare baySeq, edgeR, and DESeq. Next, relevant work and areas of potential future work is explored. Finally, we will discuss the conclusions of our report.

Chapter 2

Differential Analysis

In this chapter, we first introduce basic details of molecular biology. We place particular emphasis on the central dogma of biology. Next, we will discuss the biological explanation of differential gene expression. Third, we discuss NGS technology and how it relates to the analysis of differential gene expression. During our discussion of NGS technology we place particular emphasis on the available sequencing technologies and the contemporary abilities of the technology. Specifically, we will discuss RNA-seq data and how it relates to counts and normalized counts.

2.1 Brief Introduction to Molecular Biology

It is appropriate to discuss basic molecular biology before differential gene expression is discussed. As the name indicates, molecular biology concerns itself with the molecular details of life. We also point out that the term "molecular" is derived from the term molecule, that is, a group of two or more atoms that are held together by chemical bonds. As such, we narrow our focus on the molecular basis of biology and life known as genetics. It is well known that life is comprised of Deoxyribonucleic acid (DNA). This is a molecule that encodes all known instructions used in the development and functioning of all known

forms of life. Specifically, DNA is a nucleic acid made of monomers known as nucleotides. By definition, a nucleotide has three components:

1. 5-carbon sugar
2. phosphate group
3. nitrogenous base

The nucleotides can be categorized into two groups known as Purines and Pyrimidines [Cohen, 2007]. Purines have a two ring structure while Pyrimidines have one ring. Specifically, Adenine(A) and Guanine(T) as Purines, while Cytosine(C), Uracil(U), and Tymine(T) are Pyrimidines. The chemical structure of DNA is comprised of pairings between A and T, and pairings G and C which are held together by hydrogen bonds [Figure 2.1]. It is important to note that the nucleotides on a strand are held together by a backbone of phosphate and sugars. Consequently, the pattern that emerges is known as the double helix. The sequence of nucleotides that comprise DNA encodes biological information. A single long sequence of DNA is called a chromosome. On a chromosome it is common to refer to specific sequences as genes. It is these specific genes which are responsible for information on how to build, maintain, replicate cells. At the same time, they contain information that will be inherited during reproduction. It is important to note in animal cells, otherwise known as eukaryotes, that the DNA is contained in the nucleas of an organic cell. The entire collection of genetic information stored in genes is referred to as the genome.

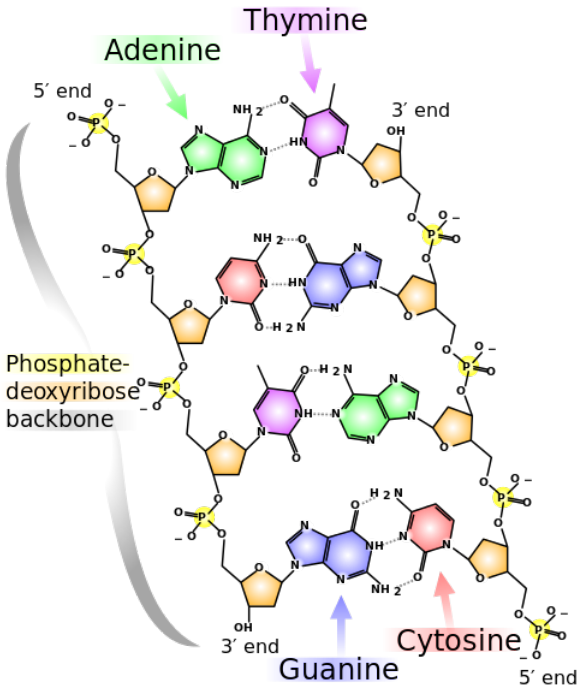

Figure 2.1: *Chemical Structure of DNA by Madprime [2014].Creative Commons*

Another important sequence of nucleotides necessary for maintaining life function is called RNA. It is comprised of ribose instead of deoxyribose and the complementary base to Thymine is Uracil. RNA is essential for acting as a catalyst for biological reactions, controlling and regulating gene expression, and participating in cellular signals [Zvelebil and Baum, 2007]. As such, it is now appropriate to briefly describe how genetic information is transmitted on a molecular level and thus providing a conceptual framework in which to interpret basic genetics. The central dogma of molecular biology arose shortly after the discovery of DNA to explain how these molecules interact with each other to transmit information. The central dogma simply states that DNA is transcribed to mRNA, which in turn makes protein [Cohen, 2007].

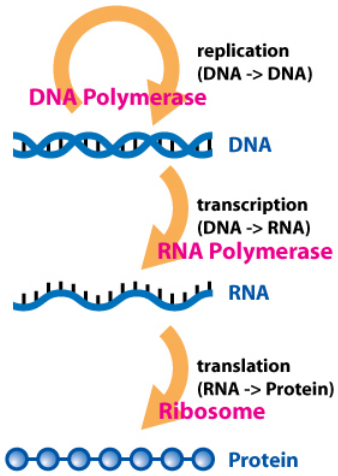

Figure 2.2: *Central Dogma of Molecular Biochemistry with Enzymes by Horspool [2014]. Creative Commons.*

It is important to note that RNA may be transferred back to DNA. However, it is more uncommon. Now that a basic framework for discussing molecular biology has been established, it is appropriate to discuss differential gene expression. Specifically, we state that genes in a DNA sequence that are transcribed to mRNA which goes to the ribosome to make a protein are considered expressed.

2.2 Biological Explanation of Differential Expression

2.2.1 Basic Biological Background

It is well understood that the genome is the same in all cells of an organism. This motivates a fundamental question in biology. Given that the genome of every cell is the same, then what makes cells different from each other? For example, red blood cells in humans are red because they contain a protein called hemoglobin. The red blood cell is red because the protein hemoglobin is bright red in color. Despite having the same genome, no other cells in the human body besides red blood cells produce hemoglobin. In other words, the genes in the red blood cell are differentially expressed to produce hemoglobin while other cells in

humans are not. There are three basic facts established by genomic research that provides a basic framework to understand differential gene expression [Gilbert, 2000]. They are as follows:

1. The DNA of all cells is identical.
2. Only a small portion of the genome in a cell is expressed.
3. The unused genes in a differentiated cell are are not destroyed or mutated. They are dormant and retained for potential expression.

It is important to note that not only are genes expressed diversely in different cells, but they can also be differentially expressed within the same cell under different conditions. For example, genes could be differentially expressed for two male red flour beetles under two treatments: one is starved and the other is fed.

It is known that a single gene can code for multiple proteins. Alternative splicing is one process that can explain protein diversity (see next section for further details). Furthermore, there are mRNA isoforms where there are several different forms of the same protein. The human being has approximately 20,500 protein producing genes [Clamp et al., 2007]. However, from these genes it is estimated that humans can produced approximately 200,000 to 300,000 proteins. However, as discussed above it has been established that each cell in the human body will only require a small portion of these genes to be expressed. In other words, different cells in the human body will express different genes. We know that specific cells in the human body produce specific proteins. For instance, the red blood cell makes globin, eye cells make crystallins, liver cells produce albumin, and melanocytes create melanin, and so on. Thus, it is of interest to study how genotype can influence phenotype. As such, exploring differential gene expression is a major research focus in modern biology. It is understood that the regulation of gene expression can occur in the following ways [Gilbert, 2000]:

1. Alternative Splicing

2. Selective RNA Processing

3. Selective messenger RNA translation

4. Differential Protein Modification

Therefore, it follows that differential gene expression can be accomplished by fairly diverse genetical processes and is critical to understanding modern biology.

2.2.2 Alternative Splicing

First, it is necessary to understand what we mean by splicing. By definition, an intron is a sequence of RNA nucleotides that are removed during splicing. Conversely, exons are the portions of RNA from a gene that remain after the introns have been removed. RNA splicing is where we take pre-messenger RNA which is comprised of introns and exons, and group the exons together. See Figure 2.3.

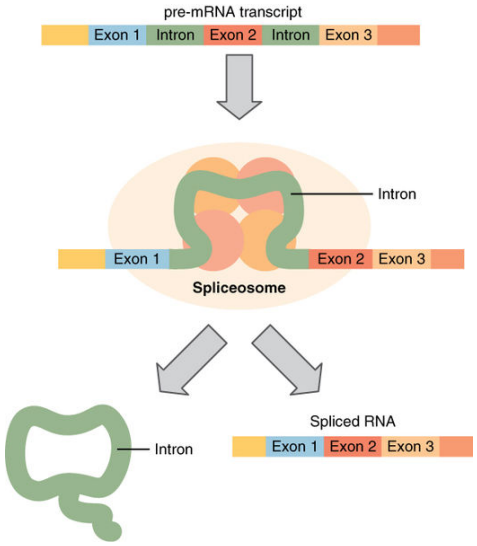

Figure 2.3: *RNA Splicing by College [2013]. Creative Commons.*

The most basic definition of alternative splicing is that one gene can code for multiple proteins. This is accomplished by the pre-mRNA being spliced so that there are alternative

spliced mRNA strands that will code for protein isoforms. See Figure 2.4.

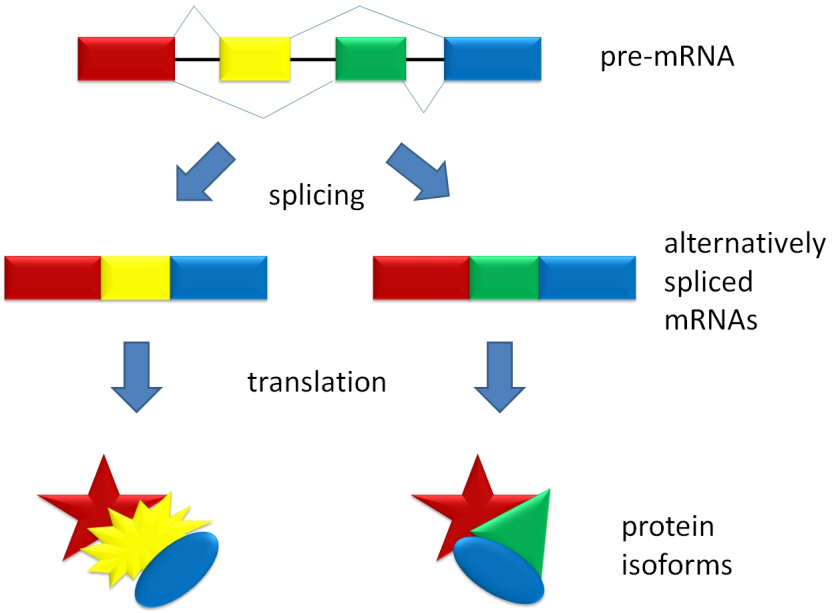

Figure 2.4: *Alternative splicing resulting in protein isoforms by Agathman [2009b]. Creative Commons.*

There are five basic modes of alternative splicing [Zvelebil and Baum, 2007]. First is exon skipping where an exon may be spliced out but the transcript is preserved. Second is when an alternative 3' splice site is used, which changes the 5' boundary in an exon downstream. The third is an alternative 5' splice junction, which changes the 3' boundary in an exon upstream. Fourth is mutually exclusive exons, given two exons one is retained in mRNA after splicing. Fifth is intron retention which means that a sequence can be spliced as an intron or it can be retained [Zvelebil and Baum, 2007]. See Figure 2.5.

