Fall 2013

Statistical Models for Gene and Transcripts Quantification and Identification Using RNA-Seq Technology

Han Wu
Purdue University

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Han Wu

Entitled
Statistical Models for Gene and Transcripts Quantification and Identification Using RNA-Seq Technology

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Michael Yu Zhu
Chair

Rebecca Doerge

Jun Xie

Dabao Zhang

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Michael Yu Zhu

Approved by: Rebecca Doerge 11/18/13
Head of the Graduate Program Date

STATISTICAL MODELS FOR GENE AND TRANSCRIPTS QUANTIFICATION AND IDENTIFICATION USING RNA-SEQ TECHNOLOGY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Han Wu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2013

Purdue University

West Lafayette, Indiana

To my parents.

ACKNOWLEDGMENTS

I would like to express my appreciation to Professor Michael Yu Zhu, who has served as the chair of my dissertation committee, Professor Dabao Zhang, Professor Jun Xie, and Professor Rebecca Doerge, for serving on my dissertation committee and for their insightful comments and encouragements.

I am particular grateful to my adviser, Professor Yu Zhu for the countless helpful discussions and guidances. Without these intuitions and discussions, I could not have completed my research and the writing of my thesis.

Last but not the least, I would like to thank my parents for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Wu, Han Ph.D., Purdue University, December 2013. Statistical Models for Gene and Transcripts Quantification and Identification Using RNA-Seq Technology. Major Professor: Michael Y. Zhu.

RNA-Seq has emerged as a powerful technique for transcriptome study. As much as the improved sensitivity and coverage, RNA-Seq also brings challenges for data analysis. The massive amount of sequence reads data, excessive variability, uncertainties, and bias and noises stemming from multiple sources all make the analysis of RAN-Seq data difficult. Despite much progress, RNA-Seq data analysis still has much room for improvement, especially on the quantification of gene and transcript expression levels. The quantification of gene expression level is a direct inference problem, whereas the quantification of the transcript expression level is an indirect problem, because the label of the transcript each short read is generated from is missing. A number of methods have been proposed in the literature to quantify the expression levels of genes and transcripts. Although being effective in many cases, these methods can become ineffective in some other cases, and may even suffer from the non-identifiability problem. A key drawback of these existing methods is that they fail to utilize all the formation in the RNA-Seq short read count data. In this thesis, we propose three model frameworks to address three important questions in RNA-Seq study. First, we propose to use finite Poisson mixture models (PMI) to characterize base pair-level RNA-Seq data and further quantify gene expression levels. Finite Poisson mixture models combine the strength of fully parametric models with the flexibility of fully nonparametric models, and are extremely suitable for modeling heterogeneous count data such as what we observed from RNA-Seq experiments. A unified quantification method based on the Poisson mixture models is developed to measure gene expression levels. Second, based on the Poisson mixture model framework, we further proposed the convolution of Poisson mixture models (CPM-Seq) to quantify the expression levels of transcripts. The maximum likelihood estimation method equipped with the EM algorithm is used to estimate model parameters and quantify transcript expression levels. Third, a penalized convolution Poisson mixture model (penCPM-Seq) is proposed to shrink transcripts with small expression levels to zero and to select transcripts that have high expression levels from the candidate set. Both simulation studies and real data applications have demonstrated the effectiveness of PMI, CPM-Seq, and penCPM-Seq. We will show that they produced more accurate and consistent quantification results than existing methods. Thus, we believe that finite Poisson mixture models provide a flexible framework to model RNA-Seq data, and methods developed based on this thesis have the potential to become powerful tools for RNA-Seq data analysis.

1. INTRODUCTION

The central dogma of molecular biology explains the process of how DNAs are transcribed to RNAs, which are further translated to proteins. Gene stores genetic information, and the genetic information is copied to RNA via transcription, which is the first step of gene expression. During this process, segments of DNA, known as exons, are copied to make the RNA. Each resulting molecule is referred to as a transcript. The same gene can encode multiple transcripts or proteins through the process of alternative splicing. The second step in central dogma of molecular biology is translation, during which mRNA uses the genetic information to produce protein via translation. Thus, it is of great importance to understand the process of both transcription and translation. In particular, the first step to understanding the central dogma of molecular biology is to understand transcription because that is where most of the regulation of gene expression occurs.

1.1 RNA-Seq Technology

Overview of the Technology

The rapid development of next generation sequencing (NGS) technologies has revolutionized the way genomic research can be conducted. Among all successful applications of the NGS technologies, RNA-Seq has become an important tool for transcriptome profiling [1]. The transcriptome is the complete set of transcripts in a cell under any given developmental stage or physiological condition. Comprehensively detecting, cataloging and quantifying all of the components in the transcriptome are grand challenges in molecular biology and functional genomics. For the past 15 years, microarray has been the technology of choice for studying transcriptome [2,3].

Despite that much insight has been gained from microarray studies, factors such as the requirement of genomic sequence information when designing probes and substantial noise caused by cross-hybridization limited the application of microarray in more in-depth study of the transcriptome.

In RNA-Seq experiments, a population of RNA is converted to a library of cDNA fragments with adaptors attached to one end. Each molecule, after amplification, is then sequenced using one of the NGS technologies. Following sequencing, the resulting reads are aligned to either the reference genome or known transcripts to produce a genome-scale transcriptional profile. Please see Figure 1.1 for an illustration of the RNA-Seq experiment. Compared to microarray, RNA-Seq is able to provide more information about the transcriptome, and possesses a list of advantages discussed below.

High resolution. The resolution of microarray expression measure is unable to go beyond the probe level. In contrast, the majority of reads generated from NGS instruments maps to the reference genome with single base resolution. Therefore, expression measure can be evaluated at single-base resolution in RNA-Seq. This makes it possible to locate the transcript's start and end, discover new transcripts, and identify alternative splicing and translocation events with RNA-Seq.

High dynamic range. In microarray, gene expression levels are represented by florescent intensity values that are known to have a limited range of signal detection. In RNA-Seq, transcript expression levels are typically evaluated using a read census approach that is known to have a much wider dynamic range.

Knowledge of genome sequences. In RNA-Seq, there is no need to design probes, hence, the knowledge of the target genome sequences is no longer needed. RNA-Seq allows the discovery of novel transcripts that have not yet been annotated, and it can also be applied to study non-model organisms for which no reference genome is available.

Therefore, RNA-Seq is able to provide a more accurate overall picture of the transcriptome and leads to a variety of improvements in transcriptome study over

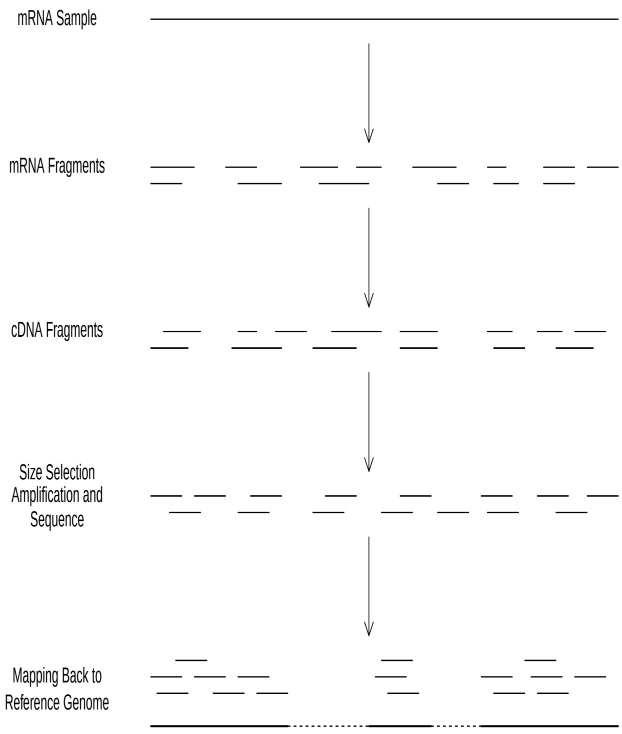

Figure 1.1. Flow of RNA-Seq Experiment

microarray. The improvements include better detection in differential expression in multiple samples, capability to discover novel transcripts, capability to discover and quantify alternative splicing events, capability to discover variants such as single nucleotide polymorphism (SNPs), insertions, deletions and translocation.

Despite the advantages and promises of RNA-Seq, RNA-Seq data are subject to a variety of sources of variation and bias. Similar to raw microarray data, RNA-Seq short reads data need to be de-noised and normalized before they can be used for downstream transcriptome analysis. An immediate challenge is how to use RNA-Seq short reads to quantify the transcriptome or gene expression levels. The accurate quantification of transcript or gene expression levels is the basis for almost all further analysis of the transcriptome. Mortazavi *et al.* proposed to use the number of reads per kilobase of a transcript per million mapped reads (RPKM) to measure the transcript's expression level [4]. The RPKM method is intuitive and takes the length of a transcript and the total number of short reads received by the transcript into consideration, but it ignores the excessive bias and variability demonstrated in RNA-Seq short reads. From the perspective of statistical modeling, the RPKM method essentially assumes that read counts in a transcript are uniformly distributed and follow a Poisson distribution of constant intensity rate, and further uses the estimate of the intensity rate as the transcript's expression measurement. However, it has been found that the assumptions of uniformity and constant intensity do not hold in real RNA-Seq data, and the simple Poisson distribution fits real data poorly. The RPKM measure may not provide the most accurate quantification of transcript expression level. Therefore, more sophisticated models and methods are needed to better characterize RNA-Seq data, separate signal and noise, and provide consistent and accurate quantification of transcript expression levels.

1.2 RNA-Seq Data Preprocessing

1.2.1 Quality Control

As discussed before, the short fragments are sequenced to produce millions of short reads. Each read contains a series of adenine (A), guanine (G), cytosine (C), or thymine (T), and a Phred quality score, encoded using ASCII codes, is assigned to each nucleotide base call. The quality score is a direct indication of the accuracy of the automatic base calling method, and can be used to filter out erroneous sequenced reads.

The quality score of each base pair is defined as $Q = -10\log_{10}(P)$, where P is the base-calling error probability. For example, a quality score of 30 results in a 99.9%

base-calling accuracy. In other words, average one base pair out of 1000 is incorrectly called. Usually, the Next Generation sequencing machine produces FASTA/FASTQ files, and in practice the quality of each read in the FASTA/FASTQ file should be checked before mapping the reads to the reference genome. To filter out reads that do not pass the minimum quality requirement or trim the base pairs with low quality scores, open source command line toolkit such as FASTA-Toolkit can be used.

1.2.2 Mapping

Mapping a large number of reads against the reference genome can be difficult. Efficient algorithms, such as BWA [5] or Tophat [6], uses either seed based method or Burrows-Wheeler transformation based method. Maq is one of the methods that uses spaced seed index. Although the gain is the mapping efficiency, it requires more than 50 gigabytes of memory. On the other hand, Burrows-Wheeler transformation based method is more popular nowadays because its efficiency and speed. For example, we can use bowtie to map non-junction reads. Bowtie [7] first indexes the reference genome and stores a memory-efficient representation of the reference annotation. The "indexing" allows one to rapidly find shorter sequences embedded within it.

Each read obtained the sequencing experiment is mapped base pair by base pair from one end to the other against the indexed reference representation. As a result, a large portion of the reads can be mapped directly and efficiently to the reference genome. However, one disadvantage of Bowtie and Maq is that they do not align reads that are generated from exon-exon junctions. As a result, some reads cannot be mapped to a continuous region on the reference genome because of alternative splicing. Taking the possibility of such biological process into consideration, Tophat is designed to map junction reads across exon-exon junctions [6].

1.2.3 Single-end and Paired-end Reads

The Illumina platform is able to generate both single-end reads and paired-end reads. Paired-end reads are more popularly used because they contain a lot of information about the spliced junctions. For example, a single-end read which contains 100 base pairs would only give relevant information about 100 base pairs. However, a read pair with 50 base pairs on each end could give more information about the fragments that are longer then 100 base pairs. The benefit of the paired-end reads is two-fold. First, paired-end reads are able to measure longer fragments without actually sequence all the base pairs in the fragments. The sequenced long fragments can be helpful in *de novo* assembly of the new species and reduce the number of ambiguous reads that map to different locations on the reference genome. Second, paired-end reads can potentially be used to measure the gene boundaries.

1.3 Bias and Variation in RNA-Seq Experiments

Recent studies have demonstrated that various types of variation and bias affect the outcome of RNA-Seq experiments. The understanding of the sources of variation and bias in RNA-Seq experiments is crucial for properly normalizing RNA-Seq data and quantifying transcript expression levels. Sources of variation and bias in RNA-Seq experiments can be classified into two major categories. The first category includes variations and biases that come from steps or protocols in a RNA-Seq experiment, which are referred to as experimental sources; and the second category includes variations and biases from the RNA sample that a RNA-Seq experiment uses, which are referred to as biological sources. In addition to these two categories, other sources may also exist.

1.3.1 Experimental Sources

A number of platforms of NGS technologies are currently available for conducting RNA-Seq experiments. Although these platforms share the same basic principles, technological details can differ. For example, the SOLiD/454 platforms amplify cDNA fragments using emulsion PCR, while the Illumina platform conducts amplification through a unique bridged amplification reaction, and the Helicos platform does not require an amplification step. Therefore, different platforms may have different sources of variation and bias. The discussion below focuses on the Illumina platform only.

Library preparation. The Illumina library preparation procedure first includes a fragmentation step in which the mRNA molecules are fragmented into small pieces. The mRNA pieces undergo a transcription step to generate a pool of cDNA fragments, which then go through size selection, DNA repair, end polishing and platform-specific adaptor ligation steps, resulting in a library of cDNA fragments that can then be sequenced by a sequencing machine. There are two potential factors that may produce variation and bias in the outcome. First, because 5' bias and mRNA secondary structures influence the primer binding sites, the synthesized cDNA fragments do not have a uniform coverage on the entire transcript. Second, DNA purification and end-polishing typically result in sample losses and limited throughput [8].

PCR amplification. Polymerase Chain Reaction (PCR) is known to produce uneven amplification of multiple templates in parallel. First, the products after the library preparation step have reduced complexity compared with the original mRNA pool, and this unevenness will be amplified. Second, not all transcripts amplify with the same efficiency; and therefore, some transcripts are excessively amplified while others remain essentially unchanged. Third, there will be PCR duplicates, which further confuse the resulting product [9]. In addition, PCR may suffer from the sample contamination and troubleshooting problems.

Base calling. The Illumina Genome Analyzer is based on parallel, fluorescence-based readout of millions of immobilized sequences [10]. Different base calling methods were compared in the literature, and it was found that they can lead to different numbers of mapped reads [11]. These reads are unevenly distributed across the genome, indicating that systematic bias can be caused by base calling methods.

Short reads mapping. After sequencing is completed, an important task is to map the large amount of short reads to the reference genome. This step can introduce variation to the outcome. First, different mapping software packages usually lead to different results. Second, polymorphism, reference sequence errors, and sequencing errors require that mismatches and indels be allowed in the mapping step, which will lower our confidence in the mapping result. Furthermore, the number of reads produced is quite large and requires a mapping program to have high efficiency. As a consequence, the accuracy of the mapping program may be compromised.

