An Isoform-free Model for Differential Expression Analysis in RNA-seq Data

A Dissertation Presented

by

Yang Liu

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2016

Stony Brook University

The Graduate School

Yang Liu

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Song Wu – Dissertation Advisor
Assistant Professor, Department of Applied Mathematics and Statistics

Wei Zhu - Chairperson of Defense
Professor, Deputy Chair, Department of Applied Mathematics and Statistics

Jie Yang
Assistant Professor, Department of Preventive Medicine

Nora Galambos – Outside Member
Ph.D., Senior Data Scientist, The Office of Institutional Research, Planning & Effectiveness

This dissertation is accepted by the Graduate School

Charles Taber

Dean of the Graduate School

Abstract of the Dissertation

An Isoform-free Model for Differential Expression Analysis in RNA-seq Data

by

Yang Liu

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2016

Next generation sequencing (NGS) technology has been widely used in biomedical research, particularly on those genomics-related studies. One of the NGS applications is high-throughput mRNA sequencing (RNA-seq), which is usually applied to discover alternative splicing events, to evaluate gene expression level and to identify differentially expressed genes. Compared with the traditional microarrays, RNA-seq is more efficient and economical. Currently, many useful software tools have been developed for RNA-seq differential expression (DE) analyses, such as edgeR, DESeq and Cufflinks; however, all these methods either ignore the isoforms of mRNA transcript, or rely on the predefined isoform structures, or depend on the De Novo isoform reconstruction from the sequencing data, which lead to less accurate inference.

In this thesis, we developed and implemented a novel splicing-graph based negative binomial (SGNB) model for gene differential expression analysis in RNA-seq data. The principle of our model is to change the expression comparisons from the unobservable transcript level to the observable read type level, according to the fundamental theory of the linear algebra. The likelihood ratio test is used for finding DE genes. Computationally, we employed the expectation-maximization (EM) and the Newton-Raphson algorithms for parameter estimation. The main advantage of our model is that it considers the isoform but does not require the pre-defined isoform structure and therefore is expected to be more robust and powerful. At the same time, our method does not ask for the De Novo procedure, which will save the time and avoid errors in reconstructing isoforms.

We performed intensive simulations to compare our new method with one of the most popular package, edgeR. Under various scenarios we examined, the results showed that our new model can achieve better power, while correctly controlling the false discovery rate. We also applied our method to a real data set to demonstrate its applicability in practice.

Table of Contents

Chapter 1 Biological Fundamentals and Next Generation Sequencing 1

1.1 The Central Dogma 1

1.2 Alternative Splicing.. 4

1.3 Introduction of Next Generation Sequencing... 7

1.4 Applications of Next Generation Sequencing 10

Chapter 2 RNA-seq Data Analyses 12

2.1 Background and Differential Expression Analysis 12

2.2 Read Mapping by TopHat 13

2.3 Summarizing the Data 19

Chapter 3 Overview of Existing Statistical Methods for DE analysis 21

3.1 Gene Level Analysis 21

3.1.1 Negative Binomial Model 21

3.2 Transcript Level Analysis 25

3.2.1 Known Transcript Structure 26

3.2.1.1 Poisson Model.. 26

3.2.1.2 Generative Model.. 30

3.2.2 Unknown Transcript Structure 34

3.2.2.1 De Novo Assembling.. 34

3.2.2.2 Transcript Abundance Estimation... 34

Chapter 4 Splicing Graph-based Negative Binomial Model 36

4.1 Motivation 36

4.2 Statistical Modeling.. 37

4.2.1 Notation 37

4.2.2 Annotation File Modification and Read Summary.................................... 38

4.2.3 Statistical Model for One Sample.. 40

4.2.4 Statistical Model for Multiple Samples 44

4.2.5 Splicing Graph based Parameter Reduction 45

4.2.6 Parameter Estimation... 49

4.2.7 Likelihood Ratio Test 55

4.3 Simulation Studies.. 55

4.3.1 Simulation Settings.. 56

4.3.2 Simulation Results 57
4.4 Real Data Analysis 75
Chapter 5 Discussion and Future Work 79

List of Figures

Figure 1-1: The central dogma of molecular biology 1
Figure 1-2: The double-helix DNA structure........ 2
Figure 1-3: Typical structure of mRNA........ 3
Figure 1-4: Genetic code table. 4
Figure 1-5: Transcription process in details........ 5
Figure 1-6: Illustration of how a Pre-mRNA is processed into mRNAs. 5
Figure 1-7: Typical alternative splicing events........ 6
Figure 1-8: NGS workflow. 7
Figure 1-9: Color based sequencing. 8
Figure 1-10: Base calling. 9
Figure 3-1: Bayesian network of sequencing procedure........ 31
Figure 4-1: Gene annotation modification. 39
Figure 4-2: Summarized data table. 39
Figure 4-3: Read types generated from one isoform. 41
Figure 4-4: Splicing graph for parameter reduction. 48
Figure 4-5: Reduced splicing graph. 49
Figure 4-6: Empirical CDF of p-value for single-end read........ 58
Figure 4-7: Empirical CDF of p-value for paired-end read. 59
Figure 4-8: Performance comparisons for single-end read with sample size 2. 61
Figure 4-9: Performance comparisons for single-end read with sample size 4. 62
Figure 4-10: Performance comparisons for single-end read with sample size 8. 63
Figure 4-11: Performance comparisons for single-end read with sample size 16. 64
Figure 4-12: Performance comparisons for paired-end read with sample size 2........ 65
Figure 4-13: Performance comparisons for paired-end read with sample size 4........ 66
Figure 4-14: Performance comparisons for paired-end read with sample size 8........ 67
Figure 4-15: Performance comparisons for paired-end read with sample size 16........ 68
Figure 4-16: Power vs. sample size for single-end read. 69
Figure 4-17: Power vs. sample size for paired-end read........ 70
Figure 4-18: False discovery rate vs. sample size for single-end read. 71
Figure 4-19: False discovery rate vs. sample size for paired-end read. 72
Figure 4-20: False positive rate vs. sample size for single-end read. 73
Figure 4-21: False positive rate vs. sample size for paired-end read........ 74
Figure 4-22: Venn diagram of significant DE genes. 76
Figure 4-23: Histogram of log fold change........ 78

Acknowledgment

I would like to express my sincere gratitude to Prof. Song Wu who is not only my advisor, but also my great friend. I would have never succeeded in finishing the PhD program without his continuous support of my study and research. Under his guidance, I've became much better in statistics, critical thinking and other skills than I entered the program. He is also an important mentor who helped me a lot in my personal life. I could not have imagined having a better advisor for my PhD study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Wei Zhu, Prof. Jie Yang, and Dr. Nora Galambos, for their encouragement and insightful comments.

Also, I would like to thank all my friends at Stony Brook University. It is you guys who gave me a colorful life during my study. I will never forget the happy times that we shared together at Stony Brook.

Last but not the least, I would like to thank my parents, Xingpo Liu and Hua Yang, for giving birth to me and supporting me spiritually throughout my life. And I would like to thank my lovely wife, Lijuan Kang, for accompanying and supporting me during my PhD study.

Chapter 1 Biological Fundamentals and Next Generation Sequencing

Genetics is one of the most important components of biology. It is the science of genes, heredity and variation in living organisms. By studying the molecular structure and functions of genes, people can gain understanding on how genes may affect different biological traits and then benefit from it. For example, being aware of the genetic basis of diseases like cancer may help us develop new strategies for their prevention and treatment. In this chapter, we will introduce essential concepts of gene structure and transcription.

1.1 The Central Dogma

In molecular biology, the central dogma is a principle that reveals the relationship among DNA, RNA and proteins (Crick, 1970). Briefly, it states that DNA makes RNA that in turn makes protein (Figure 1-1).

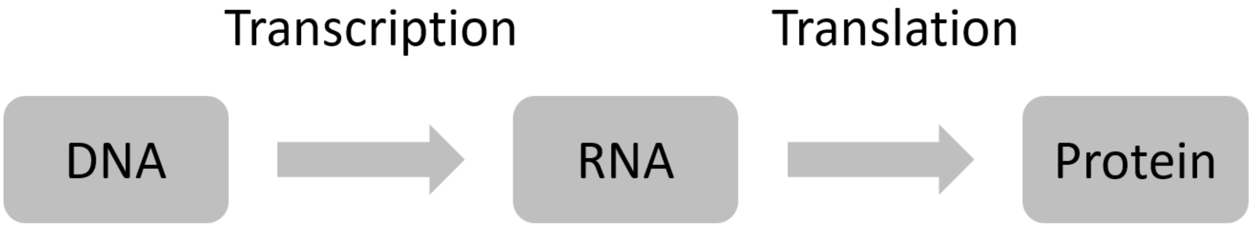


Figure 1-1: The central dogma of molecular biology. The mRNA first transcribed from DNA and then translate to Protein.

It is well known that proteins are the functional units of the human body, and deregulated proteins may cause certain diseases. By the central dogma, the protein malfunction can occur at either DNA or RNA level: at the DNA level, mutations that change the nucleotide sequences could alter protein sequences; at the RNA level, the alternative splicing (details in 1.2) mechanism may induce shifting of the coding sequences, therefore resulting different proteins.

Technically, it is much more challenging to study protein instead of DNA/RNA because of instability of the protein and easy amplification of the nucleotides. Hence, to understand the underlying mechanisms of a biological trait, researchers usually start with the genetic analysis.

In general, there are four types of deoxy nucleotides, called adenine, cytosine, guanine and thymine (uracil in RNA) and abbreviated as A, C, G, T (U), respectively. A and T (or U in RNA), and C and G pair each other to form the double-helix structure (Sinden, 1994) (Figure 1-2).

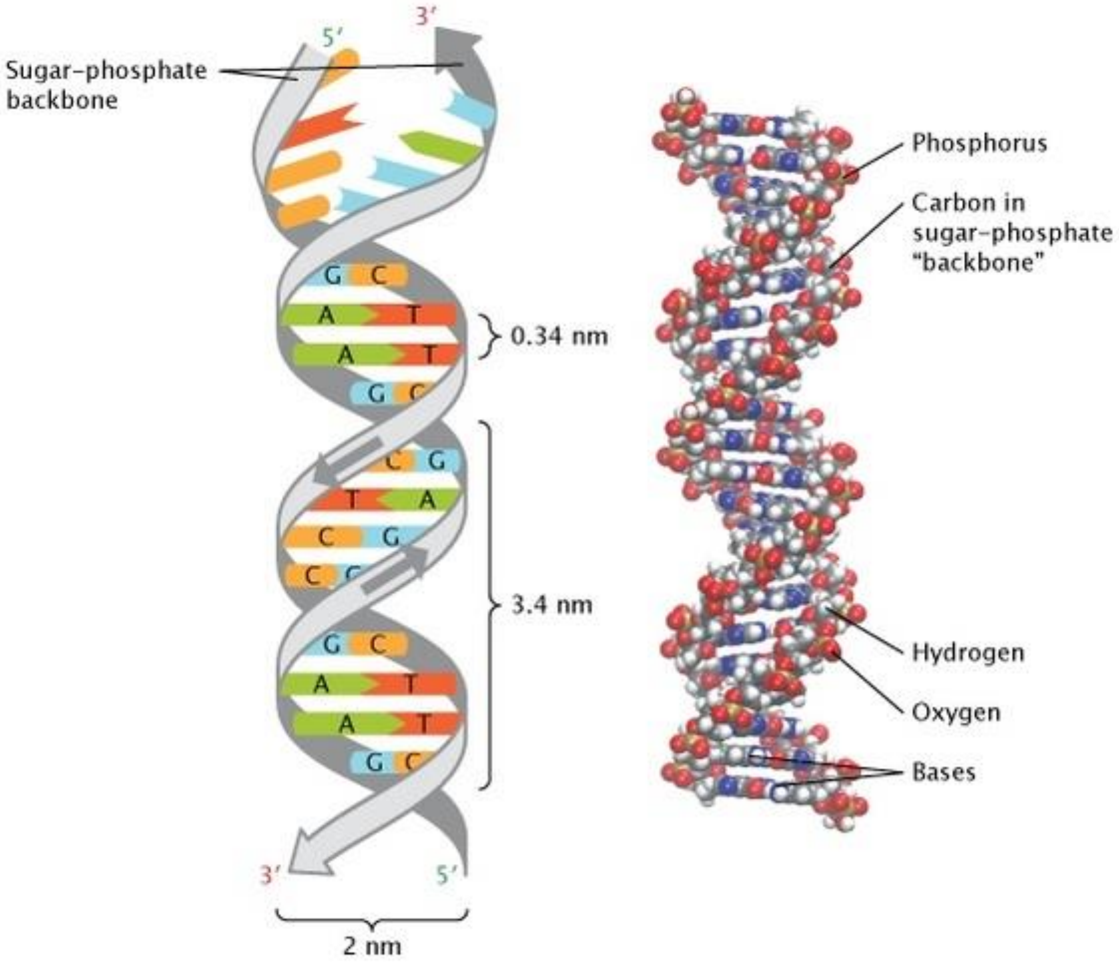


Figure 1-2: The double-helix DNA structure *(Pray, 2008)*.

From DNA to RNA, this step is called transcription, during which the information contained in a DNA sequence is transferred to an RNA sequence. The typical structure of an mRNA molecule includes 5’ cap, 3’ poly(A) tail, 5’ un-translated region (UTR), 3' UTR and coding regions. Only the coding region can be translated into a protein (Figure 1-3).

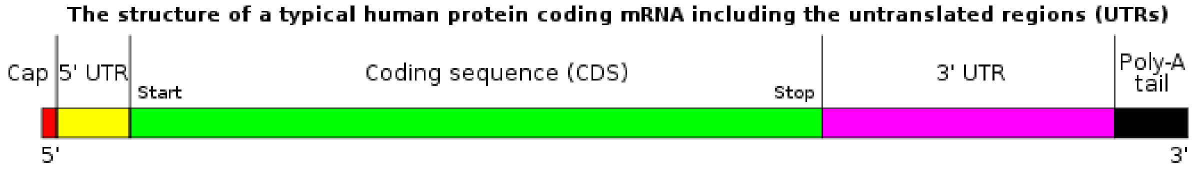
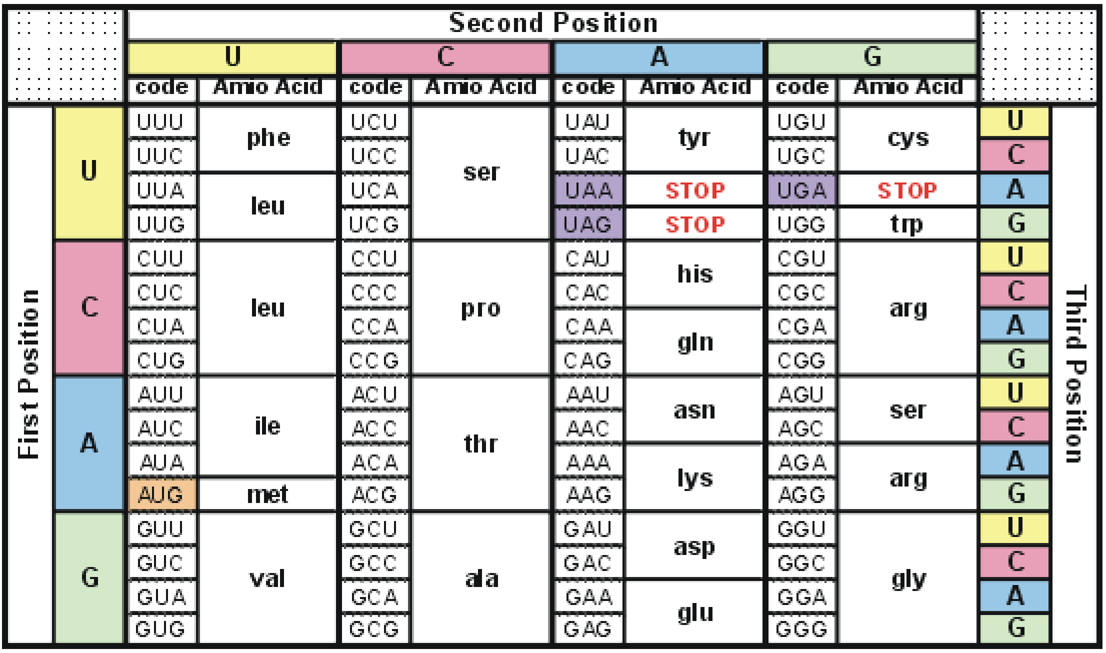

The structure of a typical human protein coding mRNA including the untranslated regions (UTRs)
Cap
5' UTR
Start
Coding sequence (CDS)
Stop
3' UTR
Poly-A tail
5'
3'

Figure 1-3: Typical structure of mRNA. Usually an mRNA starts with a 5’cap and follows by 5’ UTR, CDS, 3’ UTR and Poly-A tail. (https://en.wikipedia.org/wiki/Messenger_RNA)

The genetic information embedded in the sequence of nucleotides is arranged into codons, each of which consists of three bases and encodes a specific amino acid. The amino acids are the basic unit of proteins. The coding region begins with a start codon (AUG) and ends with one of the three stop codons (UAA, UAG, UGA), which denote the starting and ending positions for translation. Since one codon contains three bases and we have four kinds of bases in total, theoretically the number of all possible combinations will be 64, exceeding the total number of amino acids (23). The mapping from the codons to amino acids can be described by a function illustrated in the graph below (Figure 1-4), and some amino acids may have more than one codon. For example, the lysine (lys) can be coded as AAA or AAG.



		Second Position									
		U		C		A		G			
		code	Amio Acid	code	Amio Acid	code	Amio Acid	code	Amio Acid		
First Position	U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U	Third Position
		UUC		UCC		UAC		UGC		C	
		UUA	leu	UCA		UAA	STOP	UGA	STOP	A	
		UUG		UCG		UAG	STOP	UGG	trp	G	
	C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U	
		CUC		CCC		CAC		CGC		C	
		CUA		CCA		CAA	gln	CGA		A	
		CUG		CCG		CAG		CGG		G	
	A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U	
		AUC		ACC		AAC		AGC		C	
		AUA		ACA		AAA	lys	AGA	arg	A	
		AUG	met	ACG		AAG		AGG		G	
	G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U	
		GUC		GCC		GAC		GGC		C	
		GUA		GCA		GAA	glu	GGA		A	
		GUG		GCG		GAG		GGG		G	

Figure 1-4: Genetic code table. Each amino acid is corresponding to at least one codon. And there are one start codon and three stop codons. (http://passel.unl.edu/pages/informationmodule.php?idinformationmodule=956592171&topicorder=7&maxto=13)

1.2 Alternative Splicing

Alternative splicing is a post-transcriptional process that may lead to the generation of multiple proteins from single gene coding region. It is an intermediate step when DNA makes copy to mRNA. In fact, before mRNAs are produced, the pre-mRNAs are generated from DNA first. The pre-mRNAs are then processed to become mRNAs, accompanying the occurrence of alternative splicing (Figure 1-5).

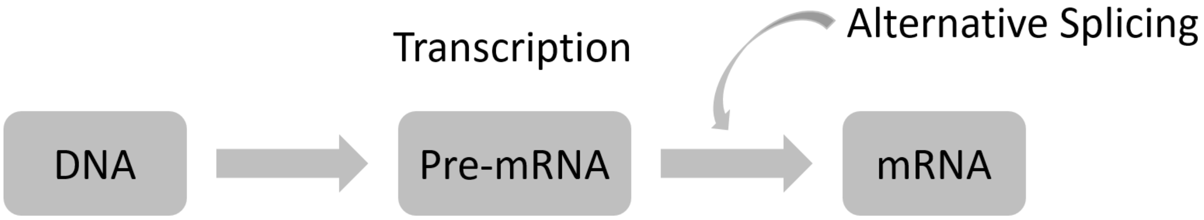


Figure 1-5: Transcription process in details. DNA first create pre-mRNA, and then pre-mRNA makes mRNA during which the alternative splicing occurs.

The pre-mRNA is a direct copy of the DNA and keeps both exons and introns of a gene. However, in mRNA, the introns that do not contain any coding information are excluded and only the exons are kept. During the process of intron exclusion, alternative splicing may occur to form different mRNA molecules (Figure 1-6). Evolutionally, this mechanism can generate larger number of transcripts, which yields better functional variety.

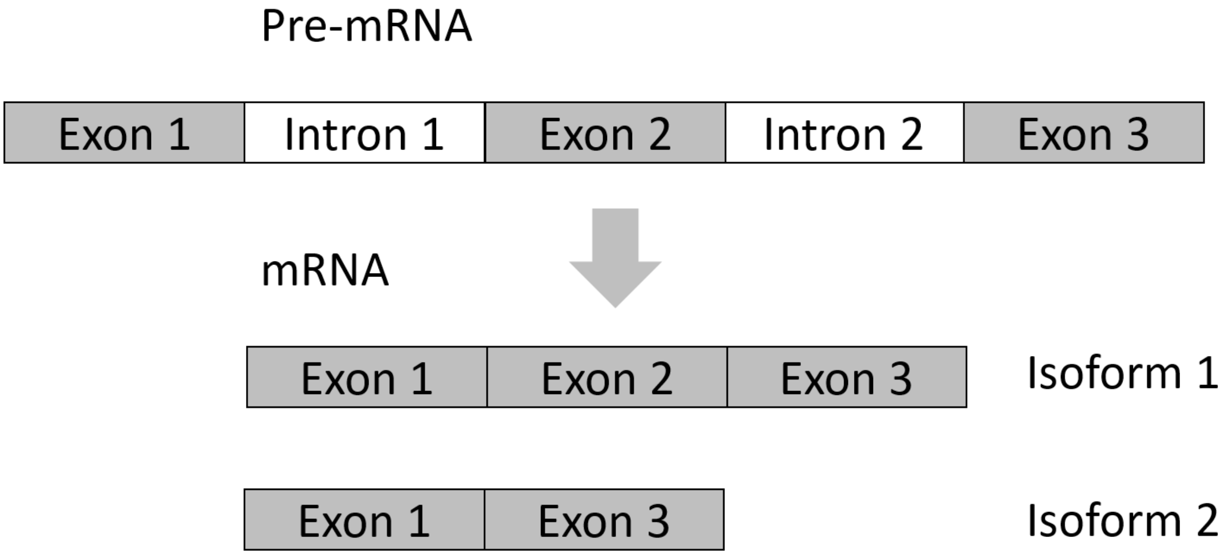


Figure 1-6: Illustration of how a Pre-mRNA is processed into mRNAs. Alternative splicing may happen during this procedure resulting in different isoforms. The isoform 1 is generated by eliminating all the introns, while the isoform 2 is generated due to one type of alternative splicing events, which directly connects exon 1 and exon 3.

Usually, there are five basic types of alternative splicing events, which are exon skipping, mutually exclusive exons, alternative 5’ donor sites, alternative 3’ acceptor sites and intron retention (Figure 1-7).

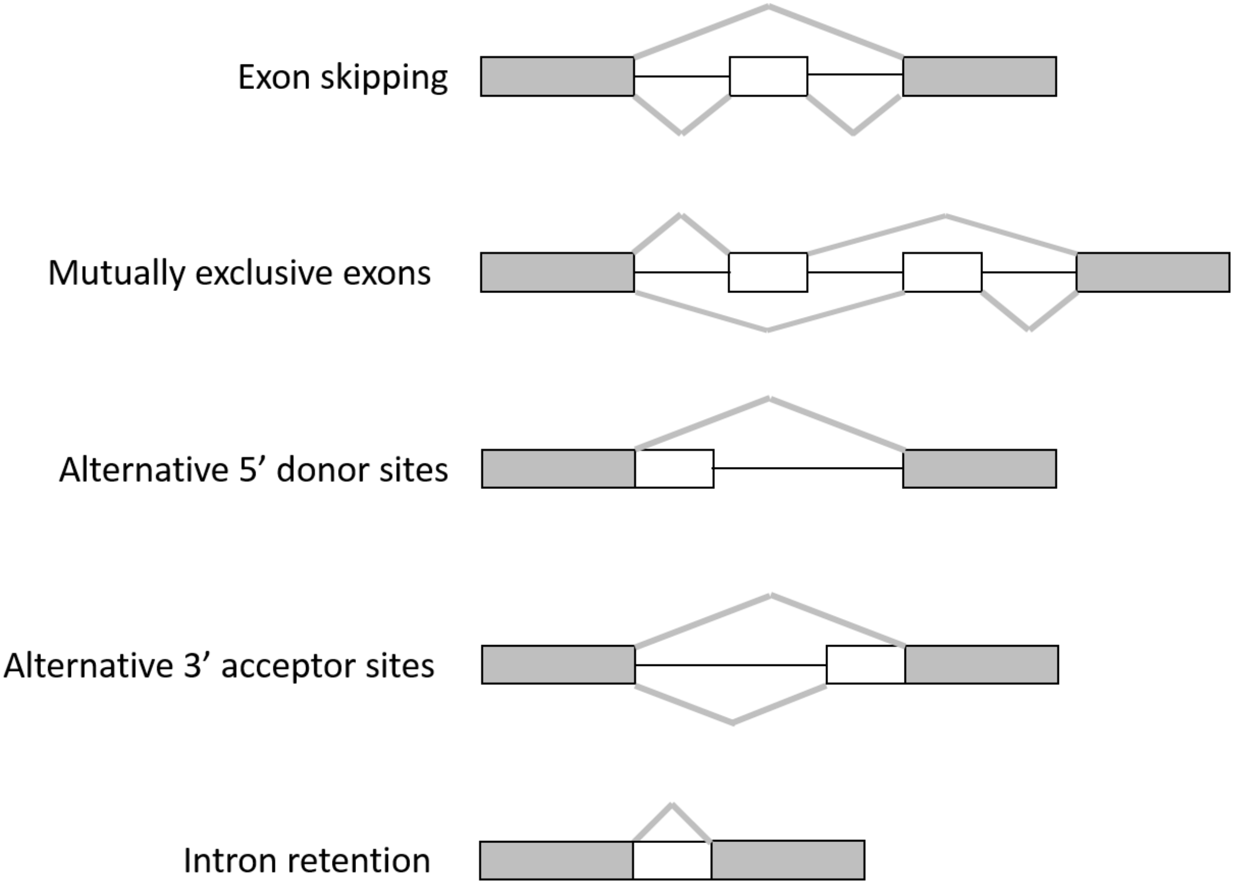


Figure 1-7: Typical alternative splicing events. The graph shows five different ways in which the alternative splicing events can happen. The rectangle with grey background denotes the exon while the one with white background denotes the intron.