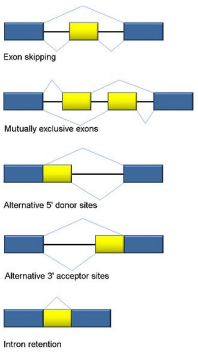

Figure 2.5: *Collection of Basic Alternative RNA splicing events by Agathman [2009a]. Creative Commons.*

Thus, it follows that similar proteins can arise from alternative splicing. This is an important observation since it explains how a genome can produce many more proteins than there are genes.

2.3 Next Generation Sequencing

2.3.1 Microarray Technology

As mentioned earlier, the origins of DNA sequencing technology are from Sanger technologies. However, before we discuss NGS sequencers it is important to mention another popular sequencing technology which was developed over ten years ago. It is important to discuss microarray technology in order to put NGS in a proper context. The technology was developed from Southern blotting, which is when a given DNA fragment is attached to a substrate and probed with a specific DNA sequence [Zvelebil and Baum, 2007]. Note that a substrate is a molecule upon which an enzyme acts. The technology had its infancy in the late 1980s and became widely available in the late 1990s. In microarray, grid of DNA segments of known sequence is used to determine differential gene expression. Thus, complementary DNA or RNA targets are introduced to the grids. Gene expression values can be obtained from the microarray through the use of heat maps which can visualize the results of the

data analysis. The nature of this data is continuous and hence the statistical methods developed in the previous decade to interpret these experiments are all founded on this basic assumption. Microarray is still very reliable and cheaper than NGS sequencers. Hence, it is still used and popular. Microarrays, however, cover less genes than NGS. Nevertheless, the normalization and interpretation of microarray data is well understood. Despite this, NGS has distinct advantages in studying alternative splicing and conducting differential gene expression analysis.

2.3.2 Next Generation Sequencing

The ability to read DNA sequences is necessary for investigating the scientific questions that are motivated by the study of differential expression. Although Sanger sequencing technology is widely available and commonly used in labs, it has drawbacks. Sanger sequencing is disadvantaged in throughput, scalability, speed and resolution [Zhang et al., 2011]. The development of Next Generation Sequencing (NGS) technology addressed many of these drawbacks and as a result has replaced Sanger at the preferred sequencing technology. NGS has many advantages over traditional methods. Namely, the ability to sequence at previous unrealized speed and the ability to produce high sequencing [Zhang et al., 2011]. We limit our current discussion in NGS technology to Illumina and 454. This does not imply that the other technologies are not important. Rather, it is appropriate to concentrate our discussion to the technologies most relevant to our data. In other words, we focus on the technologies that could produce or replicate our red flour beetle data that we use later on in analysis.

Since the dawn of NGS in the last decade, private industry has taken the lead in the development of new sequencing technology and techniques. These technologies as discussed above allow for large-scale sequencing and this trend shows no signs of abating. The trend in commercially available NGS platforms is competitive, increasing, and varied. The NGS instruments produce different base lengths, error rates, and error profiles relative to each other [Zhang et al., 2011]. These technologies can sequence both DNA and mRNA.

The Roche GS-FLX 454 Genome Sequencer is the oldest and first available NGS sequencer for commercial use. It can produce an average of 400bp with a maximum of 600bp. Many note that is half of the Sanger sequencing capabilities. Since it can produce 600bp, it also provides the longest short reads among all current NGS technologies [Zhang et al., 2011]. It has an accuracy of approximately 99% and is well known to produce raw reads with errors. The abilities of this technology make it a natural resource for de novo sequencing and isoform studies.

The Illumina Genome Analyzer came after the Roche GS-FLX 454 Genome sequencer. The Illumina Genome Analyzer produces approximately 200Gbp of short sequence reads when run. It has an accuracy of approximately 99.5%. Illumina is currently the most popular technology on the market and the most widely used. Many argue that is has superior data quality and sufficient read length. The evidence of this is that the many papers in NGS papers have described research which was dependent on Illumina [Zhang et al., 2011].

2.3.3 RNA-seq Data

Sequencing

We explain, from a high overview, the basic sequencing process as described by Brown [2013]. First, RNA is sequenced by converting it to complementary DNA (cDNA) with the reverse transcriptase enzyme. The reverse transcriptase enzyme is defined as an enzyme that generates cDNA from RNA. Since our focus is on differential gene expression, then it follows that we are interested in directly sequencing the mRNA. In summary, we begin with RNA samples and then fragment them. From these fragments, we perform reverse transcription to obtain cDNA fragments. These fragments are then given to a sequencing machine which will then output reads based the cDNA fragments provided to it.

However, it is important to note that other applications for RNA sequencing exist. For example, RNA-seq data is an excellent resource to measure alternative splicing events that

can produce different mRNA strands which will eventually arise to make different protein isoforms.

There are protocols that have been developed by every major sequencer company for sequencing RNA [Brown, 2013]. Each have to contend with some biological realities in the data. That is, ribosomal RNA (rRNA) and transfer (tRNA) are abundant in all RNA obtained from organisms, approximately 75% of RNA molecules [Brown, 2013]. Thus, it follows that such the presence of rRNA and tRNA sequences in RNA hamper the volume, accuracy, and cost of RNA-seq data. As a result, sequencers use protocols to isolate mRNA from rRNA and tRNA. This is accomplished either using a technique called poly(T) or duplex-specific nuclease (DSN). Unfortunately, even in the presence of these methods some of the rRNA and tRNA might remain. Consequently, these can be filtered out by comparing the reads to a contaminant file of rRNA and tRNA sequences from organism we are studying [Brown, 2013].

Counts for Measuring Gene Expression

Once the sequencing reads are obtained from the NGS sequencer, then it is necessary to measure for gene expression. The first step is to align the reads to a reference genome, usually with the assistance of sequence alignment software [Brown, 2013]. The most important part to note here is that expression for each gene is measured by counting the number of sequence reads that align to its coding region in the genome. This is usually accomplished by aligning millions of short reads by using an algorithm such as Bowtie. It can take considerable computational time to align the raw reads to the genome to produce this count data. Once gene expression for each gene has been counted, then we are ready to analyze the data statistically to determine whether or not differential gene expression counts vary significantly across conditions or treatment.

Normalized Counts

It is important to note that baySeq, edgeR, and DESeq rely on raw counts for measuring gene expression. However, some algorithms require the data to go through additional steps before statistical analysis of differential gene expression. For example, Cufflinks is one of these methods. Besides, normalization of count scores can aid in the reduction of variance that could exist in count data which would otherwise reduce the effectiveness in measuring change in gene expression between different conditions or treatments. Thus, we discuss simple reads per kilobase per million (RPKM) normalization. It was developed by Caltech and is the current RNA-seq normalization standard where the read count for each gene is divided by the length of reference sequence for that gene. It then scales all read counts per million reads in each lane of sequencing [Brown, 2013]. See below:

$$RPKM = \frac{10^9 \cdot C}{N \cdot L}$$

C is the number of mappable reads, N is the total number of mappable reads in an experiment, and L is the sume of the exons in base pairs. We may also describe it as:

$$RPKM = \frac{\text{Reads per transcript}}{\text{million reads} \cdot \text{transcript length}}$$

2.3.4 RNA-seq Tools

It is important to note that we are not comparing the merits of raw counts vs. normalized counts. Rather, we discuss both here as different algorithms have different RNA-seq data requirements to detect differential gene expression. Thus, regardless of whether the algorithm requires raw or normalized counts, it follows that each algorithms goal is to measure the data in such a fashion that it can measure differential gene expression in the data between biological conditions in a statistically valid manner.

Here we discuss some of the primary tools which were used in this report. They are **compcodeR**, **Galaxy**, and **cummeRBund**. Galaxy is available from the Galaxy Project

at usegalaxy.org. compcodeR and cummeRBund are both available for free from www.bioconductor.org. Bioconductor is an open source software toolkit for bioinformatics. It is often used in the analysis of NGS data to perform differential gene expression analysis among many other applications. It is specifically designed for NGS high throughput data. It is important to note that Bioconductor relies on the R statistical programming language.

2.3.5 Compcoder

compcodeR is a package in R written by Soneson and Delorenzi [2013]. It provides an interface to several NGS methods for analyzing differential gene expression already implemented in bioconductor for RNA-seq data. The package also has a framework and functions for generating simulated data [Soneson and Delorenzi, 2013]. Most importantly, it has a great interface and functions for comparing the results of data simulations.

2.3.6 Galaxy

Galaxy is a suite that is part of the Galaxy Project [Blankenberg et al., 2010], [Goecks et al., 2010], [Giardine et al., 2005]. It is an open source platform that allows a user to process raw RNA-seq data. It is comprised of many tools that allow the user to detect differential gene expression in the data. In our report, we use an analysis flow in Galaxy that utilizes Tophat and Cufflinks to produce files that can help us study differential gene expression. These files are then given to cummeRbund for visual analysis of differential gene expression.

2.3.7 Cummerbund

cummeRbund is an R package designed to aid in the analysis Cufflinks RNA-seq output. It was written by Computational Biology Group at MIT's Computer Science and Artificial Intelligence Laboratory and Rinn Lab at Harvard. The package can be found at compbio.mit.edu/cummeRbund. It allows the user to visualize differential gene expression between

genes. It helps the user navigate, with ease, the large and complicated data sets produced by studies from data produced by Cufflinks.

Chapter 3

Bayseq

This chapter will discuss one of the most complicated algorithms that can be applied to RNA-Seq output known as baySeq. First, baySeq is introduced with a brief overview. Second, we will then discuss the technical definition of the model and important details of how the algorithm functions. Third, we will demonstrate how to implement baySeq in R with a simple example.

3.1 Bayseq Method

3.1.1 A Brief Overview

One of the more complicated, but popular tools employed on RNA-Seq data is known as baySeq. It is claimed that one of the most distinct advantages of baySeq as compared to other methods is to account for multiple factor experimental designs. baySeq uses Bayesian methods to estimate posterior likelihoods of each set of models which can define differential expression for each gene [Hardcastle and Kelly, 2010]. In other words, we can assume that there are some prior distributions for genes in different groups such as differentially expressed or non-differentially expressed. BaySeq will then evaluate the posterior probability of being differentially expressed and rank the genes according to this probability.