1.3.2 Biological Sources

The RNA sample used in a RNA-Seq experiment can also introduce variation and bias into the results. The variation and bias from the RNA sample can be classified into two types. The first type is the bias caused by the local sequence composition of a transcript. For example, GC-content [12] and local secondary structures [13] are two identified sources for causing bias and variation in RNA-Seq short reads data. Furthermore, genomic regions differ in terms of their sequence complexity. Regions with more dense repetitive elements are less likely to receive reads. On the other hand, regions with higher complexity tend to have higher reads counts. The second type includes variations that come from the dynamic nature of biological processes as well as the diversity of biological conditions. The transcriptome is dynamic and transient, thus much more complicated than the mostly static genome. Due to the biological variations, transcripts composition of the same cell line from different individuals can vary, and transcripts of the same cell line but at different stages of cell development can also vary.

1.3.3 Other Sources

Some other factors could further confuse the results from RNA-Seq experiments. The first is the sequencing depth of an experiment. The sequencing depth refers to the total number of all sequenced reads in an RNA-Seq experiment. On one hand, it is directly related to the cost of an RNA-Seq experiment. The deeper the sequencing depth, the higher the cost. On the other hand, it is related to the coverage of an RNA-Seq experiment. Deeper sequencing depth would result in higher coverage. McIntyre found that approximately 64% exons can be detected for an experiment with 5-7 million reads per lane, and the percentage increases to 84% for an experiment with approximately 27 million reads [14]. Data from a lower coverage experiment suffer from higher background noise. The second is the incompleteness and errors in the reference genome and transcript annotations. For instance, although a large number of alternative splicing events have been discovered and cataloged, the collection is still incomplete [15].

1.4 Features of RNA-Seq Read Counts Data

After the short reads generated from a RNA-Seq experiment are mapped back to the genome, they need to be further summarized for subsequent analysis. In this chapter, we focus on the summarization at the base pair level. The count of reads at a base pair is defined to be the number of reads that start at the base pair. The resulting data is referred to as the base-level reads count data. Let i be the index for the ith transcript and j be the index for the jth base pair position in this transcript. Suppose the length of transcript i is n_i. For $1 \leq j \leq n_i$, Y_{ij} denotes the reads count at base pair j of transcript i. In an ideal RNA-Seq experiment free of the sources of variation and bias discussed in Section 1.3, $Y_{i1}, \ldots, Y_{in_i}$ are independent and identically distributed as Poisson or other discrete distributions. This is however hardly the case in real life RNA-Seq base-level read count data as demonstrated below.

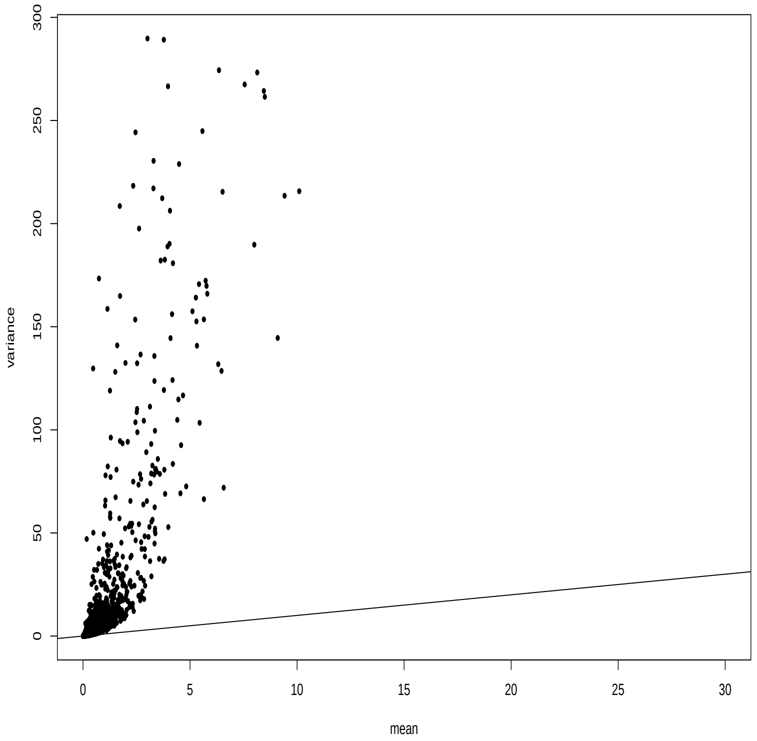

Figure 1.2. Plot of reads counts variance versus mean. (The x-axis is for the mean of reads counts of a transcript and the y-axis is for the variance of reads counts of the transcript. The solid line is for y=x.)

Over-dispersion. If the reads counts in a transcript are Poisson random variables, we would expect their mean and variance to be the same. However, it is not the case, and over-dispersion has been observed in real reads count data [16]. Figure 1.2 shows the plot of variance against mean of read counts in 500 transcripts from a real RNA-Seq data we analyze in Section 2.4.1. (Transcripts with variance larger than 300 are not included in the figure.) Each dot corresponds to a transcript, and the dotted line represents the ideal case, where the mean and variance are equal to each other. Clearly, the variances are much larger than the means, therefore models more sophisticated than the simple Poisson distribution are needed to characterize such highly dispersed count data.

Non-uniformity. Reads from the same transcript are not uniformly distributed within a transcript [16, 17]. It has been observed that certain regions in a transcript

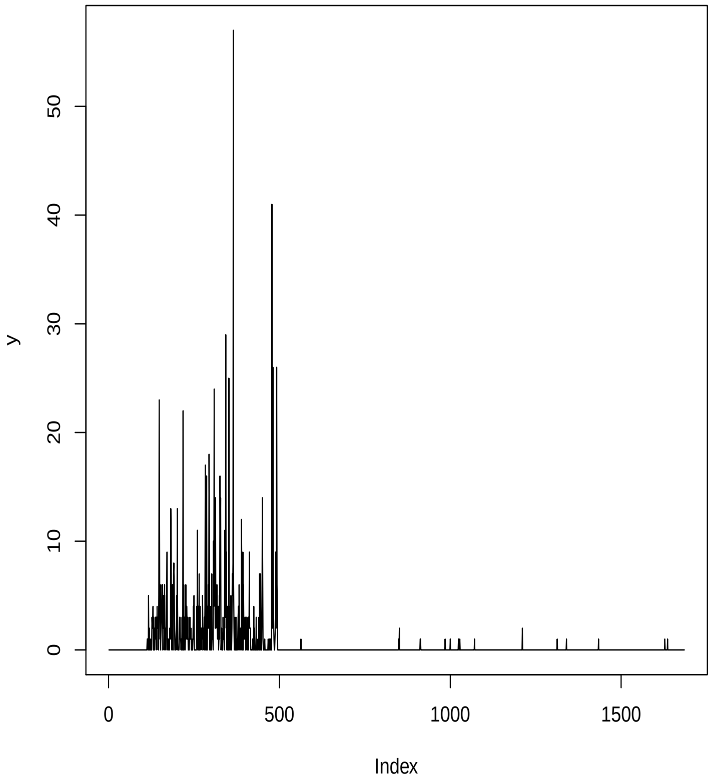

Figure 1.3. Plot of non-uniformity of reads count data of a transcript. (The x-axis is for the base pair positions of this transcript. The y-axis is for the counts at the base pairs.)

do not receive any reads or just receive a few reads, whereas other regions receive much more reads than the average. Therefore, there may exist a local sequence composition bias. Figure 1.3 demonstrates such a transcript with some base pairs heavily covered while others barely or uncovered by reads, demonstrating non-uniformity in reads count coverage.

GC-content bias. GC-content (or Guanine-Cytosine content) refers to the percentage of nitrogenous bases on a DNA molecule that are either guanine or cytosine. The count of reads a genomic region receives is found to be dependent on the GC content of the region. The association between GC-content and the reads counts of

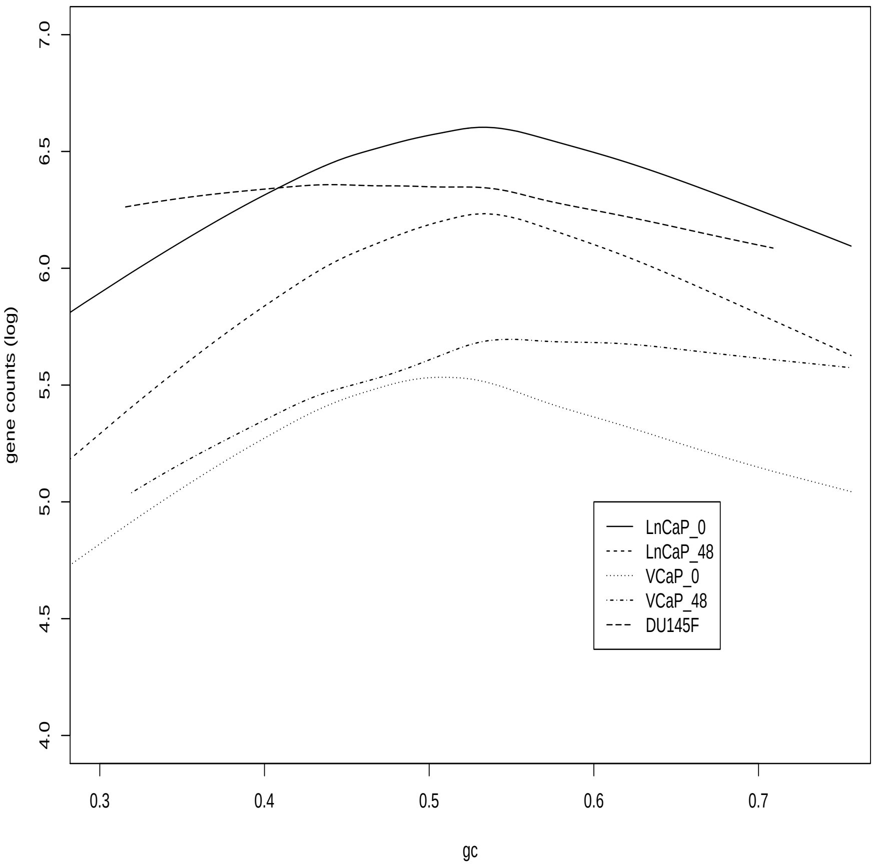

Figure 1.4. Plot of total reads count versus GC-content in 5 cell samples. (The x-axis is for the GC-content of a transcript in percentage. The y-axis is for the total reads count of the transcript in log scale. The lines are obtained by fitting lowess lines to the scatter plot of y versus x in 5 cell samples.)

a genomic region is complicated and nonlinear [12], as demonstrated in Figure 1.4. The plot shows a genomic region will, on average, have low reads count when the GC-content of this region is either too low or too high.

Transcript length bias. In addition to overdispersion, non-uniformity, and GC-content bias, other types of bias patterns are also reported in the literature, including transcript length bias [18] [17], sequence preference [16], and sequence positional

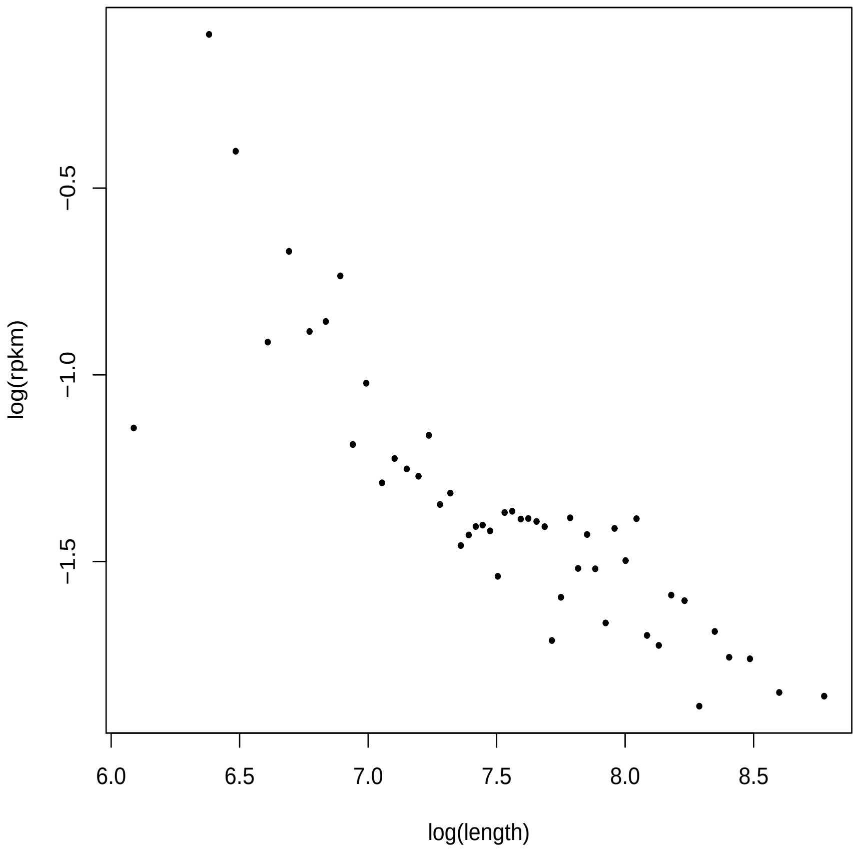

Figure 1.5. Plot of log median RPKM measure versus log median transcript length. (4964 transcripts are divided into 49 groups according to their transcript lengths. Each group contains 100 transcripts. The median RPKM measure of each group in log scale is plotted against the median transcript length in log scale.)

bias [19]. Figure 1.5 shows the relationship between the RPKM measures and the lengths of transcripts, which might be due to length bias.

The variations and biases in RNA-Seq reads count data call for proper normalization methods. Some existing model-based methods are briefly reviewed in the next section.

1.5 Normalization Method

The normalization of RNA-Seq data is a critical step in the analysis to ensure accurate inference of gene expression levels and other downstream analyses. The simplest method is the reads per kilobase per million mapped reads (RPKM), which normalizes the read counts each gene received by the gene length and library size [4]. The RPKM normalization method worked well in some cases, but is over simplified in the other cases. In the literature, there have been a relatively large number of models and methods proposed for gene and transcript expression level quantification based on RNA-Seq data. These methods primarily follow two ideas. The first is to use more sophisticated distributions to account for bias and variation in reads count data, and the second is to directly correct RNA-Seq read count data by identified sources of bias and variation. Among the methods discussed in this section, the first three methods GPseq [20], POME [21] and PMI [22] follow the first idea, and the remaining methods such as mseq, bias-adjusted Cufflinks, and the GC-content correction method mainly follow the second idea. In Chapter 2, we will review some of these methods and propose a new to quantify the gene expression levels.

2. REGION OF INTEREST EXPRESSION LEVEL QUANTIFICATION

A fundamental question in RNA-Seq is how to quantify expression levels of transcripts, or in general, regions of interest (ROI) on the genome. A number of methods have been proposed for transcript expression level quantification in the literature. In this chapter, we will review some of the existing quantification methods, and we will propose a new framework to quantify gene expression levels based on Poisson mixture models.

2.1 Existing Normalization Methods

Among all the proposed methods to quantify gene expression levels, the RPKM measure proposed by Mortazavi *et al.* [4] is the first one and is the easiest to compute. The RPKM is defined to be the number of reads per kilobase of exon model per million mapped reads. The RPKM measure is easy to compute, however, it neglects the excessive variability demonstrated in RNA-Seq data. For example, the number of reads covering a base pair (i.e., base-level count) in a ROI can vary dramatically, ranging from zero to thousands. The substantial variability suggests that uniform coverage in sequencing depth is unrealistic. Researchers have identified various types of non-uniformity and their possible causes. Oshlack and Wakefield [18] mentioned that transcript length can be a source of bias and needs to be corrected, and Hansen and Dudoit [23] pointed out that when random hexamer priming is used, the resulting sequence coverage is not uniform. The PCR amplification step in RNA-Seq can also introduce some bias [24]. These variations or biases in RNA-Seq reads count data are not accounted for in quantification methods using summary statistics such as RPKM.