The products of the alternative splicing of a pre-mRNA are called isoforms. Although the isoforms come from the same gene, the different structures lead to various proteins with different functions. A famous example is the CDKN2A gene that, through the alternative splicing mechanism, codes two distinct proteins, p16(Ink4) and p19(ARF), which function in two important pathways (Ouelle, Zindy, Ashmun, & Sherr, 1995). There are many other examples showing that isoforms may play critical roles in the development process (Gauthier, et al., 1999). So it is important to identify the isoforms for a gene in a specific tissue and also evaluate their expression levels. Through the RNA-seq, this type of analysis becomes feasible.

1.3 Introduction of Next Generation Sequencing

Next generation sequencing (NGS) is a new technology to sequence DNA, which only became commercially available in 2004. However, due to its extreme power in fast and cheap sequencing, NGS has imposed a big impact on biomedical researches at genomic level.

Currently, there are three main platforms for the next generation sequencing: the Illumina/Solexa Genome Analyzer (GA), the Roche/454 FLX, and the Applied Biosystems $SoLiD^{TM}$ System. Since the Illumina GA is becoming more popular in the research community and the data used in this proposal are generated by the Illumina GA, here we will only discuss the general workflow for the Illumina GA (Figure 1-8). The other two platforms follow similar procedures.

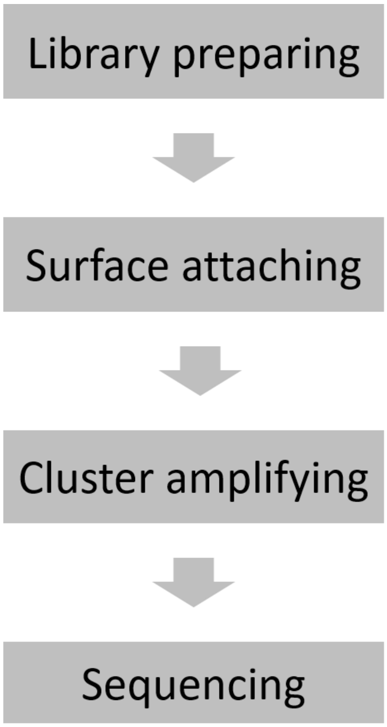


Figure 1-8: NGS workflow. First is to prepare for the sequencing samples. Then the cDNA fragments are attached on the surface of the flow cells. Next is to amplify the cDNA fragment into clusters. Finally, we sequence these clusters.

In general, DNAs are first fragmented through enzymes or sonication, and oligo adapters are then attached to these DNA fragments. Through microfluidic cluster station, these fragments are attached to the surface of a glass flow cell, which is coated with complementary sequences of the adaptors. On the flow cell, DNA fragments are amplified into a cluster through the so-called bridge PCR amplification and the inverse strands are washed away. Finally, four different fluorescence-labeled nucleotides are added to the flow cell, which have their 3'-OH chemically inactivated to ensure that only one base is incorporated per cycle. Images are taken to identify the incorporated nucleotide for each cluster and then the fluorescent groups are removed to deblock the 3' end for the next base incorporation cycle.

At each sequencing step, four color images are obtained corresponding to the four labeled nucleotides. An additional step called base calling is needed to identify the bases at each position based on the colors (Figure 1-9).

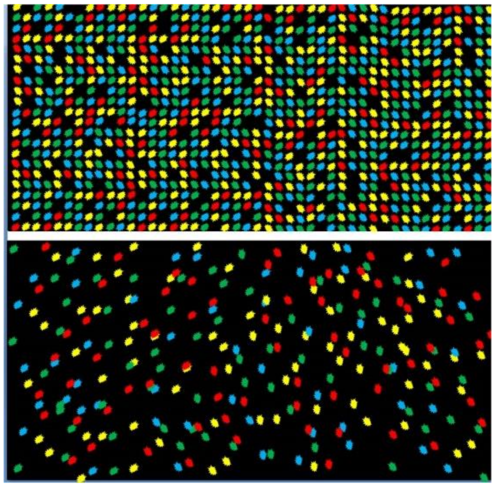


Figure 1-9: Color based sequencing. On the graph, each color denotes a unique type of nucleotides. (https://en.wikipedia.org/wiki/DNA_nanoball_sequencing).

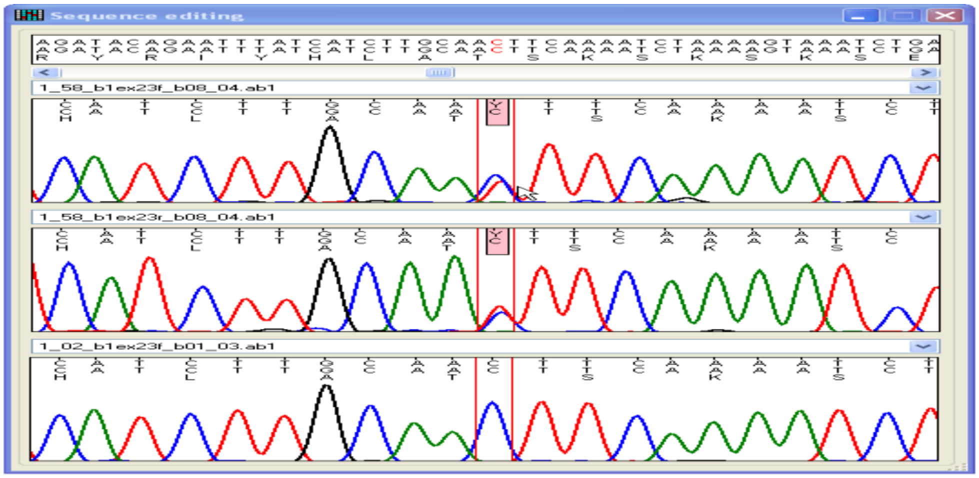

Sequence editing
AGATACAGAATTTATCATCTTGCAACTTCAAAATCTAAAAGTAAATCTGA
AGATACAGAATTTATCATCTTGCAACTTCAAAATCTAAAAGTAAATCTGA
R Y R I Y H L A T S K S K S K S E
1_58_b1ex23f_b08_04.ab1
1_58_b1ex23r_b08_04.ab1
1_02_b1ex23f_b01_03.ab1

Figure 1-10: Base calling. This graph shows the base calling procedure. The above two small red windows show a situation that two colors with similar signal strength level occur simultaneously at the same position, which makes base calling ambiguous and may lead to an error. (http://www.insilicase.co.uk/guide/GeneScreen.aspx).

However, sometimes it may be hard to get a sharp image and sequencing errors may occur (Figure 1-10). Although the nucleotide called at each position represented the best guess, sometimes the ‘best’ calling cannot be easily determined. For example, the strengths of several colors might be in a similar level. To take this information into account, a score at each position is calculated to estimate the probability of getting an error for each base-call. Commonly, a Phred score is used, which reflects the error probabilities by considering four important parameters at base calling step (Ewing & Green, 1998). Peak spacing, the ratio of the largest peak-to-peak spacing to the smallest peak-to-peak spacing in a window of seven peaks centered on the current one; uncalled/called ratio, the ratio of the amplitude of the largest uncalled peak to the smallest called peak in a window of seven peaks around the current one; type 2 uncalled/called ratio, the same as uncalled/called ratio, but using a window of three peaks; peak resolution, the number of bases between the current base and the nearest unresolved base times -1.

The formula to calculate the quality value for each base-call is:

$$q = -10 \times log_{10}(p),$$

where p is the estimated error probability for the base-call. If a base is assigned a quality value of 30, it means that the base is called incorrectly with the probability 0.001. The high quality score values correspond to low error probabilities.

From the NGS technology, millions of short reads (35—250bp) can be easily generated simultaneously. With these short sequences, the next step is to find their positions on the reference genome (details in section 3.3). There have been several efficient algorithms for mapping high-throughput short reads, such as Bowtie (Langmead, Trapnell, Pop, & Salzberg, 2009) and MAQ (Li, Ruan, & Durbin, 2008).

1.4 Applications of Next Generation Sequencing

Due to this new sequencing technology, we can improve traditional genetic analyses and also create many new ways for genome-wide researches. Some typical applications include RNA sequencing (RNA-seq), chromatin immunoprecipitation sequencing (ChIP-seq), Methylation-Seq and copy number variation sequencing (CNV-seq). Since RNA-seq is the primary focus of this thesis, we will give more detailed description of this process.

RNA-seq refers to use the high-throughput sequencing technologies to sequence the complementary DNA (cDNA) reversely transcribed from an RNA. It allows people to perform the whole transcriptome sequencing for the purpose of analyzing the transcriptome, and provides an efficient way to obtain the information, through millions of short reads, from the mRNA. The major interest of this kind of analysis is to evaluate the expression level of genes, to detect differentially expressed genes, to discover the alternative splicing events and to find single-nucleotide polymorphism.

Chapter 2 RNA-seq Data Analyses

In the previous chapter, we briefly discussed RNA-seq, one of the major applications of NGS technology. In this chapter, we will introduce more details about the goals and essential steps of analyzing the RNA-seq data.

2.1 Background and Differential Expression Analysis

Data generated from RNA-seq contains not only the information of expression levels of genes, but also the information of their structures. In this sense, RNA-seq provides more information about RNA than the traditional array-based methods.

For large-scale transcriptome analyses of thousands of genes, microarray-based method was first developed to measure their expression levels (Schena, Shalon, Davis, & Brown, 1995). Although it was a great success, this type of method has some major limitations, such as hybridization and cross-hybridization artifacts, upper bound of the signal strength, and limit of the pre-specified genes. These limitations can be easily overcome by RNA-seq. For example, by directly mapping millions of short reads to reference genome, RNA-seq provides digital (counts) expression information rather than analog signals, so the upper bound of the expression does not exist. RNA-seq also provides the chance to discover new genes or novel splicing isoforms of transcripts. In addition, because of the dramatic decrease of the sequencing cost, the cost of RNA-seq has become comparable to that of microarray-based method. Very importantly, many studies have shown that RNA-seq demonstrates the ability to efficiently create high accurate and replicable genetic data compared with traditional microarrays (Marioni, Mason, Mane, Stephens, & Gilad, 2008; Wang, Gerstein, & Snyder, 2009). Therefore, RNA-seq is gradually replacing microarrays and becoming popular among researchers.

Differential expression (DE) analysis is one of the most important applications in RNA-seq data analyses. Suppose we have a set of genes $Gene = \{g|g = 1, \ldots, G\}$. Given a gene $g \in Gene$, let k_g denote the copy number of gene g. Generally, DE study has two goals: One is to estimate the copy number k_g, which is the expression level of a specific gene; another is to find the genes with a significant changing in the copy numbers between two different conditions, like normal and disease.

2.2 Read Mapping by TopHat

As shown in section 1.3, results from RNA-seq platform are the four types of colors which represent the four types of nucleotides. After the base-calling procedure, these colors are converted to the corresponding nucleotides, which are saved in a fastq or fasta file format, containing read id, sequence letters and base-calling quality scores. In order to conduct further analyses, we need to map the short RNA-seq reads against a reference genome to find their locations.

There exist several fast and accurate mapping algorithms for locating the high-throughput short reads, such as MAQ (space seeds based) and Bowtie (Burrows-Wheeler transform based). These methods are not ideal for the RNA-seq data, because they ignore splicing junctions. For example, given a read across the junction of the exon1 and exon2 of a transcript (Figure 2-1), the Bowtie and MAQ would treat it as unmapped read, although it contains important information about how exons are joined. In order to map RNA-seq data accurately, typically a software tool called TopHat (Trapnell, Pachter, & Salzberg, 2009) is used. It can map reads from splicing junctions to the reference genome by local de novo construction of a transcript.

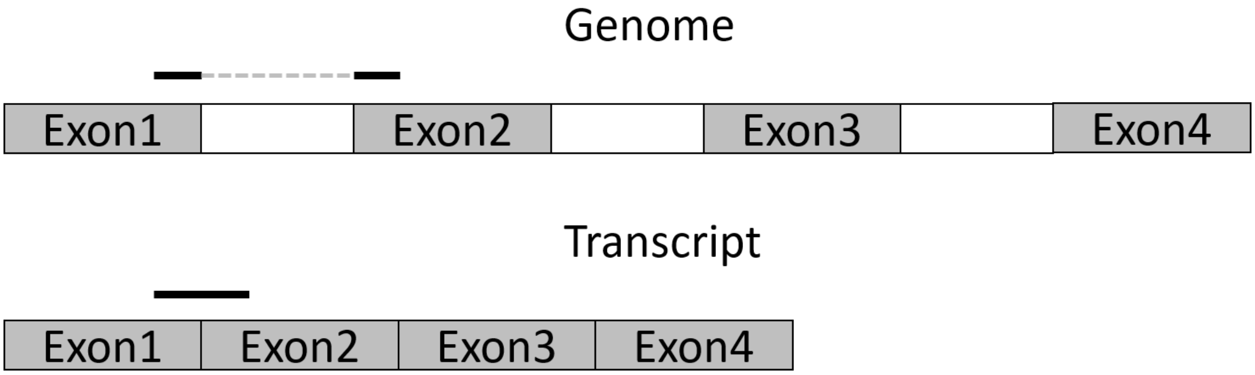
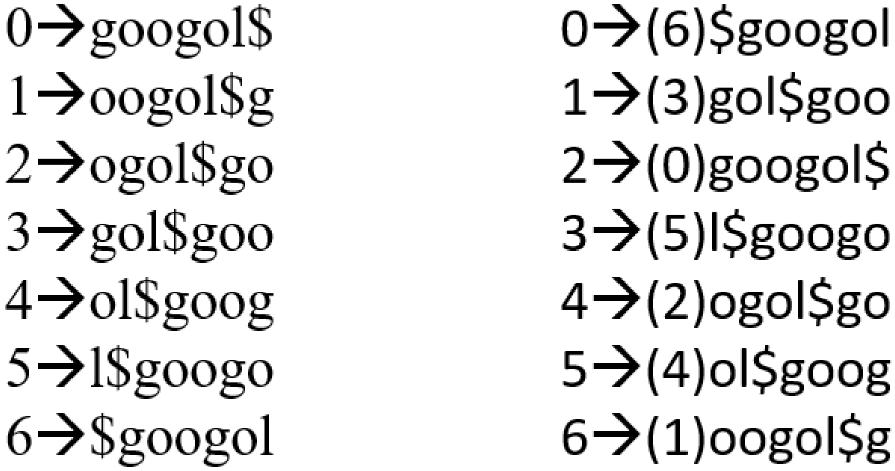


Figure 2-1: Splicing junction read. The read is not within an exon, but span two exons.

Since TopHat uses Bowtie, the Burrows-Wheeler transform based mapping algorithm, in its pipeline, we will first introduce Bowtie.

Let's consider the Burrows-Wheeler transform (BWT) for a string (Li & Durbin, 2009). Let Σ be an alphabet and suppose symbol \$ is not present in Σ and is lexicographically smaller than all the symbols in Σ. A string $X = a_0 a_1 \dots a_{n-1}$ is always ended with symbol \$ (i.e. $a_{n-1} = \$$) and this symbol only appears at the end. Let $X[i] = a_i$ be the i^{th} symbol of X, $X[i,j] = a_i \dots a_j$ be a substring and $X_i = X[i, n-1]$ be a suffix of X. Let S be the suffix array (SA) of X, which is a permutation of the integers $0 \dots n-1$. $S(i)$ then denotes the start position of the i^{th} smallest suffix. The BWT of X is a string defined as $B[i] = \$$ when $S(i) = 0$ and $B[i] = X[S(i) - 1]$ otherwise. The figure below shows an example (Figure 2-2).



0→googol$	0→(6)$googol
1→oogol$g	1→(3)gol$goo
2→ogol$go	2→(0)googol$
3→gol$goo	3→(5)l$googo
4→ol$goog	4→(2)ogol$go
5→l$googo	5→(4)ol$goog
6→$googol	6→(1)oogol$g

Figure 2-2: Burrows-Wheeler transform. The original string is ‘googol$’. The suffix array is $S = (6,3,0,5,2,4,1)$, and the BWT is ‘lo$oogg’ *(Li & Durbin, 2009).*

We can easily find that, given a string W, if it is a substring of X, the position of each occurrence of W in X will be in an interval in the suffix array. For example, if $W = 'go'$, all the positions of this substring in X is in interval [1,2] in suffix array. In general, we can define:

$$\underline{R}(W) = min\{k: W\ is\ the\ prefix\ of\ X_{S(k)}\}$$

$$\overline{R}(W) = max\{k: W\ is\ the\ prefix\ of\ X_{S(k)}\}$$

In particular, if W is an empty string, $\underline{R}(W) = 1$ and $\overline{R}(W) = n - 1$. The interval $[\underline{R}(W), \overline{R}(W)]$ then is called suffix array interval of W and the set of positions of all occurrences of W in X is $\{\mathrm{S(k)}: \underline{R}(W) \leq k \leq \overline{R}(W)\}$ (Li & Durbin, 2009). Ferragina and Manzini (Ferragina & Manzini, 2000) have proved that if W is a substring of X then we can calculate the SA interval for a new string aW based on the SA interval of W. That is:

$$\underline{R}(aW) = C(a) + O\big(a, \underline{R}(W) - 1\big) + 1$$

$$\overline{R}(aW) = C(a) + O(a, \overline{R}(W))$$

$C(a)$ is the number of symbols in $X[0, n-2]$ that are lexicographically smaller than alphabet $'a'$ and $O(a, i)$ is the number of occurrences of $'a'$ in $B[0, i]$. We will have $\underline{R}(aW) \leq \overline{R}(aW)$ if aW is a substring of X. With this result, we can find all the positions of occurrence of a short string W in a long string X by backward searching.

Bowtie is an algorithm using this transform and searching backward to map short reads against the reference genome. Let G denotes the reference genome, which can be treated as a long string. And let r be a short read. Firstly, Bowtie will calculate the suffix array for reference genome G. Secondly, the BWT string B for G and array $C(*)$ and $O(*,*)$ are calculated. Lastly, Bowtie will do the backward searching (Figure 2-3).

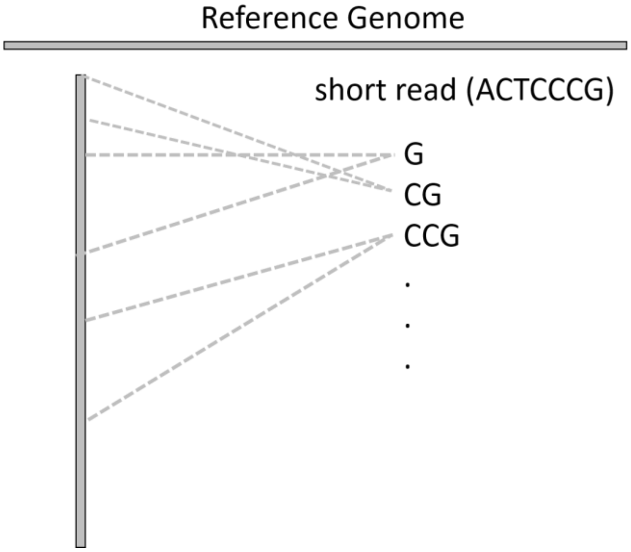


Figure 2-3: Backward searching by Bowtie.

TopHat finds junctions by mapping reads in three steps. In the first step, it maps all reads to a reference genome using Bowtie. All reads that are not mapped to the genome are set as ‘initially unmapped reads’ or IUM reads. In the second step, TopHat uses the mapped reads to generate islands, which can be treated as putative exons. At last, TopHat maps the IUM reads to the possible junctions that are built from islands to find splicing junctions (Figure 2-4).

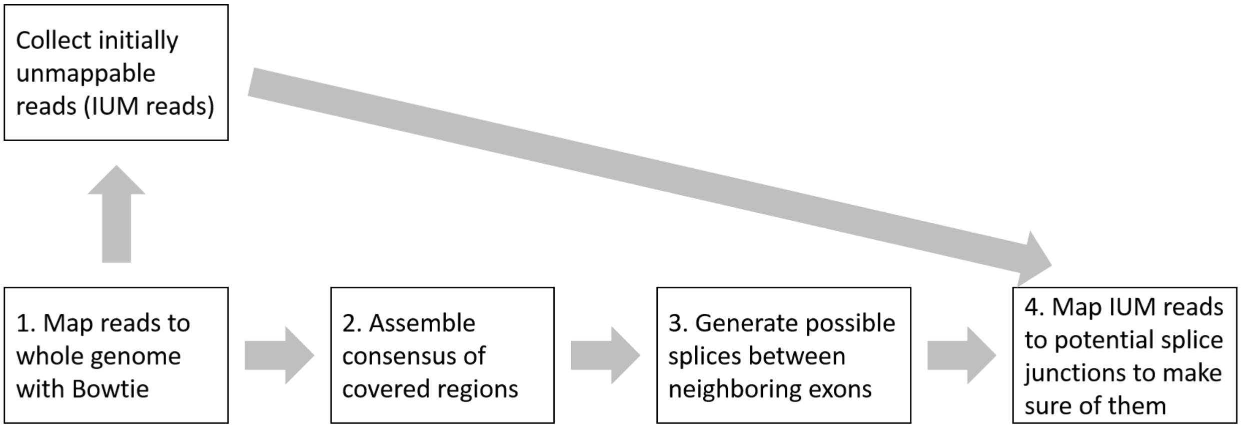
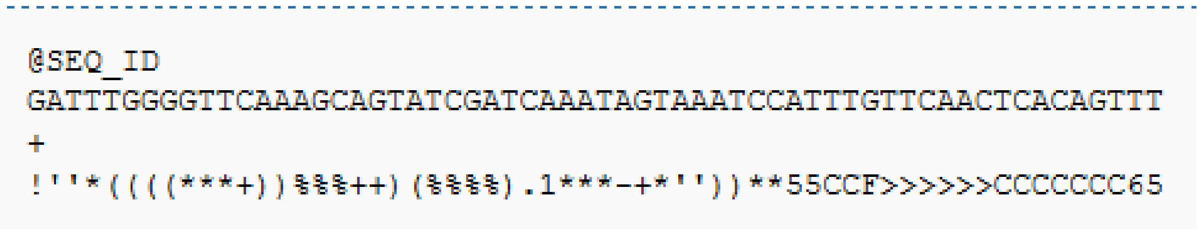

Collect initially unmappable reads (IUM reads)
1. Map reads to whole genome with Bowtie
2. Assemble consensus of covered regions
3. Generate possible splices between neighboring exons
4. Map IUM reads to potential splice junctions to make sure of them

Figure 2-4: TopHat workflow. Firstly, TopHat maps the reads by Bowtie and collects unmapped reads. Then it built potential splicing junctions based on mapped reads. Finally, it re-maps the unmapped reads to these splicing junctions.

In step 2, since the consensus may include incorrect base-call due to sequencing errors, when TopHat uses mapped reads to generate islands, it will modify them by using bases in the reference genome. Because most reads cover the end of an exon will also span splicing junctions, few reads will be mapped to the two ends of an exon in step 1. Therefore, for every generated island (potential exons), it will lose a small amount of bases on both ends. In order to capture this sequence, TopHat adds a small amount of sequence from the reference genome to both ends of each island. In step 3, TopHat uses islands to build possible splicing junction patterns. However, for the genes transcribed at low levels, there may be gaps for these genes due to low sequencing coverage. So TopHat needs to decide whether two islands need to be merged into a single exon. Because introns shorter than 70bp are rare in mammalian genomes, TopHat combines two exons into a single one if the distance between them is less than 70bp (actually use 6bp in TopHat).

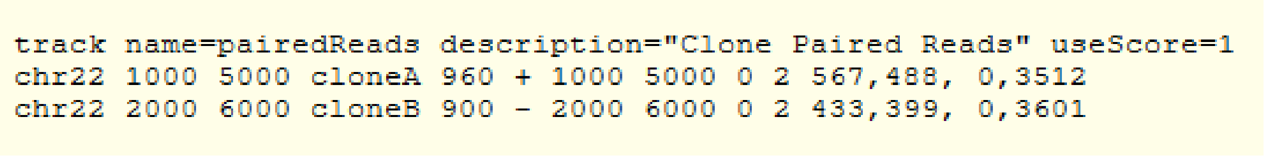
TopHat accepts files in Sanger FASTQ format. FASTQ format stores sequences and Phred quality scores in a single file (Figure 2-5).



```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

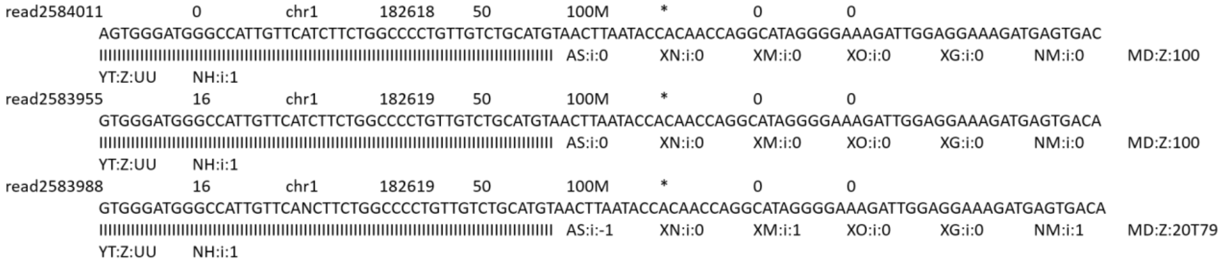
Figure 2-5: Fastq file format. This file saves the raw RNA-seq data which can be used by TopHat. For one read, there are four lines to denote its information. The first line is the read id, second is the raw sequence, third is a '+' character and the last is the sequencing quality score. (https://en.wikipedia.org/wiki/FASTQ_format).

There are two important output files form TopHat. One is for junctions in BED format (Figure 2-6), and another is all accepted hits (include junctions) in BAM format which is a compressed binary version of SAM format (Figure 2-7). The SAM stands for Sequencing Alignment/Map, which saves the mapping results.