3.1.2 Descriptions of Bayseq Method

First, baySeq estimates an empirical distribution for the parameters of the Negative Binomial distribution. It accomplishes this by bootstrapping from the data: taking individual counts and finding quasi-likelihood parameters for a Negative Binomial distribution [Hardcastle and Kelly, 2010]. Hardcastle and Kelly [2010] suggest a sample size of 10,000 iterations to estimate an empirical distribution of the parameters. Next, baySeq estimates posterior likelihoods of differential expression. We explain more in detail below.

In our report, we focus on balanced pairwise comparison of two conditions A and B with equal amount of replicates between the conditions. There are two models to potentially explain our data as follows: $M_1 = \{A_1, A_2, B_1, B_2\}$ for non-differential expression and $M_2 = \{A_1, A_2\}, \{B_1, B_2\}$. Next, define the dataset as $D_g = \{(Y_{g1}, Y_{g2}, ...), (l_1, l_2, ...)\}$ where Y_{gi} corresponds to counts of gene g such that $i \in (l_1, l_2, ...)$ where l_i is library size [Hardcastle and Kelly, 2010]. M is the user specified models that we wish baySeq to consider. We define θ_M as a vector of parameters of model M. Thus, the posterior probability of the model given the data for gene g is [Hardcastle and Kelly, 2010]

$$P(M|D_g) = \frac{P(D_g|M)P(M)}{P(D_g)}$$

We can calculate the marginal likelihood as follows:

$$P(D_g|M) = \int P(D_g|\theta_m, M)P(\theta_m|M)\mathrm{d}\theta_M$$

Hardcastle and Kelly [2010] define an empirical distribution on θ_m and estimates the marginal likelihood numerically. The priors are estimated by iteration. Thus, we have a posterior probability of the gene being differentially expressed conditional on the model for differential expression.

3.2 Bayseq Usage in R

We now consider the implementation of baySeq in R according to [Hardcastle and Kelly, 2010]. The baySeq implementation in R assumes that the data provided will be a data matrix which is comprised of counts where the rows correspond to genes and columns to samples. We assume our experiment is a balanced pairwise design. We simulate data similar to the negative binomial data generated in Hardcastle and Kelly [2010] paper. In our simulation we estimate the mean parameter for the negative binomial distribution using the normal distribution. We simulate two biological conditions. There are four biological replicates for each condition. We simulated 10000 genes where 10% are differentially expressed. Please see Appendix B for more details. The generated data matrix for our implementation in R is seen below:

```
     [,1] [,2] [,3] [,4]
[1,]    3    2   17    6
[2,]    5    3   18   38
[3,]    6    3   27   27
[4,]    0    0    5   52
[5,]    0    6   46   35
[6,]    0   10   54   42
```

Since there are two conditions A and B with two biological replicates each, then the implementation of baySeq in R will require us to define a group with this information. We elaborate as follows:

```
libs = 8
groups = list(NDE = rep(1,libs), DE = rep(1:2, each = libs/2))
```

The above defines our group which is comprised of two models. The first is assuming there is no differential expression between the conditions. The second assumes that there is differential expression between conditions. The next step is to define a count data object, this step is necessary before estimating priors and posterior probabilities. Hence, we have the code as shown below:

```
CD = new('countData', data = y, replicates = rep(1:2, each = libs/2),
                                                    groups = groups)
library(CD) = getLibsizes(CD)
```

The last function written above estimates the library sizes for the countData object created above. The next step consists of the empirical Bayes estimation process of baySeq. First, it will be necessary to estimate the prior distribution on countData. By default, the function will assume the negative binomial method. However, other methods are available. Second, once our prior has been estimated, then it follows that we will estimate posterior likelihoods for each gene conditional on each model. That is, assuming there is no differential expression and assuming there is. We implement as follows:

```
CD = getPriors.NB(CD, samplesize = 10^5, estimation = 'QL', cl = cl)
CD = getLikelihoods.NB(CD, pET = 'BIC', cl = cl)
```

The final step is to display the results. It sorts the results with the likelihood. That is, it allows us to see which genes rejected the null hypothesis where there is no differential expression. Thus, we now have evidence showing what genes are differential expressed. Please see below:

```
results = topCounts(CD, group = 'DE', number = 10)
head(results)
```

We show the results below:

	rowID	X1.1	X1.2	X1.3	X1.4	X2.1	X2.2	X2.3	X2.4	Likelihood	DE	FDR.DE
1	3163	353	353	353	353	353	353	353	353	1	1>2	0.000000e+00
2	3441	447	447	447	447	447	447	447	447	1	1>2	0.000000e+00
3	3607	406	406	406	406	406	406	406	406	1	1>2	0.000000e+00
4	5174	442	442	442	442	442	442	442	442	1	1>2	0.000000e+00
5	7337	440	440	440	440	440	440	440	440	1	1>2	0.000000e+00
6	8538	329	329	329	329	329	329	329	329	1	1>2	4.736952e-15

We have shown the top differentially expressed results. It has given the probability of the genes, associated with rowID, of being differentially expressed. This concludes the use of baySeq in R. It is important to note that the code to implement baySeq is incredible simple. However, compared to the other tools, it takes the longest to run and is probably the most complicated computationally.

Chapter 4

EdgeR

This chapter discusses the edgeR approach in detecting differential gene expression in data. We first provide a brief overview and technical description of edgeR. Second, we then discuss how to use edgeR in R and include an example of its implementation

4.1 EdgeR Method

4.1.1 A Brief Overview

The basic idea is that the count data is over-dispersed and modeled using a negative binomial distribution. We limit our consideration of edgeR to a balanced pairwise experimental design. For pairwise comparison, the authors used conditional maximum likelihood and weighted conditional likelihood to estimate the negative binomial parameters. Thus, the authors contend that an exact p-value can be obtained for determining differential expression by performing an exact test similar to Fischer.

4.1.2 Description of EdgeR Method

The data is assumed to be organized as a matrix comprised of counts where rows correspond to genes g and columns represent biological sample corresponding to i. These counts are assumed to be distributed as a negative binomial as described by Robinson et al. [2010] as follow:

$$Y_{gi} \sim \mathcal{NB}(\mu_{gi}, \phi_g)$$

where the mean is denoted by μ_{gi} and ϕ_g is dispersion. Please note that l_i is the library size represented by the total number of reads and μ_{gi} is the proportion of sequenced gene. As described above, it follows that the read counts Y_{gi} are nonnegative integers. This distribution is used in order to model count data when overdispersion could be present [Robinson et al., 2010]. It is important to note that Robinson et al. [2010] assume the mean and the variance are related by ϕ such that $\sigma^2 = \mu + \phi\mu^2$. First, Robinson et al.

[2010] propose estimating ϕ_g by finding a common dispersion estimate of ϕ. Robinson et al. [2010] use a weighted likelihood approach for shrinking gene-wise dispersion so that it converges. Hence, in order to test for differential expression, an exact test analogous to Fisher's exact test is used. The difference is that instead of a hypergeometric distribution, a negative binomial distribution is used. Therefore, the p-value is defined as the probability of observing counts at or more extreme than we observed [Robinson et al., 2010].

4.2 EdgeR Usage In R

The following is an example on how to implement edgeR [Robinson et al., 2010]. We use the same simulated data as described in Chapter 3. That is, we will consider a balanced pairwise experimental design between four conditions with two replicates each. edgeR package in R requires the user to provide it with the original data, not normalized. The count data will be comprised of rows that correspond to genes and columns to samples. edgeR models the count data as negative binomial. It estimates the mean for each gene and sample. It also estimates the dispersion which is the same across all genes. Once these estimates are obtained, an exact test is performed [Robinson et al., 2010]. See below:

```
d = DGEList(counts = y, group = c(1,1,2,2), lib.size = rep(1000,4))
de = exactTest(d, dispersion = 0.2)
topTags(de)
```

Thus, we obtain the following results.

```
Comparison of groups:  2-1
       logFC   logCPM       PValue          FDR
821 6.256040 11.80315 1.455105e-17 1.455105e-13
591 7.888743 10.70937 1.433265e-14 6.336287e-11
88  5.135611 11.60906 1.900886e-14 6.336287e-11
101 5.223563 11.46346 3.069242e-14 7.673105e-11
15  4.577731 11.99201 9.207976e-14 1.841595e-10
59  4.917946 11.41146 2.367778e-13 3.946297e-10
214 5.082740 11.06954 9.954818e-13 1.422117e-09
549 4.496906 11.54415 1.494889e-12 1.868611e-09
10  4.974733 10.97420 3.279654e-12 3.300366e-09
427 4.365906 11.56214 3.300366e-12 3.300366e-09
```

Note, we have performed an exact test to determine if genes from condition A and B are differentially expressed. The method topTags() sort by the highest ranked p-values for each gene. Simply put, we expect a p-value for each pairwise comparison of a given gene. Thus, the implementation of edgeR in R can be accomplished in a few lines.

Chapter 5

DESeq

The DESeq is a popular tool used to detect differential analysis in RNA-Seq output. It is similar to edgeR, but estimates the variance differently. We first discuss the DESeq method by providing an overview and some technical details. Finally, we will present a simple example of DESeq usage in R.

5.1 DESeq Method

5.1.1 A Brief Overview

DESeq is a method used for detecting differentially expressed genes in data. It is known to be similar to other tools such as edgeR. However, one of the unique distinctions of DESeq is that it is not limited to RNA-Seq data but information obtained from high-throughput experiments. The method is primarily known for estimating variance between genes. Therefore, the variance is conditional on the gene and replicate.

5.1.2 Description of DESeq Method

The basic assumptions for the model is that data is distributed according to a negative binomial distribution. Let the counts be assumed to represented by Y_{gi} given gene g and sample i. Let ρ denote the biological condition.

$$Y_{gi} \sim \mathcal{NB}(\mu_{gi}, \sigma^2_{gi})$$

where mean $\mu_{gi} = l_i q_{gi}$ such that l_i is library size and q_{gi} is the gene and condition specific expression rate. The variance is $\sigma^2_{gi} = \mu_{gi} + l_i^2 v_\rho(q_{gi})$ such that v_ρ is a smooth function per gene condition. The expected read count is given by $\hat{q}_{g\rho} = \frac{1}{N_p} \sum_{i:\rho(i)=\rho} \frac{Y_{gi}}{\hat{l}_i}$ where ρ indicates the biological condition and $i : \rho(i) = \rho$ represents the condition of sample i. Please note N_p is the total amount of replicates for condition ρ. The library size $\hat{l}_i$ is estimated using medians of the counts per gene and sample. We estimate variance as follows:

$$w_{gp} = \frac{1}{l_p - 1} \sum_{i \in p} (\frac{Y_{gi}}{\hat{l}_i} - \hat{q_{ip}})^2 \text{ and } z_{gp} = \frac{\hat{q_{gp}}}{N_p} \sum_{i \in p} \frac{1}{l_i}$$

Thus, we obtain estimate of the v_p equation as $\hat{v}_p(\hat{q}_{gi}) = w_p(\hat{q}_{gi}) - z_{gp}$. Similar to edgeR, differential gene expression is also tested in analogous for Fischer's exact test. Once again, the hypergeometric distribution is replaced with negative binomial probability [Anders and Huber, 2010].