To accommodate these variations and biases, statistical model-based approaches are preferred.

It is widely accepted that the standard Poisson distribution with constant intensity λ is not an appropriate model for RNA-Seq reads count data because the property of the distribution that the mean and variance are equal is often invalid. Marioni *et al.* [25] suggested to use the quasi-Poisson distribution or the negative binomial distribution. Bullard *et al.* [11] considered the quantification problem in the generalized linear model framework, and took into account of different biological conditions and technical effects. Srivastava and Chen [20] proposed a Generalized Poisson (GP) model to account for over- and under-dispersion in reads count data and subsequently developed a quantification method called GPseq for transcript expression level measurement. Li *et al.* [16] proposed a Poisson model for base level count, which incorporates sequence patterns in the neighborhood of a base pair into the intensity, and the quantification method based on the model is called mseq. Recently, Hu *et al.* [21] proposed a Poisson mixed effects model for base level reads count data, which uses two types of random effects to account for variation specific to each base pair and possible correlation between adjacent base pairs, and further developed the quantification method POME based on the proposed model.

Despite reported improvements in accuracy over the RPKM measure, the aforementioned model-based methods all try to use models or distributions with known shapes to fit the reads count data of all transcripts. They may be either too simple such as the GP model that does not have enough parameters to fit real data well, or too complex such as mseq that may suffer from the problem of overfitting. These existing methods fail to properly characterize the complexity and variability of RNA-Seq data, which limits their potentials for transcript expression level quantification. Therefore, a more flexible modeling strategy is needed to portrait observed RNA-Seq data. In Statistics, a powerful and commonly-used strategy to model distributions with unknown shapes is to use mixture models.

A mixture model consists of a parametric component distribution and a nonparametric mixing distribution, therefore, it combines the strength of parametric models with the flexibility of nonparametric models. When the mixing distribution is assumed to have a finite support, the mixture model is referred to as a finite mixture model. It is known that finite mixture of normal distributions can approximate any continuous distributions, and finite mixture of Poisson distributions can be used to approximate discrete distributions for count data [26]. Furthermore, in RNA-Seq data, we observed that in many transcripts, base pairs are clustered or grouped in terms of the reads count intensities they are subject to, and these groups may reflect different reads generating mechanisms in RNA-Seq experiments. Finite mixture models are suitable for modeling data with intrinsic grouping structures.

In this chapter, we propose to use finite Poisson mixture models to analyze RNA-Seq base-level reads count data and quantify transcript expression levels. The Poisson mixture models provide a flexible framework for modeling and analyzing RNA-Seq data. We applied EM algorithms to fit these models; we also used a BIC-based model selection procedure to adapt the models to individual transcripts. Additionally, we proposed a unified method for quantifying transcript expression levels based on the three models. When applying the proposed methods to analyze two RNA-Seq data sets, we found our methods demonstrated excellent performance in model fitting and expression quantification in comparison with other existing methods. We believe that the Poisson mixture models and the proposed methods have great potential for RNA-Seq data analysis.

2.2 Poisson Mixture Model

2.2.1 PMI Model

Suppose a ROI consists of n base pairs, indexed by $1, 2, \ldots, n$, from left to right. Let $y = (y_1, \ldots, y_n)^T$ be the vector of observed reads counts at base pairs $1, 2, \ldots, n$.

Assume y_i $(1 \leq i \leq n)$ are independent and follow the same Poisson mixture distribution

$$f\left(y_i|\lambda, \pi\right) = \sum_{k=1}^{K} \pi_k \mathrm{Poi}\left(y_i; \lambda_k\right), \tag{2.1}$$

where K is the number of components, $\pi = (\pi_1, \pi_2, \ldots, \pi_K)^T$ is the vector of mixing proportions satisfying $\sum_{k=1}^{K} \pi_k = 1$, $\lambda = (\lambda_1, \ldots, \lambda_K)^T$ is the vector of intensities of the K Poisson components, and $\mathrm{Poi}(y_i; \lambda_k) = \lambda_k^{y_i} \exp(-\lambda_k)/y_i!$ is the mass function of the kth component. We refer to (2.1) as the Poisson mixture model with assumption of independence between the base pairs, or in short the PMI model. The PMI model provides a flexible way to model count data from distributions with arbitrary shapes, and the mixture structure may also represent some natural grouping or clustering structure in the data. Once the parameters λ and π are estimated for a transcript, they can be used to quantify the expression level of the transcript.

A critical assumption in the PMI model is that reads counts of different base pairs are independent. We have calculated lag one autocorrelations between base level reads counts within transcripts in a number of RNA-Seq data sets, and found that they are not negligible and can become substantial in a large number of transcripts, indicating dependence between reads counts of adjacent base pairs. When the dependence is strong, more general Poisson mixture models capable of accommodating such dependence need to be considered. Thus we provide two extensions of PMI model to account for such correlations in Chapter 2.3.

2.2.2 EM Algorithms for PMI Model

We apply the EM algorithm to calculate the maximum likelihood estimates (MLEs) of the parameters in the PMI model and the maximum partial likelihood estimates (MPLEs) [27] of the parameters in the PMAI and PMAIP models. The algorithms are described separately for the models below.

Under the PMI model, the MLEs of π and λ are defined to be

$$(\hat{\pi}, \hat{\lambda}) = \arg\max_{\pi,\lambda} \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \text{Poi}\left(y_i; \lambda_k\right). \tag{2.2}$$

In order to develop the EM algorithm, a two-step data generating scheme needs to be introduced. For each observed count y_i, we define a membership vector $z_i = (z_{i1}, \ldots, z_{iK})^T$ indicating from which Poisson component y_i is sampled. Suppose y_i is sampled from the kth component, then $z_{ik} = 1$ and $z_{ij} = 0$ for $j \neq k$. Let $z = (z_1, z_2, \ldots, z_k)^T$, which is referred to as the membership matrix of y. With both y and z, the complete likelihood function for π and λ is

$$L_c = \prod_{i=1}^{n} \prod_{k=1}^{K} (\pi_k \text{Poi}(y_i; \lambda_k))^{z_{ik}}. \tag{2.3}$$

Suppose the current parameter estimates are $\hat{\pi}^{cur}$ and $\hat{\lambda}^{cur}$. The E-step is to calculate the expected log complete likelihood function

$$Q(\pi, \lambda | \hat{\pi}^{cur}, \hat{\lambda}^{cur}) = E_z(\log(L_c(\pi, \lambda)) | \hat{\pi}^{cur}, \hat{\lambda}^{cur}, y), \tag{2.4}$$

where the expectation is over the conditional distribution of z, given $\hat{\pi}^{cur}$, $\hat{\lambda}^{cur}$ and y. Essentially, it is to compute

$$\bar{z}_{ik} = E\left(z_{ik} | y_i, \hat{\pi}_k^{cur}, \hat{\lambda}_k^{cur}\right) = \frac{\hat{\pi}_k^{cur} \text{Poi}(y_i; \hat{\lambda}_k^{cur})}{\sum_{k=1}^{K} \hat{\pi}_k^{cur} \text{Poi}(y_i; \hat{\lambda}_k^{cur})}.$$

The M-step is to maximize Q with respect to π and λ, and the resulting maximizers are the updates of $\hat{\pi}^{cur}$ and $\hat{\lambda}^{cur}$, which are, respectively,

$$\begin{cases} \hat{\lambda}_k^{new} = \sum_{i=1}^{n} \bar{z}_{ik} y_i / \sum_{i=1}^{n} \bar{z}_{ik} \text{ for } k = 1, 2, \ldots, K \\ \hat{\pi}_k^{new} = \sum_{i=1}^{n} \bar{z}_{ik} / n \text{ for } k = 1, 2, \ldots, K-1 \\ \hat{\pi}_K^{new} = 1 - \sum_{k=1}^{K-1} \hat{\pi}_k^{new}. \end{cases} \tag{2.5}$$

2.2.3 Quantification Rule for PMI Model

Suppose the PMI model with K components is chosen by the BIC criterion as the best model for a transcript, and the MLEs of the model parameters are calculated to be $\hat{\pi}$ and $\hat{\lambda}$. We need to design a rule to quantify the expression level of the transcript. Because for the majority of transcripts, the number of components needed to fit the reads count data well is below 4, we only consider the cases with $K = 1$, $K = 2$, or $K = 3$. When $K = 1$, the PMI model reduces to the simple Poisson model, and the estimated intensity $\hat{\lambda}_1$ can be used as the expression measure of the transcript. When $K = 2$, the estimated mixing propositions are $\hat{\pi}_1$ and $\hat{\pi}_2$, and correspondingly, the estimated intensities are $\hat{\lambda}_1$ and $\hat{\lambda}_2$ with $\hat{\lambda}_1 < \hat{\lambda}_2$, we propose to use the following weighted average of the intensities as the measure of the transcript expression level,

$$g_s = \frac{(s\hat{\pi}_1\hat{\lambda}_1 + \hat{\pi}_2\hat{\lambda}_2)}{(s\hat{\pi}_1 + \hat{\pi}_2)}$$

where s is a pre-specified number between 0 and 1. When $K = 3$, the estimated mixing proportions are $\hat{\pi}_1$, $\hat{\pi}_2$ and $\hat{\pi}_3$, and the estimated intensities are $\hat{\lambda}_1$, $\hat{\lambda}_2$, and $\hat{\lambda}_3$ with that $\hat{\lambda}_1 < \hat{\lambda}_2 < \hat{\lambda}_3$, we propose to quantify the expression level of the transcript as

$$g_s = \frac{(s\hat{\pi}_1\lambda_1 + \hat{\pi}_2\lambda_2 + \hat{\pi}_3\lambda_3)}{(s\hat{\pi}_1 + \hat{\pi}_2 + \hat{\pi}_3)}. \tag{2.6}$$

where s again is a pre-specified number between 0 and 1.

Notice that in the proposed measures above, the proportion for the lowest intensity is down weighted by a factor of s. The major justification for down weighting the smallest intensity is that base level reads count data are dominated by zero counts and the Poisson component with the lowest intensity in the PMI model primarily is used to account for these excessive zero counts. For example, for the first cell line LnCaP0 in the prostate cancer data we analyze in the next section, roughly more than 75% of all its transcripts have less than 20% of their base pairs receiving at least one read. Zero counts in general do not contain much information about the expression level of a transcript, therefore, their weight in the measure should be reduced. The benefit from down-weighting the lowest intensity is to increase the weight of the higher intensities, especially the medium intensity, because the proportion for the highest intensity is often small or close to zero when $K = 3$. In the literature, some researchers propose to discard all the zero counts. This however may lead to some information loss about the data-generating mechanism in RNA-Seq experiments. The quantification method we proposed above can be considered as a model-based approach to coping with zero counts in RNA-Seq data. Furthermore, the proposed quantification measure is believed to be robust with respect to extremely large counts present in RNA-Seq data, because the highest intensity instead of the largest counts is used in the measure.

We propose to use the Bootstrap method to determine the value of the tuning parameter s in practice. The Bootstrap method was originally proposed by Efron [28] as a general procedure to assess the accuracy of a sample estimate. We use Bootstrap to compute the mean squared error (MSE) of g_s defined above. For each $s \in (0, 1)$, we draw B Bootstrap samples from the original transcript-level reads count data. For the bth sample ($1 \leq b \leq B$), the PMI model is fitted, and the transcript expression level is quantified using the proposed method as g_s^b. Then the variance of g_s is estimated as $\hat{\text{var}}(g_s) = \sum_{b=1}^{B}(g_s^b - \bar{g}_s^B)/B$, where $\bar{g}_s^B = \sum_{b=1}^{B} g_s^b/B$ is the Bootstrap mean. The bias of g_s is estimated as the difference between the Bootstrap mean and the original sample estimate, that is, $\hat{\text{bias}}(g_s) = \bar{g}_s^B - g_s$. Then the MSE of g_s is estimated by the sum of the estimated variance and the squared estimated bias, that is,

$$\hat{\text{MSE}}(g_s) = \hat{\text{var}}(g_s) + \hat{\text{bias}}^2(g_s). \tag{2.7}$$

After the estimated MSE of each g_s is obtained, we plot it against the tuning parameter s and refer to the resulting curve as the quantification performance (QP) curve of g_s. As s increases from 0 to 1, the QP curve is expected to demonstrate an overall decreasing pattern. When s is close to zero, g_s is dominated by the components with higher intensities and is sensitive to the change in nonzero reads counts; therefore, the estimated variance and MSE of g_s are large, indicating high variability or poor quantification performance of g_s. When s is close to 1, g_s is dominated by the component with the lowest intensity or the zero counts and becomes insensitive to the change in nonzero reads counts; therefore, even though the estimated MSE of g_s is small, the quantification performance of g_s is still poor due to its insensitivity. The best quantification performance of g_s should be achieved at an s value that balances

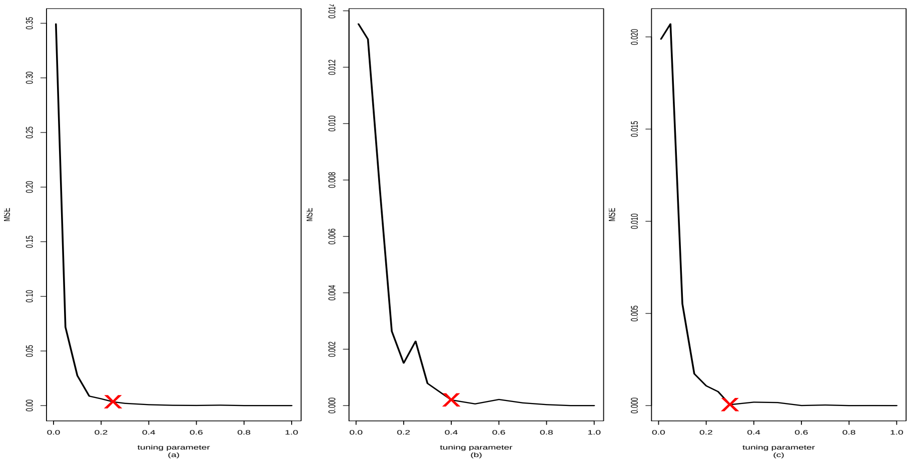

Figure 2.1. An example of the QP curve of g_s and the selection of s^*. Plots (a)–(c) show the QP curves of g_s in three transcripts. The s^* values for the three transcripts are 0.2, 0.4, and 0.3, respectively, indicated by red circles in the plots.

the variability and sensitivity of g_s. We propose to choose the elbow point of the QP curve of g_s as such a value and denote it by s^*. Figure 2.2.3 demonstrates the QP curves of g_s in three transcripts in the prostate cancer RNA-Seq data we analyze and discuss in Section 3.1. The elbow points of the three curves were determined to be 0.2, 0.4, and 0.3 in Plots (a)–(c) of Figure 2.2.3, respectively.

In practice, however, it is computationally intensive to determine s^* for each individual transcript. Instead, we propose to adopt a sampling scheme to select a common tuning parameter s to be used by all the transcripts as follows. First, a subset of 100 to 200 transcripts is randomly sampled; second, the Bootstrap method is applied to each selected transcript to determine the value of s^*; and third, the average value of s^* over the sample is calculated and denoted by $\bar{s}^*$. We recommend to use $\bar{s}^*$ when quantify all the transcripts.