```
track name=pairedReads description="Clone Paired Reads" useScore=1
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

Figure 2-6: BED format. The first column is the name of the chromosome and then are the start position of the feature in the chromosome, the ending position of the feature in the chromosome, the name of the BED line, the score between 0 and 1000, the strand either "-" or "+", the starting position at which the feature is drawn thickly (for example, the start codon in gene displays), the ending position at which the feature is drawn thickly (for example, the stop codon in gene displays), the RGB value of the form R,G,B (e.g. 255,0,0), the number of blocks (exons) in the BED line, a comma-separated list of the block sizes and a comma-separated list of block starts. (https://genome.ucsc.edu/FAQ/FAQformat.html#format1).



```
read2584011           0           chr1        182618      50          100M        *           0           0
            AGTGGGATGGGCCATTGTTCATCTTCTGGCCCCTGTTGTCTGCATGTAACTTAATACCACAACCAGGCATAGGGGAAAGATTGGAGGAAAGATGAGTGAC
            IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII  AS:i:0      XN:i:0      XM:i:0      XO:i:0      XG:i:0      NM:i:0      MD:Z:100
            YT:Z:UU     NH:i:1
read2583955           16          chr1        182619      50          100M        *           0           0
            GTGGGATGGGCCATTGTTCATCTTCTGGCCCCTGTTGTCTGCATGTAACTTAATACCACAACCAGGCATAGGGGAAAGATTGGAGGAAAGATGAGTGACA
            IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII  AS:i:0      XN:i:0      XM:i:0      XO:i:0      XG:i:0      NM:i:0      MD:Z:100
            YT:Z:UU     NH:i:1
read2583988           16          chr1        182619      50          100M        *           0           0
            GTGGGATGGGCCATTGTTCANCTTCTGGCCCCTGTTGTCTGCATGTAACTTAATACCACAACCAGGCATAGGGGAAAGATTGGAGGAAAGATGAGTGACA
            IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII  AS:i:-1     XN:i:0      XM:i:1      XO:i:0      XG:i:0      NM:i:1      MD:Z:20T79
            YT:Z:UU     NH:i:1
```

Figure 2-7: SAM format. The first column is the query template/pair name followed by bitwise flag, reference sequence name, one-based left most position, mapping quality, extended CIGAR string, mate reference sequence name, one-based mate position, inferred template length, query sequence, query quality and optional fields.

2.3 Summarizing the Data

For the purpose of evaluating gene expression, we need to summarize the output files of TopHat. Usually, we summarize the SAM files by counting the number of reads attributed to each gene regions based on the chromosome id, start positon and CIGAR string (indicating how the reads map to the genome) in the file. We can use HTseq (Anders, Pyl, & Huber, 2015) for doing this, which provides different ways for counting RNA-seq reads according to diverse gene structure models.

Suppose we have summarized data into a two-way table, in which rows are the gene names, and columns are sample individuals. The cell values of this table will be the number of reads mapped to the corresponding gene and individual. One way to make a simple evaluation of gene expression level is to calculate RPKM (Mortazavi, Williams, McCue, Schaeffer, & Wold, 2008) or FPKM. That is, for a gene g, its RPKM value is defined as below

$$RPKM_g = \frac{number\ of\ mapped\ reads\ to\ g}{length\ of\ \ g(kilobase) \times total\ mapped\ reads(million)}$$

RPKM normalizes the raw count number by the length of gene and the total mapped reads. In this way, it allows us to compare expression measurements across different genes and different experiments.

Similarly, the FPKM is defined as

$$FPKM_g = \frac{number\ of\ fragments\ to\ g}{length\ of\ g(kilobase) \times total\ mapped\ fragments(million)}$$

The reason of using fragment number rather than the read number is to modify the RPKM for the paired-end read. In paired-end data, we will have two mate pairs sequenced from one cDNA fragment. FPKM count the two reads that comprise a paired-end read as one fragment, which is more accurate than doubling the read number. And also, the normalizing factor ensure that the comparisons through genes and experiments are fair.

Chapter 3 Overview of Existing Statistical Methods for DE analysis

In chapter 2, we introduced the RPKM and FPKM, the two popular measurements of gene expression level. Although they provide a concise way to evaluate the expression level, they still belong to the group of preliminary methods to represent gene expression. In order to get more accurate results from DE analysis, we need to employ more complicated statistical models to handle the RNA-seq data. In this chapter, we will review several widely-used statistical methods for modeling the RNA-seq data.

3.1 Gene Level Analysis

The first group of methods is for the DE analysis at gene level, which means that only the gene expression level needs to be estimated but the isoform information is ignored. Although these methods do not consider the detailed information about the isoform, it actually has pretty good performance in practice, so it is still widely used for DE problems nowadays.

3.1.1 Negative Binomial Model

For the gene level DE analysis, the most popular distribution used is the negative binomial model. Several state-of-arts software packages are based on this distribution, such as edgeR (Robinson, McCarthy, & Smyth, 2010), DESeq (Anders & Huber, 2010) and baySeq (Hardcastle & Kelly, 2010). Since these packages share the similar probabilistic model and only differ in some trivial aspects such as algorithms for parameter estimation, in this section, we only give a detailed review on the application of negative binomial model in DE analysis for one of them, which is the edgeR package.

Let $GS = \{g|g = 1,2, \dots, G\}$ be a set of genes, where G is the total number of genes. Let k_g and l_g be the copy number and the length of gene g respectively. Let j be the sample id and X_{gj} be the number of reads mapped to gene g in sample j.

Since the RNA sequencing can be treated as a random sampling procedure that the reads are independently and uniformly sequenced from every possible nucleotide (Jiang & Wong, 2009), it can be shown that X_{gj} follows a Poisson distribution,

$$X_{gj} \sim Poisson\left(N_j \theta_g\right)$$

where N_j is the total mapped reads or the library size of sample j and $\theta_g = \frac{k_g l_g}{\sum_{g=1}^{G} k_g l_g}$ is the interested expression level parameter before adjusting for the gene length. If we denote $S = \sum_{g=1}^{G} k_g l_g$, which is the transcriptome size, then we have $\theta_g = \frac{k_g l_g}{S}$. We know that the point estimation of θ_g is $\frac{X_{gj}}{N_j}$, which means $\frac{\widehat{k_g l_g}}{S} = \frac{X_{gj}}{N_j}$. If we divide both side of $\frac{\widehat{k_g l_g}}{S} = \frac{X_{gj}}{N_j}$ by the length of the gene, we will get $\frac{\widehat{k_g}}{S} = \frac{X_{gj}}{N_j l_g}$. It's clear that the right side, $\frac{X_{gj}}{N_j \times l_g}$, is just the RPKM and the left side, $\frac{\widehat{k_g}}{S}$, is the relative expression level of gene g. So according to the Poisson model, the RPKM is an estimation of the relative expression level.

However, in reality, the Poisson distribution is not enough to account for the variation among the samples. Since usually the sequencing procedure is done on several different individuals, the sample variance of X_{gj} actually is larger than its sample mean due to the biological replication. In order to overcome the issue of over-dispersion, the copy number k_g can be assumed to be a random variable K_{gj} with mean k_g. Then we will have a hierarchical model:

$$X_{gj}|K_{gj} \sim Poisson\left(N_j \frac{K_{gj} l_g}{S_j}\right)$$

$$K_{gj} \sim Gamma\left(\mu = k_g, \phi_g\right)$$

where

$$S_j = \sum_{g=1}^{G} K_{gj} l_g$$

Here S_j is treated as a fixed number. Then we can get a Gamma-Poisson mixture model. By integrating out the K_{gj}, the marginal distribution of X_{gj} is a negative binomial distribution.

$$X_{gj} \sim NB\left(N_j \frac{k_g l_g}{S_j}, \varphi_g\right)$$

The observable variables of this model are X_{gj} and N_j. The known parameter is l_g and the parameters that need to be estimated are k_g, S_j and φ_g. It is easy to show that k_g and S_j are actually not estimable, however, their ratio, that is, $\frac{k_g}{S_j}$, is estimable.

Another important step is the transcriptome size normalization. In the model above, although we can compare $\frac{k_{g0}}{S_{j0}}$ with $\frac{k_{g1}}{S_{j'1}}$ to detect differential expression, what we really want to compare is k_{g0} and k_{g1}. If the denominators S_{j0} and $S_{j'1}$ are not the same across the different samples j and j', the comparison would not be fair. So we need to normalize the transcriptome size.

The TMM normalization is a popular normalization procedure (Robinson & Oshlack, 2010). The idea is that, among all the genes that need to be tested, most of them are not differentially expressed. If two individuals share the same copy number of a gene, then $\frac{X_{gj}/N_j}{X_{gj'}/N_{j'}}$ would be a good estimation of $\frac{S_{j'}}{S_j}$. So we can set a particular individual to be a baseline sample, and calculate the normalization factor $f_j = \frac{S_j}{S_{ref}}$ for all other samples. Then this factor can be applied into the model, which has the following form:

$$X_{gj} \sim NB\left(N_j \frac{S_j}{S_{ref}} \frac{S_{ref}}{S_j} \frac{k_g l_g}{S_j}, \varphi_g\right) = NB\left(N_j \frac{S_{ref}}{S_j} \frac{k_g l_g}{S_{ref}}, \varphi_g\right) = NB\left(N_j^* \theta_g^*, \varphi_g\right)$$

where $N_j^* = \frac{N_j}{f_j}$ and $\theta_g^* = \frac{k_g l_g}{S_{ref}}$. If we consider about samples under different conditions, then we will have the model below

$$X_{gj0} \sim NB\left(N_j^* \theta_{g0}^*, \varphi_g\right)$$

$$X_{gj'1} \sim NB\left(N_{j'}^* \theta_{g1}^*, \varphi_g\right)$$

The parameter estimation of edgeR is based on the quantile-adjusted conditional maximum likelihood (Robinson & Smyth, 2008) and weighted conditional likelihood (Robinson & Smyth, 2007).

The quantile-adjusted conditional maximum likelihood (quantile-adjusted CML) is used to overcome the underestimation problem for estimating parameter φ_g. Although the dispersion φ_g is a nuisance parameter, its estimation still affects the accuracy of estimating θ_g. It has been shown that by traditional likelihood estimation, especially for a small sample size, φ_g is underestimated (Robinson & Smyth, 2008). In this case, the estimation of θ_g is usually inaccurate. The quantile-adjusted CML method is first to generate a pseudo data, which have the same library size N across all samples, from the real data. In order to make them equivalent to the real data, the pseudo data points are set to share the similar quantiles with the corresponding real data points. Then the summation of the pseudo data turns out to be a sufficient statistics of θ_g. By conditioning on this sum, we can eliminate θ_g and only keep the φ_g in the likelihood function. After estimating φ_g from this conditional likelihood function, we can insert the estimated value into the original likelihood function and estimate the parameter θ_g. In this way, the accuracy of the estimation of φ_g and θ_g is dramatically increased.

The weighted conditional likelihood method is used to modify the dispersion φ_g for each gene. It tries to adjust each φ_g to a common φ shared by all genes. The idea is to add a weighted conditional likelihood through the genes to the likelihood of individual φ_g.

$$l_w(\varphi_g) = l_g(\varphi_g) + \alpha l_{all}(\varphi_g)$$

$$l_{all}(\varphi_g) = \sum_{g=1}^{G} l_g(\varphi_g)$$

where $l_g(\varphi_g)$ comes from the quantile-adjusted CML and α is the weight. This method is extremely useful when there are not enough samples, since it is more reliable to estimate a common φ for all genes than to estimate φ_g separately for each gene.

By employing the techniques shown above, the edgeR provides a stable and reasonable performance in DE analysis and is probably the most widely used package.

3.2 Transcript Level Analysis

The next group of methods for DE analyses is to take the isoform information into consideration. That is, instead of estimating the general gene level expression, we try to estimate expressions at the transcript level and find the significantly differentially expressed genes according to these isoform expressions. These methods can be further categorized into two subgroups according to their assumptions: one is to assume all transcript structures are known, and another assumes them unknown. Both are illustrated.

3.2.1 Known Transcript Structure

With well-defined isoforms, the model aims to obtain the estimators of expression level for each isoform. The main challenge is that isoforms of a gene usually have common regions. If a read is mapped to these regions, it is hard to tell which isoform the read originally came from. In this section, we discuss three types of models that have been proposed to solve this problem, which all achieve reasonable estimations of isoform expression levels, but based on different statistical models.

3.2.1.1 Poisson Model

The first method is the Poisson model (Salzman, Jiang, & Wong, 2011; Jiang & Wong, 2009). This model can be applied on both single and paired-end RNA-seq data. Since the sequencing procedure is independent across genes, we can focus on one gene each time.

Let $GS = \{g|g = 1 \ldots G\}$ be the set of all genes, where G is the total number of genes. Let $T = \{t_i|i = 1, \ldots, I\}$ be the set of all transcripts, where I is the total number of transcripts. Let $T_g = \left\{t_{gi}|i = 1, \ldots, I_g\right\}$ be the set of all isoforms of gene g, where I_g is the total number of isoforms of gene g. Denote k_i and l_i as the copies and the length of isoform i respectively. Let

$\{h|h = 1, \dots, H_g\}$ be the set of unique read types that can be sequenced from isoforms of gene g. A read type is identified by the $5'$ end for a single end read or both $5'$ and $3'$ end for a paired end read. Given a sample j, let X_{ih} be the number of type h read generated from isoform i and $X_h = \sum_{i=1}^{I} X_{ih}$ be the number of type h read generated from all isoforms. Finally, let N denote the number of total mapped reads.

For single-end reads, the uniform sampling model is used, which assumes that each read is sequenced independently and uniformly from all possible nucleotides and the read is small enough compared with the whole transcriptome. Under this assumption, it can be shown that

$$X_{ih} \sim Bin(N, \theta_{ih})$$

where $\theta_{ih} = \frac{k_i}{\sum_{i=1}^{I} k_i l_i}$ is the probability of getting a read h after sequencing the sample once. And since $N \to \infty$ and $\theta_{ih} \to 0$, the binomial distribution can be approximated by a Poisson distribution

$$X_{ih} \sim Poisson(N\theta_{ih})$$

The distribution of X_h can be derived as a sum of independent Poisson random variable.

$$X_h \sim Poisson\left(N \sum_{i=1}^{I_g} \theta_{ih}\right)$$

The likelihood function is

$$f(x_1, \dots, x_H) = \prod_{h=1}^{H} \frac{\left(N \sum_{i=1}^{I_g} \theta_{ih}\right)^{x_h} e^{-n \sum_{i=1}^{I_g} \theta_{ih}}}{x_h!}$$

The parameter θ_{ih} represents the expression level of each isoform.

For paired-end reads, the insertion length model is used, which assumes that given the length of a paired-end read on isoform i, the read is sequenced uniformly on this isoform. Actually, the information about the insertion length is important. In the sample preparation step of the sequencing, there is a step to filter out cDNA fragments with extreme length (too long or too short). So only fragments with proper length are left for sequencing. Since a paired-end read comes from both side of a fragment, its insertion length should be in a suitable scope. Although most of the reads are not uniquely mapped to the isoforms, they are very likely to have different insertion lengths on different isoforms. So by considering about the insertion length, it is much easier to make sure of the destination of the reads. The model is built by the following steps.

Let's denote A as getting a read with type h from isoform i by sequencing the sample once, B as getting a read from isoform i by sequencing the sample once, and C as the read being type h given that the read is sequenced from isoform i. It is obvious that

$$P\{A\} = P\{B\ \}P\{C\} = \frac{k_i l_i}{\sum_{i=1}^{I} k_i l_i} q(l_{ih}) \frac{1}{l_i - l_{ih}}$$

where l_{ih} is the length of type h read on isoform i and $q(\cdot)$ is the distribution of the read length. In reality, we can use an empirical pdf of $q(\cdot)$ in the model. So for paired-end read data, the model becomes

$$X_{ih} \sim Poisson\left(N, \theta_{ih} q(l_{ih}) \frac{l_i}{l_i - l_{ih}}\right)$$

where $\theta_{ih} = \frac{k_i}{\sum_{i=1}^{I} k_i l_i}$ is the same as in uniform sampling model.

Generally, if we set $C_{ih} = q(l_{ih})\frac{l_i}{l_i - l_{ih}}$ for paired end read and $C_{ih} = 1$ for single end read, then the model can be written as

$$X_h \sim Poisson\left(N\sum_{i=1}^{I_g} C_{ih}\theta_{ih}\right)$$

And the likelihood function for one sample is

$$f(x_1, \dots, x_H) = \prod_{h=1}^{H} \frac{\left(N\sum_{i=1}^{I_g} C_{ih}\theta_{ih}\right)^{x_h} e^{-n\sum_{i=1}^{I_g} C_{ih}\theta_{ih}}}{x_h!}$$

The log-likelihood function is

$$l(\theta) = \sum_{h=1}^{H}\{x_h log(NC_h^T\theta_h) - NC_h^T\theta_h - log(x_h!)\}$$

where $C_h = \left[C_{1h}, \dots, C_{I_g h}\right]^T$ and $\theta_h = \left[\theta_{1h}, \dots, \theta_{I_g h}\right]^T$.

It can be shown that the function $l(\cdot)$ is a concave function. Let's consider about the function

$$l_h(\theta) = x_h log(nC_h^T\theta_h) - nC_h^T\theta_h - log(x_h!)$$

In fact, the Hessian matrix of $l_h(\cdot)$ has the form

$$H_{ij} = \frac{\partial^2 l_h(\theta)}{\partial\theta_{ih}\partial\theta_{jh}} = -\frac{x_h C_{ih} C_{jh}}{\left(\sum_{i=1}^{I_g} C_{ih}\theta_{ih}\right)^2}$$

which can be written as

$$H = -daa^T$$

where $a = \left[C_{1h}, \dots, C_{I_g h}\right]^T$ and $d = -\frac{x_h}{\left(\sum_{i=1}^{I_g} c_{ih}\theta_{ih}\right)^2}$. For any vector y, we have

$$y^T H y = -y^T d a a^T y = -d(a^T y)^T a^T y = -d(a^T y)^2 \leq 0$$

So the hessian matrix H is negative semi-definite, which means the function $l_h(\theta)$ is concave. Since we know that

$$l(\theta) = \sum_{h=1}^{H} l_h(\theta)$$

So function $l(\cdot)$ is also concave, by the theorem that the summation of a group of concave functions is still concave.

Based on this model, statistical inferences can be conducted by solving the maximum likelihood estimation (MLE), which is an asymptotically unbiased and consistent if the true parameter is in the interior of the parameter space (Lehmann, 1998). However, if the true parameter is close to the boundary of the parameter space, the MLE will lose its good properties. The author of this model proposed another way to estimate the parameter. They set a prior density to θ and generate random samples from posterior distribution based on importance sampling (Liu, 2002). By doing so, the estimation is more robust if the true parameter is indeed around the boundary.

3.2.1.2 Generative Model

Compared to the Poisson model that summarizes the count number of each type of read, the generative model tries to model each individual read directly without summarizing them together. In this method (Li, Ruotti, Stewart, Thomson, & Dewey, 2010), a model is designed to solve the mapping uncertainty problem under both gene and isoform levels.

The generative model is best illustrated with a Bayesian network (Figure 3-1), which models the sequencing procedure.

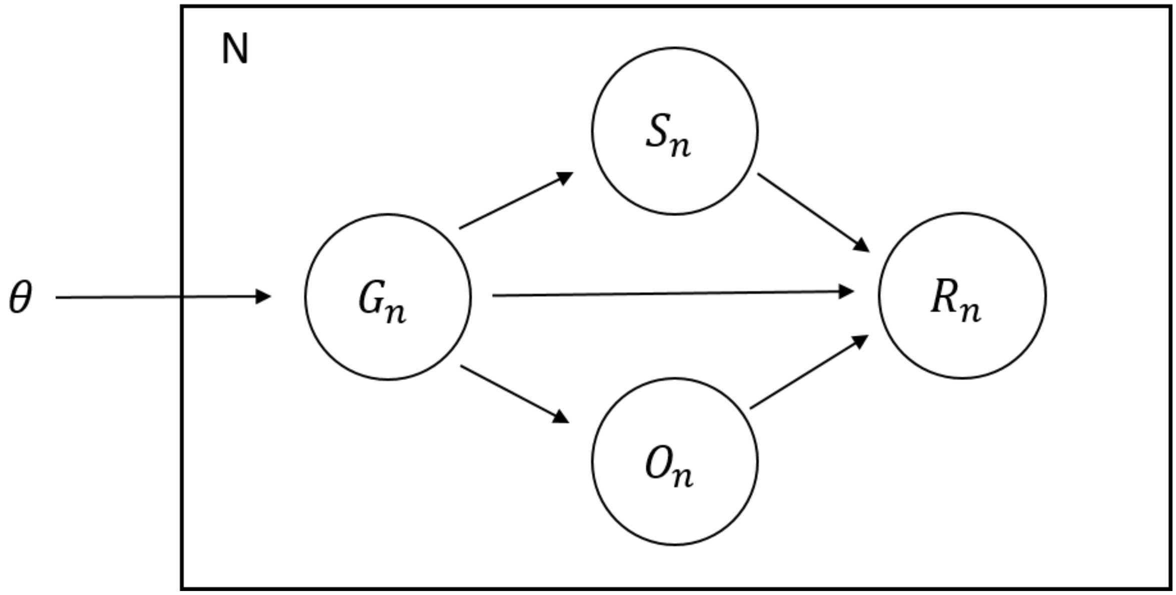


Figure 3-1: Bayesian network of sequencing procedure. R_n represents the n^{th} sequenced read, and all the reads are treated as the i.i.d random variables generated from this graph model. G_n, S_n and O_n denote the isoform, start position and orientation of the n^{th} read respectively *(Li, Ruotti, Stewart, Thomson, & Dewey, 2010).*

The complete likelihood function of the random vector (R_n, G_n, S_n, O_n) is then:

$$p(\rho, i, j, k; \theta) = \prod_{n=1}^{N} P(G_n = i; \theta) P(S_n = j | G_n = i) P(O_n = k | G_n = i) P(R_n = \rho | G_n = i, S_n = j, O_n = k)$$

The value of G_n is taken from a set of isoforms $[0, M]$. The value of S_n is taken from a set of positions $[1, l_i - L + 1]$, where l_i is the length of the isoform i and L is the read length. The value of O_n is 0 if the sequence of the n^{th} read is the same orientation with its parent isoform and is 1 otherwise. R_n takes a value from the set of sequenced reads $\{\rho\}$. They can be formulated as follows:

$$p_G(i;\theta) = P(G_n = i;\theta) = \theta_i$$

$$p_{S|G}(j|i) = P(S_n = j|G_n = i) = \frac{1}{l_i - L + 1}$$

$$p_{O|G}(0|i) = p_{O|G}(1|i) = P(O_n = 0|G_n = i) = \frac{1}{2}$$

$$p_{R|G,S,O}(\rho|i,j,k) = P(R_n = \rho|G_n = i, S_n = j, O_n = k) = f(x) = \begin{cases} \prod_{t=1}^{L} \omega_t\left(\rho_t, \gamma_{j+t-1}^i\right), k = 0 \\ \prod_{t=1}^{L} \omega_t\left(\rho_t, \bar{\gamma}_{j+t-1}^i\right), k = 1 \end{cases}$$

The $\theta_i = \frac{k_i l_i}{\sum_i k_i l_i}$ is the probability of getting a read from isoform i after one sequencing. The function $\omega_t(a, b)$ is the probability that the character of sequenced read at position t is a given that the corresponding character of the reference isoform is b. ρ_t is the character of the read ρ at position t and γ^i is the sequence of isoform i, while $\bar{\gamma}^i$ is the sequence of its reverse complement. The function $\omega_t(a, b)$ allows the incorporation of sequencing error by setting different values at each position along the read. The larger the value of $\omega_t(a, b)$ is, the more accurate the sequencing procedure is. At the same time, by changing the way of formulating $p_{S|G}(j|i)$, the model can deal with the sequencing bias issue. For the illustration purpose, we will reduce the complexity of the model and only show how to estimate the parameters in a simple version of this model. As mentioned previously, if assuming the sequencing to be a strand-specific protocol, then we have $p_{O|G}(0|i) = 1$, which means we can ignore the orientation problem. Under this situation, R_n is the observable variable and G_n and S_n are the latent variables. The marginal distribution of R_n is given by

$$p(\rho;\theta) = \prod_{n=1}^{N} p(\rho_n;\theta) = \prod_{n=1}^{N} \sum_{i,j} p(i,j;\theta) p(\rho_n|i,j) = \prod_{n=1}^{N} \sum_{i,j} \frac{\theta_i}{l_i - L + 1} p(\rho_n|i,j)$$

where only the θ_i is the unknown parameter and all others are known. It can be shown that it is convenient to use EM algorithm to get the MLE of θ. At E step, we compute

$$Q\left(\theta|\theta^{(t)}\right) = E\left[\sum_{n=1}^{N} log\left(p(\rho_n, i, j;\theta)\right)|\theta^{(t)}, \rho\right] = \sum_{n=1}^{N} E\left[log\left(p(\rho_n, i, j;\theta)\right)|\theta^{(t)}, \rho_n\right]$$

$$= \sum_{n=1}^{N} p\left(i,j|\theta^{(t)}, \rho_n\right) log\left(\frac{\theta_i}{l_i - L + 1} p(\rho_n|i,j)\right)$$

where

$$p\left(i,j|\theta^{(t)}, \rho_n\right) = \frac{p(i,j;\theta^{(t)})p(\rho_n|i,j)}{\sum_{i,j} p(i,j;\theta^{(t)})p(\rho_n|i,j)} = \frac{\left(\theta_i^{(t)}/l_i - L + 1\right)p(\rho_n|i,j)}{\sum_{i,j}\left(\theta_i^{(t)}/l_i - L + 1\right)p(\rho_n|i,j)}$$

At M step, we compute the MLE of θ by setting the partial derivatives to zero. It can be shown that the likelihood function of R_n is a concave function with respect to the parameter θ by the similar argument given in section 3.2.1.1.