5.2 DESeq Usage In R

The following is an implementation of DESeq in R as developed by Anders and Huber [2010]. As in edgeR, DESeq assumes that the data is structured as a data matrix where the rows correspond to genes and columns are the samples. The values in the matrix are assumed to be raw counts. The authors caution that the data should not be normalized before implementing DESeq. The same data matrix as described in Chapter 3 and Chapter 4 is simulated here. That is, a pairwise experimental design with two biological conditions with two replicates each. There are 1000 genes simulated where 10% are differentially expressed. It is now necessary to define the biological conditions of the data and create a counts data set. This is a necessary step in order to estimate the dispersion in the data and conduct an exact test. See the code below:

```
condition = c("trt1","trt1","trt2","trt2")
cds = newCountDataSet(y,condition)
```

Thus, we are now ready to estimate size factors. It estimates the size factor by finding the median. Thus, we have the following:

```
cds = estimateSizeFactors(cds)
sizeFactors(cds)
```

We now estimate the dispersion estimates for the count data with the command given below. For each condition, DESeq will compute an dispersion value. After this, fits by regression a dispersion and mean relationship. DESeq chooses for each gene the dispersion parameter which will be used in tests [Anders and Huber, 2010]. See below:

```
cds = estimateDispersions(cds)
```

We are now ready to proceed with testing for differential gene expression. In order to accomplish this, we only need to invoke a test as described below:

```
res = nbinomTest(cds, 'trt1', 'trt2')
resSig = res[ res$padj < 0.1, ]
head(resSig[order(resSig$pval),])
```

The output produced from this is another data matrix which contains both a p-value and an adjusted p-value. We sort by the most significant adjusted p-value.

```
     id baseMean baseMeanA baseMeanB foldChange log2FoldChange         pval
59   59   25.250      1.50     49.00   32.66667       5.029747 1.897339e-06
88   88   29.250      1.50     57.00   38.00000       5.247928 5.496997e-06
191 191   15.125      0.75     29.50   39.33333       5.297681 6.408957e-06
661 661   19.000      1.25     36.75   29.40000       4.877744 7.130403e-06
68   68   14.500      0.75     28.25   37.66667       5.235216 8.629669e-06
859 859   18.250      1.25     35.25   28.20000       4.817623 9.482088e-06
          padj
59  0.00980764
88  0.00980764
191 0.00980764
661 0.00980764
68  0.00980764
859 0.00980764
```

This concludes our example. It can be seen by the above, how DESeq can provide in a few lines a complete test for differential gene expression given raw RNA-seq data.

Chapter 6

Data Simulation

We now turn our focus towards the simulation of the data which was used to compare edgeR, DESeq, and baySeq. First, we will discuss the approach used for data simulation, that is, the design of the data simulation and how the counts were modeled and parameters were estimated. Second, we will discuss the results produced by the edgeR, DESeq, and baySeq on the simulated data. Finally, we will summarize the results and discuss what general conclusions may be inferred regarding a comparison between the methods. It is important to note that we are no longer considering the data simulation used in Chapters 3, 4, and 5.

6.1 Methods for Data Generation

Our goal is to duplicate the approach for simulating data outlined by Soneson and Delorenzi [2013]. In turn, it is important to note that the work of Soneson and Delorenzi [2013] is based also on Robles et al. [2012]. The simulation was accomplished by using the package compcodeR in R found at `bcf.isb-sib.ch//data//compcodeR`. Let us now discuss how the the data was simulated.

6.1.1 Design of Data Simulation

The experiment is based on balanced pairwise experimental design. That is, the data is comprised of replicates which are evenly divided between two conditions S_1 and S_2. In our implementation, this means $|S_1| = |S_2|$ indicating that each condition has a balanced sample size. For each condition $i \in \{1, 2\}$, we represent samples in S_i such that $S_i = \{s_{i1}, ..., s_{iT}\}$ where T is the size of replicates. Let G denote the set of genes such that $G = \{g_1, ..., g_N\}$, where $N = 12,500$. Of these genes, 10% were selected to be differentially expressed. The first sample set can be considered a control, while the second sample set can be considered some sort of unusual phenotype [Soneson and Delorenzi, 2013]. Soneson and Delorenzi [2013] denote $G_{DE}^{up} \subseteq G$ as set of differentially expressed genes between S_1 and S_2 where genes are upregulated in S_2. Conversely, $G_{DE}^{down} \subseteq G$ denotes the set of differentially expressed genes in S_2 that were downregulated. In our experiment, both S_1 and S_2 have replicate size

5. Furthermore, based on results demonstrated by Robles et al. we replicated the data simulation 12 times, as this replicate size is optimum in reducing false discovery rates and conversely having a demonstrably higher true discovery rate compared to smaller replicate sizes. In summary, this means we have a data set which is comprised of 12,500 genes and 5 replicates for each condition. There are $1,250$ differentially expressed genes. In general, we can visualize the data generated as follows:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} \end{pmatrix}$$

Figure 6.1: *Balanced pairwise design with two conditions with two biological replicates where a_{ij} represents count for gene i and sample j*

6.1.2 Counts as Negative Binomial Model

First,we consider how the gene counts can be modeled. Note that Y_{gi} represents a negative binomial random variable count for gene g and replicate i. Specifically, we assume that it is distributed as described below:

$$Y_{gi} \sim \mathcal{NB}(\mu_{gi}, \mu_{gi}(1 + \mu_{gi}\phi_{gi}))$$

where the mean is μ_{gi} and the variance is $\mu_{gi}(1 + \mu_{gi}\phi_{gi})$. The parameter ϕ_{gi} is called the overdispersion parameter. The mean is estimated by,

$$\mu_{gi} = E[Y_{gi}] = \frac{\hat{\mu}_{gc(i)}}{\sum_{g \in G} \hat{\mu}_{gc(i)}} M_i$$

Soneson and Delorenzi [2013] define M_i as the sequencing depth where $M_i = 10^7 U_i$ for $U_i \sim \mathcal{U}[0.7, 1.4]$ and $c(i) \in \{S_1, S_1\}$. That is, $c(i)$ represents either the control condition S_1 or the abnormal condition S_2. We defined $\phi_{gi} = \phi_g$ for all samples. A discussion of how $\hat{\mu}_{gc(i)}$ and ϕ_{gi} were estimated is provided in the next section. Finally, we define $\hat{\mu}_{gS_2} = \gamma_g^{v_g} \hat{\mu}_{gS_1}$ given $\gamma_g = 1.5 + e^1$ where,

$$v_g = \begin{cases} 1 & \text{if } g \in G_{DE}^{up} \\ -1 & \text{if } g \in G_{DE}^{down} \\ 0 & \text{otherwise} \end{cases}$$

This means if the gene is differentially expressed and upregulated, it will increase the count value for $\hat{\mu}_{gS_2} > \hat{\mu}_{gS_1}$. Conversely, if the gene is differentially expressed but downregulated then $\hat{\mu}_{gS_2} < \hat{\mu}_{gS_1}$. If there is no differential expression, then it follows that $\lambda_{gS_2} = \lambda_{gS_1}$.

6.1.3 Estimation of $\hat{\mu}_{gc(i)}$ and $\hat{\phi}_{gi}$

The parameters $\hat{\mu}_{gc(i)}$ and $\hat{\phi}_{gi}$ are estimated by the R package **compcodeR**. The authors Soneson and Delorenzi [2013] estimate μ and ϕ parameters for simulated data by using real RNA-seq data. They obtained this real RNA-seq data from two sources. The first dataset is from **tweeDEseqCountData** by Prickrell et al.. The second dataset is from from Frazee et al. at `http://bowtie-bio.sourceforge.net/recount/`.
The Prickrell et al. [2010] dataset consists of RNA-seq from 69 Nigerian individuals who are not related. In the second dataset by Frazee et al. [2012], Soneson and Delorenzi only used the Cheung data which consists of 41 Caucasian individuals of European ancestry. The following approach was conducted by Soneson and Delorenzi [2013] on both datasets to estimate the μ and ϕ parameters from real data. First, filtering was performed on the data by removing all samples for which the library size was smaller than 2 million reads. Also, they filtered out genes whose average count across all given replicates was less than 1. Second, the reads for each sample were resampled so that all library sizes were equal to the smallest library size[Soneson and Delorenzi, 2013]. Third, for each gene, maximum likelihood estimates of $\hat{\mu}$ and ϕ are obtained for N *iid* variable from a Negative Binomial distribution given the $y_1, \cdots, y_N$ counts in the datasets [Robles et al., 2012]

$$L(\mu, \phi | y_1, \cdots, y_N) = \sum_{i=1}^{N} \ln(Pr(Y_i = y_i | \mu, \phi)$$

$$= \sum_{i=1}^{N} \ln(\Gamma(y_i + \frac{1}{\phi})) - N \ln(\Gamma(\frac{1}{\phi})) - \sum_{i=1}^{N} \ln(\Gamma(y_i + 1)) + \sum_{i=1}^{N} y_i \ln(\frac{\mu\phi}{1 + \mu\phi}) - \frac{N}{\phi} \ln(1 + \mu\phi)$$

Solving for the equation above, one can obtain the MLE of μ_g as follows, as the average count for a given gene across all samples

$$\hat{\mu_g} = \frac{1}{N} \sum_{i=1}^{N} Y_{gi}$$

Similarly, we can find the MLE of ϕ_g by solving for ϕ by maximizing the equation given above for the counts $y_1, \cdots, y_N$. Soneson and Delorenzi obtained the estimates of μ_g and ϕ_g from Prickrell et al. [2010] and Frazee et al. [2012] data sets. The authors combined the estimates of μ_g and ϕ_g from both of these data sets into set of parameters. They then sampled pairs of (μ_g, ϕ_g) from the dataset to obtain estimates for the simulation.