2.3 Two Extensions of Poisson Mixture Models

2.3.1 PMAI Model

The Poisson mixture model with autoregressive intensities or the PMAI model was originally proposed for time series count data [29]. For $i = 2, \ldots, n$, conditional on y_{i-1}, y_i is assumed to follow the following Poisson mixture distribution,

$$g\left(y_i|y_{i-1}; \pi, \alpha, \beta\right) = \sum_{k=1}^{K} \pi_k \text{Poi}\left(y_i; \exp\left(\alpha_k + \beta_k \log\left(y_{i-1}+1\right)\right)\right), \tag{2.8}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)^T$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_K)^T$. Note that for $k = 1, \ldots, K$, the intensity $\lambda_{k,i}$ of the kth Poisson component at base pair i depends on the transformed reads count at base pair $i-1$ through a linear function with intercept α_k and slope β_k, that is,

$$\log \lambda_{k,i} = \alpha_k + \beta_k \log\left(y_{i-1}+1\right). \tag{2.9}$$

Through this autoregressive relation, the PMAI model imposes a global correlation structure for y. The use of $\log(y_{i-1}+1)$ in (2.9) ensures that y as a series is stationary and the correlation between y_i and y_j depends on their distance $|i-j|$ only [29]. The stationarity is important because it provides the flexibility to index the base pairs of an transcript from left to right or the other way around. In the PMAI model, the mixing proportions π_k's are assumed to be constant throughout the transcript. Next, we further consider an extension of the PMAI model that allows varying mixing proportions.

2.3.2 PMAIP Model

For $i = 2, \ldots, n$, conditional on y_{i-1}, the reads count y_i at base pair i is assumed to follow the following Poisson mixture distribution

$$h\left(y_i|y_{i-1}; u, v, \alpha, \beta\right) = \sum_{k=1}^{K} \pi_k(y_{i-1}; u_k, v_k) \text{Poi}\left(y_i; \exp\left(\alpha_k + \beta_k \log(y_{i-1}+1)\right)\right) \tag{2.10}$$

where

$$\pi_k(y_{i-1}; u_k, v_k) = \frac{\exp(u_k + v_k \log(y_{i-1} + 1))}{\sum_{k=1}^{K} \exp(u_k + v_k \log(y_{i-1} + 1))}, \tag{2.11}$$

for $k = 1, 2, \ldots, K$, where $u = (u_1, u_2, \ldots, u_K)^T$, and $v = (v_1, v_2, \ldots, v_K)^T$. Note that an autoregressive relation has been postulated between the mixing proportions at base pair i and the observed reads count y_{i-1} at base pair $i-1$. To be more specific, we treat the Kth component as the baseline component and set u_K and v_K to be zero, and then (2.11) is equivalent to assuming an autoregressive model for the baseline logit $\log \pi_k/\pi_K$,

$$\log \frac{\pi_k}{\pi_K} = u_k + v_k \log(y_{i-1} + 1) \tag{2.12}$$

for $k = 1, 2, \ldots, K-1$. As discussed in the previous subsection, the autoregressive structure imposed on the mixing proportions can further help characterize the clustering patterns of reads count demonstrated in some transcripts. Because it assumes autoregressive relations in both intensities and mixing proportions, we refer to the extended PMAI model as the Poisson mixture model with autoregressive intensities and mixing proportions, or in short the PMAIP model.

2.3.3 EM Algorithms for PMAI Model

We let $\phi = (\phi_1, \ldots, \phi_K)^T$, where $\phi_k = (\pi_k, \alpha_k, \beta_k)^T$, and $x_i = \log(y_{i-1} + 1)$. Under the PMAI model, the MPLEs of the model parameters are defined as

$$\hat{\phi} = \arg\max_{\phi} \prod_{i=2}^{n} \sum_{k=1}^{K} \pi_k \text{Poi}\left(y_i; \exp(\alpha_k + \beta_k x_i)\right). \tag{2.13}$$

With the membership matrix z, the complete partial likelihood is

$$L_c = \prod_{i=2}^{n} \prod_{k=1}^{K} \left[\pi_k \text{Poi}(y_i; \exp(\alpha_k + \beta_k x_i))\right]^{z_{ik}}. \tag{2.14}$$

Suppose the current parameter estimates are $\hat{\phi}^{cur} = (\hat{\pi}^{cur},\ \hat{\alpha}^{cur}, \hat{\beta}^{cur})^T$. The E-step is to calculate $Q(\phi|\hat{\phi}^{cur}) = E_z(\log(L_c(\phi))|\hat{\phi}^{cur}, y)$, which is essentially to calculate

$$\bar{z}_{ik} = E(z_{ik}|y_i, \hat{\psi}_k^{cur}) = \frac{\hat{\pi}_k^{cur} \text{Poi}(y_i; \exp(\hat{\alpha}_k^{cur} + \hat{\beta}_k^{cur} x_i))}{\sum_{k=1}^{K} \hat{\pi}_k^{cur} \text{Poi}(y_i; \exp(\hat{\alpha}_k^{cur} + \hat{\beta}_k^{cur} x_i))}.$$

The M-step is to update the parameter estimates by maximizing Q with respect to ϕ. The updated mixing proportions have explicit expressions, which are $\hat{\pi}_k^{new} = \sum_{i=2}^{n} \bar{z}_{ik}/n$ for $k = 1, 2, \ldots, K-1$, and $\hat{\pi}_K^{new} = 1 - \sum_{k=1}^{K-1} \hat{\pi}_k^{new}$. For each k $(1 \leq k \leq K)$, to obtain $\hat{\alpha_k}^{new}$ and $\hat{\beta_k}^{new}$, we need to solve the following system of nonlinear equations,

$$\begin{cases} \frac{\partial Q}{\partial \alpha_k} = \sum_{i=2}^{n} \bar{z}_{ik}(-\exp(\alpha_k + \beta_k x_i) + y_i) = 0, \\ \frac{\partial Q}{\partial \beta_k} = \sum_{i=2}^{n} x_i \bar{z}_{ik}(-\exp(\alpha_k + \beta_k x_i) + y_i) = 0. \end{cases}$$

Note that the updating of $\hat{\alpha}_k$ and $\hat{\beta}_k$ can be done separately for each k. Software packages such as the R package nleqslv can be used to solve these equations.

2.3.4 EM Algorithms for PMAIP Model

Let $\psi = (\psi_1, \psi_2, \ldots, \psi_K)^T$, where $\psi_k = (u_k, v_k, \alpha_k, \beta_k)^T$ for $k = 1, 2, \ldots, K$. Again let $x_i = \log(y_{i-1}+1)$. Under the PMAIP model, the MPLEs of the model parameters are defined as

$$\hat{\psi} = \arg\max_{\psi} \prod_{i=2}^{n} \sum_{k=1}^{K} \pi_k(x_i, u_k, v_k)\text{Poi}(y_i; \exp(\alpha_k + \beta_k x_i)). \tag{2.15}$$

With the membership matrix z, the complete partial likelihood function is

$$L_c = \prod_{i=2}^{n} \prod_{k=1}^{K} \left[\pi_k(x_i, u_k, v_k)\text{Poi}(y_i; \exp(\alpha_k + \beta_k x_i))\right]^{z_{ik}}. \tag{2.16}$$

Suppose the current parameter estimates are $\hat{\psi}^{cur} = (\hat{u}^{cur}, \hat{v}^{cur}, \hat{\alpha}^{cur}, \hat{\beta}^{cur})^T$. The E-step is to calculate $Q(\psi|\hat{\psi}^{cur}) = E_z(\log(L_C(\psi))|\hat{\psi}^{cur}, y)$, which is essentially to calculate

$$\bar{z}_{ik} = E(z_{ik}|y_i, \hat{\psi}_k^{cur}) = \frac{\hat{\pi}_k(x_i, \hat{u}_k^{cur}, \hat{v}_k^{cur})\text{Poi}(y_i; \exp(\hat{\alpha}_k^{cur} + \hat{\beta}_k^{cur} x_i))}{\sum_{k=1}^{K} \hat{\pi}_k(x_i, \hat{u}_k^{cur}, \hat{v}_k^{cur})\text{Poi}(y_i; \exp(\hat{\alpha}_k^{cur} + \hat{\beta}_k^{cur} x_i))}.$$

The M-step is to update $\hat{u}^{cur}$, $\hat{v}^{cur}$, $\hat{\alpha}^{cur}$, and $\hat{\beta}^{cur}$ by maximizing Q. To obtain $\hat{\alpha}_k^{new}$ and $\hat{\beta}_k^{new}$ for $1 \leq k \leq K$, we need to solve the following system of two nonlinear equations,

$$\begin{cases} \frac{\partial Q}{\partial \alpha_k} = \sum_{i=2}^{n} \bar{z}_{ik} \left[- \exp(\alpha_k + \beta_k x_i) + y_i \right] = 0, \\ \frac{\partial Q}{\partial \beta_k} = \sum_{i=2}^{n} x_i \bar{z}_{ik} \left[- \exp(\alpha_k + \beta_k x_i) + y_i \right] = 0. \end{cases}$$

To obtain $\hat{u}_1^{new}, \ldots, \hat{u}_{K-1}^{new}$ and $\hat{v}_1^{new}, \ldots, \hat{v}_{K-1}^{new}$, we need to solve the following system of $2(K-1)$ nonlinear equations simultaneously,

$$\begin{cases} \frac{\partial Q}{\partial u_k} = \sum_{i=2}^{n} \left[- \sum_{m=1}^{K} \bar{z}_{im} \frac{\exp(u_k + v_k x_i)}{\sum_{l=1}^{K} \exp(u_l + v_l x_i)} + \bar{z}_{ik} \right] = 0 \\ \frac{\partial Q}{\partial v_k} = \sum_{i=2}^{n} \left\{ x_i \left[- \sum_{m=1}^{K} \bar{z}_{im} \frac{\exp(u_k + v_k x_i)}{\sum_{l=1}^{K} \exp(u_l + v_l x_i)} + \bar{z}_{ik} \right] \right\} = 0. \end{cases}$$

Software packages such as the R package nleqslv can be used to solve the systems of nonlinear equations above.

2.3.5 Quantification Rules for PMAI Model

The quantification rule based on the PMI model can be extended to the PMAI model with some modification. For the ith base pair in a transcript, the conditional component intensities $\exp(\hat{\alpha}_k + \hat{\beta}_k x_i)$ $(1 \leq k \leq K)$ are functions of the true signals $\exp(\hat{\alpha}_k)$ and the noise from the previous count $\exp(\hat{\beta}_k x_i)$. We propose to only use the true signals for the purpose of quantifying transcript expression level. Let $\hat{\lambda}_k = \exp(\hat{\alpha}_k)$ for $1 \leq k \leq K$. Then, the formula for calculating g_s defined above can be directly applied. Furthermore, the Bootstrap procedure for determining the value of s under the PMI model needs to be replaced by a Bootstrap procedure that can draw samples from dependent data, because reads counts are assumed to be dependent under the PMAI model. One of the possible choices is the block Bootstrap method [30].

2.3.6 Quantification Rules for PMAIP Model

Under the PMAIP model, the intensity of the kth component can be modified and calculated in the same way as under the PMAI model, i.e. $\hat{\lambda}_k = \exp(\hat{\alpha}_k)$. However, under the PMAIP model, the mixing proportions vary from base pair to base pair. At the ith base pair of a transcript, the mixing proportions are $\hat{\pi}_k(x_i, \hat{u}_k, \hat{v}_k)$ for $1 \leq k \leq K$. We propose to apply the rule proposed under the PMI model to each base pair to generate a base pair-level expression measure, and then the expression measure of the transcript is defined to be the average of the expression measures of all its base pairs. Similar as under the PMAI model, the block Bootstrap method can be used to draw samples from the dependent base level reads count data to determine the s value for g_s.

2.4 Comparison Studies

2.4.1 Data Description

Prostate Cancer Data. We applied the Poisson mixture models to analyze a RNA-Seq data set of 12 prostate cancer related cell and tissue samples generated by Chinnaiyan's lab [31]. The 12 samples are LnCaP0, LnCaP24, LnCaP48, VCaP0, VCaP24, VCaP48, DU145F, aT34, DU145F2, aT34N, VCaP, and RWPE. Each sample was sequenced using two RNA-Seq platforms, an Illumina platform and a Helicos platform. The Illumina platform uses an amplification-based sequencing technology, whereas the Helicos platform uses a single-molecule sequencing technology. The Helicos platform does not use any PCR-based amplification, and it directly measures transcripts instead of their amplified copies. As a result, the data generated by the Helicos platform or the Helicos data are not subject to biases caused by PCR. Therefore, transcript expression measurements based on the Helicos data are believed to be more accurate than those based on the data generated by the Illumina platform or the Illumina data. When comparing our methods with other existing quantification methods, we treated transcript expression measurements based on the Helicos data as the gold standard.

Human Brain and UHR Data. We analyzed another RNA-Seq data set from the MicroArray Quality Control (MAQC) project [32], which were generated for two RNA samples, namely, the Universal Human Reference (UHR) RNA sample from Stratagene and the Human Brain Reference (Brain) RNA sample from Ambion, by the Illumina platform. A total of seven lanes of sequencing data for each sample were available. The data set can be downloaded from NCBI Read Archive (SRA) at http://www.ncbi.nlm.nih.gov/sra under the accession numbers SRA010153 and SRA008403. As part of the MAQC project, the expression levels of 1044 genes in each sample were also measured by TaqMan Gene Expression Assay based on quantitative real time polymerase chain reaction (qRT-PCR) technology. The qRT-PCR measures are treated as the gold standard in the analysis.

2.4.2 Quantification Accuracy Comparison

Model Fitting and Selection. The Poisson mixture models can better fit and characterize RNA-Seq data than the simple Poisson model. We use Pearson's Chi-square test to show that simple Poisson model cannot fit the data well. A small p-value from the test indicates lack of fit, whereas a large p-value indicates good fit. We use 5% as the significance level. The simple Poisson model and the PMI model were applied to all the transcripts in the prostate cancer data. The percentage of transcripts for which each model provides good fit is used as an overall goodness of fit measure of the model. On average, in each cell sample, the simple Poisson model only managed to fit about 15.7% of the transcripts well, whereas the MPI model fit about 85.7% of the transcripts well; and the remaining 14.3% transcripts require more sophisticated models such as the PMAI and PMAIP models to characterize them well.

In order to demonstrate the diversity of transcript reads count data as well as the adaptivity of the proposed Poisson mixture models, we applied the BIC model selection procedure to the top 5000 highly expressed transcripts of each cell sample in the prostate cancer data. For convenience, the PMI, PMAI, and PMAIP models with K up to three were considered, which are denoted by, respectively, PMI_1, PMI_2, PMI_3, PMAI_2, PMAI_3, PMAIP_1, PMAIP_2, and PMAIP_3, where the subscript number indicates the number of components. The relative frequencies of the models selected by BIC or equivalently, the proportions of transcripts for which the models were selected are given below.

PMI_1	PMI_2	PMI_3	PMAI_2	PMAI_3	PMAIP_1	PMAIP_2	PMAIP_3	Not Fit
1.98%	29.61%	7.76%	17.34%	13.83%	0.84%	5.75%	10.85%	12.03%

It is clear that a large proportion of transcripts require more sophisticated models to fit them well.