In general, the generative model can not only solve the problem of evaluating isoform expression level, but also can take the reads mapped to multiple genomic loci into consideration. Moreover, it can also adjust for the sequencing bias and sequencing error, which is typically a concern about during the DE analysis.

3.2.2 Unknown Transcript Structure

In this section, we will give a quick summary about DE analysis without pre-defined isoform. This problem is conceptually harder, since we need to reconstruct the isoform structures first before evaluating their expression levels. The procedure of reconstructing the isoform based on the RNA-seq reads is called De Novo assembling. We will illustrate this technique according to Cufflinks (Trapnell, et al., 2010), which is a very popular software tool for DE analysis, and also show how they estimate the expression level for the assembled isoforms.

3.2.2.1 De Novo Assembling

Usually, De Novo assembling is an inevitable step for DE analysis if isoform structures are unknown. The basic idea is to construct a graph model by connecting overlapped reads, and then seek for trustable routes according to some specific principles. These routes will be the assembled isoforms. In details, there are four basic rules followed by Cufflinks assembler. First, every valid read must be mapped to at least one assembled isoforms. Second, every assembled isoform must be represented by a chain of the valid reads. Third, the number of assembled isoforms should be as small as possible while satisfying the first requirement. Last, given the assembled isoforms, the statistical model used for estimating expression level must be identifiable. Based on these rules, the problem of finding the assemblies is transformed to a problem of finding a minimum partition of a partial order, which is the set of reads with a binary relation, into chains. This problem in turn can be converted to an even simpler problem of finding a maximum matching in a weighted bipartite graph, which shows how the reads are connected, based on the Dilworth's theorem (Dilworth, 1950).

3.2.2.2 Transcript Abundance Estimation

The statistical model used by Cufflinks is also the generative model, but a little bit different from the model discussed in section 3.2.1.2. The Cufflinks introduces the distribution of the read length into the model, which makes it more powerful to model the paired-end RNA-seq data; however, the model doesn't make the use of the sequencing accuracy and read orientation information as shown in the previous generative model.

Cufflinks models each read independently and mimic the sequencing procedure step-by-step (Trapnell, et al., 2010). Let's denote $p_r(\cdot)$ be the read length distribution, $I_t(r)$ be the length of read r on transcript t, l_t be the length of the transcript t and $\tilde{l}_t = \sum_{i=1}^{l_t} p_r(i)(l_t - i + 1)$ be the adjusted length of transcript t. The likelihood function is then:

$$L(\theta|r) = \prod_{r=1}^{n} \sum_{t=1}^{T} \theta_t \left(\frac{p_r\big(I_t(r)\big)}{l_t - I_t(r) + 1} \right)$$

The $\theta_t = \frac{k_t \, \tilde{l}_t}{\sum_{t=1}^{T} k_t \, \tilde{l}_t}$ is the probability of getting a read from transcript t after one sequencing. The $\frac{p_r\left(I_t(r)\right)}{l_t - I_t(r) + 1}$ is the probability that the read sequence is r given that the read is sequenced from transcript t. For the single end read, the read length is fixed so that $p_r(\cdot) \equiv 1$; for the paired end read, the read length distribution $p_r(\cdot)$ can be approximated by the empirical distribution similar to section 3.2.1.1.

Chapter 4 Splicing Graph-based Negative Binomial Model

In the previous chapter, we discussed several statistical methods for the DE analysis. Some of them ignore the isoform structure, some of them rely on the well-defined isoforms, and some of them ask for the De Novo assembling technique. Our research tries to overcome these problems by building a powerful, robust and efficient method. In this chapter, we will introduce a new isoform-free model for DE analysis and we will show the motivations, the structure and the advantages of our model.

4.1 Motivation

Our purpose is to develop an efficient and accurate method to detect differentially expressed genes under transcript level, but without relying on the pre-defined isoform structure and the De Novo assembling procedure.

For the transcript level analysis, we change the null hypothesis from the gene level

$$H_{01}: \sum_{i=1}^{I_g} k_{0i} = \sum_{i=1}^{I_g} k_{1i}$$

to the transcript level

$$H_{02}: k_{0i} = k_{1i}, i = 1, \dots, I_g$$

The first hypothesis is tested by the gene level analysis tools, such as edgeR and DESeq. However, it is not exactly equivalent to 'the gene being not differentially expressed'. For instance, suppose a gene has two isoforms a and b. Under one condition we assume that the copy numbers of the isoforms are $k_a = n$ and $k_b = 5n$, so that $k_a + k_b = 6n$. Under another condition, we set $k_a = 5n$ and $k_b = n$, and so $k_a + k_b = 6n$. This means that based on the first null hypothesis, the gene is not differentially expressed, however, the copies of isoforms, k_a and k_b, do change a lot across different conditions. Therefore, we need to redefine the hypothesis to make the test more powerful. That is, we call a gene is not differentially expressed if and only if $k_{0i} = k_{1i}$ for all i of the gene.

Moreover, although we want to perform a global test on all isoforms of one gene, we do not want to make any assumption on the isoform structure or use the De Novo assembling technique. The reason is that if the model depends on the assumed or assembled isoforms, the performance will be related to the accuracy of the isoform structures, which should be less robust than an isoform-free model. We will show how to achieve this goal based on the linear algebra theory in next section.

4.2 Statistical Modeling

In this section, we will illustrate our modeling procedure and the numerical algorithms for parameter estimation.

4.2.1 Notation

Let G be a set of genes to be tested. Given a gene $g \in G$, let $T_g = \{t_{gi} | i = 1, \dots, I_g\}$ be a set of transcripts that can be transcribed from gene g. In another word, T_g denotes a set of all isoforms related to gene g. Let $R_g = \{r_{gh} | h = 1, \dots, H_g\}$ denote the set of all unique read types that can be sequenced from the isoforms of gene g and $R_{gi} = \{r_{gih} | h = 1, \dots, H_{gi}\}$ be the read types sequenced from isoform i (‘read type’ will be defined later). Denote k_{gi} to be the copy number of isoform i of gene g, and l_{gi} be its length. Let j be the sample id, l_f be the average length of cDNA fragment in our experiment.

4.2.2 Annotation File Modification and Read Summary

In order to detect differentially expressed genes at transcript level, we need to summarize RNA-seq reads in a more detailed way. For this purpose, we need to use Tophat for read mapping, which can identify exon-exon junctions from RNA-seq reads.

Due to the alternative splicing events (Wang, et al., 2008; Matlin, Clark, & Smith, 2005), exons of different isoforms may have overlaps in the annotation file. For example, if an exon has an alternative $5'$ donor site, there would be two transcripts in the annotation file. One contains this exon with its first $5'$ donor site and another contains the exon with its second $5'$ donor site (Bernard, Jacob, Mairal, & Vert, 2014). This will lead to some difficulties when we summarize reads into the read types. To avoid this issue, we first construct blocks, which are non-overlapping DNA segments. This modification needs to be applied gene by gene. Given a gene g, let $POS = \{pos_i | i = 1, \dots, I_g^{pos}\}$ be the set of all the nucleotide positons within the exon regions in the annotation file. And let $E = \{e_i | i = 1, \dots, I_g^{e}\}$ be the set of all exons. Our goal is to group all nucleotide positions in POS into non-overlapping regions. Let's denote these regions as blocks with notation b. Let A_{pos} denote the set of all exons that contain position pos. We group two positions $pos_i < pos_j$ to the same block if they satisfy:

$$a).\, pos_j - pos_i = 1. \quad b).\, A_{pos_i} = A_{pos_j}.$$

Finally, we index these blocks based on their start and end positions. An example is shown below (Figure 4-1).

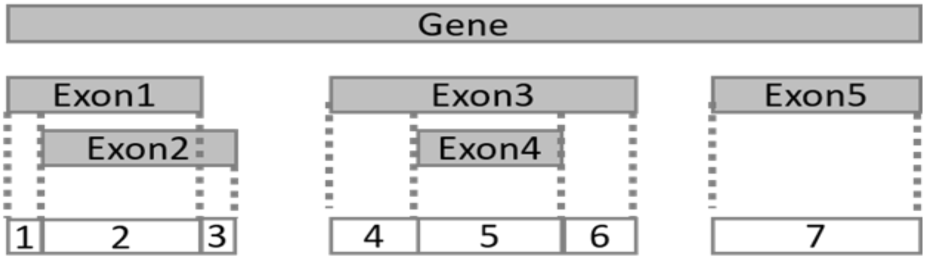
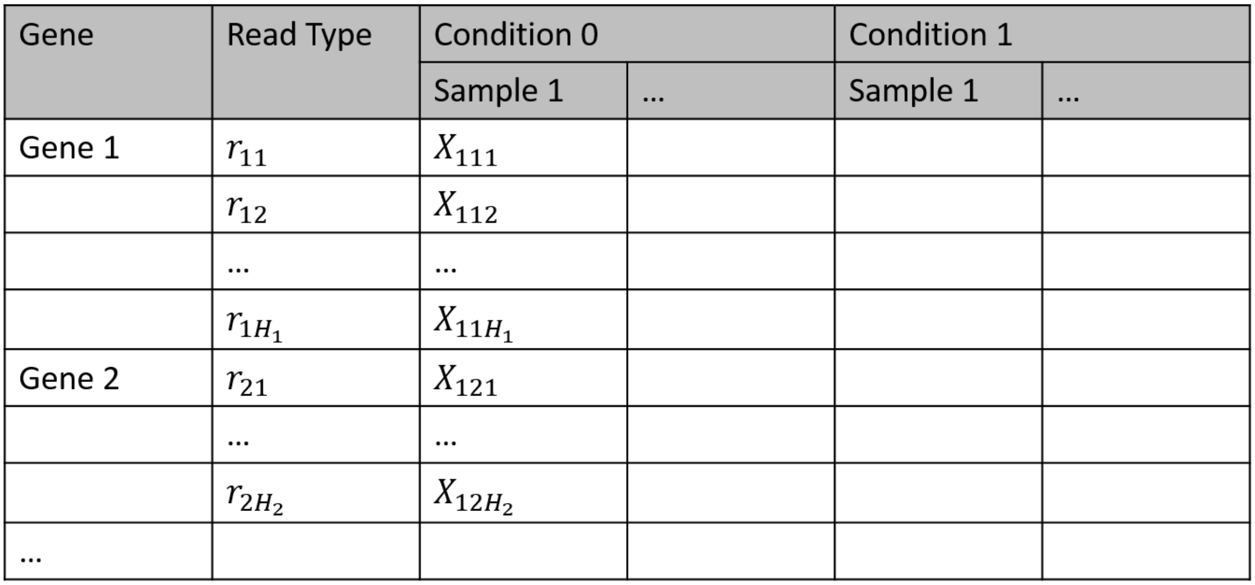


Figure 4-1: Gene annotation modification. Given a gene, its exon regions in the annotation file are shown in the figure. We re-group the nucleotide positions based on the rules described above. The boxes with white background are the created blocks.

Once the new annotation file is constructed, we can summarize mapped reads to read types based on it.

Given a single end read, we define a read type to be a chain of ordered index of blocks where the read is mapped to, i.e. 1, 1-2, 1-3-5 (Bernard, Jacob, Mairal, & Vert, 2014). The summarized data is a two-way table. Each row denotes a read type and each column denotes a sample. The cell value is the count number of a read type sequenced from a sample (Figure 4-2).



Gene	Read Type	Condition 0		Condition 1	
		Sample 1	…	Sample 1	…
Gene 1	r_{11}	X_{111}			
	r_{12}	X_{112}			
	…	…			
	r_{1H_1}	X_{11H_1}			
Gene 2	r_{21}	X_{121}			
	…	…			
	r_{2H_2}	X_{12H_2}			
…					

Figure 4-2: Summarized data table. The cell value is the count number of a specific read type in the corresponding sample.

For paired-end RNA-seq data, we simply treat it as two single end reads and summarize them in the same way as the single-end read.

4.2.3 Statistical Model for One Sample

First, let's consider modeling the count data of read types for one sample. Since the genes are independently sequenced, we can perform the analysis gene by gene. For simplicity, we only keep the uniquely mapped reads and eliminate reads that are mapped to multiple gene regions. Studies have shown that the Poisson distribution can fit the read count well when there exist only technical replications (Jiang & Wong, 2009; Salzman, Jiang, & Wong, 2011; Mortazavi, Williams, McCue, Schaeffer, & Wold, 2008). According to this, we assume

$$X_{gi} \propto \frac{N_c k_{gi} l_{gi}}{l_f}$$

where X_{gi} is the number of reads sequenced from isoform i of gene g, and N_c is the number of cells being sequenced. We assume that X_{gi} follows a Poisson distribution

$$X_{gi} \sim Poisson(\frac{aN_c k_{gi} l_{gi}}{l_f}) \tag{1}$$

They are also independent because each transcript is independently sequenced during the experiment. We also assume

$$\left(X_{gi1}, \dots, X_{giH_{gi}} \middle| X_{gi}\right) \sim Multinomial\left(X_{gi}, p_{gi1}, \dots, p_{giH_{gi}}\right) \tag{2}$$

where X_{gih} is the number of read type h sequenced from the isoform i of gene g, and p_{gih} is the probability of getting a read type h after sequencing the isoform i once. Usually, p_{gih} depends on l_{gih} and the sequencing bias (Hansen, Brenner, & Dudoit, 2010; Roberts, Trapnell, Donaghey, Rinn, & Pachter, 2011; Li, Jiang, & Wong, 2010). l_{gih} is the number of all possible start positions of read type h on isoform i of gene g. For example, given an isoform, if we let the start position of a single-end read go from the left to the right, we will obtain a unique read type at each position. So given an isoform, the number of read types is finite and every read type has its own set of possible start positions (Figure 4-3).

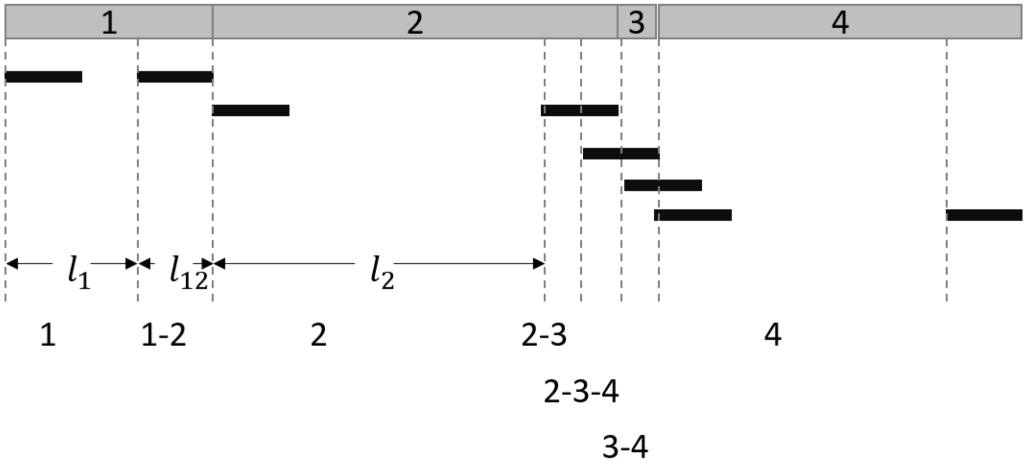


Figure 4-3: Read types generated from one isoform. When we move a read from left to the right on the isoform, we will discover different read types. l_h is the number of the start positions that can generate read type h.

We can model p_{gih} as $p_{gih} = \frac{w_{gih} l_{gih}}{\sum_{h=1}^{H_{gi}} w_{gih} l_{gih}}$, where w_{gih} accounts for the sequencing bias and $\sum_{h=1}^{H_{gi}} w_{gih} = 1$. When $w_{gih} = w_{gi}$, it represents a special case that the sequencing procedure is unbiased, that is, every nucleotide on a transcript is equally likely to be sequenced as a starting position of a read (Jiang & Wong, 2009). In order to get (2), we assume

$$X_{gih} \sim Poisson\left(\frac{aN_c p_{gih} k_{gi} l_{gi}}{l_f}\right), independently \qquad (3)$$

It is clear that $X_{gi} = \sum_{h=1}^{H_{gi}} X_{gih} \sim Poisson\left(\frac{aN_c k_{gi} l_{gi}}{l_f}\right)$, so that (2) will be held. The observable variable X_{gh} then follows

$$X_{gh} = \sum_{i \in \{i | i\ has\ h\}} X_{gih} \sim Poisson\left(\frac{aN_C \sum_{i \in \{i|i\ has\ h\}} p_{gih} k_{gi} l_{gi}}{l_f}\right) \tag{4}$$

The total number of reads mapped to gene g follows:

$$X_g = \sum_{i=1}^{I_g} \sum_{h=1}^{H_{gi}} X_{gih} \sim Poisson\left(\frac{aN_c \sum_{i=1}^{I_g} k_{gi} l_{gi}}{l_f}\right) \tag{5}$$

The total number of mapped reads of the sample follows

$$N = \sum_{g=1}^{G} X_g \sim Poisson\left(\frac{aN_c \sum_{g=1}^{G} \sum_{i=1}^{I_g} k_{gi} l_{gi}}{l_f}\right) \tag{6}$$

Let $S = \sum_{g=1}^{G} \sum_{i=1}^{I_g} k_{gi} l_{gi}$ and $\theta_{gh} = \frac{\sum_{i \in \{i|i\ has\ h\}} p_{gih} k_{gi} l_{gi}}{S}$, we will have

$$\begin{aligned} &\left(X_{11}, \dots, X_{1H_1}, \dots, X_{G1}, \dots, X_{GH_G} | N\right) \\ &\sim Multinomial(N, \theta_{11}, \dots \theta_{1H_1}, \dots, \theta_{G1}, \dots, \theta_{GH_G}) \end{aligned} \tag{7}$$

By the central limit theory we have

$$\begin{bmatrix} X_{11} \\ \vdots \\ X_{1H_1} \\ \vdots \\ X_{G1} \\ \vdots \\ X_{GH_G} \end{bmatrix} \overset{D}{\rightarrow} N\left(\begin{bmatrix} N\theta_{11} \\ \vdots \\ N\theta_{1H_1} \\ \vdots \\ N\theta_{G1} \\ \vdots \\ N\theta_{GH_G} \end{bmatrix}, \begin{bmatrix} N\theta_{11}(1-\theta_{11}) & \dots & -N\theta_{11}\theta_{GH_G} \\ & \ddots & \\ -N\theta_{11}\theta_{GH_G} & \dots & N\theta_{GH_G}(1-\theta_{GH_G}) \end{bmatrix}\right)$$

Since $\theta_{gh} \to 0$, $N \to \infty$ and $N\theta_{gh} = \lambda_{gh}$, we have

$$\begin{bmatrix} X_{11} \\ \vdots \\ X_{1H_1} \\ \vdots \\ X_{G1} \\ \vdots \\ X_{GH_G} \end{bmatrix} \overset{D}{\to} N\left(\begin{bmatrix} \lambda_{11} \\ \vdots \\ \lambda_{1H_1} \\ \vdots \\ \lambda_{G1} \\ \vdots \\ \lambda_{GH_G} \end{bmatrix}, \begin{bmatrix} \lambda_{11} & \dots & 0 \\ & \ddots & \\ 0 & \dots & \lambda_{GH_G} \end{bmatrix}\right)$$

Asymptotically, X_{gh} can be treated as independent Poisson random variable given the total number of mapped reads.

$$X_{gh}|N \sim Poisson(N\theta_{gh}) \tag{8}$$

We will use this conditional distribution as our model, and simply treat N, the total mapped reads, as a constant.

Based on this model, we can test $H_0\colon \theta_{0gh} = \theta_{1gh}, h = 1, \dots, H_g$. Testing $H_0\colon \theta_{0gh} = \theta_{1gh}, h = 1, \dots, H_g$ is equivalent to test $H_0\colon k_{0gi} = k_{1gi}, i = 1, \dots, I_g$, with one extra assumption. In fact, we can write the relationship between θ_{gh} and k_{gi} in a matrix form according to (8).

$$\overrightarrow{\theta_g} = \begin{bmatrix} \theta_{g1} \\ \vdots \\ \theta_{gH_g} \end{bmatrix} = \begin{bmatrix} p_{g11} & \cdots & p_{gI_g1} \\ \vdots & \ddots & \vdots \\ p_{g1H_g} & \cdots & p_{gI_gH_g} \end{bmatrix} \begin{bmatrix} \frac{k_{g1}l_{g1}}{S} \\ \vdots \\ \frac{k_{gI_g}l_{gI_g}}{S} \end{bmatrix} = P_g\overrightarrow{k_g} \tag{9}$$

Some of p_{gih} will be zero in matrix P_g, since it is possible that some isoforms cannot generate some of read types. We know that, if the rank of P_g equals its column number, then $H_0\colon \theta_{0gh} = \theta_{1gh}, h = 1, \dots, H_g \Leftrightarrow H_0\colon k_{0gi} = k_{1gi}, i = 1, \dots, I_g$. So we need to make the assumption that the column rank of P_g is full, which is reasonable in most cases.

4.2.4 Statistical Model for Multiple Samples

For multiple samples, we need to consider issues related to biological replications and transcriptome size normalization.

When samples are collected from different individuals, the Poisson model fails to account for the large variation among these biological replications (Hansen, Wu, Irizarry, & Leek, 2011; Glaus, Honkela, & Rattray, 2012). In this case, the negative binomial model is a more flexible for fitting the data and usually performs better than the Poisson model.

We extend our Poisson model to the negative binomial model and do transcriptome size adjustment by using TMM normalization.

We assume k_{gi} in (8) is a random variable rather than a constant. So we have

$$X_{jgh}|\mathrm{K}_{jgi} \sim Poisson\left(N_j \frac{\sum_{i\in\{i|i\ has\ h\}} p_{gih}K_{jgi}l_{gi}}{S_j}\right) \tag{10}$$

where X_{jgh} is the number of read type h sequenced from gene g in sample j. By using TMM normalization, we can estimate $\frac{S_j}{S_{j'}}$ for any two samples j and j'. So we select a sample as a reference sample r, and calculate $\frac{S_r}{S_j}$ for all j. Then (10) can be re-formulated as

$$X_{jgh}|\mathrm{K}_{jgi} \sim Poisson\left(N_j^* \frac{\sum_{i\in\{i|i\ has\ h\}} p_{gih}K_{jgi}l_{gi}}{S_r}\right) \tag{11}$$

where $N_j^* = N_j\frac{S_r}{S_j}$ is the normalized library size. If we denote $\Theta_{jgh} = \frac{\sum_{i\in\{i|i\ has\ h\}} p_{gih}K_{jgi}l_{gi}}{S_r}$, then we have

$$X_{jgh}|\Theta_{jgh} \sim Poisson(N_j^* \Theta_{jgh}) \tag{12}$$

We assume that the prior density function of Θ_{jgh} follows a gamma distribution

$$\Theta_{jgh} \sim Gamma(\alpha_{gh}, \beta_{gh}) \tag{13}$$

with the mean $E[\Theta_{jgh}] = \alpha_{gh}\beta_{gh} = \theta_{gh} = \frac{\sum_{i\in\{i|i\ has\ h\}} p_{gih} k_{gi} l_{gi}}{S_r}$. Then we can get the marginal distribution of our data X_{jgh} by integrating out Θ_{jgh}, which is a negative binomial distribution.

$$X_{jgh} \sim NB(N_j^* \theta_{gh}, \varphi_{gh}) \tag{14}$$

where $\varphi_{gh} = \frac{1}{\alpha_{gh}}$.

4.2.5 Splicing Graph based Parameter Reduction

Usually, we will have a large number of parameters, which can lead to a high variation in the model. In order to make the model more stable and powerful, we need to reduce the number of parameters as much as possible. We try to do this according to a splicing graph model.

Recall the relation between the read type expression and the isoform expression.

$$\vec{\theta}_g = \begin{bmatrix} \theta_{g1} \\ \vdots \\ \theta_{gH_g} \end{bmatrix} = \begin{bmatrix} p_{g11} & \cdots & p_{gI_g1} \\ \vdots & \ddots & \vdots \\ p_{g1H_g} & \cdots & p_{gI_gH_g} \end{bmatrix} \begin{bmatrix} \frac{k_{g1} l_{g1}}{S} \\ \vdots \\ \frac{k_{gI_g} l_{gI_g}}{S} \end{bmatrix} = P_g \vec{k}_g$$

We have shown that if $rank(P_g)$ equals the number of its columns, we can convert the test of k_{gi} to the test of θ_{gh}. We know that for a matrix, its column rank equals to its row rank. If we sum two rows that are proportional to each other, the row rank will not change, and so is the column rank. So our basic strategy is to aggregate rows in matrix P_g as many as possible, while maintaining the same rank. In this way, we can group several read types to a new read type, which leads to a reduction in the number of parameters.