6.2 Results

We compared three methods for differential expression analysis on simulated RNA-seq data. All three methods considered will accept raw count data for analysis. The methods were compared solely on simulated data. This is meaningful since we understand how the data was precisely modeled. As described above, the data was simulated from counts using the Negative Binomial distribution where the mean and dispersion were estimated from real

data. We performed the data simulation for 12 replications. Our focus in this study was to examine how the three methods could detect true differentially expressed genes over non-differentially expressed genes. Hence, we produced a Receiver Operating Characteristic (ROC) curve and a boxplot of AUC values to summarize the AUC values across all data set replicates included in our comparison. Finally, we examined a false discovery curve to depict the number of false positives encountered while stepping through a list of genes ranked by their p-values which represent their statistical significance [Soneson and Delorenzi, 2013]. The ROC curve is a comparison of the percentage of true positives (TPR) out of total positives vs the percentage of false positives (FPR) out of the actual total negative at given different thresholds. In other words, it is ideal to have a high TPR vs a low FPR. We compare edgeR, DESeq, and baySeq in Figure 6.1.

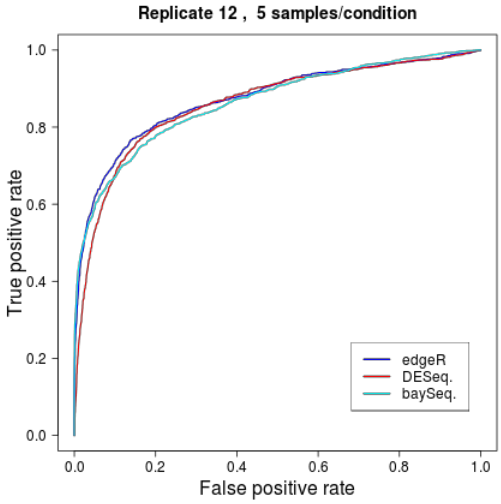

Figure 6.2: *Comparing edgeR, DESeq, and baySeq when* $625 \in G_{DE}^{up}$ *and* $625 \in G_{DE}^{up}$ *showing comparable ROC curves.*

We can see that both edgeR and DESeq outperform baySeq slightly here. However, they are fairly comparable as the AUC values in the boxplots will reveal in Figure 6.2.
The x-axis labeled AUC represents the AUC values observed in Figure 6.1. edgeR slightly outperforms DESeq and baySeq. This can be seen because edgeR value is higher than both DESeq and baySeq. DESeq and baySeq performed almost equivalently under these experimental conditions. Next, we looked at the highest ranked p-values and observed the false discovery rate (FDR) as can be seen in Figure 6.3.
All methods compared FDR increased with the number of genes selected. DESeq has the highest FDR rate regardless of the number of genes selected. EdgeR performed in the middle, while baySeq has the lowest number of false discoveries. These results are consistent with other results reported in the literature [Soneson and Delorenzi, 2013]. We will discuss relevant work in the next chapter.

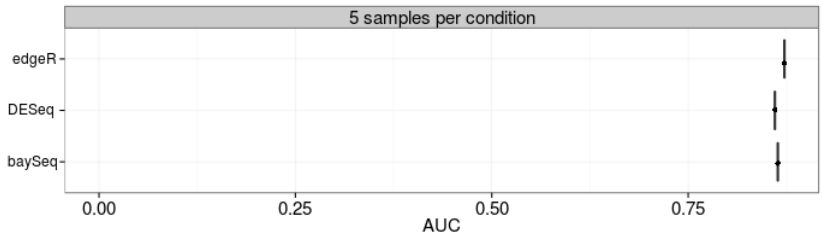

Figure 6.3: *Comparing edgeR, DESeq, and baySeq when* $625 \in G_{DE}^{up}$ *and* $625 \in G_{DE}^{up}$ *showing edgeR has highest AUC value over others.*

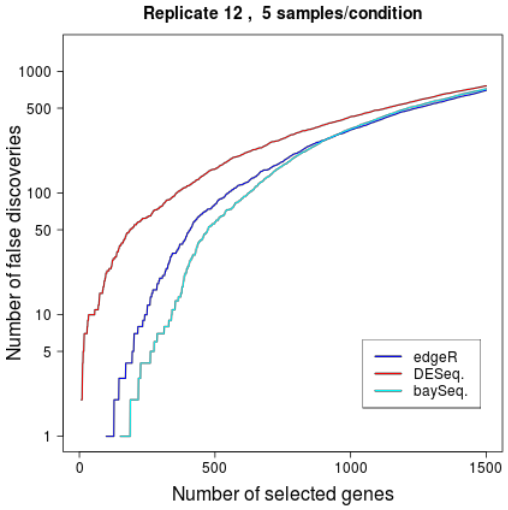

Figure 6.4: *Comparing FDR given number of selected genes for edgeR, DESeq, and bayseq when* $625 \in G_{DE}^{up}$ *and* $625 \in G_{DE}^{up}$.

Chapter 7

Analysis of Red Flour Beetle Data

7.1 Red Beetle Data

The data analyzed was collected from the red four beetle, also known by its scientific name as *Tribolium castaneum*. There were two treatments considered respectively: starved male and fed male. They were separated from pupae and not mated. Each beetle was individually placed in a tube. The beetles were starved 24 hours before a treatment.

Data	
Treatment	# of Seq. Reads
Male Starved	9,878K
Male Fed	5,571K

Figure 7.1: *Red flour beetle raw data with starved male and fed male*

It is of biological interest that the male beetle produces a pheromone, when fed, which attracts female beetles. mRNA was collected from the abdomen of the red flour beetle and sequenced by Illumina. The experiment was interested in measuring potential differential gene expression between the fed and starved male beetle [Park, 2013].

7.2 Galaxy Pipeline Flow Analysis

Since the data had no biological replicates, we could not apply baySeq, edgeR, or DESeq to the data. Instead, we used Galaxy to analyze the data, which can be applied to data with no biological replicates [Goecks et al., 2010], [Blankenberg et al., 2010], [Giardine et al., 2005]. The analysis flow of Galaxy is a series of steps where raw reads are converted into data where differential gene expression analysis can then be conducted. First, raw reads are examined for basic quality control; quality was assessed with the use of box plots. If necessary, data is trimmed and cleaned. Second, the processed reads will be aligned to the

red beetle genome. Galaxy uses Tophat to help align the reads to our reference genome. After this, we use Cufflinks to assemble the transcripts and test for differential analysis.

7.2.1 Quality Control

The first necessary step in the analysis was to compute basic quality statistics. That is, create summary statistics with a boxplot that has a base pair quality score for each set of reads. For each read, we have three scores: quartile range, median score, and outliers. The quartile range will indicate where 50% of the data scores are distributed. The median score will report where half of the scores are above or below without being susceptible to outliers. The outlier scores indicate unusual score values for the read position. We now examine the boxplot of raw read scores starved male and fed male. See Figure 7.1 and Figure 7.2.

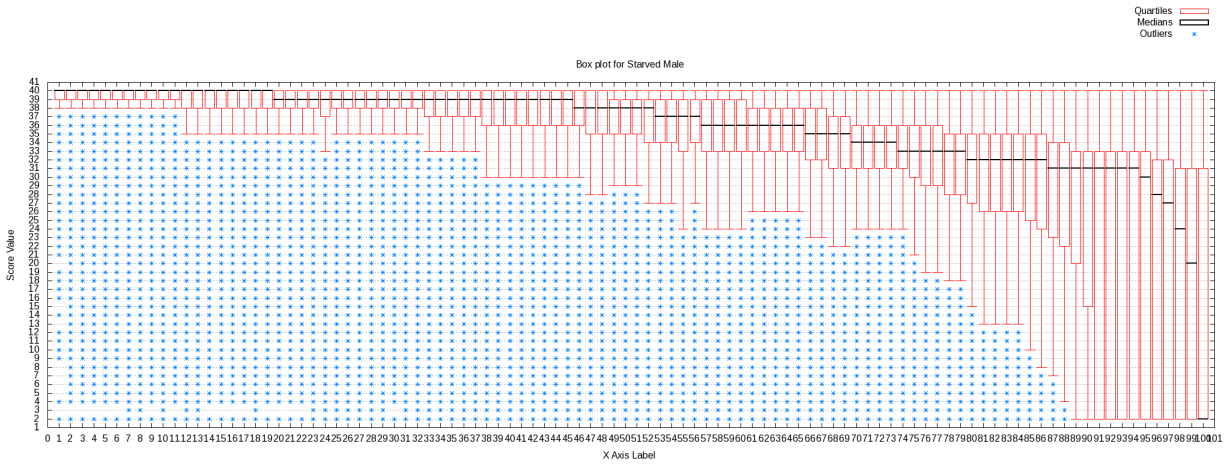

Figure 7.2: *Box plot of raw reads for starved male. Median and quantile score decline observed.*

In Figure 7.2 the median scores are consistently above a score of thirty for most of the data. However, the median and quantile scores decline toward the end. There are also outliers present for every base pair read. In Figure 7.3 the median scores are stable for most of the base pair reads, but decline below 30 range at the end. The quantile range continues to expand. Thus, it is necessary to trim these files to improve median and quantile ranges. We do this by using the sliding window trimmer. This simply takes the observations and calculates the mean score. If the observation causes a score to go below an acceptable mean score, then it is discarded. See the cleaned box plots in Figure 7.3 and Figure 7.4.The median scores are all above 30 for all bp read scores in Figure 7.4. The quantile ranges have also been reduced indicating an improvement in the ranges of the data. Thus, it appears that cleaning the data was successful. In Figure 7.5 the data has been cleaned for fed male as well. It can be seen that the boxplot median scores and quantile ranges have improved. That is, the scores are consistently above 30 and the quantile ranges have been reduced. Thus, it follows that the data has been cleaned and improved.