In Figure 2.4.2, we demonstrate the reads count data of a transcript and the results from fitting the PMAIP mode with three components to the data. The upper-left plot shows that the original reads count data of the transcript are clustered into three groups, which include the large counts (illustrated by red circles), medium counts (blue squares), and small counts (green diamonds), respectively. The upper-right plot shows the estimated three Poisson component distributions with de-noised intensities $\hat{\lambda}_1 = .105$ (green solid curve), $\hat{\lambda}_2 = 2.79$ (blue dashed curve), and $\hat{\lambda}_3 = 20.544$ (red dotted curve), respectively. The estimated mixing proportions $\hat{\pi}_1$, $\hat{\pi}_2$, and $\hat{\pi}_3$ vary from base pair to base pair. The plot of $\hat{\pi}_2$ versus base pair index is given in the lower-left panel of the figure, and the plot of $\hat{\pi}_3$ versus base pair index is given in the lower-right panel. Because $\hat{\pi}_1 = 1 - \hat{\pi}_2 - \hat{\pi}_3$, the plot of $\hat{\pi}_1$ is not presented. In essence, the estimated intensities together with the mixing proportions have captured most information in the reads count data.

Expression Level Quantification. We applied the PMI, PMAI, and PMAIP models with up to three components to analyze the prostate cancer data, and then used the quantification rules proposed in Section 2.2.3, 2.3.5, and 2.3.6 to obtain

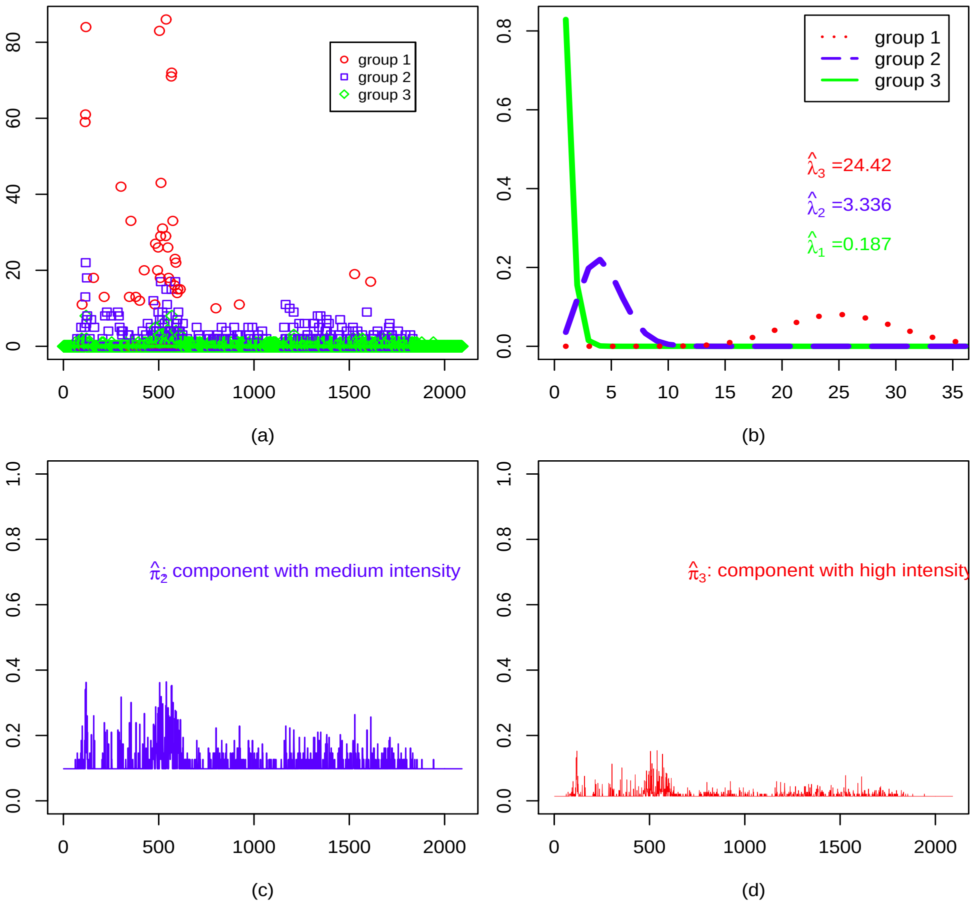

Figure 2.2. Read Counts, Estimated Intensities, and Mixing Proportions of a Transcript. Plot (a) shows the clustering of the reads counts into three groups labeled by 1 to 3 and indicated by red circles, blue squares, and green diamonds, respectively. Plot (b) gives the estimated distribution functions of the Poisson components of the PMAIP$_3$ model, indicated by green solid line, blue long dash line, and red dotted line, respectively. Plot (c) shows the estimated proportion of the second component with medium intensity at each base pair, according to the PMAIP$_3$ model. Plot (c) shows the estimated proportion of the third component with highest intensity at each base pair, according to the PMAIP$_3$ model.

expression measurements of the transcripts. Corresponding to the models used in transcript expression quantification, the resulting measurements are referred to as the PMI, PMAI, and PMAIP measurements. We further compared the proposed methods with other existing gene expression quantification methods including the effective length based RPKM (eRPKM) method, GPseq, mseq, and POME. The difference between eRPKM and the original RPKM (oRPKM) measure is that eRPKM uses the number of base pairs receiving at least one read as the length of a transcript, whereas oRPKM uses the total number of base pairs as the length. To demonstrate and compare the quantification results, we focused on the highly expressed genes. For each cell sample, we first selected the top 5000 highly expressed transcripts according to their oRPKM measurements based on the Helicos data; and then a filter was used to remove those transcripts with low read coverage [31] or extremely high read counts in the Illumina data. The number of remaining transcripts in each cell sample is listed in Table 2.1.

As discussed in Chapter 2.4.1, the oRKPM measurement of the expression level of a transcript based on the Helicos data is treated as the gold standard. The Spearman rank correlation coefficients between the gold standard and the eRPKM, GPseq, mseq, POME, PMI, PMAI, and PMAIP measurements were calculated and reported in Table 2.1. Among all the methods, PMI performed best in 8 out of 12 cell samples, POME performed best in 4 out of 12 cell samples, and eRPKM performed best in one cell sample (The best correlation coefficients are highlighted). The performances of PMAI and PMAIP are slightly worse than the best but still comparable. We observed that the improvement of PMI over eRPKM is substantial.

To further compare the performances of the methods, we focused on transcripts for which the BIC procedure selected the PMI model with three components as the most appropriate model. The Spearman correlation coefficients between the expression measurements of these transcripts and the gold standard were calculated and reported in Table 2.2. For this subgroup of transcripts, the performances of PMI, PMAI, PMAIP, and POME have improved, but the performances of eRKPM, GPseq,

Table 2.1
Comparison of Quantification Methods in Highly Expressed Transcripts. Each row is for one cell line. The quantification methods are listed from column 3 to column 10. The highest spearman rank correlation coefficients with the gold standard are highlighted in bold.

Cell line	Transcripts	eRPKM	GP	mseq	POME	PMI	PMAI	PMAIP
LnCaP0	4607	0.631	0.539	0.621	0.694	**0.720**	0.697	0.694
LnCaP24	4578	0.633	0.562	0.617	0.692	**0.723**	0.694	0.694
LnCaP48	4405	0.622	0.510	0.617	0.667	**0.671**	0.643	0.648
VCaP0	3780	0.581	0.610	0.557	0.620	**0.690**	0.681	0.681
VCaP24	4312	0.594	0.613	0.543	0.632	**0.690**	0.679	0.678
VCaP48	4175	0.637	0.637	0.498	0.677	**0.743**	0.725	0.726
DU145F	4605	0.620	0.482	0.603	**0.653**	0.642	0.621	0.628
aT34	3913	0.649	0.470	0.632	**0.662**	0.622	0.587	0.599
DU145F2	4615	0.620	0.481	0.602	**0.653**	0.642	0.617	0.620
aT34N	2924	0.534	0.521	0.526	0.544	**0.593**	0.577	0.579
VCaP	3357	0.545	0.410	0.477	**0.552**	0.535	0.513	0.517
RWPE	4505	0.536	0.463	0.526	0.581	**0.593**	0.577	0.565

Table 2.2
Comparison of Quantification Methods in Highly Expressed and Highly Variable Transcripts. Each row is for one cell line. The quantification methods are listed from column 3 to column 10. The highest spearman rank correlation coefficients are highlighted in bold

Cell line	Transcripts	eRPKM	GPseq	mseq	POME	PMI	PMAI	PMAIP
LnCaP0	2824	0.594	0.537	0.585	0.682	**0.751**	0.721	0.711
LnCaP24	2468	0.614	0.563	0.598	0.696	**0.767**	0.735	0.729
LnCaP48	2220	0.598	0.459	0.594	0.663	**0.698**	0.661	0.656
VCaP0	1282	0.594	0.604	0.588	0.670	**0.754**	0.751	0.745
VCaP24	1699	0.536	0.625	0.517	0.613	**0.716**	0.697	0.687
VCaP48	1298	0.602	0.630	0.545	0.672	**0.780**	0.756	0.746
DU145F	2262	0.594	0.462	0.573	0.647	**0.679**	0.657	0.652
aT34	1455	0.674	0.461	0.651	0.689	**0.712**	0.666	0.661
DU145F2	2318	0.600	0.466	0.584	0.650	**0.683**	0.655	0.647
aT34N	976	0.623	0.613	0.610	0.643	**0.747**	0.715	0.717
VCaP	1174	**0.645**	0.397	0.615	0.642	**0.645**	0.598	0.606
RWPE	2631	0.525	0.440	0.515	0.588	**0.632**	0.611	0.587

and mseq have deteriorated. The PMI measurements have achieved the highest correlation with the gold standard in all the 12 cell samples. The relative sub-optimal performances of PMAI and PMAIP may be due to the quantification rule currently used, which is mainly motivated by the PMI model. Different quantification rules need to be further developed to unleash the potential of the PMAI and PMAIP models.

We also applied eRPKM, GPseq, and PMI to quantify the expression levels of the genes in the Human Brain and UHR data set, in particular, the 1044 genes with qRT-PCR measurements. We filtered out those genes that do not receive any reads or have multiple matched names in the University of California, Santa Cruz (UCSC) genome browser [33]. The numbers of remaining genes in the Human Brain and

Table 2.3
Comparison of Quantification Methods In Human Brain and UHR data. The results for the UHR sample and Brain sample are separated into the left panel and right panel, respectively. Each row represents a lane in the data. The highest spearman rank correlation coefficients for each lane are highlighted in bold.

UHR					Brain				
Lane	Genes	eRPKM	GPseq	PMI	Lane	Genes	eRPKM	GPseq	PMI
1	894	0.836	0.825	**0.843**	1	856	0.761	0.770	**0.788**
2	889	0.830	0.826	**0.839**	2	865	0.759	0.781	**0.796**
3	893	0.811	0.821	**0.837**	3	864	0.758	0.782	**0.800**
4	881	0.820	0.820	**0.835**	4	860	0.767	0.776	**0.793**
6	890	0.824	0.822	**0.838**	6	867	0.757	0.781	**0.794**
7	880	0.810	0.821	**0.835**	7	872	0.766	0.784	**0.797**
8	883	0.815	0.819	**0.829**	8	862	0.745	0.784	**0.796**

UHR samples are listed in Table 2.3. The qRT-PCR measurements were considered of high quality and commonly used as the gold standard, so we will use them here to compare the other quantification methods. As before, the Spearman correlation coefficients between the measurements by eRPKM, GPseq and PMI and the qRT-PCR measurements were calculated and reported in Table 2.3. The PMI measurements have achieved the highest correlation coefficients with the gold standard in both the Brain and UHR samples and across all the lanes. Because qRT-PCR measurements are a more reliable and objective gold standard, the good performance of PMI in this data set corroborates its good performance in the prostate cancer data.

3. TRANSCRIPT EXPRESSION LEVEL QUANTIFICATION

Transcription is the first step of gene expression, and all transcription products, which are called RNA molecules or transcripts, form the transcriptome of a cell under a given developmental stage or physiological condition. The largest family of transcripts are mRNAs. The number of transcripts is much larger than the number of genes due to gene alternative splicing.

Transcriptome profiling, which is to comprehensively detect, catalog and quantify all transcripts in the transcriptome, is a grand challenge in molecular biology and functional genomics. In the past two decades, microarray has been used as the major technology for interrogating the transcriptome. Recently, the development of next generation sequencing (NGS) technology has revolutionized the way genomic research is conducted. In particular, NGS technology provides a new venue for mapping and quantifying the transcriptome. As one of such new technologies, RNA-Seq directly measures the abundance of transcripts and has become an attractive alternative for profiling the transcriptome [4].

3.1 Background and Existing Transcript Quantification Methods

In a typical RNA-Seq experiment, RNA molecules are fragmented into small pieces and converted to a library of cDNA fragments with adapters attached to one end or both ends. Each fragment, after amplification, is then sequenced using one of the NGS technologies, generating hundreds of millions of short nucleotide sequences or short reads. After sequencing, the resulting reads are either assembled *de novo* or aligned to the reference genome to produce a genome-scale transcriptional profile. Using the mapped short reads (single-end reads or paired-end reads), the number of reads that each base pair of the reference genome receives can be calculated, and the resulting counts are collectively referred to as the base level read counts data.

The read counts data can be used to quantify either the expression level of a ROI on the reference genome, such as an exon or a gene, or the expression levels of transcripts. On the one hand, the expression level of a ROI is relatively easier to quantify because the total number of reads received by the ROI directly reflects its abundance and thus can be used to measure its expression level after proper normalization. Various statistical model based quantification methods for ROI's have been proposed in the literature, which include GPseq [20], PMI [22] and POME [21]. On the other hand, the expression levels of transcripts are more difficult to quantify because one may not be able to allocate the short reads uniquely to transcripts. Due to alternative splicing, multiple transcripts can give rise to identical reads. In other words, the labels of the transcripts that identical reads are generated from are missing. Therefore, the quantification of transcript expression levels is an indirect problem as in comparison with the quantification of ROI expression levels.

A number of methods have been proposed for transcript expression level quantification in the literature. Jiang *et al.* proposed to use a Poisson distribution to model the total number of reads in each exon or exon-exon junction [34]. The intensity of the Poisson distribution is further assumed to be a linear combination of the expression levels of the transcripts that contain the exon or exon-exon junction. Trapnell *et al.* proposed a method called Cufflinks [35]. Cufflinks uses a probabilistic model to represent the generating scheme for each read. The probabilistic model involves the expression levels of the transcripts that can produce this read. Li *et al.* proposed a Lasso regression approach (called IsoLasso) to quantifying the expression levels of transcripts [36]. IsoLasso first divides a gene into a set of segments based on exon-intron boundaries and then applies Lasso to regress read counts in the segments against the transcripts' expression indexes. Li *et al.* proposed to use a sparse linear model for isoform discovery and abundance estimation (SLIDE) [37]. SLIDE is similar to IsoLasso except that SLIDE uses read counts in bins instead of segments and the mapped reads are considered in the same bin if their starting and ending positions belong to the same exons, respectively. All these four methods discussed above use the typical approach for solving indirect problems, which is to use a probabilistic or statistical model to relate observations (i.e. observed reads) to unobserved quantities (i.e. transcript abundances). They differ from each other in terms of the units of observations and the type of models they used. In particular, Jiang *et al.* used exon or exon-exon junction as the unit, Cufflinks treats each observed read as the unit, IsoLasso used each segment as the unit, and SLIDE uses each bin as the unit.