We have already assumed that

$$p_{gih} = \begin{cases} \dfrac{w_{gih} l_{gih}}{\sum_{h=1}^{H_{gi}} w_{gih} l_{gih}}, & if\ i\ can\ generate\ h \\ 0, & if\ i\ can\ not\ generate\ h \end{cases} \tag{15}$$

Notice that l_{gih}, the number of possible start positions of read type h, only depends on the index h. So we have $l_{gih} = l_{gh}$ for all i. Now let's figure out that the conditions under which two rows in P_g may be proportional to each other.

First, consider the unbiased sequencing model, that is $p_{gih} = \frac{l_{gh}}{\sum_{h=1}^{H_{gi}} l_{gh}} = \frac{l_{gh}}{L_{gi}}$, where $L_{gi} = \sum_{h=1}^{H_{gi}} l_{gh}$. For any two read types h and h', if the set of isoforms that can generate h is the same as the set of isoforms that can generate h', we call these two read types 'always show together'. In other words, if an isoform can provide read type h, it must also be able to provide read type h'; if it cannot provide read type h, it also shall not provide read type h', and vice versa. Under this situation, the rows in P_g for these two read types look like the following

$$\begin{bmatrix} \dfrac{l_{gh}}{L_{g1}} & \dfrac{l_{gh}}{L_{g2}} & \cdots & \dfrac{l_{gh}}{L_{gI_g}} \\ \dfrac{l_{gh'}}{L_{g1}} & \dfrac{l_{gh'}}{L_{g2}} & \cdots & \dfrac{l_{gh'}}{L_{gI_g}} \end{bmatrix} \tag{16}$$

which are proportional to each other.

For a general case, which $p_{gih} = \frac{w_{gih} l_{gih}}{\sum_{h=1}^{H_{gi}} w_{gih} l_{gih}}$ and $L_{gi} = \sum_{h=1}^{H_{gi}} w_{gih} l_{gih}$. If we still select two read types meet the situation described above, then the rows will be

$$\begin{bmatrix} \frac{w_{g1h} l_{gh}}{L_{g1}} & \frac{w_{g2h} l_{gh}}{L_{g2}} & \dots & \frac{w_{gI_gh} l_{gh}}{L_{gI_g}} \\ \frac{w_{g1h'} l_{gh'}}{L_{g1}} & \frac{w_{g2h'} l_{gh'}}{L_{g2}} & \dots & \frac{w_{gI_gh'} l_{gh'}}{L_{gI_g}} \end{bmatrix} \tag{17}$$

If we assume $\frac{w_{g1h}}{w_{g1h'}} = \frac{w_{g2h}}{w_{g2h'}} = \cdots = \frac{w_{gI_gh}}{w_{gI_gh'}}$, then these rows are proportional to each other.

In conclusion, for the unbiased sequencing model, any two read types that are ‘always show together’ represent two linearly dependent rows in P_g; for the general sequencing model, if the assumption of the weights is true, this statement is still valid. Therefore, in order to find read types whose rows are linearly dependent in P_g, we only need to find read types that always show together.

For this purpose, we can create a splicing graph to help us search for these read types. Let $B = \{b_i | i = 1, \dots, B_g\}$ be the set of blocks belonging to gene g, which comes from the modified annotation file. Then a read type is denoted as $b_{i_1} b_{i_2} \dots b_{i_N}$, where $b_{i_k} \in B$ and $b_{i_j} < b_{i_{j'}}$ if $j < j'$. We define a prefix-string of a read type to be any substring starting from the left most block, that is $b_{i_1} b_{i_2} \dots b_{i_n}$, where $n = 1,2, \dots, N$. We also define a suffix-string to be any substring ending at the right most block, that is $b_{i_n}, b_{i_{n+1}}, \dots, b_{i_N}$, where $n = 1,2, \dots, N$. We compare read types based on lexicographic order. Then two read types can connect to each other if and only if there exist a suffix-string in the smaller read type equaling to a prefix-string in the bigger read type. Let $PS_r = \{all\, prefix - string\ of\ read\ r\}$ and $SS_r = \{all\, suffix - string\ of\ read\ r\}$.

So any $r < r'$ can connect together if and only if $SS_r \cap PS_{r'} \neq \emptyset$. If $r < r'$ and they can connect to each other, we denote the pair as $r \rightarrow r'$. Moreover, given $r \rightarrow r'$ and $r \rightarrow r''$, if $r' < r''$ and $r' \rightarrow r''$, we will delete the connection between r and r''. The splicing graph is a directed graph whose path go from the smaller read types to the larger read types. A simple example is shown below (Figure 4-4).

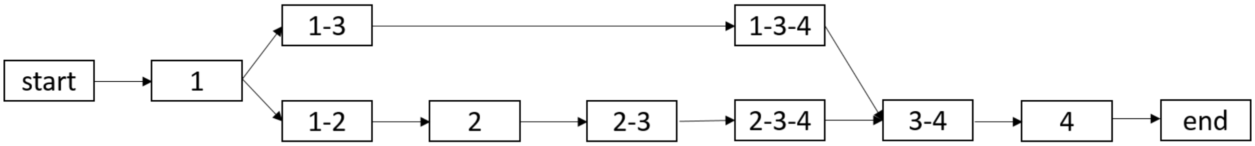


Figure 4-4: Splicing graph for parameter reduction. The ‘start’ and ‘end’ nodes are the pseudo read types to imply the possible starting and ending read types. There will be a path between two read types if they satisfy the connection conditions.

Notice that, we have two pseudo nodes on the graph, i.e. the ‘start’ and ‘end’. Their function is to imply the starting and ending read types (LeGault & Dewey, 2013). Since it is impossible to know the real starting and ending read types from RNA-seq data, the path between any read type and the ‘start’ or ‘end’ nodes will be a potential path. We can regularize the graph complexity by modifying the number of the potential paths, which will result in a different level of parameter reduction.

If two read types have a path between them and the out degree of the smaller read type and the in degree of the larger read type both equal to one, then these two read types will always show together, i.e. 1-2 and 2 in the graph above. In other word, if $h \rightarrow h'$ and the in degree of h and the out degree of h' are one, where $h = b_{i_1} \dots b_{i_j} \dots b_{i_k}$ and $h' = b_{i_j} \dots b_{i_k} \dots b_{i_N}$, it means that there must be some isoforms have the partial structure $b_{i_1} \dots b_{i_j} \dots b_{i_k} \dots b_{i_N}$ and this partial structure is the only one that involves h and h'. That is, h and h' always show together. So we combine read types satisfying such criterion to a new read type, which will significantly reduce the number of parameters in our model. Figure 4-5 shows the reduced version of Figure 4-4.

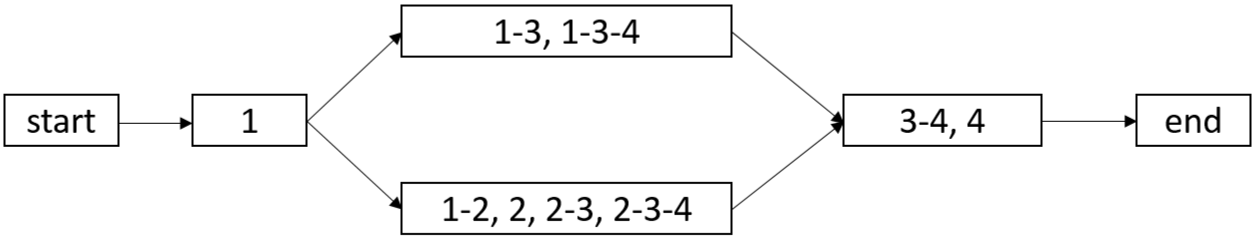


Figure 4-5: Reduced splicing graph. The number of parameters decrease from 9 to 4.

4.2.6 Parameter Estimation

We use a combination of the EM and Newton-Raphson algorithms for parameter estimation.

Given a gene g, let's denote $e = 0$ or 1 to be the condition id. We then have

$$X_{ejgh}|\Theta_{ejgh} \sim Poisson(N_j^* \Theta_{ejgh})$$

$$\Theta_{ejgh} \sim Gamma(\alpha_{gh}, \beta_{egh})$$

where $E[\Theta_{ejgh}] = \alpha_{gh}\beta_{egh} = \frac{\sum_{i=1}^{I_g} p_{gih} k_{egi} l_{gi}}{s_r} = \theta_{egh}$. The marginal distribution of the observable variable X_{ejgh} is

$$X_{ejgh} \sim NB(N_j^* \theta_{egh}, \varphi_{gh})$$

where $\varphi_{gh} = \frac{1}{\alpha_{gh}}$. The posterior distribution of Θ_{ejgh} is

$$\Theta_{ejgh}|X_{ejgh} \sim Gamma\left(X_{ejgh} + \varphi_{gh}^{-1}, \frac{\theta_{egh}}{N_j^*\theta_{egh} + \varphi_{gh}}\right)$$

Then we can have

$$E\left[\Theta_{ejgh}\middle|X_{ejgh} = x_{ejgh}\right] = \left(x_{ejgh} + \varphi_{gh}^{-1}\right)\left(\frac{\theta_{egh}}{N_j^*\theta_{egh} + \varphi_{gh}^{-1}}\right)$$

$$E\left[\ln\left(\Theta_{ejgh}\right)\middle|X_{ejgh} = x_{ejgh}\right] = \psi\left(x_{ejgh} + \varphi_{gh}^{-1}\right) + \ln\left(\frac{\theta_{egh}}{N_j^*\theta_{egh} + \varphi_{gh}^{-1}}\right)$$

$$= \frac{\Gamma'\left(x_{ejgh} + \varphi_{gh}^{-1}\right)}{\Gamma(x_{ejgh} + \varphi_{gh}^{-1})} + \ln\left(\frac{\theta_{egh}}{N_j^*\theta_{egh} + \varphi_{gh}^{-1}}\right)$$

The hypotheses are

$$H_0: \theta_{0gh} = \theta_{1gh}, h = 1, \dots, H_g, \quad H_1: H_0 \; is \; not \; true$$

Under $H_0 \cup H_1$, the marginal log-likelihood function is

$$l_X\left(\theta_{egh}, \varphi_{gh}\right) = \sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\Big\{\log\left(\Gamma\left(x_{ejgh} + \varphi_{gh}^{-1}\right)\right) - \log\left(\Gamma\left(x_{ejgh} + 1\right)\right) - \log\left(\Gamma\left(\varphi_{gh}^{-1}\right)\right)$$

$$+ x_{ejgh}\log\left(N_j^*\theta_{egh}\right) - x_{ejgh}\log\left(N_j^*\theta_{egh} + \varphi_{gh}^{-1}\right) - \varphi_{gh}^{-1}\log\left(\varphi_{gh}N_j^*\theta_{egh} + 1\right)\Big\}$$

The joint log-likelihood function is

$$l_{X,\Theta}(\theta_{egh},\varphi_{gh})$$

$$=\sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\Bigg\{x_{ejgh}\log(N_j^*)-\varphi_{gh}^{-1}\log(\varphi_{gh})-\log\Big(\mathit{\Gamma}(x_{ejgh}+1)\Big)$$

$$-\log\Big(\Gamma(\varphi_{gh}^{-1})\Big)-\varphi_{gh}^{-1}\log(\theta_{egh})+(x_{ejgh}+\varphi_{gh}^{-1}-1)\log(\mathit{\Theta}_{ejgh})$$

$$-\left(N_j^*+\frac{1}{\varphi_{gh}\theta_{egh}}\right)\Theta_{ejgh}\Bigg\}$$

According to the EM algorithm, we have the following procedures.

E step:

$$Q\Big(\theta_{egh},\varphi_{gh}\Big|\theta_{egh}^{(t)},\varphi_{gh}^{(t)}\Big)=E_{\Theta|X}\big[l_{X,\Theta}(\theta_{egh},\varphi_{gh})\big]$$

$$=\sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\Bigg\{x_{ejgh}\log(N_j^*)-\varphi_{gh}^{-1}\log(\varphi_{gh})-\log\Big(\mathit{\Gamma}(x_{ejgh}+1)\Big)$$

$$-\log\Big(\Gamma(\varphi_{gh}^{-1})\Big)-\varphi_{gh}^{-1}\log(\theta_{egh})$$

$$+(x_{ejgh}+\varphi_{gh}^{-1}-1)\left[\psi\left(x_{ejgh}+\left(\varphi_{gh}^{(t)}\right)^{-1}\right)+\ln\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)}+\left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]$$

$$-\left(N_j^*+\frac{1}{\varphi_{gh}\theta_{egh}}\right)\left[\left(x_{ejgh}+\left(\varphi_{gh}^{(t)}\right)^{-1}\right)\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)}+\left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\Big]\Big]\Bigg\}$$

M step:

$$\theta_{egh}^{(t+1)},\varphi_{gh}^{(t+1)}=argmax_{\theta_{egh},\varphi_{gh}}Q\Big(\theta_{egh},\varphi_{gh}\Big|\theta_{egh}^{(t)},\varphi_{gh}^{(t)}\Big)$$

By setting the partial derivatives to zero, we can update θ with the formula

$$\theta_{egh}^{(t+1)} = \frac{\sum_{j=1}^{J_e} \frac{\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\theta_{egh}^{(t)}}{N_j^* \theta_{egh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}}{J_e}$$

However, for φ, there is no closed form solution. We will use the Newton-Raphson algorithm to calculate the solution of the equation

$$\begin{aligned}(J_0 + J_1)\, log(\varphi_{gh}) - (J_0 + J_1) + (J_0 + J_1)\psi(\varphi_{gh}^{-1}) + J_0\, log\left(\theta_{0gh}^{(t+1)}\right) + J_1 log\left(\theta_{1gh}^{(t+1)}\right) \\ - \sum_{e=0}^{1}\sum_{j=1}^{J_e}\left[\psi\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right) + log\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right] \\ + \sum_{e=0}^{1}\left(\theta_{egh}^{(t+1)}\right)^{-1}\sum_{j=1}^{J_e}\left[\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right] = 0\end{aligned}$$

through an iterative formula

$$\varphi_{gh,t+1} = \varphi_{gh,t} - \frac{f(\varphi_{gh,t})}{f'(\varphi_{gh,t})}$$

where

$$\begin{aligned}f(\varphi_{gh}) = (J_0 + J_1)\, log(\varphi_{gh}) - (J_0 + J_1) + (J_0 + J_1)\psi(\varphi_{gh}^{-1}) + J_0\, log\left(\theta_{0gh}^{(t+1)}\right) + J_1 log\left(\theta_{1gh}^{(t+1)}\right) \\ - \sum_{e=0}^{1}\sum_{j=1}^{J_e}\left[\psi\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right) + log\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right] \\ + \sum_{e=0}^{1}\left(\theta_{egh}^{(t+1)}\right)^{-1}\sum_{j=1}^{J_e}\left[\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\left(\frac{\theta_{egh}^{(t)}}{N_j^*\theta_{egh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]\end{aligned}$$

$f'(\varphi_{gh}) = (J_0 + J_1)\varphi_{gh}^{-1} - (J_0 + J_1)\psi'(\varphi_{gh}^{-1})\varphi_{gh}^{-2}$

Then we can update

$$\varphi_{gh}^{(t+1)} = \varphi_{gh,t+1}$$

if the iteration satisfies the stopping strategy.

Under H_0, the marginal likelihood function is

$$l_X(\theta_{gh}, \varphi_{gh}) = \sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\Big\{\log\Big(\Gamma\big(x_{ejgh} + \varphi_{gh}^{-1}\big)\Big) - \log\Big(\Gamma\big(x_{ejgh} + 1\big)\Big) - \log\Big(\Gamma\big(\varphi_{gh}^{-1}\big)\Big)$$

$$+ x_{ejgh}\log\big(N_j^*\theta_{gh}\big) - x_{ejgh}\log\big(N_j^*\theta_{gh} + \varphi_{gh}^{-1}\big) - \varphi_{gh}^{-1}\log\big(\varphi_{gh}N_j^*\theta_{gh} + 1\big)\Big\}$$

The joint log-likelihood function is

$$l_{X,\Theta}(\theta_{gh}, \varphi_{gh}) = \sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\Big\{x_{ejgh}\log\big(N_j^*\big) - \varphi_{gh}^{-1}\log\big(\varphi_{gh}\big) - \log\Big(\Gamma\big(x_{ejgh} + 1\big)\Big)$$

$$- \log\Big(\Gamma\big(\varphi_{gh}^{-1}\big)\Big) - \varphi_{gh}^{-1}\log\big(\theta_{gh}\big) + \big(x_{ejgh} + \varphi_{gh}^{-1} - 1\big)\log\big(\Theta_{jgh}\big)$$

$$- \left(N_j^* + \frac{1}{\varphi_{gh}\theta_{gh}}\right)\Theta_{jgh}\Big\}$$

Again, according to the EM algorithm, we have

E step:

$$Q\left(\theta_{gh}, \varphi_{gh} \middle| \theta_{gh}^{(t)}, \varphi_{gh}^{(t)}\right) = E_{\Theta|X}\left[l_{X,\Theta}\left(\theta_{gh}, \varphi_{gh}\right)\right]$$

$$= \sum_{e=0}^{1}\sum_{j=1}^{J_e}\sum_{h=1}^{H_g}\left\{x_{ejgh}\log\left(N_j^*\right) - \varphi_{gh}^{-1}\log\left(\varphi_{gh}\right) - \log\left(\Gamma\left(x_{ejgh}+1\right)\right)\right.$$

$$- \log\left(\Gamma\left(\varphi_{gh}^{-1}\right)\right) - \varphi_{gh}^{-1}\log\left(\theta_{gh}\right)$$

$$+ \left(x_{ejgh} + \varphi_{gh}^{-1} - 1\right)\left[\psi\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right) + \ln\left(\frac{\theta_{gh}^{(t)}}{N_j^*\theta_{gh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]$$

$$\left. - \left(N_j^* + \frac{1}{\varphi_{gh}\theta_{gh}}\right)\left[\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\left(\frac{\theta_{gh}^{(t)}}{N_j^*\theta_{gh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]\right\}$$

M step:

$$\theta_{gh}^{(t+1)}, \varphi_{gh}^{(t+1)} = argmax_{\theta_{gh},\varphi_{gh}} Q\left(\theta_{gh}, \varphi_{gh} \middle| \theta_{gh}^{(t)}, \varphi_{gh}^{(t)}\right)$$

We can calculate θ by

$$\theta_{gh}^{(t+1)} = \frac{\sum_{e=0}^{1}\sum_{j=1}^{J_e}\frac{\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\theta_{gh}^{(t)}}{N_j^*\theta_{gh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}}{\left(J_0 + J_1\right)}$$

and φ by

$$\varphi_{gh,t+1} = \varphi_{gh,t} - \frac{f\left(\varphi_{gh,t}\right)}{f'\left(\varphi_{gh,t}\right)}$$

where

$$f\left(\varphi_{gh}\right) = (J_0 + J_1)\log\left(\varphi_{gh}\right) - (J_0 + J_1) + (J_0 + J_1)\psi\left(\varphi_{gh}^{-1}\right) + (J_0 + J_1)\log\left(\theta_{gh}^{(t+1)}\right)$$

$$-\sum_{e=0}^{1}\sum_{j=1}^{J_e}\left[\psi\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right) + \ln\left(\frac{\theta_{gh}^{(t)}}{N_j^*\theta_{gh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]$$

$$+\left(\theta_{gh}^{(t+1)}\right)^{-1}\sum_{e=0}^{1}\sum_{j=1}^{J_e}\left[\left(x_{ejgh} + \left(\varphi_{gh}^{(t)}\right)^{-1}\right)\left(\frac{\theta_{gh}^{(t)}}{N_j^*\theta_{gh}^{(t)} + \left(\varphi_{gh}^{(t)}\right)^{-1}}\right)\right]$$

$$f'\left(\varphi_{gh}\right) = (J_0 + J_1)\varphi_{gh}^{-1} - (J_0 + J_1)\psi'\left(\varphi_{gh}^{-1}\right)\varphi_{gh}^{-2}$$

Then we can update

$$\varphi_{gh}^{(t+1)} = \varphi_{gh,t+1}$$

if the iteration satisfies the stopping strategy.

4.2.7 Likelihood Ratio Test

After parameter estimation, we can test the hypothesis by likelihood ratio test (LRT). Given a gene g, the test statistics is

$$\frac{\sup\limits_{H_0} \sum_{e=0}^{1}\sum_{h=1}^{H_g} l\left(\hat{\theta}_{gh}, \hat{\varphi}_{gh}\right)}{\sup\limits_{H_0 \cup H_1} \sum_{e=0}^{1}\sum_{h=1}^{H_g} l\left(\hat{\theta}_{egh}, \hat{\varphi}_{gh}\right)}$$

If the true parameters are the interior points of the parameter space, then under H_0, the test statistics follows a Chi-square distribution asymptotically with a degree of freedom H_g.

4.3 Simulation Studies

In this section, we will compare the performance of our method to that of the edgeR based on the simulation studies. The simulated data are generated by an R package ‘polyester’ according to the human transcript FASTA file, ‘Homo_sapiens.GRCh38.cdna.all.fa’, downloaded from Ensembl database. For convenience, we set our transcriptome to be only the protein coding transcripts on chromosome 1 with a transcript support level 1 or 2, which means that these transcripts are with a high confidence level. The number of selected genes is about 1800. Particularly, for paired-end reads, we ran the edgeR twice, one by treating a paired-end reads as two single-end reads and another by treating it as one long read. The detailed simulation settings are described below.

4.3.1 Simulation Settings

We simulated both single-end and paired-end RNA-seq reads from ‘polyester’. The distribution of the transcript copy number follows a negative binomial distribution with $\mu = 5$ and $\sigma^2 = 10$. The fragment length is taken to be 250 base-pair with standard error in 25 base-pair. The sequencing error is 0.005, which means that the probability of having a wrong nucleotide for one position is about 0.5%. We set two different conditions, 0 and 1, and set the percentage of differentially expressed genes to be around 30%. For each DE gene, the fold changes of its isoforms are selected equally likely from the set $\{0.25,0.5,1,2,4\}$. So if the fold changes of the isoforms are all selected to be 1 by chance, then that gene will not be treated as a differentially expressed gene. We assign equal number of samples to each condition, which vary in the set $\{2, 4,8,16\}$. We set the read length to be 100 base-pair for the single-end read and set both end to be this number for the paired-end read. The coverage is set to be 60 for the single-end read, so each nucleotide on transcriptome is covered by 60 short reads in average. For paired-end reads, each mate pair is treated as two single end reads and the coverage is controlled to be 60, too. We use the default setting of the dispersion parameter, which is

$reads\ per\ transcript \times fold\ changes/3$. For each simulation setting, we repeated 12 runs to make our conclusion more robust.

4.3.2 Simulation Results

We first checked the performance of type I error control by simulating the RNA-seq data without DE events. In this study, we only ran the simulation once and set 8 samples to both conditions. We draw the empirical CDF of the p-value for both single-end reads (Figure 4-6) and paired-end reads (Figure 4-7).

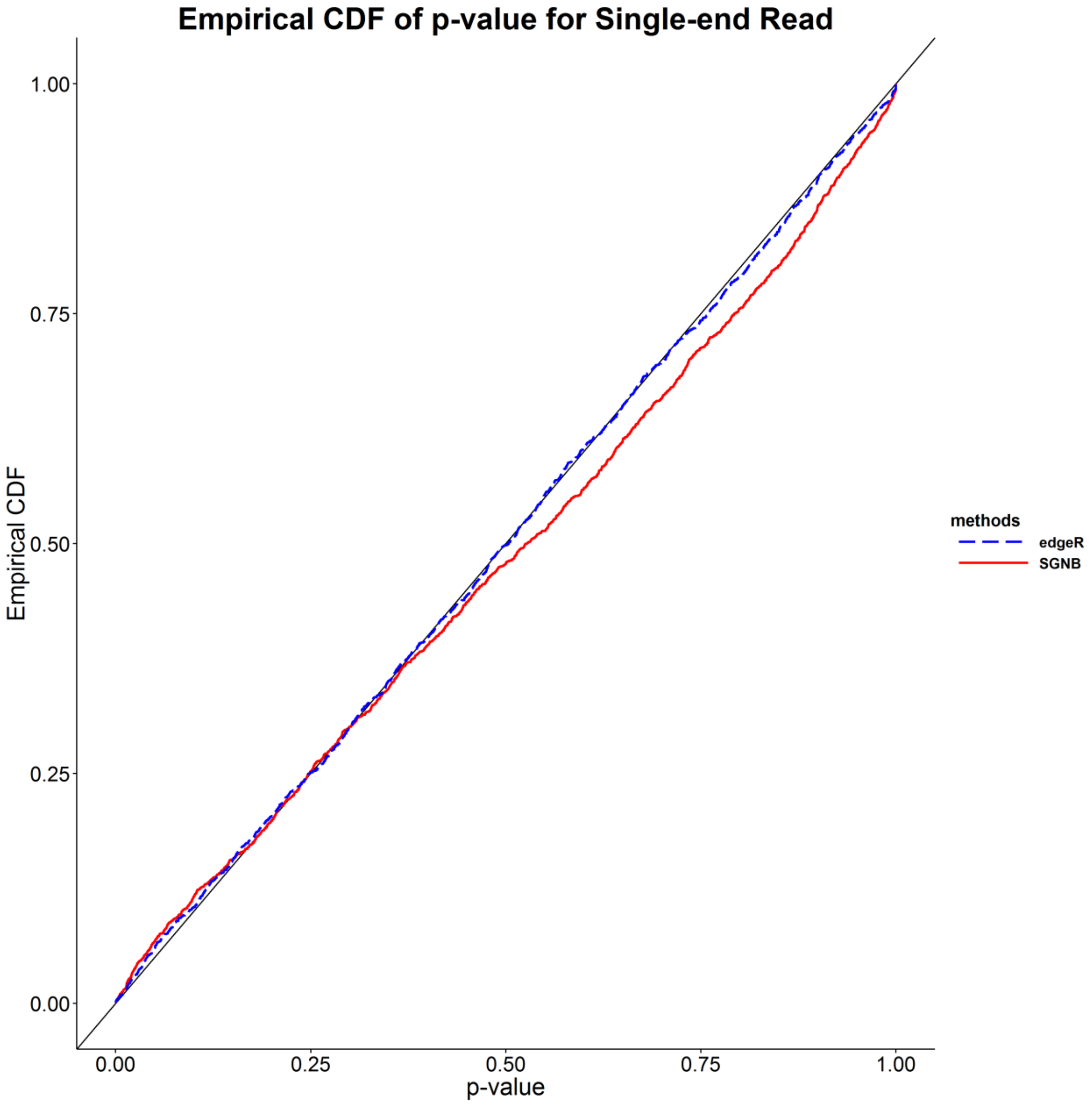


Figure 4-6: Empirical CDF of p-value for single-end read. The blue dash line is the CDF of edgeR and the red solid line is our method SGNB.