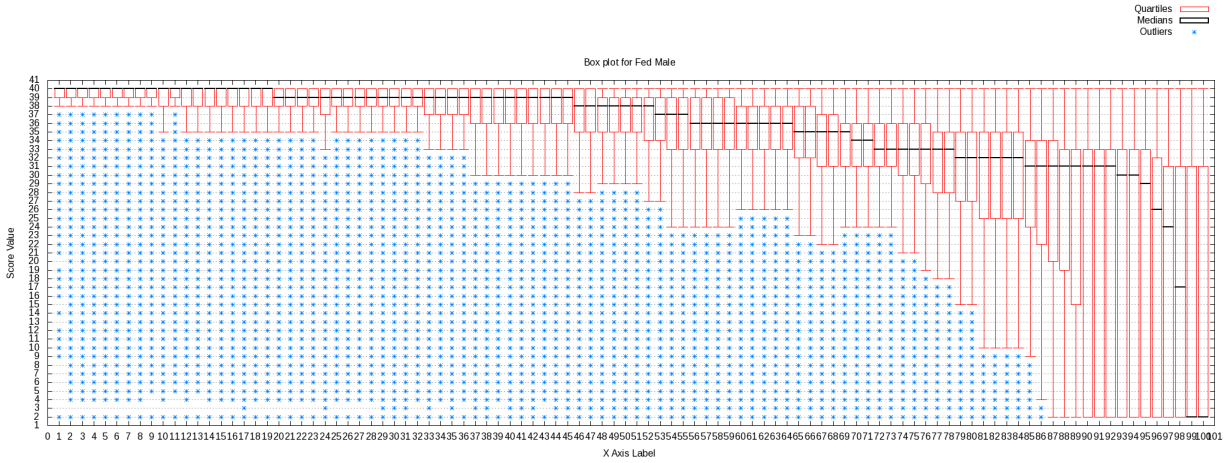

Figure 7.3: *Box plot of raw reads for fed male. Median and quantile score decline observed.*

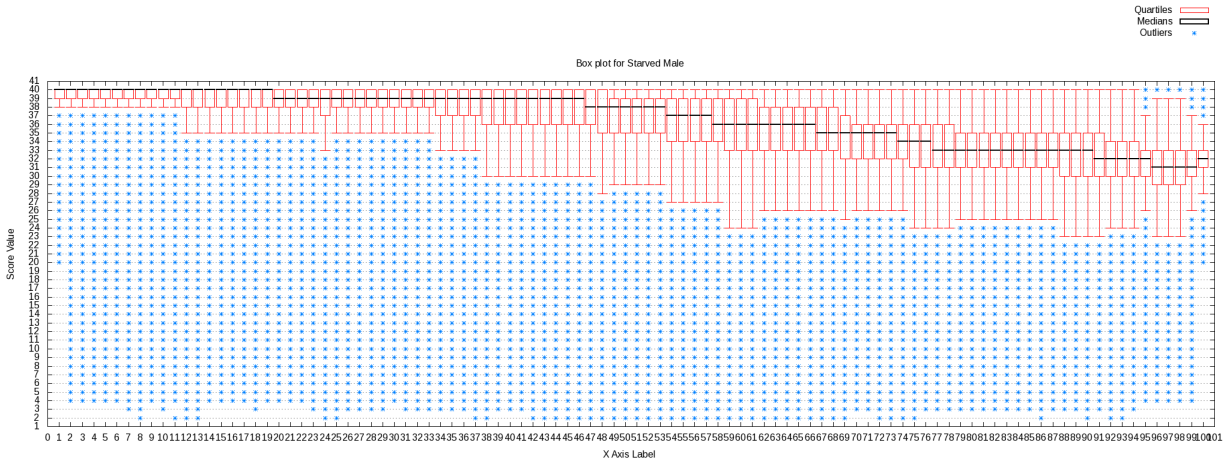

Figure 7.4: *Box plot of processed reads for starved male. Improved median and quantile scores.*

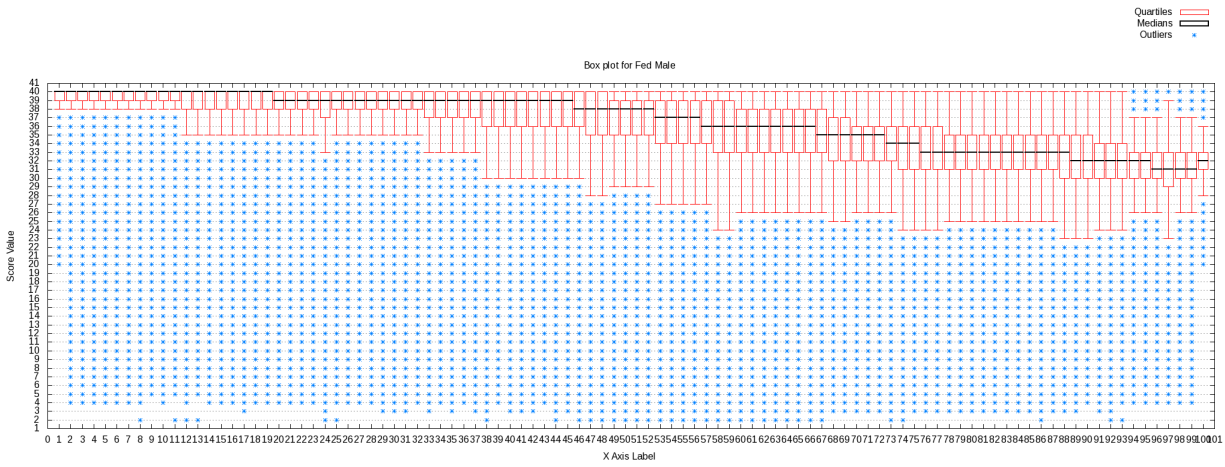

Figure 7.5: *Box plot of processed reads for fed male. Improved median and quantile scores.*

7.2.2 Aligning Reads Using Tophat and Cufflinks

Tophat accepts the RNA-seq data as input. This algorithm and program aligns the input to a reference genome in order to identify exon splice junctions [Trapnell et al., 2012]. Cufflinks is the next tool in the analysis flow and requires output from Tophat [Roberts et al., 2005]. It is a powerful tool which allows the analyst to assemble transcripts and test for differential gene expression. It is arguable one of the most well known. Hence, it is instrumental in allowing us to assess differential gene expression in our data. The analysis pipeline was implemented as shown in Figure 7.6. Cuffdiff analysis can find significant changes in the transcript expression, splicing, and the promoter use [Brown, 2013]. That is, alternative splicing transcription start sites.

7.3 CummeRbund Analysis for Differential Expression

We wish to determine from the results above what differential gene expression, if any, has been detected in the data. First, we count the genes that have statistically significant p-value. Second, we visualize the data and the genes. In order to visualize difference we use cummeRbund `compbio.mit.edu\cummeRbund` to generate various plots. We consider three plots consecutively: density plot, scatter plot, and the volcano plot. See Appendix D for further details.

First, we wished to visualize the data as a distribution in Figure 7.7. The density plot can be thought of as a smoothed representation of a histogram to examine the distributions of FPMK values across gene samples. Figure 7.7 density plot shows that the distribution of FPMK scores across gene samples is extremely similar. The shape for starved male and fed male are similar. Therefore, this is an indication that their distributions are very similar and visually confirms our observation that there are not many differentially expressed genes between the two biological conditions.

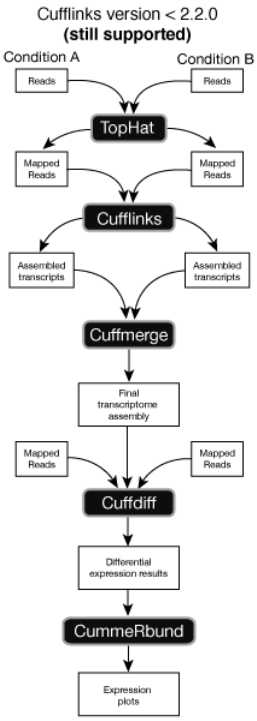

Figure 7.6: *DE analysis flow by Trapnell et al. [2014]. Reprinted with Permission.*

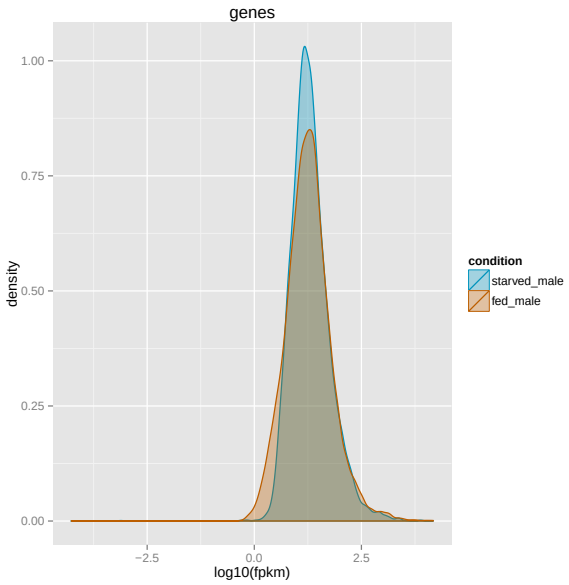

Figure 7.7: *CummeRbund density plot of genes comparing starved vs. fed male showing similar distribution.*

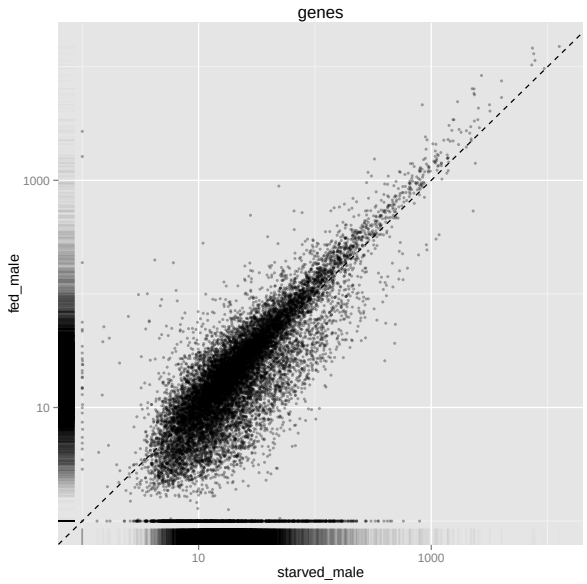

Figure 7.8: *Scatter plot comparing starved male vs. fed male showing general similarities.*

The next graph we consider is a scatter plot in Figure 7.8. It compares the FPKM values from the starved male and fed male gene samples. The majority of scores follow a linear pattern across the gene samples. Hence, it follows that the graph above shows general similarities between starved male and fed male given FPKM scores and some outliers. If there was differential expression present between starved male and fed male, then we would expect to see a series of dots following a parallel line above or below the diagonal line. No such pattern is observed.

The last plot to be considered is a volcano plot in Figure 7.9. The volcano plot is designed to show changes between large datasets comprised largely of replicate data. The plot compares fold-change in scores on the x-axis vs. an negative logarithmic scaled p-values. Consequently, values found towards the top of the plot that are both on the far left or right represent values of interest. This is because they represent large scale fold changes as p-values with more statistical significance. The results from the above plot indicate no differentially expressed genes. There are some values above that have large scale fold changes, but none have statistically significant p-values. Consequently, the conclusion is this plot shows little change between the starved male and fed male. This is consistent with the other plots observed. Given the results presented, we can only come to the simple conclusion that there is no evidence for differential gene expression between the starved male and the fed male.