The types of units used by the four methods discussed in the previous paragraph may not be statistically adequate, and may lead to some drawbacks. One direct consequence of using the bin, exon, or segment as the observation unit is the loss of information. A well-known advantage of NGS technologies is that they provide single base resolution. In a typical RNA-Seq experiment, due to fragmentation and sampling, some base pairs of a gene receive reads, while others do not receive reads. We all agree that base pairs with positive read counts can reflect the abundance level of transcripts. However, base pairs without reads also contain information about the abundance of transcripts and reflect various uncertainties in a RNA-Seq experiment. Thus, using only base pairs with read counts but not those with zero read counts again results in information lose. The aggregation of positive read counts would lead to further information loss. Both the methods proposed by Jiang *et al.* and IsoLasso model the aggregated read counts in exon, segment, or junction counts rather than base pair level counts. Cufflinks models only observed reads, but fail to model zero read counts. SLIDE only models aggregated bin counts, whereas base pairs with zero read counts and bins without observed reads are ignored. Therefore, these methods fail to utilize all information contained in RNA-Seq read counts data, and as a consequence may fail to provide accurate quantification of transcript expression levels. In some cases, they suffer from the non-identifiability problem, and fail to distinguish transcripts that are distinguishable.

To improve upon these existing methods, in this paper, we treat each base pair as the unit and propose a novel approach that models both zero and non-zero read counts for quantifying transcript expression levels. The generating scheme for read count at each base pair can be considered involving two steps. In the first step, short reads are generated from each transcript that contains this base pair according to a certain distribution; and in the second step, all these shorts reads are mapped to the same base pair on the reference genome, which give rise to the observed read count. In the first step, the label of the transcript each short read is generated from is conceptually available, whereas after the second step, this label becomes missing. Therefore, the second step can be regarded as a convolution step, in which short reads from different transcripts are mixed and the information about their origins is lost.

Instead of directly proposing models for the observed read counts on the genome as the four existing methods discussed above do, we propose to model the short read counts of transcripts, and refer to the resulting model as the transcript level model. After the transcript level models are available, the models for the base level read counts on the reference genome can then be derived as the convolution of the transcript level models, which is referred to the genome level models. In particular, we propose to use the mixture of Poisson models at the transcript level, which lead to the convolution of mixture of Poisson models at the genome level.

In this article, we do not consider the transcripts assembly problem. Instead we focus on the quantification of a given set of candidate transcripts. The candidate transcripts set can include annotated transcripts, novel transcripts of current research interest, or those assembled by other methods.

3.2 Convolution of Poisson Mixture Model

3.2.1 Unique Exon Annotations

The exons of a gene form a partition of the gene's exonic region. In general, these exons represent the smallest units that can be entirely transcribed or skipped during the transcription of the gene. However, it can also happen that only a part of an exon is transcribed. When this happens, the involved exon needs to be further divided into sub-exons so that each exon or sub-exon is either completely retained or excluded in a transcript. Suppose gene g contains k_g exons or sub-exons, which are labeled as $e_1, \ldots, e_{k_g}$ from the left end to the right end of the gene, respectively. Suppose the number of base pairs in e_i is n_i for $1 \leq i \leq k_g$. Let $n_0 = 0$ and $n = n_1 + \ldots + n_{k_g}$. We index the exonic base pairs of gene g from left to right as $1, \ldots, n$. It is clear that exon e_i consists of the base pairs $\{\sum_{j=0}^{i-1} n_j + 1, \ldots, \sum_{j=0}^{i} n_j\}$ for $1 \leq i \leq k_g$.

3.2.2 Isoforms Notations and Read Types

As discussed in the Introduction, the transcription of gene g can produce different transcripts, which are referred to as alternatively spliced isoforms. For example, if only e_1 and e_5 are kept during the transcription while all the other exons are skipped, the resulting transcript consists of e_1 and e_5, which can be denoted as $T_1 = e_1 e_5$; and if only e_1, e_2, and e_6 are kept and the others are skipped, the resulting transcript is $T_2 = e_1 e_2 e_6$. Let $\mathcal{T}_g = \{T_1, \ldots, T_N\}$, where $N = |\mathcal{T}_g|$, be a set of candidate transcripts. This article will focus on the quantification of the expression levels of T_1, $\ldots$, and T_N, using RNA-Seq base level counts data.

For every short read mapped to the annotated region of gene g, its starting and ending positions can be obtained. The starting position and ending position of a single end read are denoted as *start* and *end*, respectively. Each paired-end read contains two mate pairs, and we denote the starting positions and ending positions of the two mate pairs as $start_1$, end_1, $start_2$, and end_2, respectively. For any two single-end reads, if their starting positions and ending positions belong to exons e_{l_1} and e_{l_2}, respectively, they are said to be of the same type, which is denoted as $r_{l_1 l_2}$. Similarly for a paired-end read, if its starting and ending positions $start_1$, end_1, $start_2$, and end_2 are in exons e_{l_1}, e_{l_2}, e_{l_3}, and e_{l_4}, it is said to be of type $r_{l_1 l_2 l_3 l_4}$. Let $\mathcal{R}$ denote the collection of all possible read types. The mapped reads can also be classified into non-junction reads and junction reads. Non-junction reads are those reads whose starting and ending positions are in the same exon, whereas junction reads are those whose starting and ending positions are not in the same exon. Correspondingly, all of the possible read types can be classified into non-junction read types and junction read types. For example, r_{1111} is non-junction read type because reads of type r_{1111} have their starting and ending positions in the same exon e_1, whereas r_{1122} is a junction read type because reads of type r_{1122} have the starting and ending positions of the first mate pair in exon e_1 but the starting and ending positions of the second mate pair in exon e_2. Let $\mathcal{N}$ denote the collection of non-junction read types and $\mathcal{J}$ the collection of junction read types. Then we have $\mathcal{R} = \mathcal{N} \cup \mathcal{J}$. At base pair m of gene g, the number of reads or the total read count starting at this base pair can be obtained, which is denoted as S_m. Furthermore, this total read count can be partitioned into read counts of different types as $S_m = \sum_{r \in \mathcal{R}} Y_m^r$, where Y_m^r denote the total count of type r ($r \in \mathcal{R}$) reads starting at base pair m.

As discussed in the Introduction, mapped reads and their corresponding counts can be used to quantify the overall expression level of a gene, but they may not be directly used to quantify the expression levels of transcripts because the information of which transcript each read is generated from may not always be available. The counts of different types of reads Y_m^r's contain more information than the total read counts S_m, but they may not be directly used for quantifying transcript expression levels due to the same reason. To properly model the distribution of Y_m^r, a convolution model is needed. In essence, the quantification of transcripts' expression levels is an indirect statistical inference problem, which is to infer the transcripts' expression levels from the genome level read counts data.

To facilitate the inference, we first propose statistical models to represent the reads-generating mechanism for each individual transcript, which are refer to as the transcript level models, then, use a convolution model to characterize the read counts of different types at the gene level.

3.2.3 CPM-Seq Model

The majority of current NGS technologies sequence fragments of length around 200 base pairs or less and the average length of exon on human genome is 170 base pairs long [38]. Therefore, counts of junction reads that involve more than two exons are usually low. In this paper, we only consider reads that involve no more than two exons. As a result, the notation of different types of paired-end reads can be much simplified.

Consider a paired-end read of type $r_{l_1 l_2 l_3 l_4}$. If the read is a non-junction read, then $l_1 = l_2 = l_3 = l_4 = l$, and $r_{l_1 l_2 l_3 l_4}$ can be simplified to be r_{ll}; if the read is a junction read, there can be three possible scenarios, which are $l_1 = l_2 = l_3 < l_4$, $l_1 = l_2 < l_3 = l_4$, and $l_1 < l_2 = l_3 = l_4$. Since in all three scenarios, the read involves the same exon junction between e_{l_1} and e_{l_4}, we do not distinguish them and instead collapse them into one type and denote it as $r_{l_1 l_4}$. Therefore, only two indices are needed to indicate the type involving no more than two exons.

For gene g with k_g exons, there are in total $k_g(k_g + 1)/2$ all possible types of reads among which k_g types are non-junction types and $k_g(k_g - 1)/2$ types are junction types. We still use $\mathcal{N}$ and $\mathcal{J}$ to denote the collection of all non-junction types and the collection of all junction types under consideration, respectively. It is clear that $\mathcal{R} = \mathcal{N} \bigcup \mathcal{J}$ is the collection of all possible types. We use E_g to denote the exonic region of gene g, that is, $E_g = e_1 \cup e_2 \cup \cdots \cup e_{k_g}$. For any type $r \in \mathcal{R}$ and base pair $m \in E_g$, recall Y_m^r denotes the count of type r reads starting at base pair m. We use $Y^r = \{Y_m^r; m \in E^r\}$ to represent the collection of type r read counts, where E^r is the collection of all possible base pairs that can become the starting positions of type r reads. We use $\mathbf{Y} = \{Y^r; r \in \mathcal{R}\} = \{Y_m^r; r \in \mathcal{R}, m \in E^r\}$ to represent the counts for all types of reads defined on all possible base pairs.

Recall $\mathcal{T}_g = \{T_1, T_2, \ldots, T_N\}$ be a collection of N candidate transcripts of gene g under consideration. Consider transcript $T_t \in \mathcal{T}_g$ for $1 \leq t \leq N$. For base pair m of transcript T_t, i.e. $m \in T_t$, and read type $r \in \mathcal{R}$, we define X_{tm}^r to be the count of type r reads generated from T_t in a RNA-Seq experiment, and assume X_{tm}^r follows a two-component mixture of Poisson distribution with the probability mass function

$$f\left(X_{tm}^r = x|\lambda_t, p_t\right) = \sum_{i=1}^{2} p_{ti} Poi\left(x; \lambda_{ti}\right), \tag{3.1}$$

where $p_t = (p_{t1}, p_{t2})'$ is the vector of mixing proportions satisfying $\sum_{i=1}^{2} p_{ti} = 1$, $\lambda_t = (\lambda_{t1}, \lambda_{t2})'$ is the vector of intensity rates of the two Poisson components, $Poi(x; \lambda_{ti}) = (\lambda_{ti})^x \exp(-\lambda_{ti})/x!$, and x is any non-negative integer.

The first Poisson component with intensity rate λ_{t1} is used to model the base pairs that either are not covered in RNA-Seq experiment or covered with an abnormally smaller number of reads due to various sequencing uncertainties, whereas the second Poisson component with intensity rate λ_{t2} is used to model the base pairs that are normally covered by RNA-Seq experiment and λ_{t2} represents the abundance of the transcript. As we discussed in the Introduction, using a background intensity to model zero or small counts is crucial because zero counts contain information, and the modeling of zero counts will further help us separate transcripts. The proposed mixture of Poisson distribution can also be considered as a zero-inflated Poisson distribution with the first component accounting for zero or low counts in RNA-Seq data.

Note that X_{tm}^r for $m \in T_t$, $T_t \in \mathcal{T}_g$, and $r \in \mathcal{R}$ may not be always directly observable. After the reads are mapped to the annotated region of gene g, the transcript label t is missing as discussed previously. Instead of observing X_{tm}^r, we may only observe Y_m^r, which is the total count of type r reads for $m \in E^r$. There however exists a relationship between Y_m^r and X_{tm}^r, which can be obtained explicitly when the collection of transcript $\mathcal{T}_g$ is given. Consider a base pair $m \in E^r$ for $r \in \mathcal{R}$. Suppose a total of N_r transcripts $\{T_{i_1} \ldots T_{i_{N_r}}\} \subset \mathcal{T}_g$ can give rise to type r reads. Let $X_{i_1 m}^r, \ldots, X_{i_{N_r} m}^r$ be the counts of type r reads at base pair m from the candidate transcripts $T_{i_1}, \ldots, T_{i_{N_r}}$, respectively. Then Y_m^r is the sum of X_{km}^r for $k = i_1, \ldots, i_{N_r}$, i.e.

$$Y_m^r = X_{i_1 m}^r + \ldots + X_{i_{N_r} m}^r. \tag{3.2}$$

Therefore, the distribution Y_m^r is the convolution of the distributions of $X_{i_1 m}^r, \ldots, X_{i_{N_r} m}^r$. Because X_{km}^r follows the two-component mixture of Poisson distribution as defined previously in model (3.1) with $f(X_{km}^r = x) = \sum_{i=1}^{2} p_{ki} Poi(x; \lambda_{ki})$ for $k \in \{i_1, i_2, \ldots, i_{N_r}\}$, the distribution of Y_m^r can be derived explicitly, which is a 2^{N_r}-component mixture of Poisson distribution with the following probability mass function

$$p(Y_m^r = y) \equiv f(y|\{T_{i_1} \ldots T_{i_{N_r}}\}) = \prod_{k=i_1}^{i_{N_r}} [\sum_{j_k=1}^{2} p_{kj_k} Poi(y; \lambda_{kj_k})] \qquad (3.3)$$

$$= \sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} \left[p_{i_1 j_{i_1}} \ldots p_{i_{N_r} j_{i_{N_r}}} Poi(y; \lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}}) \right],$$

where y is a non-negative integer. There are in total $4N_r$ unknown parameters in the model above, which include $2N_r$ proportion parameters p_{k1} and p_{k2} satisfying $p_{k1} + p_{k2} = 1$ for $k = i_1, i_2, \ldots, i_{N_r}$ and $2N_r$ intensity rates λ_{k1} and λ_{k2} for $k = i_1, i_2, \ldots, i_{N_r}$. Although the intensity rates λ_{t1} may vary from transcript to transcript, because they mainly depend on the coverage uncertainty in RNA-Seq experiment as discussed before, we further assume that they are equal, that is, $\lambda_{i_1 1} = \ldots = \lambda_{i_{N_r} 1}$.

The two-component mixture of Poisson distribution of X_{km}^r, for $k \in \{i_1, \ldots, i_{N_r}\}$, plays a key role in making the inference of transcripts' expression levels from Y_m^r possible. Suppose X_{km}^r follows a simple Poisson distribution $Poi(\lambda_k)$ with intensity rate λ_k. Then, $Y_m^r = X_{i_1 m}^r + \cdots + X_{i_{N_r} m}^r$ follows a simple Poisson distribution with intensity rate $\lambda_{i_1} + \cdots + \lambda_{i_{N_r}}$. Given y_m^r, $\lambda_{i_1} + \cdots + \lambda_{i_{N_r}}$ can usually be estimated but the individual intensity rates λ_{i_1}, $\ldots$, $\lambda_{i_{N_r}}$ usually cannot be uniquely identified or estimated. The two-component mixture of Poisson model for X_{km}^r not only characterizes the reads-generating mechanism at transcript level but also makes the inference of transcript expression levels from the gene level read counts data possible. As will be shown later, given the realization y_m^r for $m \in E^r$ and $r \in \mathcal{R}$, the parameters in model (3.3) can be estimated using the maximum likelihood methods, and the resulting estimates can then be further used to quantify the transcript expression levels. We will also show that it is not necessary to allocate reads to different transcripts anymore, as is commonly done in the literature. In theory, X_{km}^r can be assumed to follow a mixture of Poisson distributions with more than two components, and all methods developed in this article can be extended accordingly.