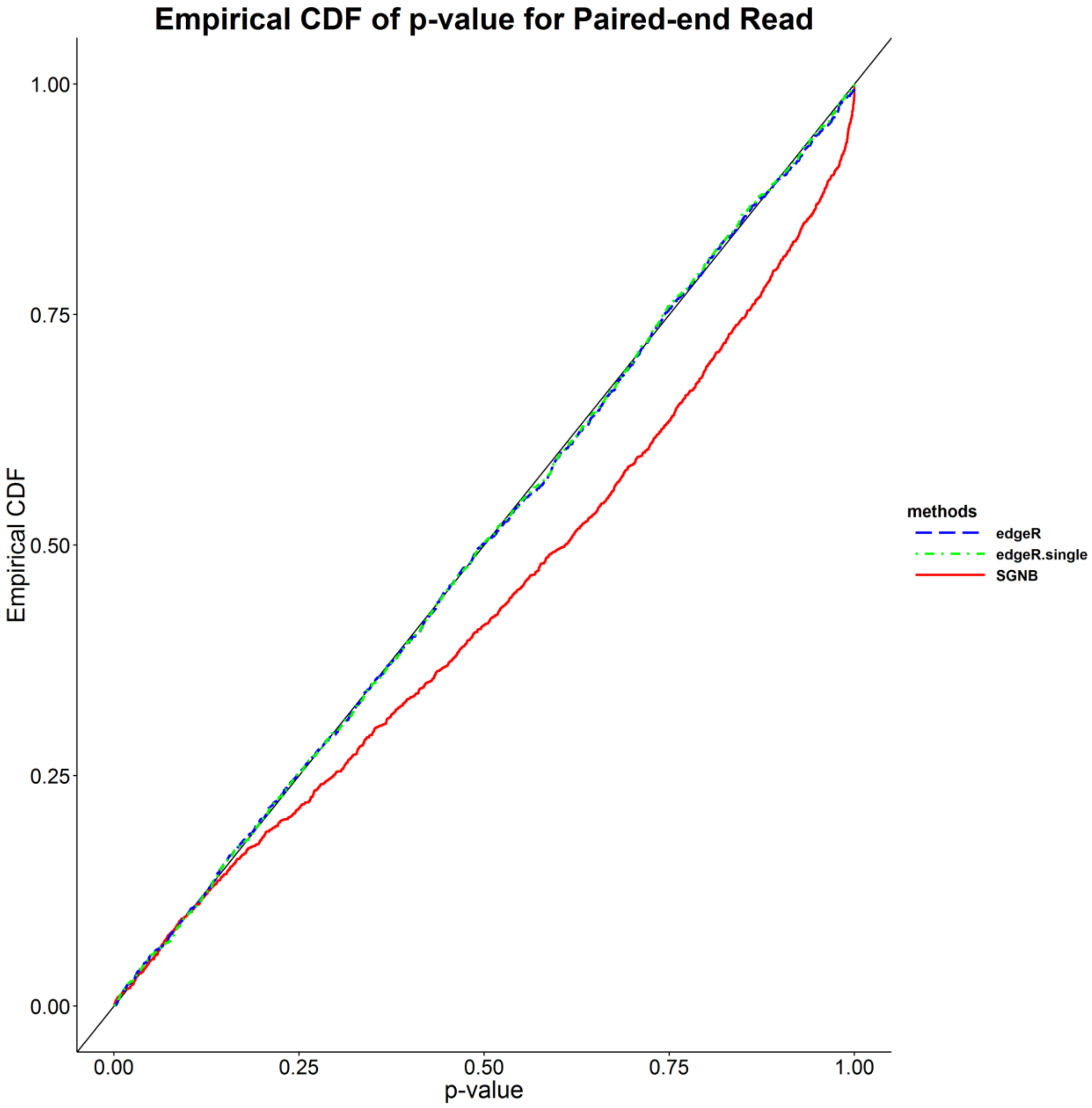


Figure 4-7: Empirical CDF of p-value for paired-end read. The blue dash line is the CDF of edgeR by treating a mate pair read as one long read. The green dot dash line is the CDF of edgeR by treating the paired-end read as the single-end read. The red solid line is for our method.

Theoretically, the plot of valid p-values vs their CDF is expected to follow the diagonal line. For the single-end reads, SGNB and edgeR share a similar CDF before p-value reaches 0.5 and SGNB is a little bit lower than edgeR after 0.5. For the paired-end reads, SGNB is almost always below the diagonal line, suggesting that our method is more conservative, while edgeR can control the type-I error pretty well.

Next, we checked the hypothesis testing performance under each sample size setting through the ROC curves (Figures 4-8, 4-9, 4-10 and 4-11). The ROC curve is draw by plotting the true positive rate against the false positive rate. A better model should have a larger area under the ROC curve.

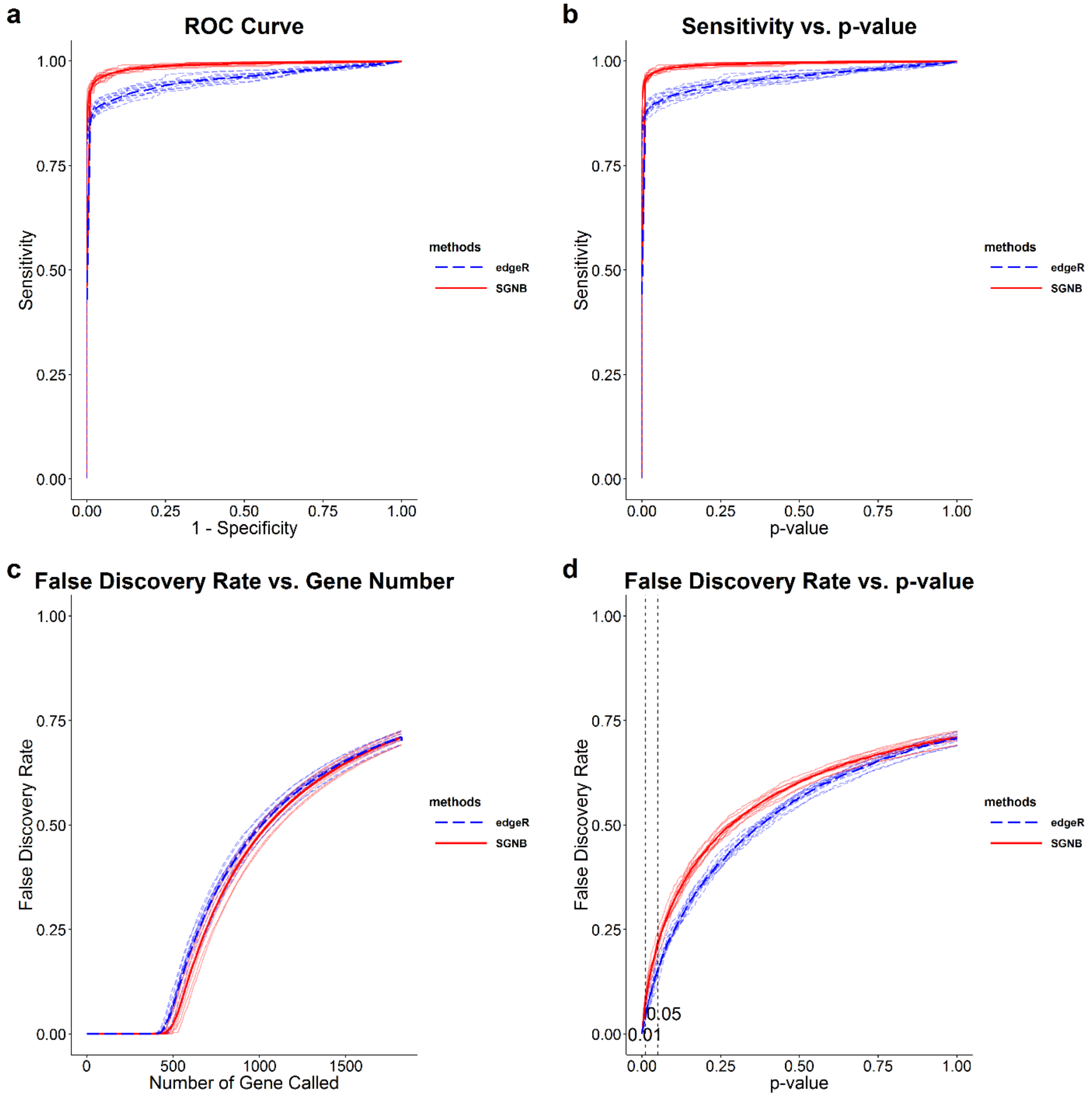


Figure 4-8: Performance comparisons for single-end read with sample size 2. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

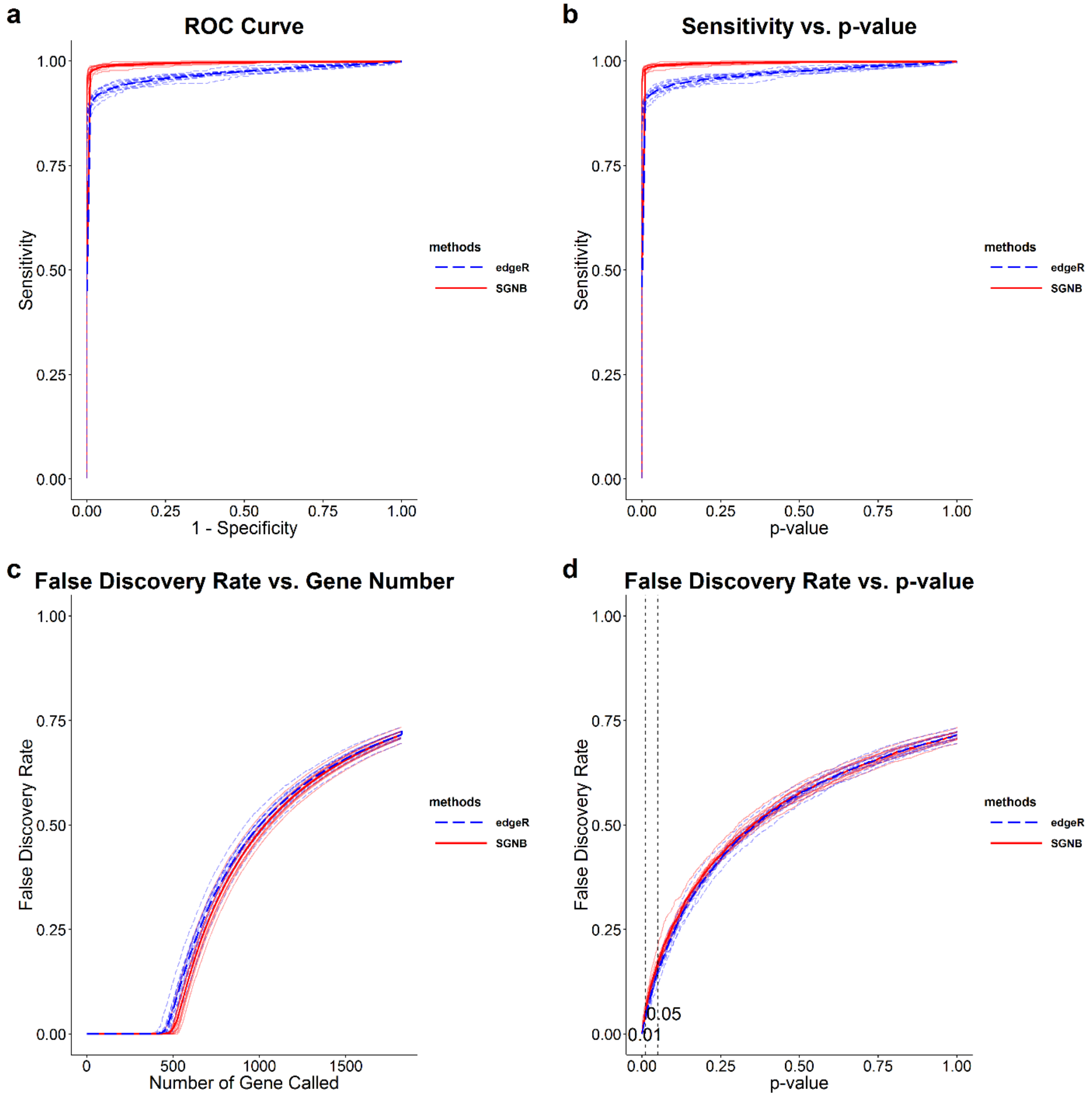


Figure 4-9: Performance comparisons for single-end read with sample size 4. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

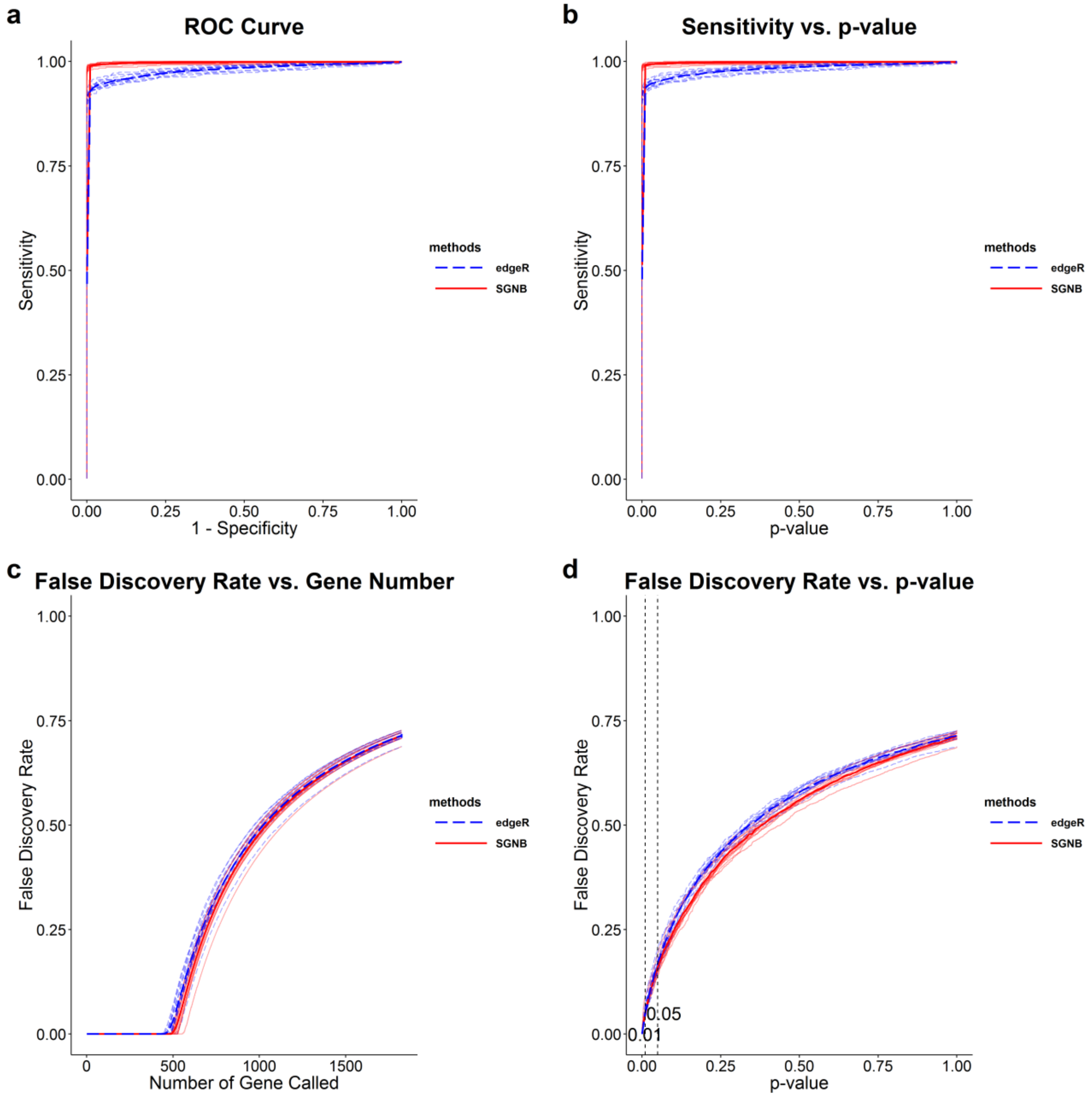


Figure 4-10: Performance comparisons for single-end read with sample size 8. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

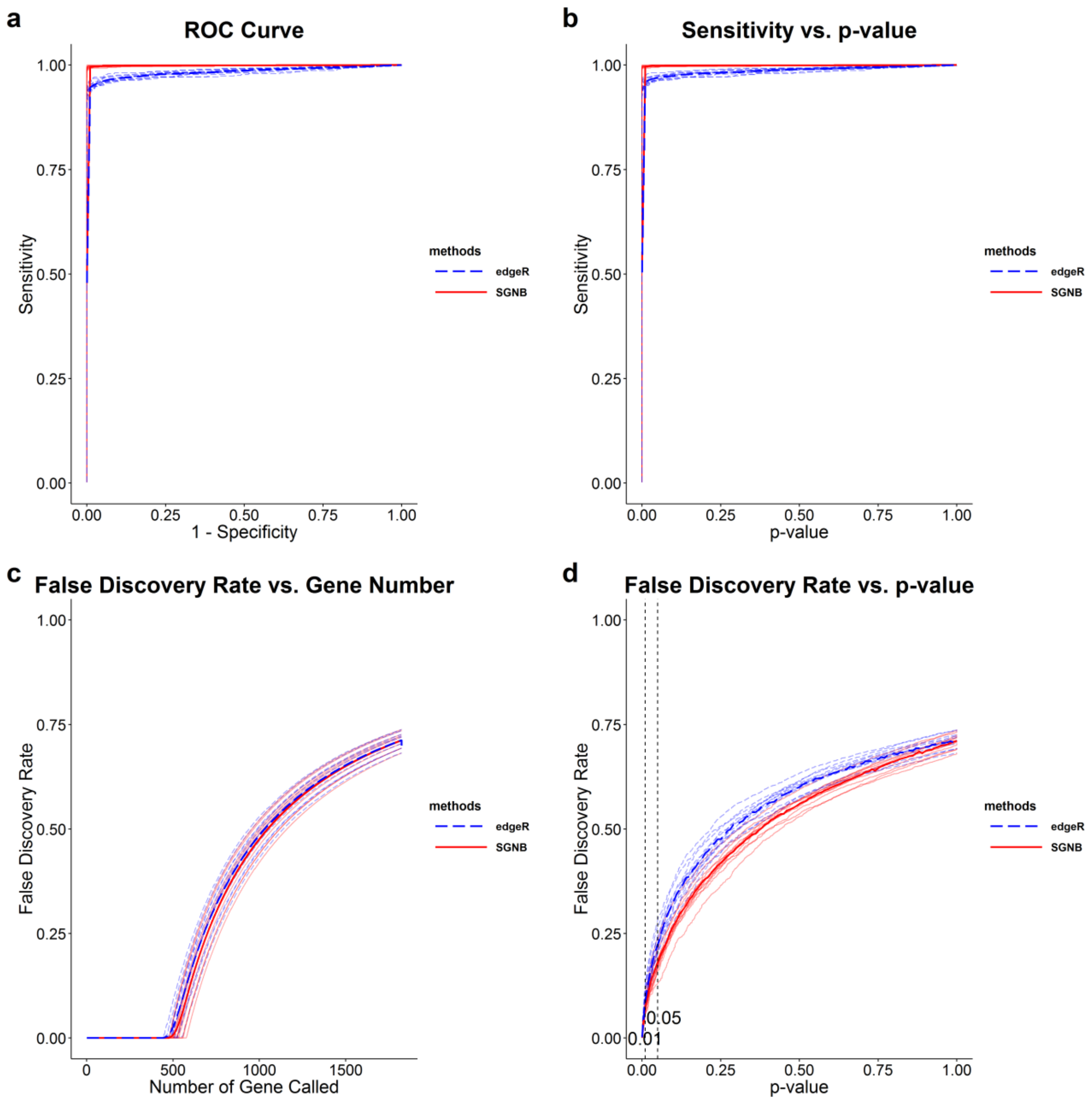


Figure 4-11: Performance comparisons for single-end read with sample size 16. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

As we can see from the graphs above, our method always has a higher ROC curve than the edgeR package. That is, we can achieve a higher true positive rate at a fixed p-value level. When the sample size is small (i.e. 2), our method has a little bit worse false discovery rate controlled by p-value. But when the sample size increases, SGNB becomes better and can beat edgeR after sample size 8. The similar results can be seen for the paired-end read (Figures 4-12, 4-13, 4-14, 4-15).