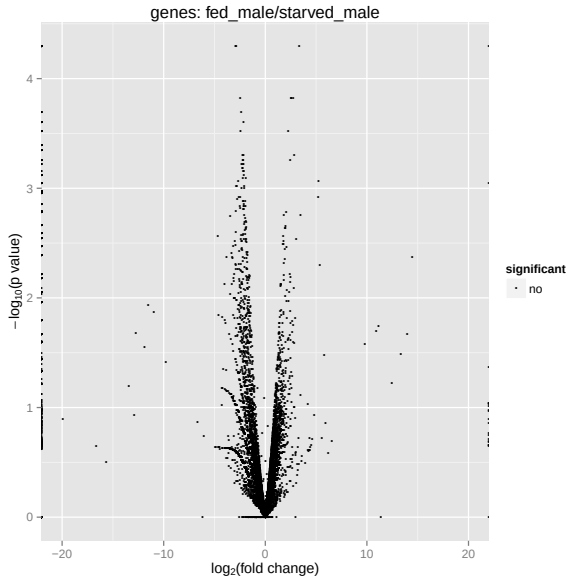

Figure 7.9: *Volcano plot comparing starved male vs. fed male showing*

Chapter 8

Relevant Work

8.1 Review of Data Simulations and Comparisons of Methods

We will review three relevant studies which all analyzed NGS RNA-seq data and compared baySeq, edgeR, and DESeq methods. These papers all conducted data simulations which probed the relative merits of these algorithms. We will briefly discuss each paper below.

8.1.1 Soneson Study

The study by Soneson and Delorenzi [2013] is important to this report. Soneson and Delorenzi [2013] wrote the compcodeR package and conducted their own data simulation which compared many tools including baySeq, edgeR, and DESeq methods. The authors conducted a data simulation which varied the sample sizes and replicates. They also simulated distributions such as Poisson and Negative Binomial. For data modeled as Negative Binomial, edgeR, baySeq, and DESeq were comparable. They only considered a balanced pairwise experimental design. Soneson and Delorenzi [2013] concluded that small sample sizes, such as 2 replicates, which are common in RNA-seq experiments are still a problem for the methods we have considered. They recommended, based on their data simulations with replicates 2, 5, and 10 that experiments be conducted with larger replicates so that the algorithms have lower FDR. [Soneson and Delorenzi, 2013].

8.1.2 Robles Study

The study by Robles et al. [2012] focused on generating simulated NGS RNA-seq data from Negative Binomial distribution. Robles et al. [2012] compared DESeq and edgeR and studied different experimental designs that focused on biological replicates and sequencing depth. They made an interesting observation that reduction of sequencing depth by 15% could still avoid negative effects on FDR and TPR [Robles et al., 2012]. In other words,

Robles et al. [2012] suggested that biological replicates are more influential in experimental design than sequencing depth.

8.1.3 Kvam Study

The authors Kvam et al. [2012] conducted a sophisticated statistical data simulation comparing edgeR, DESeq, and baySeq. The authors compared the methods given under different distributions. The first was that the counts were generated from a Negative Binomial distribution. The second was that the data was generated from a Poisson distribution. The authors explored FDR rates and concluded that none of the methods were controlled well under the conditions examined [Kvam et al., 2012]. Furthermore, the authors concluded that the methods baySeq, edgeR, and DESeq were comparable. Kvam et al. [2012] claimed that baySeq performed well in detecting highly expressed genes. That is, when a gene is upregulated and downregulated across biological conditions.

8.2 Review of Relevant Work

We summarize the results of the analyses done by Robles et al. [2012], Soneson and Delorenzi [2013], and Kvam et al. [2012] as described above. The consensus is that edgeR, baySeq, and DESeq were fairly comparable to each other. Thus, the choice in choosing one method over the other should be motivated by experimental design and other factors to ease the analysis for the scientist.

8.2.1 BaySeq

The strengths of baySeq is that it is excellent in detecting differential expression when genes are regulated in different directions [Kvam et al., 2012]. There is consensus that it performs well with large replication and low outliers in controlling for FDR [Soneson and Delorenzi, 2013]. There are, however, some acute and distinct weaknesses. It is the most complicated and untenable algorithm to understand. Thus, it comes as no surprise that it is computationally slow. It has a lower TPR and FDR for around two replicates then would be observed for replicates around 5 or 10 [Soneson and Delorenzi, 2013] .

8.2.2 EdgeR

The strength of edgeR is generally higher TPR rate compared to other methods. It is also robust in dealing with low sample sizes and performs well with outliers in this situation as well. As for weaknesses, it has medium computation time requirements [Soneson and Delorenzi, 2013]. A weakness is that the FDR is larger when data contains outliers [Soneson and Delorenzi, 2013].

8.2.3 DESeq

The strengths of DESeq is that it can be a conservative estimation algorithm [Soneson and Delorenzi, 2013]. It is also a fairly quick algorithm in comparison to the others, but time efficiency degrades proportional to the size of data. Given a large sample size, it has a larger FDR rate than would be observed for smaller sample size [Soneson and Delorenzi, 2013]. In this sense it has similar behavior to baySeq. The FDR rate can increase when replicate size is below 5.

Chapter 9

Future Work and Conclusion

9.1 Future Work

It is now appropriate to discuss the framework for future work. First, it is important to note that we limited ourselves to pairwise balanced experimental design in order to duplicate and simulate work prevalent in the field. However, methods such as baySeq support experimental designs that are multifactor. Furthermore, other methods are extending their abilities to account for more complicated designs. Hence, it is worth exploring more complicated experiments in simulation to see if the methods still remain comparable.

The data simulation in this report utilized the compcodeR package simulate raw data counts. The compcodeR package is easy to use, versatile, and is a powerful tool. Nevertheless, it is of future interest to explore simulations using the R statistical programming language directly. Also, it is important to explore not only Poisson and Negative Binomial generated data but more hierarchical data in an effort to simulate realistic data better. As NGS RNA-seq technology and data continues to evolve, this is a distinct area of future potential in future research.

An interesting consensus from this study and those explored is that FDR can be addressed by using a large number of biological replicates. That is, something around 5 or more replicates. However, as we discussed above, small samples often lead to methods having elevated FDR. It is almost certain the NGS technology will continue to develop, become cheaper, and more common. However, it is unknown if this will impact the ease and cost of having biological replicates. Thus, an area of high interest is to explore how methods might be extended to account for controlling FDR better given lower and more realistic sample numbers. For example, between 2 to 3 replicates.

Finally, as the field continues to develop at a fast pace, it is necessary to expand the scope of this study to new methods that are being proposed and published.

9.2 Conclusion

In this report, we evaluated and compared three NGS methods: edgeR, DESeq, and baySeq for differential gene expression. We simulated data given there are 12,500 genes, where 1,250 genes are differentially expressed, and 5 replicates for two biological conditions. In our simulation, we did not observe any superior method. edgeR and DESeq are based on similar techniques and showed similar ROC, AUC, and FDR rates. baySeq relies on Empirical Bayes methods and had comparable AUC values to edgeR and DESeq. It did have a lower FDR rate compared to edgeR and DESeq. These results are similar to the study conducted by Soneson and Delorenzi [2013].

No differential gene expression was detected in the red flour beetle data. Since the data had no biological replicates, it was necessary to use Cufflinks to analyze the data. The data had two biological conditions: starved male and fed male. We conducted the analysis using the Galaxy project pipeline. We normalized the counts and cleaned the data. The report visualized the results with cummeRbund which revealed that there was no evidence of differential gene expression being present in the data. As a result, we recommend the experiment be conducted again in the future with biological replicates and starving the male for a longer period of time in hopes of observing changes.

This report has raised interesting questions that should be investigated in future work. It is important that simulated data be used to compare new and existing NGS methods. FDR in NGS studies and experiments should be investigated further for replicates less than 3. Furthermore, more complicated experimental designs should be explored as biological experiments continue to evolve and become more sophisticated. This report has been successful in replicating results observed in literature [Soneson and Delorenzi, 2013], [Robles et al., 2012], [Kvam et al., 2012].

Bibliography

Agathman. Collection of basic alternative rna splicing events. Retrieved from http://en.wikipedia.org/wiki/File:Alt_splicing_bestiary2.jpg, 2009a.

Agathman. Splicing overview. Retrieved from http://en.wikipedia.org/wiki/File:Splicing_overview.jpg, 2009b.

S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11:R106, 2010.

D. Blankenberg, G. Von Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor. Galaxy: a web-based genome analysis tool for experimentalists. *Current Protocols in Molecular Biology*, pages 19.10.1–21, 2010.

S. Brown. *Next-Generation DNA-Sequencing Informatics*, volume 1st Edition. New York: Cold Spring Harbor Laboratory Press, 2013.

M. Clamp, B. Fry, M. Kamal, X. Xie, X. Cuff, J. Lin, M. Kellis, K. Lindblad-Toh, and E. Lander. Distringuishing protein-coding and noncoding genes in the human genome. *Proceedings of the National Academy of Sciences*, 104(49):19428–19433, 2007.

W. W. Cohen. *A Computer Scientist's Guide to Cell Biology*, volume 1st Edition. New York: Springer Science, 2007.

OpenStax College. Rna splicing overview. Retrieved from http://en.wikipedia.org/wiki/File:0326_Splicing.jpg, 2013.

A. C. Frazee, B. Langmead, and J. T. Leek. Recount:a multi-experiment resource of analysis-ready rna-seq gene count datasets. *BMC bioinformatics*, 12(1):449, 2012.

B. Giardine, C. Riemer, RC. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, and J. Taylor. Galaxy: a platform for interactive large-scale genome analysis. *Genome Research*, 15(10):1451–5, 2005.

S. F. Gilbert. *Development Biology*, volume 7th Edition. Sunderland: Sinauer Associates, 2000.

J. Goecks, J. Nekrutenko, A. Taylor, and Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8):R86, 2010.

T. J. Hardcastle and K. Kelly. bayseq: Empirical bayesian methods for identifying differential expression in sequence count data. *BMC bioinformatics*, 11(1):422, 2010.

D. Horspool. Central dogma of molecular biochemistry with enzymes. Retrieved from http://en.wikipedia.org/wiki/File:Central_Dogma_of_Molecular_Biochemistry_with_Enzymes.jpg, 2014.

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

Madprime. Chemical structure of dna. Retrieved from http://en.wikipedia.org/wiki/File:DNA_chemical_structure.svg, 2014.

O. Morozova and M. A. Marra. Applications of next-generation sequencing technologies in functional genomics. *Genomics*, 92(5):255–264, 2008.

Y. Park. Red flour beetle data, 2013. Unpublished Data.

J. K. Prickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nikadori, and J. K. Pritchard. Understanding mechanisms underlying human gene expression variation with rna sequencing. *Nature*, 464(7289):768–772, 2010.

A. Roberts, H. Pimentel, C. Trapnell, and L. Pachter. Identification of novel transcripts in annotated genomes using rna-seq. *Bioinformatics*, 27(17):2325–2329, 2005.

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

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

F. Sanger and A. Coulson. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. *Nature*, 94(3):441–448, 1975.

Schlurcher. Negative binomial. Retrieved from https://commons.wikimedia.org/wiki/File:Negativ_Binomial_Distribution.PNG, 2009.

J. Shendure and J. Hanlee. Next-generation dna sequencing. *Nature*, 26(10):1135–1145, 2008.

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

C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. Kelly, H. Pimentel, S. Salzberg, J. Rinn, and L. Pachter. Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks. *Nature Protocols*, 7:562–578, 2012.