In the discussion above, we do not distinguish junction reads from non-junction reads. In real RNA-Seq data, non-junction reads and junction reads demonstrate different characteristics. First of all, junction reads contain more information about which transcripts they are generated from than non-junction reads. Consider junction reads of type $r_{l_1l_2} \in \mathcal{J}$ with $l_1 < l_2$ and non-junction reads of type $r_{l_1l_1} \in \mathcal{N}$. Let $\mathcal{T}^{r_{l_1l_1}}$ be the collection of candidate transcripts that can generate reads of non-junction type $r_{l_1l_1}$, and $\mathcal{T}^{r_{l_1l_2}}$ the collection of candidate transcripts that can generate reads of junction type $r_{l_1l_2}$. It is clear that $\mathcal{T}^{r_{l_1l_2}}$ is a subset of $\mathcal{T}^{r_{l_1l_1}}$, and hence reads of type $r_{l_1l_2}$ are less convoluted and therefore contain more direct information about the transcripts than reads of type $r_{l_1l_1}$. Due to the same reason, junction reads are often used for assembling novel transcripts. Secondly, because the exon-exon junctions are on average much longer than the fragments sequenced in RNA-Seq experiment, the number of junction reads is much smaller than the number of non-junction reads. In other words, given a junction read type $r \in \mathcal{J}$, for $m \in E^r$, the number of positive y_m^r's is small. We postulate that the excessive large number of base pairs with $y_m^r = 0$, for $r \in \mathcal{J}$, maybe caused by other unknown missing mechanisms, which cannot be properly modeled. Therefore, when estimating the model parameters, it may not be appropriate to use the original distribution of Y_m^r, for $r \in \mathcal{J}$. One approach to solving this difficulty is to consider only positive counts $y_m^r > 0$ and the conditional distribution of y_m^r given $y_m^r > 0$. Another advantage of using the positive counts and their conditional distributions is to avoid the ambiguity in the definition of E^r for

$r \in \mathcal{J}$. We define $y^r_+ = \{y^r_m : m \in E^r_+\}$, where $E^r_+ = \{m : y^r_m > 0\}$. For $r \in \mathcal{J}$, the conditional distribution of Y^r_m given $Y^r_m > 0$ for $m \in E^r$ is given as follows.

$$p(Y^r_m = y|Y^r_m > 0) = \frac{\prod_{k=i_1}^{i_{N_r}} [\sum\limits_{j_k=1}^{2} p_{kj_k} Poi(y; \lambda_{kj_k})]}{1 - p(Y^r_m = 0)} \tag{3.4}$$
$$= \frac{\sum\limits_{j_{i_1}} \cdots \sum\limits_{j_{i_{N_r}}} \left[p_{i_1 j_{i_1}} \ldots p_{i_{N_r} j_{i_{N_r}}} Poi(y; \lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}}) \right]}{1 - \sum\limits_{j_{i_1}} \cdots \sum\limits_{j_{i_{N_r}}} p_{i_1 j_{i_1}} \ldots p_{i_{N_r} j_{i_{N_r}}} e^{-\lambda_{i_1 j_{i_1}} - \ldots - \lambda_{i_{N_r} j_{i_{N_r}}}}}.$$

3.2.4 EM Algorithm for CPM-Seq

Directly optimizing the original composite likelihood function $L(\theta|\tilde{\mathbf{y}})$ is difficult and time consuming. Instead we apply the EM algorithm to calculate the MCLE $\hat{\theta}$. In order to develop the EM-algorithm, we introduce a two-step data generating scheme as follows. For type $r \in \mathcal{R}$ and $m \in E^r$, recall that Y^r_m follows a 2^{N_r}-component mixture of Poisson distribution, and we index the components by $i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}}$, for $j_{i_k} \in \{1, 2\}$ and $k \in \{1, \ldots, N_r\}$. We define membership indicator variables $Z^r_{m,(i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}})}$ such that

$$\begin{cases} Z^r_{m,(i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}})} = 1 \text{ if } Y^r_m \sim Poi\Big(\lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}}\Big) \\ Z^r_{m,(i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}})} = 0 \text{ if } Y^r_m \not\sim Poi\Big(\lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}}\Big). \end{cases} \tag{3.5}$$

Let $z^r = \{z^r_{m,(i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}})}, m \in E^r\}$, for $r \in \mathcal{N}$ and $z^r_+ = \{z^r_{m,(i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}})}, m \in E^r_+\}$ for $r \in \mathcal{J}$. Let $\tilde{\mathbf{z}} = \{z^r : r \in \mathcal{N}\} \cup \{z^r_+ : r \in \mathcal{J}\}$, which is the membership indicator of $\tilde{\mathbf{y}}$. With both $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$, the complete composite log-likelihood for θ can be written as

$$l(\theta|\tilde{\mathbf{y}}, \tilde{\mathbf{z}}) = \log(L(\theta^r|\mathbf{y}, \mathbf{z})) \tag{3.6}$$

$$= \sum_{r \in \mathcal{N}} l^r(\theta^r|y^r, z^r) + \sum_{r \in \mathcal{J}} l^r_c(\theta^r|y^r, z^r, y^r > 0),$$

where for non-junction type $r \in \mathcal{N}$, the complete log-likelihood is

$$\begin{aligned} l^r(\theta^r|y^r, z^r) &= p(y^r, z^r|\theta^r) \\ &= \sum_{m \in E_r} \Bigg\{ \sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} (z^r_{m,i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}}}) \cdot \Bigg[\log(p_{i_1 j_{i_1}} \\ &\ldots p_{i_{N_r} j_{i_{N_r}}}) + \log(Poi(\lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}})) \Bigg] \Bigg\}, \end{aligned} \tag{3.7}$$

and for junction read type $r \in \mathcal{J}$, the complete conditional log-likelihood is

$$\begin{aligned} &l^r_c(\theta^r|y^r, z^r_+, y^r > 0) = p(y^r, z^r_+|\theta^r, y^r > 0) = \sum_{m \in E^r_+} \\ &\Bigg\{ \sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} (z^r_{m,i_1,j_{i_1} \ldots i_{N_r} j_{i_{N_r}}}) \cdot \Bigg[\log(p_{i_1 j_{i_1}} \ldots p_{i_{N_r} j_{i_{N_r}}}) + \\ &\log(Poi(\lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}})) \Bigg] - \sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} (z^r_{m,i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}}}) \cdot \\ &\log \left(1 - p_{i_1 j_{i_1}} \ldots p_{i_{N_r} j_{i_{N_r}}} \cdot e^{-\lambda_{i_1 j_{i_1}} \ldots - \lambda_{i_{N_r} j_{i_{N_r}}}} \right) \Bigg\}. \end{aligned}$$

Suppose the current parameter estimate is $\hat{\theta}^{cur} = (\hat{\lambda}^{cur}, \hat{p}^{cur})'$. The E-step is to calculate the expected complete log-likelihood function

$$Q(\theta|\hat{\theta}^{cur}, \tilde{\mathbf{y}}) = E_{\tilde{\mathbf{z}}} \Big[\log \big(l(\theta|\hat{\theta}^{cur}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) \big) \Big], \tag{3.8}$$

where the expectation is over the conditional distribution of $\tilde{\mathbf{z}}$ given $\hat{\lambda}^{cur}$, $\hat{p}^{cur}$, and $\tilde{\mathbf{y}}$. Notice that $Q(\theta|\hat{\theta}^{cur}, \tilde{\mathbf{y}})$ can also be written as

$$E[\sum_{r \in \mathcal{N}} l^r(\theta^r|y^r, \hat{\theta}^{cur}, z^r) + \sum_{r \in \mathcal{J}} l^r_c(\theta^r|y^r, \hat{\theta}^{cur}, z^r_+, y^r > 0))]. \tag{3.9}$$

The function Q consists of two parts, one of which involves the non-junction types, and the other involves the junction types. The expectation of z^r for $r \in \mathcal{N}$ and z^r_+ for $r \in \mathcal{J}$ can be calculated separately. For $r \in \mathcal{N}$, it is to compute

$$
\begin{aligned}
&E(z^r_{m,i_1k_{i_1}\ldots i_{N_r}k_{i_{N_r}}}|\hat{\lambda}^{cur}, \hat{p}^{cur}, y^r_m) \\
&= \frac{\hat{p}^{cur}_{i_1k_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r}k_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1k_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r}k_{i_{N_r}}})}{\sum\limits_{j_{i_1}} \cdots \sum\limits_{j_{i_{N_r}}} \left[\hat{p}^{cur}_{i_1j_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r}j_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1j_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r}j_{i_{N_r}}})\right]},
\end{aligned}
\tag{3.10}
$$

and for $r \in \mathcal{J}$, it is to compute

$$
\begin{aligned}
&E(z^r_{m,i_1k_{i_1}\ldots i_{N_r}k_{i_{N_r}}}|\hat{\lambda}^{cur}, \hat{p}^{cur}, y^r_m, y^r_m > 0) \\
&= \frac{\hat{p}^{cur}_{i_1k_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r}k_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1k_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r}k_{i_{N_r}}})}{\sum\limits_{j_{i_1}} \cdots \sum\limits_{j_{i_{N_r}}} \left[\hat{p}^{cur}_{i_1j_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r}j_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1j_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r}j_{i_{N_r}}})\right]}.
\end{aligned}
\tag{3.11}
$$

The M-step is to maximize Q with respect to λ and p, and the resulting maximizers can be used to update $\hat{\theta}^{cur} = (\hat{\lambda}^{cur}, \hat{p}^{cur})'$. We use a block coordinate descent algorithm to optimize Q. First, we fix the value of p at $\hat{p}^{cur}$ and maximize Q with respect to λ. Specifically, we can either solve the following system of $N+1$ gradient equations,

$$\frac{\partial Q}{\partial(\lambda_{11}, \lambda_{12}, \ldots, \lambda_{N2})'} = 0,$$

or directly maximize the Q function with respect to λ. The resulting maximizer is $(\hat{\lambda}_{11}, \hat{\lambda}_{12}, \ldots, \hat{\lambda}_{N2}) = \arg\max_{\lambda_{11},\lambda_{12}\ldots,\lambda_{N2}} Q$, and the current estimate of $\hat{\lambda}^{cur}$ is updated to be $(\hat{\lambda}_{11}, \hat{\lambda}_{12}, \ldots, \hat{\lambda}_{N2})$. Although, for each gene with multiple transcripts, we can write down the gradient function and provided it in the optimization package, it is generally not practical to write down the gradient function for each gene model. Instead, we either numerically evaluate the gradient or use the Automatic Differentiation Model Builder (ADMB), which will be described in more details in Section 4.5.2. Second, we fix the the value of λ at $\hat{\lambda}^{cur}$, and optimize Q with respect to p_{t2} for $1 \le t \le N$ by solving the following gradient function,

$$\frac{\partial Q}{\partial p_{t2}} = 0 \text{ for } 1 \le t \le N. \tag{3.12}$$

Alternatively, we can also directly maximize the Q function with respect to the p. Let the resulting solution be $\hat{p}_{t2} = \arg\max_{p_{t2}} Q$ and then $\hat{p}_{t1} = 1 - \hat{p}_{t2}$ for $1 \leq t \leq N$. Then current estimate $\hat{p}^{cur}$ is updated to be $(\hat{p}_{12}, \ldots, \hat{p}_{N2})$.

The EM algorithm iterates between the E-step and M-step until some convergence criterion is satisfied. It is worth pointing out that as the number of candidate transcripts increases, the computational complexity also increases. The EM algorithm for the CPM model suffers from the curse of dimensionality and the problem of local optima. To deal with the first problem, more sophisticated optimization algorithms or parallel computing techniques could be implemented. To deal with the second problem, we adopt the strategy of using multiple initializations.

We repeat the EM algorithm with different initial values of the parameters and choose the estimates that achieve the largest likelihood value.

3.2.5 Quantification rule

Suppose the MCLEs of the model parameters for transcript T_t are calculated to be $\hat{\lambda}_{t1}$, $\hat{\lambda}_{t2}$, $\hat{p}_{t1}$, and $\hat{p}_{t2}$. Following the quantification procedure proposed in section 2.2.3, the expression level of transcript T_t is quantified to be $g_t^s = (s\hat{\lambda}_{t1}\hat{p}_{t1} + \hat{\lambda}_{t2}\hat{p}_{t2})/(s\hat{p}_{t1} + \hat{p}_{t2})$, where s is a pre-specified number between 0 and 1. When s is close to 0, g_t^s is dominated by the component that mainly represents transcript's abundance, and g_t^s is sensitive but suffers from high variability. When s is close to 1, g_t^s is dominated by the component that mainly represents background noise or zero counts, and g_t^s is stable but insensitive to the transcript's expression level. A proper value of s can avoid the two extremes and lead to a sensitive as well as robust quantification result of the transcript. As proposed in section 2.2.3, using single isoform genes, a bootstrap procedure can be used to find the proper value of s. In practice, we found that a s value between 0.2 and 0.3 will in general be a good choice.

3.2.6 Identifiability

In this section, we provide a simple simulation example to show why the CPM-Seq model is able to correctly estimate the transcript expression levels. Consider a hypothetical gene with two exons e_1 and e_2 and three transcripts T_1, T_2, and T_3. The first transcript T_1 contains only the first exon, and the second transcript T_2 only includes the second exon. The third transcript T_3 contains both e_1 and e_2. We used transcript level mixture of Poisson model to simulate counts. We set $\lambda_{12} = 5$, $\lambda_{22} = 10$, and $\lambda_{32} = 15$. For each transcript, the background noise is set to be 0.04, and the proportions for the background noises are set to be 0.7, 0.8, and 0.9. If no junction reads are generated in the simulation, and only non-junction reads are generated, most of the existing methods would not be able to uniquely estimate the transcript expression levels. Once the transcript level counts are generated, they are convolution as $Y_m^{r_{11}} = X_{1m}^{r_{11}} + X_{3m}^{r_{11}} = \sum_{i=1}^{2} \sum_{j=1}^{2} p_{1i} p_{3j} \text{Poi}\left(y; \lambda_{1i} + \lambda_{3i}\right)$ and $Y_m^{r_{22}} = X_{2m}^{r_{22}} + X_{3m}^{r_{22}} = \sum_{i=1}^{2} \sum_{j=1}^{2} p_{2i} p_{3j} \text{Poi}\left(y; \lambda_{2i} + \lambda_{3i}\right)$. CPM-Seq model is fitted to estimate the expression levels of three transcripts, and it is able to identify the expression levels correctly in 98 out of 100 simulation runs. The reason that the CPM-Seq model is identifiable is because the Poisson mixture model is identifiable [39], and as a result, the convolution of Poisson mixture model is again a Poisson mixture model with more components. Thus, the convolution of Poisson mixture model is also identifiable. Additionally, the identifiability can be explained by the unique solution of linear system of equations. As is demonstrated above that both $Y_m^{r_{11}}$ and $Y_m^{r_{22}}$ follow Poisson mixture distribution with four components. We can first fit a Poisson model with four components to the r_{11} and r_{22} types of data. Let the estimated Poisson intensities be $b_{11}, b_{12}, b_{13}, b_{14}, b_{21}, b_{22}, b_{23}$, and b_{24}. Then the expression level of each transcript can be obtained by solving the following system of linear equations $Ax = b$, where

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \tag{3.13}$$

$x = (\lambda_{11}, \lambda_{12}, \lambda_{22}, \lambda_{32})^T$ and $b = (b_{11}, b_{12}, b_{13}, b_{14}, b_{21}, b_{22}, b_{23}, b_{24})^T$. Since $rank(A) = 4$, which is the number of unknown parameters of x, the linear system will have unique solutions. In general, when we have multiple exons and multiple transcripts, we can show that the rank of the A matrix will be equal to the number of unknown parameters of x. Thus the convolution Poisson mixture model is always identifiable.