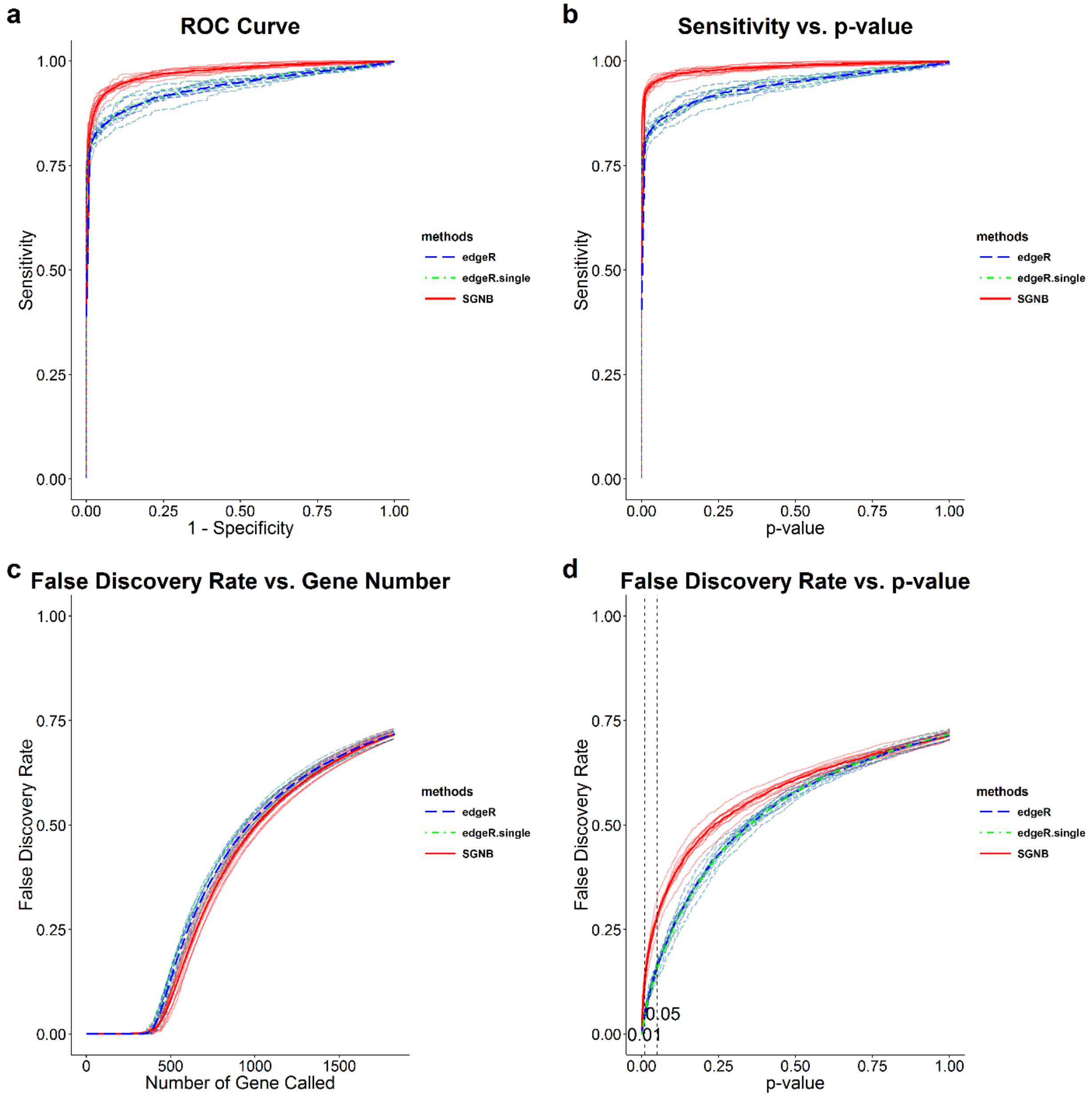


Figure 4-12: Performance comparisons for paired-end read with sample size 2. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

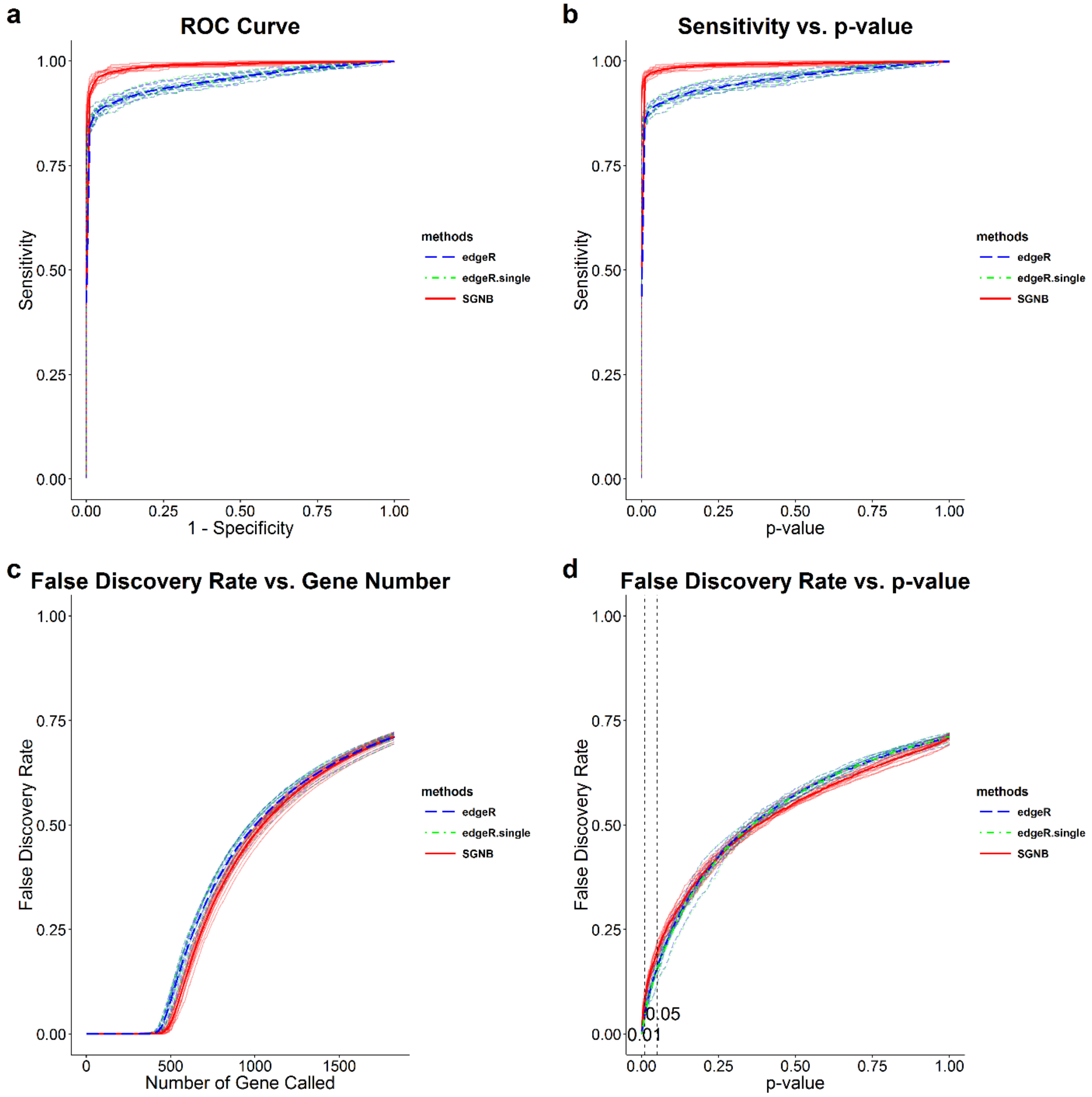


Figure 4-13: Performance comparisons for paired-end read with sample size 4. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

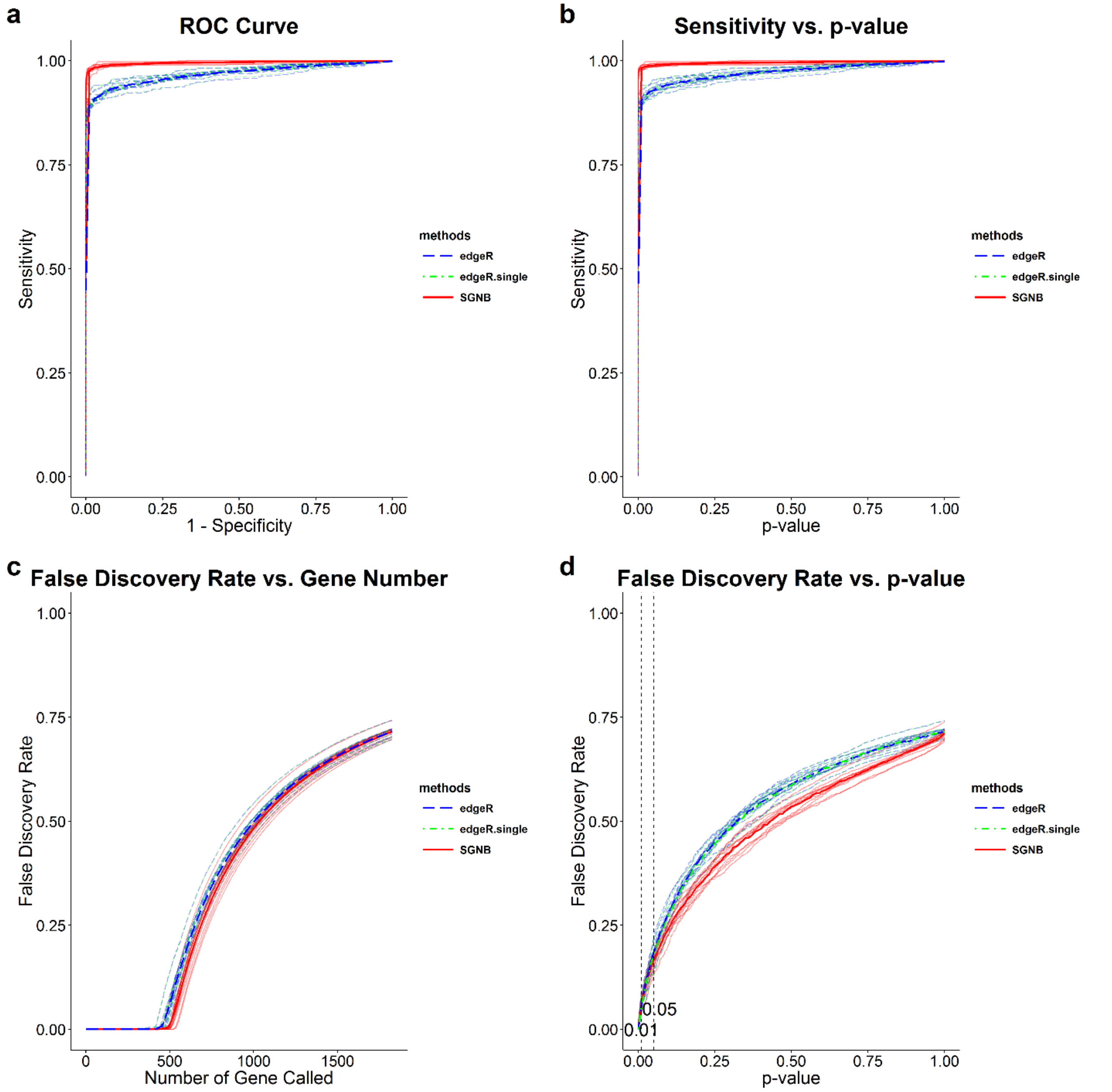


Figure 4-14: Performance comparisons for paired-end read with sample size 8. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

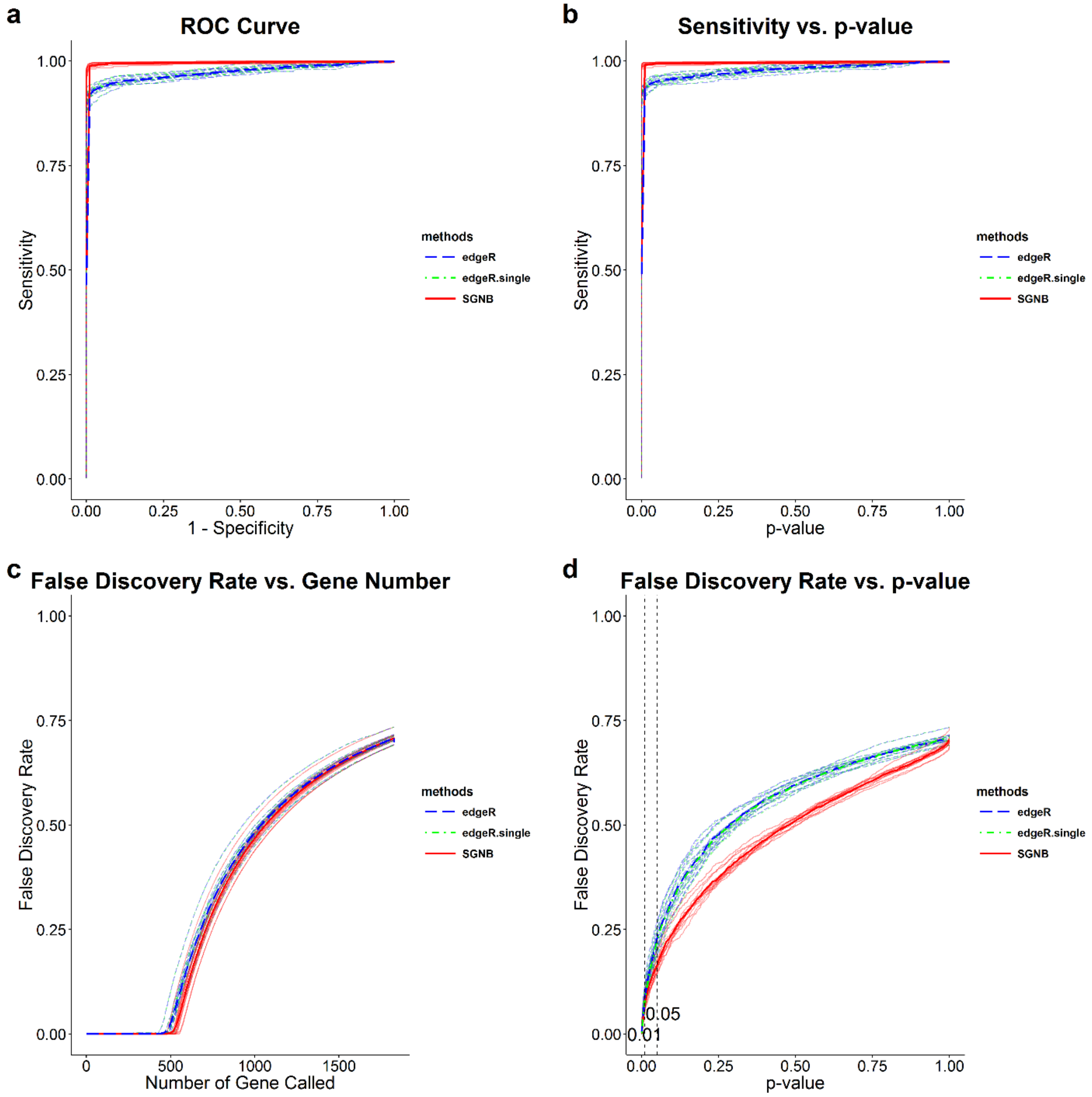


Figure 4-15: Performance comparisons for paired-end read with sample size 16. a). ROC curve. b). Sensitivity vs. p-value. c). False Discovery Rate vs. Gene Calling Number. d). False Discovery Rate vs. p-value.

Finally, we controlled the type one error rate at 0.05 significant level and calculated the corresponding true positive rate under different sample sizes. Then we plotted these true positive rates against the sample sizes (Figures 4-16 and 4-17).

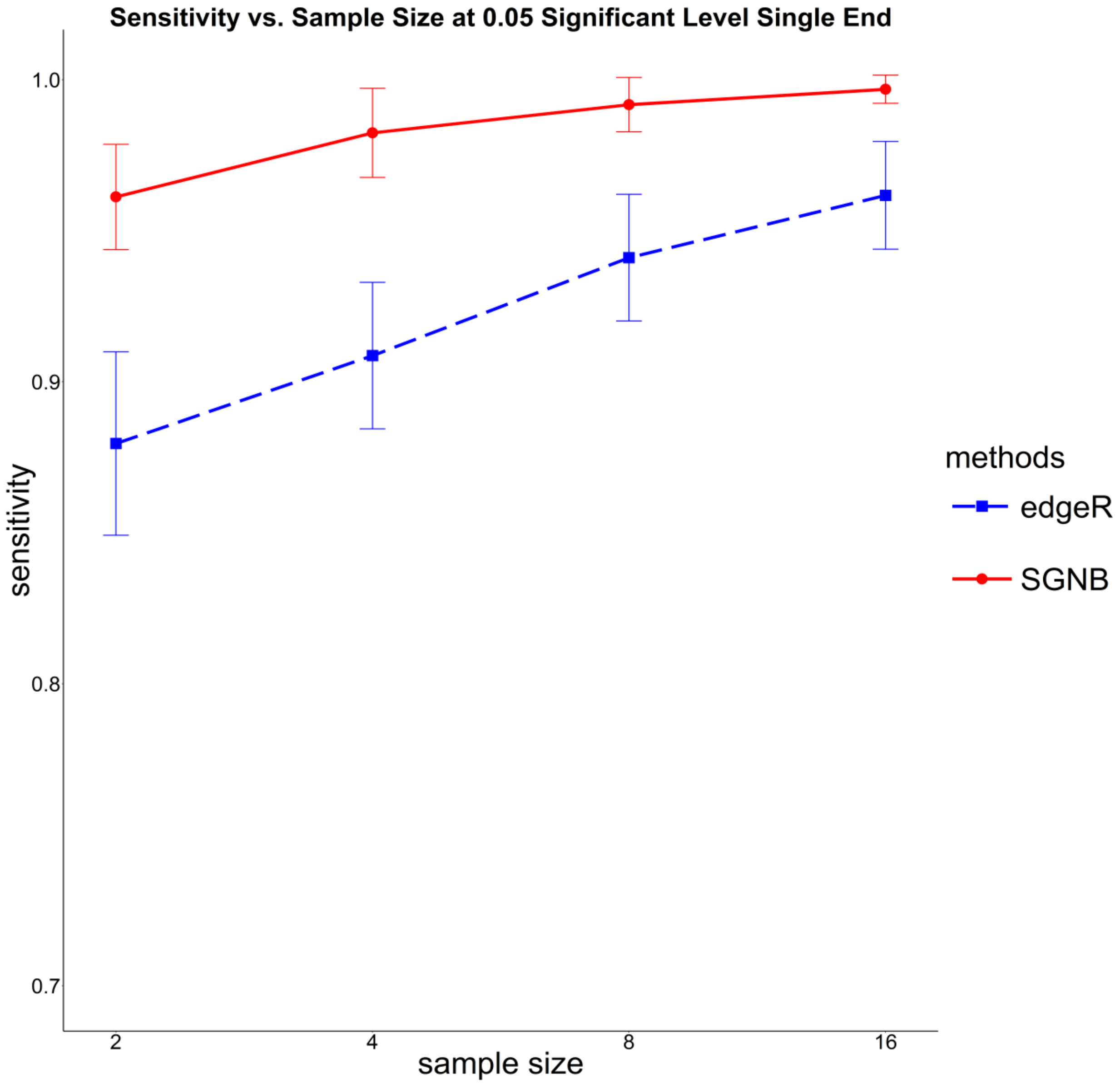


Figure 4-16: Power vs. sample size for single-end read. The points are the average power calculated form the 12 runs and the error bar denotes its 95% CI.

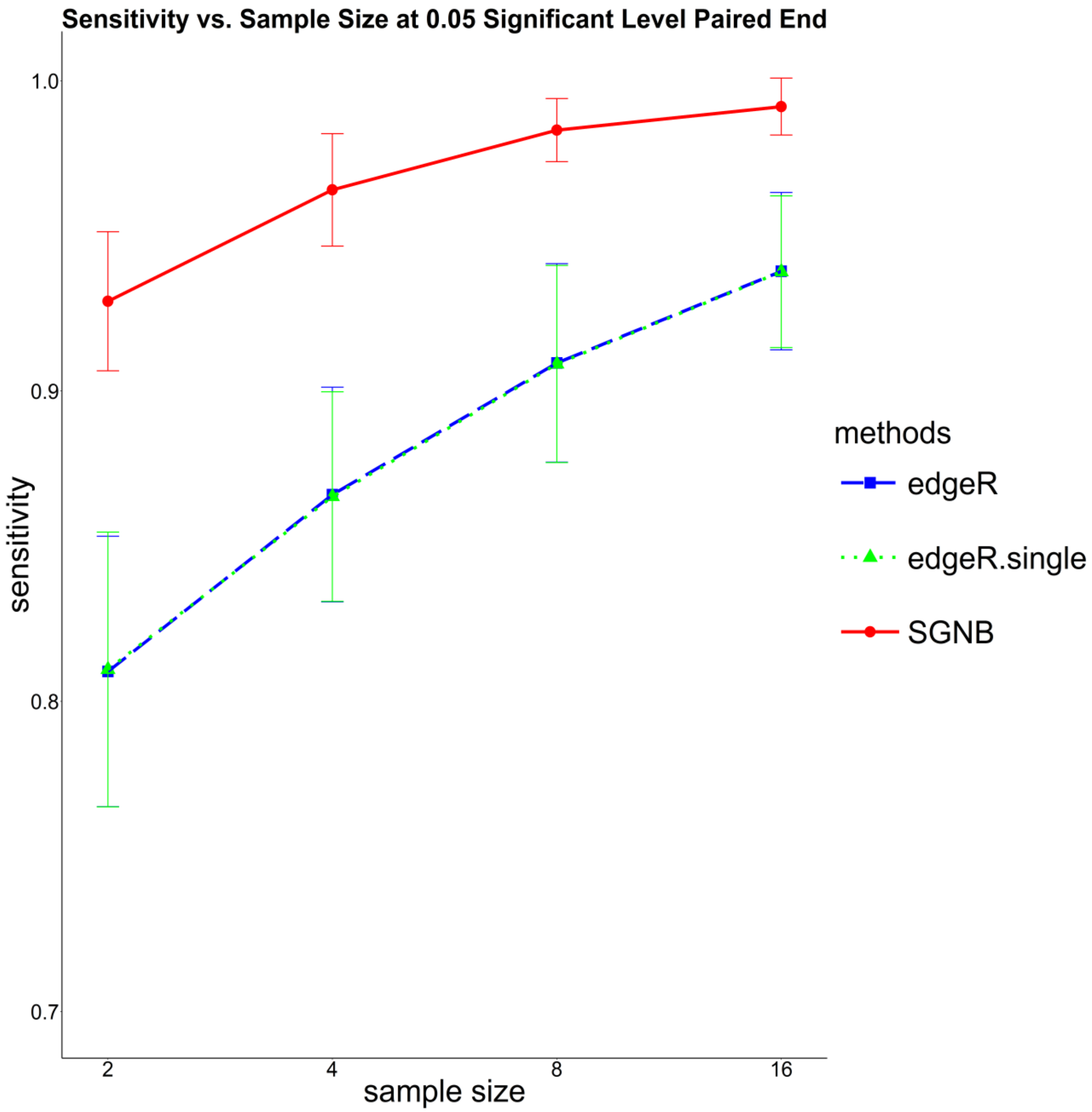


Figure 4-17: Power vs. sample size for paired-end read. The points are the average power calculated form the 12 runs and the error bar denotes its 95% CI.

We can see that our method always has higher powers with smaller standard errors. At the same time, with the increase of the sample size, both edgeR and SGNB seem like to increase the power.

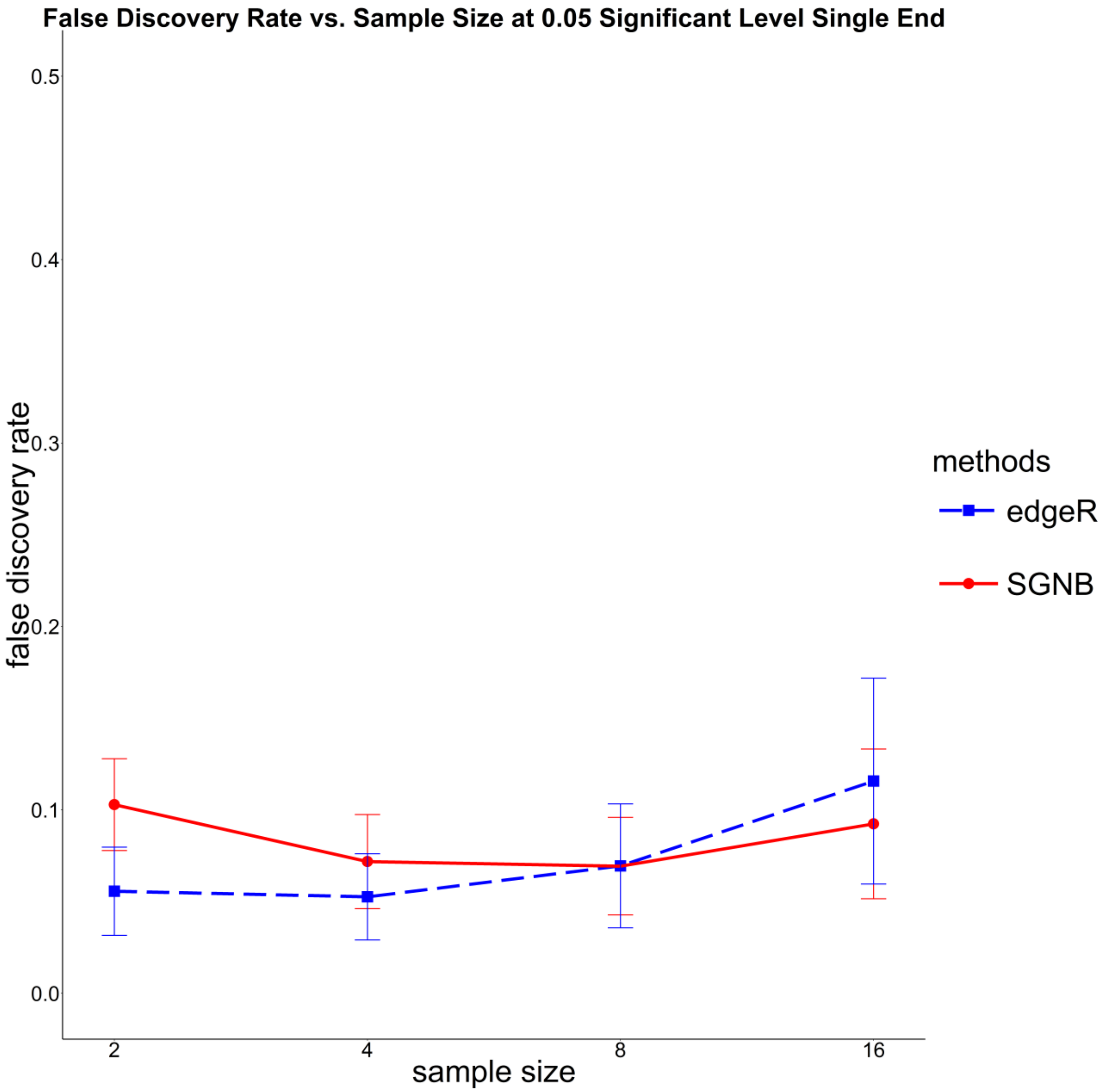


Figure 4-18: False discovery rate vs. sample size for single-end read. The points are the average false discovery rate calculated from 12 runs and the error bar denotes its 95% CI.