C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. Kelly, H. Pimentel, S. Salzberg, J. Rinn, and L. Pachter. Cufflinks. Retrieved from http://cufflinks.cbcb.umd.edu/, 2014. Used with permission. All Rights Reserved.

J. Zhang, R. Chiodini, A. Badr, and G. Zhang. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 38:95–109, 2011.

M. Zvelebil and J. Baum. *Understanding Bioinformatics*, volume 1st Edition. London; New York: Garland Science, 2007.

Appendix A

Negative Binomial Distribution

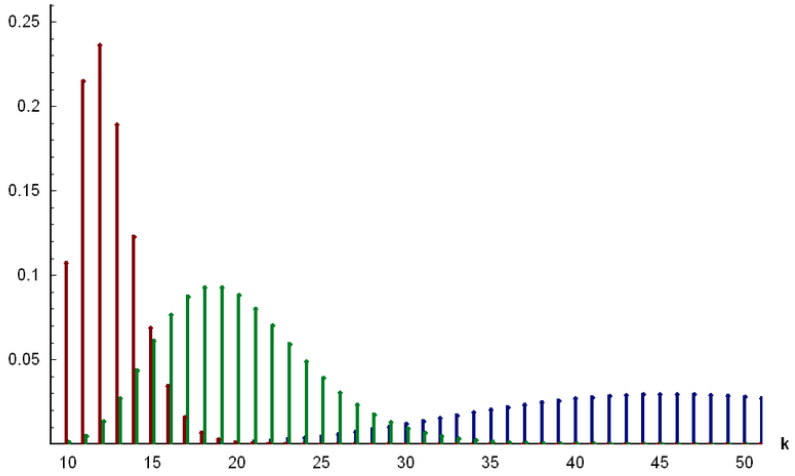

Figure A.1: *Distribution of Negative Binomial with* $\mu = 10$, $\phi = 0.8$*(red),* $\phi = 0.2$*(blue), and* $\phi = 0.5$*(green). Creative Commons Schlurcher [2009].*

The negative binomial is a discrete probability distribution which has two parameters, the mean and the dispersion. We model counts in RNA-seq experiments as Y as follows:

$$Y \sim NB(\mu, \phi)$$

The distribution is useful in situations where RNA-seq has biological replicates where the dispersion expands quicker than the mean [Anders and Huber, 2010]. The negative binomial estimates the expected mean as $\frac{\phi\mu}{1-\phi}$ and expected dispersion as $\frac{\phi\mu}{(1-\phi)^2}$.

Appendix B

NGS Tools Usage in R

We simulated data to demonstrate the usage of baySeq, edgeR, and DESeq tools in R as shown below. The approach was based on the data simulation as outlined by Hardcastle and Kelly [2010]. The notable difference is the means are estimated much simpler. We assume a balanced pairwise design with two biological conditions with 2 replicates each. The first 10% of genes are simulated to are assumed to be differentially expressed with mean 5 for the first condition and mean 25 for the second. The dispersion is not assumed to be consistent and distributed from a gamma distribution. For the non-differentially expressed genes, we assumed a mean of 10 and an overall dispersion of 0.5. The R code is provided below:

```
#Define the seed
set.seed(2003)

#Define the simulation parameters
n = 10000
samples = 8
percentDE = 0.10
count1 = rnorm(n, mean = 5, sd = 1)
count2 = rnorm(n, mean = 25, sd = 1)
count3 = rnorm(n, mean = 10, sd = 1)
dispersions = rgamma(n, shape = 0.85, scale = 0.5)

y = matrix(0, nrow = n, ncol = samples)

for(i in 1:(percentDE*n))
{

  for(j in 1:(samples/2))
  {
```

```
    y[i,j] = rnbinom(1, size = 1/dispersions[i], mu = count1[i])
  }
  for(j in ((samples/2) + 1): samples)
  {
    y[i,j] = rnbinom(1, size = 1/dispersions[i], mu = count2[i])
  }
}

for(i in ((percentDE * n) + 1): n)
{
  y[i,] = rnbinom(1, size = dispersions[i], mu = count3[i])
}
```

We now turn out attention to the tools baySeq, edgeR, and DESeq usage in R with the simulated data above. See the code below:

```
#baySeq usage in R
#library size
libs <- 4

#define the groups, first for NGE and second for DE, assume balanced size
groups <- list(NDE = rep(1,libs), DE = rep(1:2, each = libs/2))

#Create a count data object of counts with associated group
CD = new("countData", data = y,
     replicates = rep(1:2, each = libs /2), groups = groups)

libsizes(CD) = getLibsizes(CD)
cl = NULL

#Estimate priors and likielihood given a particular model
CD = getPriors.NB(CD, samplesize = 10^5, estimation = "QL", cl = cl)
CD = getLikelihoods.NB(CD, pET = 'BIC', cl = cl)

#Report the highest probabilties of differential expression given "DE" model
results = topCounts(CD, group = "DE", number = 10)
head(results)
```

```
#edgeR usage in R
d <- DGEList(counts = y, group = c(1,1,2,2), lib.size = rep(n, samples))
de <- exactTest(d, dispersion = 0.2)
topTags(de)

#DESeq usage in R
#Conditions of the data
condition <- c("trt1", "trt1", "trt2", "trt2")

#Counts
cds <- newCountDataSet(y, condition)
cds

#Normalization
cds <- estimateSizeFactors(cds)
sizeFactors(cds)

#Variance Estimation
cds <- estimateDispersions(cds)

#Estimate differential expression (currently produces an error)
res <- nbinomTest(cds, "trt1", "trt2")
head(res)
```

Appendix C

Data Simulation Using CompcodeR

The following code in R was used to conduct the data simulation as described in Chapter 8 of the report. See below:

```
library(compcodeR)

n=12
i=1

for(i in 1:n)
{

    printStatement<-paste("dat",i,sep="")
    printStatement
    dir = "~/Dropbox/Programming/masters/
          dataSim/dataSimNoDispersion/simDataSets/"
    dataSet <- "dataSimulation"
    dataSetRSD<-paste(dir,"dat",i,".rds",sep="")
    generateSyntheticData(dataset = dataSet, n.var=12500,
                          samples.per.cond=5,
                          n.diffexp=1250,
                          repl.id = i,
                          seqdepth = 1e+07,
                          fraction.upregulated=0.5,
                          between.group.diffdisp=FALSE,
                          filter.threshold.total =1,
                          filter.threshold.mediancpm = 0,
                          fraction.non.overdispersed = 0,
                          output.file = dataSetRSD)
```

```
}


for(i in 1:n)
{

    #The input data set is the files from above produced
    # by generateSyntheticData method
    dataSetInput <-paste(dir,"dat",i,".rds",sep="")

    #We define the results of our analysis in the following folder
    outputDir = "~/Dropbox/Programming/masters/dataSim/
                 dataSimNoDispersion/simDataDEResults/"

    #1. edgeR analysis
    runDiffExp(data.file = dataSetInput,
               result.extent = "edgeR.exact",
               Rmdfunction = "edgeR.exact.createRmd",
               output.directory = outputDir,
               norm.method = "TMM",
               trend.method = "movingave",
               disp.type = "tagwise")

    #We now save the results to be used in comparing our results,
    #deREsults stands for differential express results
    deResults <- c(paste(outputDir,"dat",i,"_edgeR.exact.rds",sep=""))

    #2. DESeq analysis
    runDiffExp(data.file=dataSetInput,
               result.extent = "DESeq.nbinom",
               Rmdfunction = "DESeq.nbinom.createRmd",
               output.directory = outputDir,
               sharing.mode = "fit-only",
               disp.method = "pooled",
               fit.type = "local")

    deResults <- c(deResults,
                  paste(outputDir,"dat",i,"_DESeq.nbinom.rds",sep=""))

    #3. baySeq Analysis
    runDiffExp(data.file=dataSetInput,
               result.extent = "baySeq",
```

```
                Rmdfunction = "baySeq.createRmd",
                output.directory = outputDir,
                norm.method = "quantile",
                distr.choic = "NB",
                equaldisp = TRUE,
                sample.size = 1e5,
                estimation = "QL",
                pET = "BIC")

    #4. Save the output from the analysis in the following
    deResults <- c(deResults,
                   paste(outputDir,"dat",i,"_baySeq.rds",sep=""))
    deResults

}


##############################################
#Compare results                             #
##############################################

#Build a data frame with input files,
#these are the outputs from the results above
file.table <- data.frame (input.files = deResults,
              stringsAsFactors = FALSE)

#Define a list of parameteres for the comparison study.
parameters <- list (inc.nbr.samples = NULL,
                    incl.replicates = NULL,
                    incl.dataset = dataSet,
                    incl.de.methods = NULL,
                    fdr.threshold = 0.05,
                    tpr.threshold = 0.05,
                    typeI.threshold = 0.05,
                    ma.threshold = 0.05,
                    fdc.maxvar = 1500,
                    overlap.threshold = 0.05,
                    fracsign.threshold = 0.05,
                    comparisons = c("auc", "fdcurvesone", "rocone"))
outputDirectory <- "~/Dropbox/Programming/masters/
                   dataSim/dataSimNoDispersion/simDataResults/"
runComparison(file.table = file.table,
              parameters = parameters,
```

```
output.directory = outputDirectory)
```

Appendix D

CummeRbund Code

The following code given below was used for the visual analysis of gene data produced by cufflinks for the red flour beetle data. See below:

```
#Load the library
library("cummeRbund")

#Set the working directory
setwd("/home/hlyates/Dropbox/Programming/masters/cuffdiff/RScripts")
#getwd()


#Step 2: Create a cuffSet object using readCufflinks().
#The files are in tutorial and established as working directory
#cuffData <- readCufflinks("data")
cuffData = readCufflinks(dir = "data", dbFile = "cuffData.db",
           geneFPKM = "genes.fpkm_tracking", geneDiff = "gene_exp.diff")

#Count the significant genes in cummerBund
sigGeneIds <- getSig(cuffData, alpha = 0.05, level = "genes")
length(sigGeneIds)


#Step 3: Plot the density map and export as a PDF file
den <- csDensity(genes(cuffData))

#Now export the plot as a pdf
pdf("myDen.pdf")
plot(den)
dev.off()
```

```
#compare expression of each gene in two conditions with scatter plot
sca <- csScatter(genes(cuffData), "starved_male", "fed_male")
pdf("mySca.pdf")
plot(sca)
dev.off()


#Step 4: Find the significantly differentially expressed genes
#between starved male and fed female, assign to "sigGene"
sigGene <- csVolcano(genes(cuffData), "starved_male", "fed_male")
pdf("sigGene.pdf")
plot(sigGene)
dev.off()
```