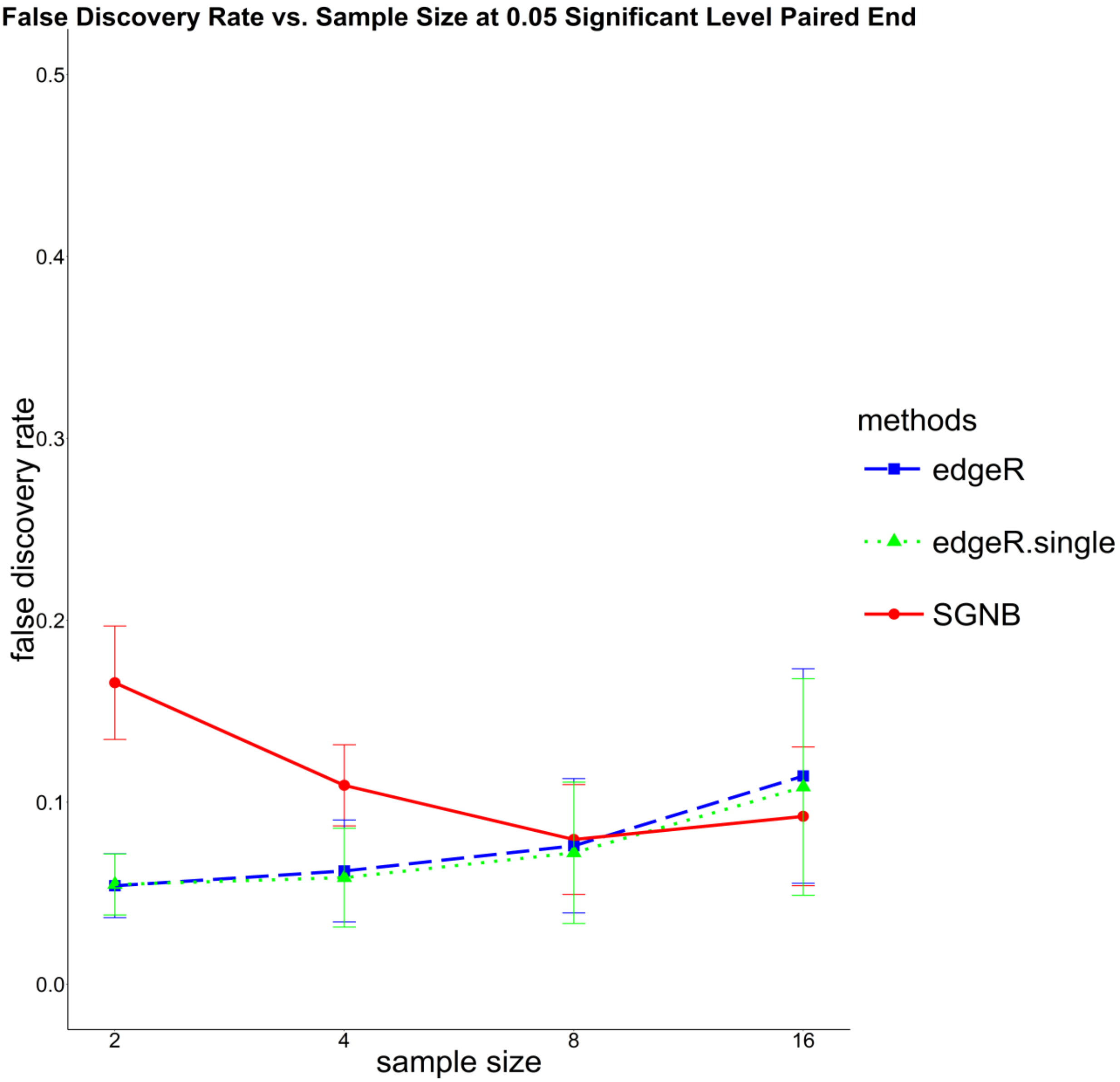


Figure 4-19: False discovery rate vs. sample size for paired-end read. The points are the average false discovery rate calculated from 12 runs and the error bar denotes its 95% CI.

For both single-end and paired-end reads, our method has a little bit worse false discovery rate when the sample size is small (i.e. 2, 4). However, the performance becomes better when the sample size increases, and after sample size 8, SGNB is better than edgeR.

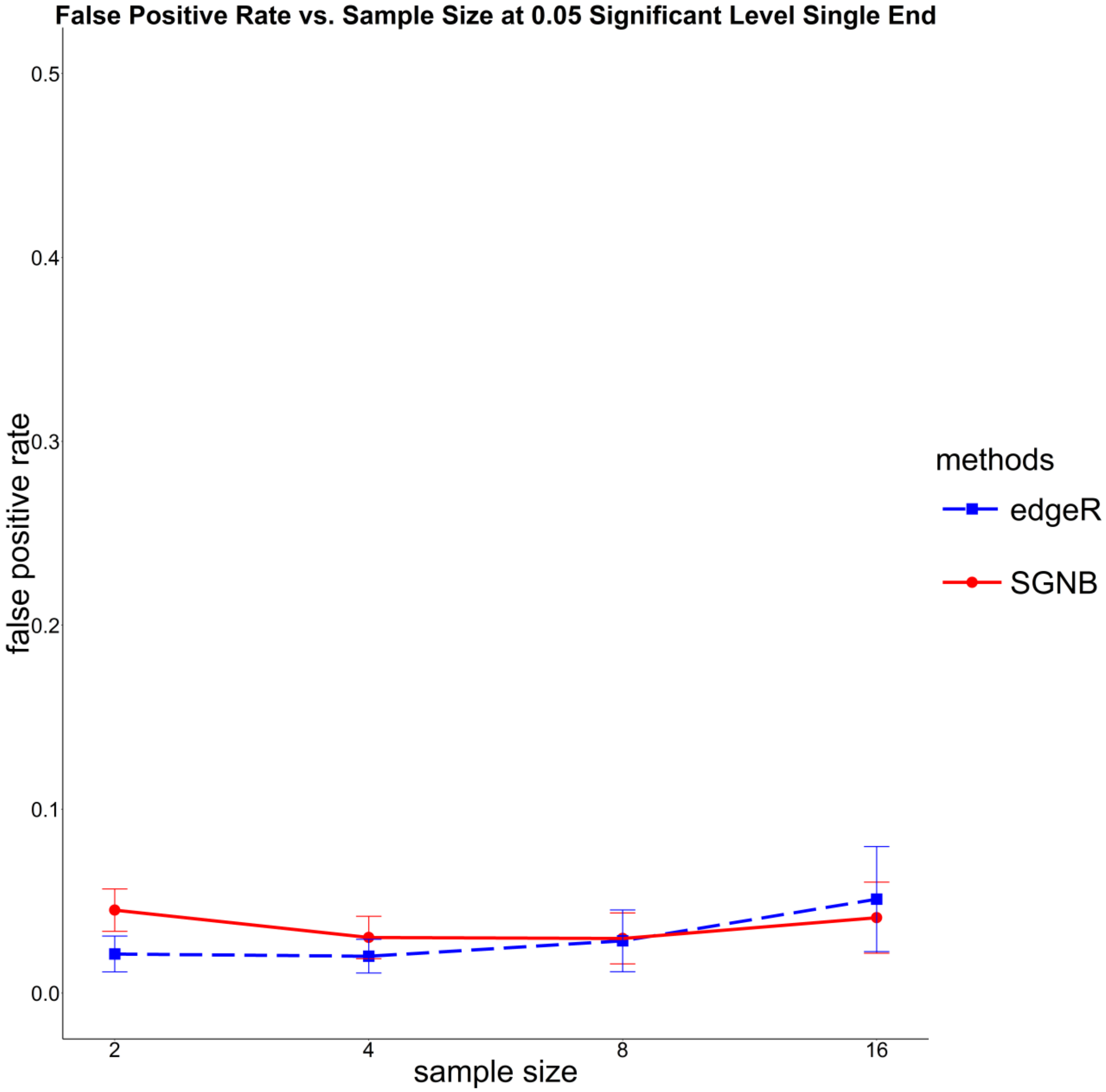


Figure 4-20: False positive rate vs. sample size for single-end read. The points are the average false positive rate calculated from 12 runs and the error bar denotes its 95% CI.

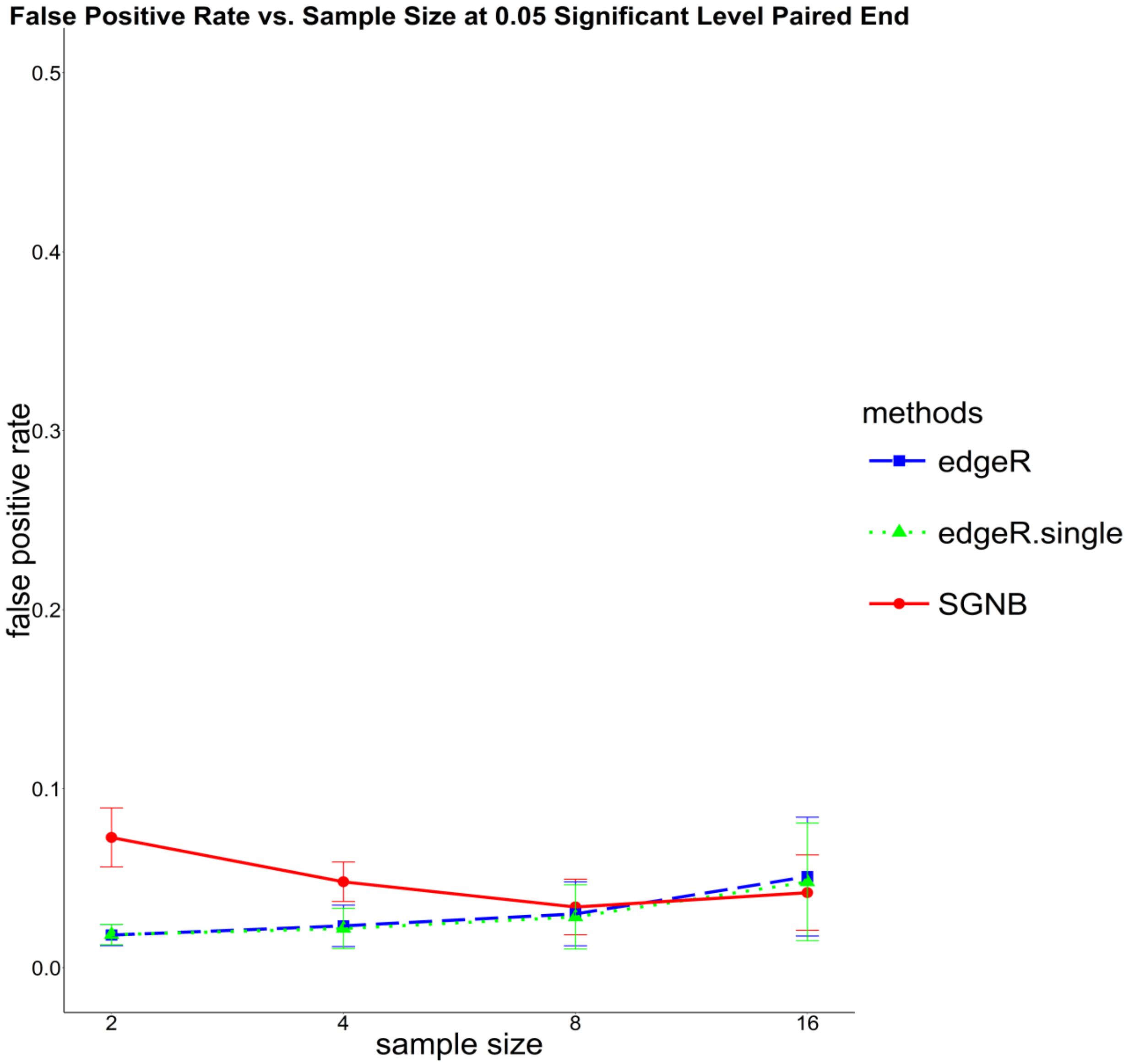


Figure 4-21: False positive rate vs. sample size for paired-end read. The points are the average false positive rate calculated from 12 runs and the error bar denotes its 95% CI.

The performance of controlling false positive rate is similar to false discovery rate. Our method is a little bit worse than edgeR before sample size 8 and is better than edgeR after sample size 8.

From these results, it is clear that under the same sample size and sequencing depth, our model is able to achieve a higher power without losing the control of both false positive rate and false discovery rate.

4.4 Real Data Analysis

We analyzed a real RNA-seq data by both SGNB and edgeR. The goal is to compare gene expression levels in platelet samples from 5 healthy individuals (control group) and 7 ET (essential thrombocythemia) patients (study group). Totally 26586 genes were considered. The Venn Diagram was created to compare the significant genes called by edgeR and SGNB at 0.05 significant level (Figure 4-22).

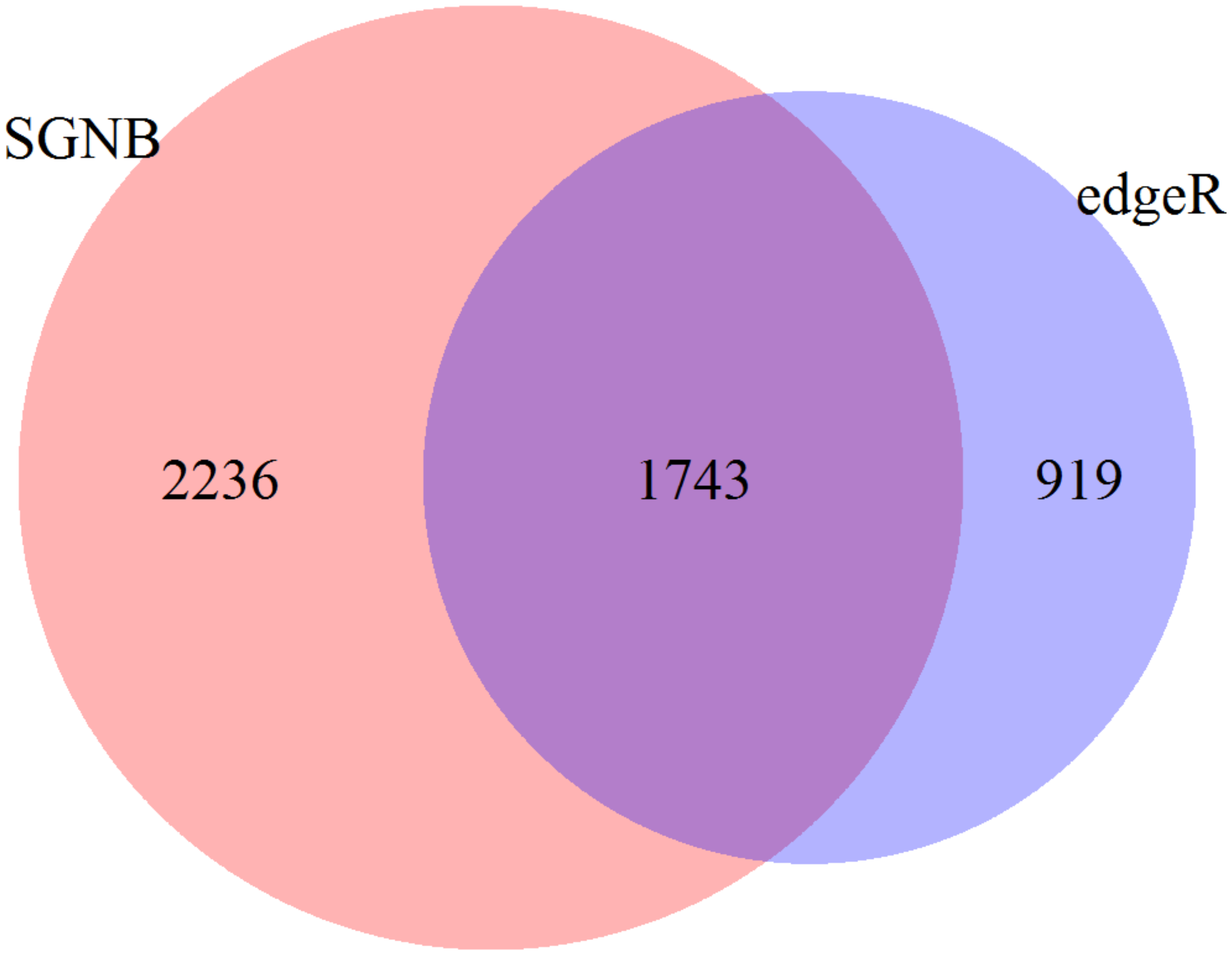


Figure 4-22: Venn diagram of significant DE genes. The red area denotes the number of DE genes called by SGNB while the blue area is for edgeR.

Within the 26586 genes, there are 3979 (15%) DE genes called by SGNB and 2662 (10%) DE genes called by edgeR. The number of common genes is 1743, which is about 65% of the total number of genes called by edgeR. There are 2236 genes detected by SGNB only and 919 genes detected by edgeR only. For these genes, it is hard to tell if they are true or not; however, we compared the distribution of the log fold changes of the gene expression level for these genes (Figure 4-23). As shown in Figure 4-23 b and c, genes only called by SGNB usually have a smaller absolute log-fold change compared with the genes only called by edgeR. It should be a reasonable result, since the edgeR is good at detecting the DE genes with a large total amount changing, while SGNB is able to detect both total amount and structure changing. That is, although the genes only called by SGNB have smaller fold changes in the total amount, they might have a significant changing in their isoform structures, which cannot be identified by edgeR. It is also possible that we might get some false positives among the genes only called by SGNB due to the low sequencing coverage. At the same time, there were genes with a large fold change that could be detected by edgeR but not SGNB. The reason might be that SGNB tests multiple hypotheses while edgeR only tests one. That is, in some situation, we might loss the power due to test the multiple hypotheses.

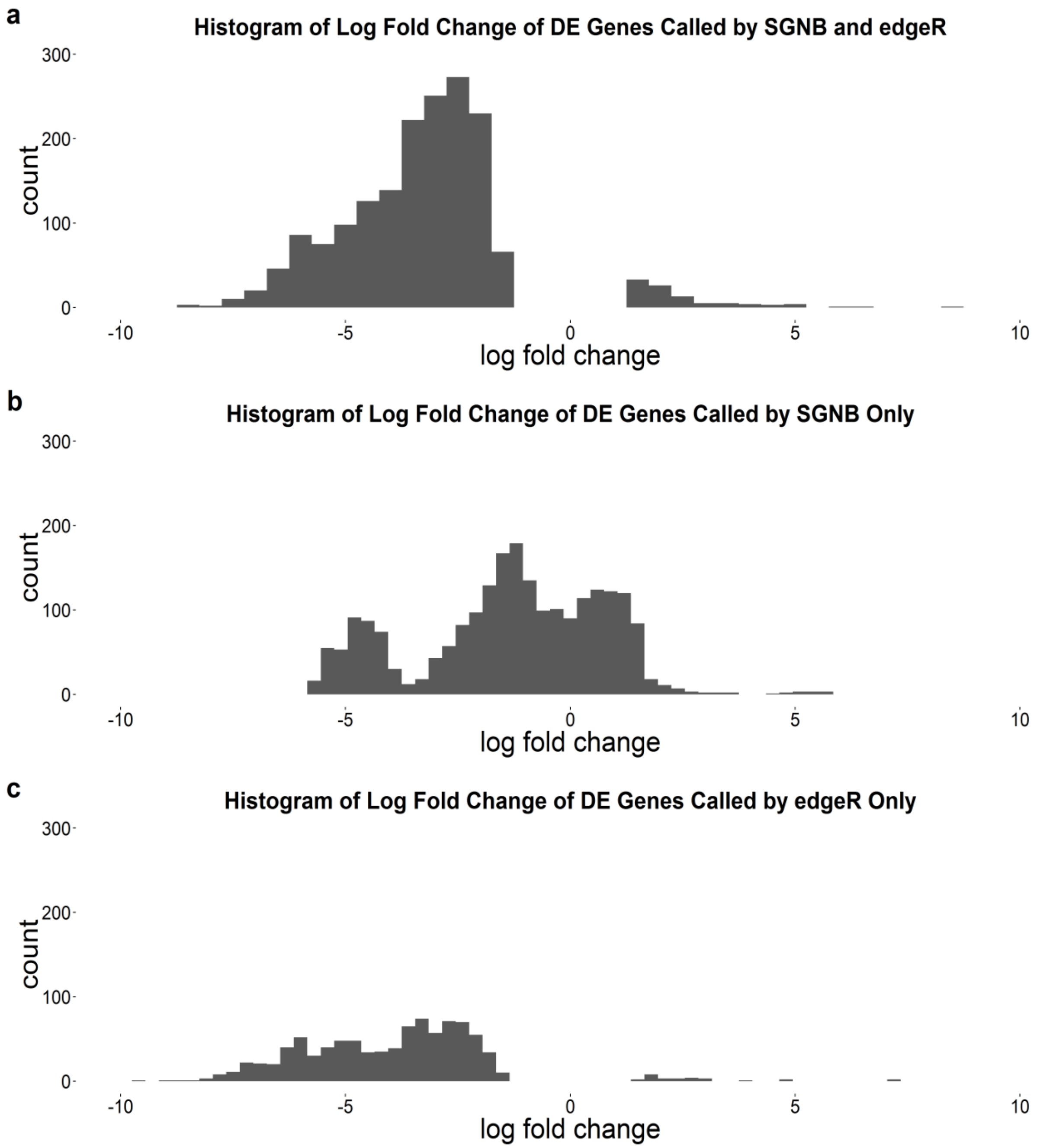


Figure 4-23: Histogram of log fold change. a). The distribution of log fold change of DE genes called by both SGNB and edgeR. b). The distribution of log fold change of DE genes called by SGNB only. c). The distribution of log fold change of DE genes called by edgeR only.

Chapter 5 Discussion and Future Work

We have shown that, for the purpose of quantifying both the amount and structure changing of isoforms, the SGNB is more powerful than edgeR. At the same time, although we cannot estimate am expression level for each isoform without a well-defined structure, we can give an estimation at gene level by simply summing read type expression levels together. However, there are still more work that need to be done with our method.

Firstly, for one paired-end read, we simply treat it as two single-end reads. By doing so, we ignore the information provided by the distribution of the insertion length. According to research results from other groups, making a good usage of the insertion length could lead to a better model fitting (Salzman, Jiang, & Wong, 2011), since the guessing of where the read comes from could be more accurate. So we may try to figure out how to define the read type by taking into consideration of the insertion length.

Secondly, in this work, we only proposed our method for two condition comparisons. Our method should be able to be extended for multiple group comparisons. One way is to simply write down the likelihood functions under each condition, then multiply them together to get the joint likelihood function. We can perform the hypothesis testing based on the likelihood ratio test statistics. However, the performance needs to be checked for this extension. We need to show that the model can control the type I error and the false discovery rate.

Finally, our current model is only good for independent samples and it may be extended to paired samples. Since paired samples usually yield a higher power, it will be very valuable to generalize our method to account for the paired samples.

References

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol.*, 11(10): R106.

Anders, S., Pyl, P. T., & Huber, W. (2015). HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics.*, 31(2): 166-169.

Bernard, E., Jacob, L., Mairal, J., & Vert, J.-P. (2014). Efficient RNA isoform identification and quantification from RNA-Seq data with network flows. *Bioinformatics.*, 30(17): 2447-2455.

Crick, F. (1970). Central dogma of molecular biology. *Nature.*, 227(5258): 561-563.

Dilworth, R. P. (1950). A decomposition theorem for partially ordered sets. *Annals of Mathematics.*, 51(1): 161-166.

Ewing, B., & Green, P. (1998). Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.*, 8(3): 186-94.

Ferragina, P., & Manzini, G. (2000). Opportunistic data structures with applications. *IEEE.*, 390 - 398.

Gauthier, K., Chassande, O., Plateroti, M., Roux, J.-P., Legrand, C., Pain, B., . . . Samarut, J. (1999). Different functions for the thyroid hormone receptors TRα and TRβ in the control of thyroid hormone production and post-natal development. *The EMBO Journal.*, 18: 623-631.

Glaus, P., Honkela, A., & Rattray, M. (2012). Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics.*, 28(13): 1721-1728.

Hansen, K. D., Brenner, S. E., & Dudoit, S. (2010). Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucl. Acids Res.*, 38(12): e131.

Hansen, K. D., Wu, Z., Irizarry, R. A., & Leek, J. T. (2011). Sequencing technology does not eliminate biological variability. *Nature Biotechnology.*, 29: 572–573.

Hardcastle, T. J., & Kelly, K. A. (2010). baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics.*, 11: 422.

Jiang, H., & Wong, W. H. (2009). Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics.*, 25(8): 1026-1032.

Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, 10(3): R25.

LeGault, L. H., & Dewey, C. N. (2013). Inference of alternative splicing from RNA-Seq data with probabilistic splice graphs. *Bioinformatics.*, 29(18): 2300-2310.

Lehmann, E. L. (1998). *Theory of point estimation.* Springer.

Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., & Dewey, C. N. (2010). RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics.*, 26(4): 493–500.

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics.*, 25(14): 1754-1760.

Li, H., Ruan, J., & Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, 18(11): 1851-1858.

Li, J., Jiang, H., & Wong, W. H. (2010). Modeling non-uniformity in short-read rates in RNA-Seq data. *Genome Biology.*, 11: R50.

Liu, J. (2002). Monte Carlo strategies in scientific computing. *Springer*.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., & Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.*, 18(9): 1509-1517.

Matlin, A. J., Clark, F., & Smith, C. W. (2005). Understanding alternative splicing: towards a cellular code. *Nat Rev Mol Cell Biol.*, 6(5): 386-398.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods.*, 5(7): 621-628.

Ouelle, D. E., Zindy, F., Ashmun, R. A., & Sherr, C. J. (1995). Alternative reading frames of the INK4a tumor suppressor gene encode two unrelated proteins capable of inducing cell cycle arrest. *Cell.*, 83(6): 993-1000.

Pray, L. A. (2008). Discovery of DNA structure and function: Watson and Crick. *Nature Education.*, 1(1): 100.

Roberts, A., Trapnell, C., Donaghey, J., Rinn, J. L., & Pachter, L. (2011). Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology.*, 12: R22.

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology.*, 11: R25.

Robinson, M. D., & Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics.*, 23(21): 2881-2887.

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

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics.*, 26(1): 139–140.

Salzman, J., Jiang, H., & Wong, W. H. (2011). Statistical Modeling of RNA-Seq Data. *Statist. Sci.*, 26(1): 62-83.

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science.*, 270(5235): 467-470.

Sinden, R. R. (1994). DNA structure and function. *Academic Press*.

Trapnell, C., Pachter, L., & Salzberg, S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics.*, 25(9): 1105-1111.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., Baren, M. J., . . . Pachter, L. (2010). Transcript assembly and abundance estimation from RNA-Seq reveals thousands of new transcripts and switching among isoforms. *Nat Biotechnol.*, 28(5): 511–515.

Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., . . . Burge, C. B. (2008). Alternative isoform regulation in human tissue transcriptomes. *Nature.*, 456(7221): 470-476.

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet.*, 10(1): 57-63.