3.2.7 Simulation Study and Real Application

In order to further compare CPM-Seq with Cufflinks, we need to have a gold standard as the benchmark. In simulation study, we can simulate expression levels and treat them as the gold standard. In real data application, however, the true expression levels of transcripts are in general not available; and the qRT-PCR measurements are instead popularly used as the gold standard. In this section, we first use simulation study to compare our proposed method with Cufflinks and discuss their concordance and discrepancies. The simulation study is conducted at two different scales, which are the small and large scales, respectively. Examples 1 and 2 present the small and large scale studies, respectively. We further use two real datasets to compare our proposed method with Cufflinks. The first dataset contains the single-end sequencing data and qRT-PCR measurements of eight transcripts. We use the qRT-PCR measurements as the gold standard. The second dataset contains the paired-end sequencing data of a brain sample. The comparison results based on these two datasets are presented in Example 3 and Example 4, respectively.

Simulation Study

There are two possible ways to generate RNA-Seq read counts data. One way is to simulate the data from a pre-specified parametric model, and the other way is to use a RNA-Seq simulator. To make our simulation study more convincing, we follow the latter approach. We choose the Flux simulator to generate RNA-Seq short reads. The Flux simulator was developed by Gabriel *et al.* [40] to simulate RNA-Seq experiments *in silico* and is among the most sophisticated simulators. Given a set of transcripts and their expression levels, the Flux simulator simulates the protocols of RNA-Seq experiment step-by-step to generate the short reads.

Example 1 We conducted a small scale simulation study to compare the performances of CPM-Seq and Cufflinks. Five genes were selected from chromosome 1 of human genome. Each gene contains three annotated isoforms. Using the Flux simulator, 75 bp paired-end reads are generated for these 15 isoforms as follows. Firstly, the simulator randomly assigned expression levels to all 15 isoforms in the annotation. Secondly, the simulator randomly fragmented these isoform molecules into small pieces, which were then amplified *in silico*. Thirdly, the simulator sequenced these fragments and generated three thousand 75 bp paired-end reads. Once the reads were obtained, we mapped them back to the reference genome using Tophat [6]. We converted the mapped reads to counts data of different types. Based on the counts data of different types, we applied CPM-Seq and Cufflinks separately to quantify the expression levels of the 15 transcripts. We refer to the resulting measurements as the CPM-Seq measurements and Cufflinks measurements, respectively. The expression levels of the transcripts assigned by the simulator in the first step were treated as the gold standard.

The Pearson correlation coefficient between the CPM-Seq measurements and the gold standard is 0.715, and the Pearson correlation coefficient between the Cufflinks measurements and the gold standard is 0.665. The scatter plots of the CPM-Seq and Cufflinks measurements against those of the gold standard are given in Figure 3.1. We also calculated the Spearman rank correlation coefficient between CPM-Seq and the gold standard (0.871) and the Spearman rank correlation coefficient between Cufflinks and the gold standard (0.275). The scatter plots of the ranks of the CPM-Seq and Cufflinks measurements against those of the gold standard are given in Figure 3.2.

We can see that in terms of Pearson correlation coefficient, CPM-Seq slightly outperforms Cufflinks. However, in terms of Spearman rank correlation coefficient, CPM-Seq outperforms Cufflinks dramatically. We believe that Spearman rank correlation coefficient characterizes the performances of the two different methods much better than the Pearson correlation coefficient. The relatively high Pearson correlation coefficient between the Cufflinks measurements and the gold standard is attributed to one influential case, which is plotted as a red diamond in the upper right corner of Figure 3.1. After this influential point is removed, the Pearson correlation coefficient between Cufflinks and the gold standard is reduced significantly from 0.665 to -0.059, whereas the Pearson correlation coefficient between CPM-Seq and the gold standard only decreases slightly from 0.871 to 0.809. We also calculated the Spearman correlation coefficients between the CPM-Seq and Cufflinks measurements and the gold standard after the influential point is removed. The resulting Spearman correlation coefficients are 0.842 for CPM-Seq and 0.108 for Cufflinks. The overall superior performance of CPM-Seq over Cufflinks is further demonstrated by the strong linear pattern in plot (b) of Figure 3.2 for CPM-Seq, and the lack of linear pattern in plot (a) for Cufflinks. When comparing the 15 transcripts pairwise (105 pairs in total), CPM-Seq ranked 90 pairs correctly, whereas Cufflinks only ranks 63 pairs correctly. This example suggests that Spearman rank correlation coefficient provides a more

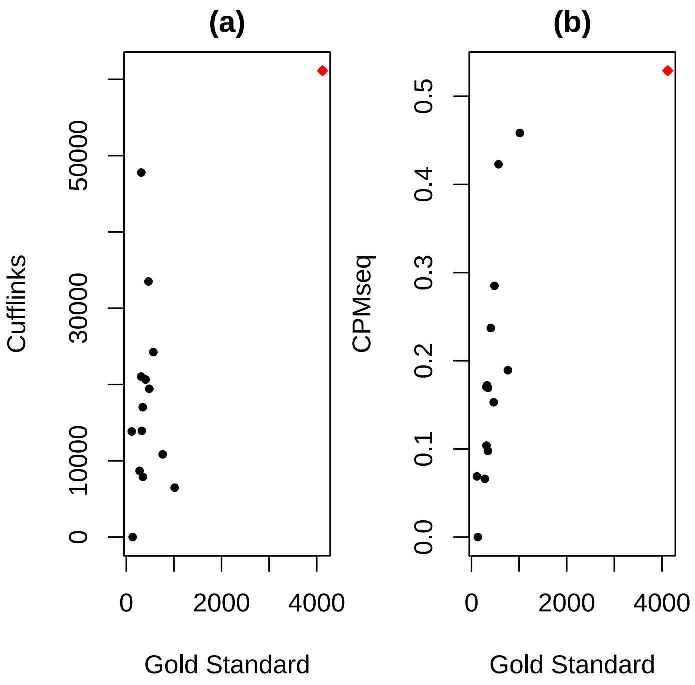

Figure 3.1. Expression level quantification of 15 transcripts using simulated paired-end data. Plot (a) and (b) show Cufflinks measurements and CPM-Seq measurements against the gold standard, respectively.

reliable measure of the performance of a quantification method, and thus we will use it in the other examples in the rest of the paper.

Example 2 We also conducted a large scale simulation study with ten replicated runs. In each run, the Flux simulator randomly assigns expression levels to all isoforms of human chromosome 1 in refseq hg 18, and generates three million 75 bp paired-end reads. Once the reads are obtained, they are mapped back to the reference genome using Tophat [6]. We filtered out genes that have received in total less than 20 reads, and genes that received more than 60 reads at least at one base pair. In the first run, there were 987 genes left after filtering. Among these genes, 710 genes have single isoform, 156 genes have two isoforms, 74 genes have three isoforms, 27 genes have four isoforms, and 27 genes have five isoforms. We applied CPM-Seq and

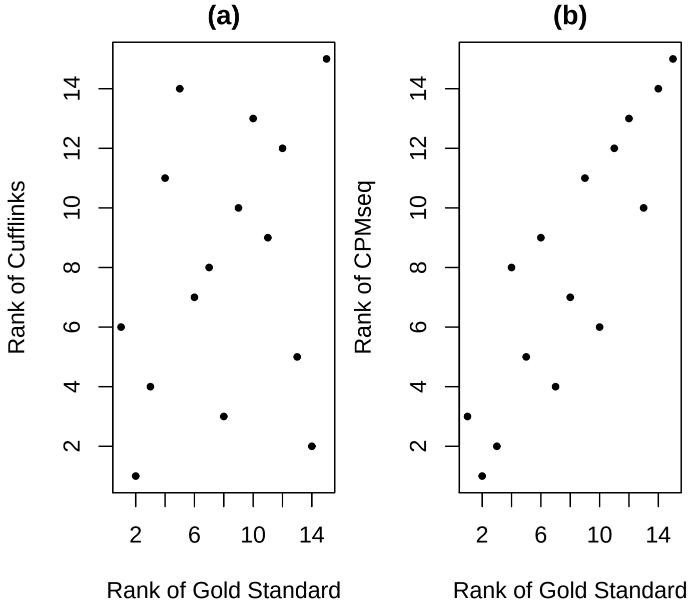

Figure 3.2. Rank of expression level quantification of 15 transcripts using simulated paired-end data. Plot (a) and (b) show rank of Cufflinks measurements and rank of CPM-Seq measurements against the rank of the gold standard, respectively (The Spearman rank correlation coefficients are 0.275 vs. 0.871)

Table 3.1
Spearman rank correlation coefficients for CPM-Seq and Cufflinks of ten simulation replicates

	CPM-Seq with Gold	Cufflinks with Gold	CPM-Seq with Cufflinks
run 1	0.616	0.538	0.498
run 2	0.551	0.540	0.436
run 3	0.606	0.550	0.471
run 4	0.612	0.555	0.464
run 5	0.589	0.548	0.511
run 6	0.547	0.527	0.464
run 7	0.603	0.562	0.490
run 8	0.610	0.532	0.449
run 9	0.602	0.569	0.498
run 10	0.594	0.585	0.536

Cufflinks separately to quantify the expression levels of these isoforms and calculated their Spearman rank correlation coefficients with the gold standard. The Spearman rank correlation coefficient for CPM-Seq is 0.616 and the Spearman rank correlation coefficient for Cufflinks is 0.538. Therefore, CPM-Seq outperformed Cufflinks in this run. The simulation results of the other 9 runs are reported in Table 3.1. To compare the performances of CPM-Seq and Cufflinks in all ten runs, we applied the paired t-test, and the resulting p-value is ≤ 0.0001, suggesting a significant improvement of CPM-Seq over Cufflinks.

Real Data Application

As discussed previously, two real datasets are further used to compare CPM-Seq and Cufflinks, and the corresponding results are presented as Example 3 and Example 4 below. Example 3 is based on a small scale study with qRT-PCR measurements, which are used as the gold standard. Example 4 is based on the large scale study that does not have qRT-PCR measurements. Therefore, we do not have a gold standard in Example 4. Instead, we use the characteristics of the read counts data themselves to facilitate the comparison of the two methods.

Example 3 Two human cell lines named MCF7 and HME were studied by Wang *et al.* using RNA-Seq [41]. The resulting data can be downloaded from the NCBI Short Read Archive at http://www.ncbi.nlm.nih.gov/sra under accession number GSE12946. There are 21.6 million 32 bp reads for the MCF7 cell line and the 17.8 million 32 bp sequenced reads for the HME cell line. Using bowtie, we mapped the reads to the ucsc hg18 reference genome and obtained the base level read counts data for both cell lines [7]. We applied CPM-Seq and Cufflinks to quantify the transcripts' expression levels.

The original study of Wang *et al.* did not provide the qRT-PCR measurements of the transcripts. Fortunately, Kim *et al.* [42] used the qRT-PCR technology to measure eight transcripts of four genes of these two cell lines in a separate study. We used these qRT-PCR measurements as the gold standard to compare the performances of CPM-Seq and Cufflinks. The eight transcripts and their qRT-PCR, CPM-Seq and Cufflinks measurements are reported in Table 3.2 and the Spearman rank correlation coefficients between the gold standard and the two quantification methods are calculated and reported in Table 3.3. We can see that CPM-Seq achieves higher correlation with the gold standard than Cufflinks in both cell lines.

Example 4 In this example, we will see that overall, CPM-Seq is concordant with Cufflinks, but their quantification results can be quite different from each other for some genes. We analyzed a RNA-Seq data of the Human Brain Reference RNA (Brain) sample, which was originally generated by Wong's lab using the Illumina Genome Analyzer platform [43]. We processed one lane of eight millions 50 bp paired-end reads. The data set can be downloaded from NCBI Short Read Archive

Table 3.2
Expression level of 8 transcripts of HME and MCF7 cell lines

	HME			MCF7		
transcript ID	qRT-PCR	CPM-Seq	Cufflinks	qRT-PCR	CPM-Seq	Cufflinks
uc002cvs.1	423.5	0.9771	1.3396	595.4	1.7365	2.7964
uc002cvt.2	234.7	1.4227	33.6959	302.3	1.5082	51.6319
uc002qlp.1	277.9	1.1251	7.8938	381.3	2.0921	17.1806
uc002qlq.1	621.8	1.3131	18.9976	755.9	2.7526	44.7393
uc002xmo.1	8.1	0.0019	0.1141	189.6	1.6722	7.7976
uc002xmn.1	10.7	0.0039	0.2860	530.0	3.8728	18.7706
uc003ngr.1	12.4	0.8350	14.3832	317.8	3.9782	125.3420
uc003ngs.1	538.2	0.0472	1.6722	19207.9	4.4089	45.9333

Table 3.3
Spearman rank correlation for 8 transcripts of HME and MCF7 cell lines

	CPM-Seq	Cufflinks
HME	0.571	0.476
MCF7	0.619	-0.024

(SRA) at http://www.ncbi.nlm.nih.gov/sra under the accession numbers GS475204 and GSM475205 [43]. Tophat was used to map the reads to refseq hg18 [6]. We filtered out genes that have received in total less than 20 reads, genes that have received more than 60 reads at least at one base pairs, and genes with more than 5 exons. After filtering, 433 genes on chromosome 1 are left and these genes contain 743 isoforms. Among the 433 genes, there are 277 single isoform genes, 87 two-isoform genes, 51 three-isoform genes, and 18 four-isoform genes. Because these multi-isoform genes contain many sub-exons, it is not possible to observe every type of junction reads even if all of the junctions are expressed. Therefore, we used the composite likelihood, which includes all non-junction reads, and the positive junction reads. We applied CPM-Seq and Cufflinks to quantify the expression level of each transcript. The Spearman rank correlation coefficient between CPM-Seq and Cufflinks in this example was calculated to be 0.589, which shows that overall CPM-Seq has a good concordance with Cufflinks in this example. Despite their general concordance, the quantification results of CPM-Seq and Cufflinks are different for a large number of genes. A more careful comparison between the CPM-Seq and Cufflinks measurements of these genes indicates that the CPM-Seq measurements are more reasonable. We give such an example below.

According to human refseq hg18, gene ZNF238 contains two exons, which we denote as e'_1 and e'_2, and it has two annotated transcripts labeled as NM205768 and NM006352. Transcript NM205768 consists of e'_1 and a part of e'_2, and transcript NM006352 consists of the entire e'_2. In order to make the transcripts either contain or skip an exon entirely, we split exon e'_2 into two sub-exons denoted as e_2 and e_3. We re-denote exon e'_1 as e_1. Therefore, exons e_1, e_2 and e_3 form a partition of the exonic region of gene ZNF238. The total length of gene ZNF238's exonic region is 4387, and the exonic base pairs are indexed as $1, \ldots, 4387$. Exons e_1, e_2, and e_3 contain base pairs $\{1, \ldots, 187\}$, $\{188, \ldots, 695\}$, $\{696, \ldots, 4387\}$, respectively. NM205768 consists of e_1 and e_3, and NM006352 consists of e_2 and e_3. We re-label the two transcripts NM205768 and NM006352 as T_1 and T_2, respectively, and assume that they form the collection of candidate transcripts, that is, $\mathcal{T} = \{T_1, T_2\}$.

As discussed previously, not all junction reads will be observed due to insufficient coverage or technological limitations of RNA-Seq experiment. In the 50 bp paired-end reads data generated by Wong's lab, we only observed five types of reads for gene ZNF238. The frequency of each type of reads is summarized in Table 3.4, and the base level counts are plotted in Figure 3.3.

We applied CPM-Seq to quantify the expression levels of T_1 and T_2, and the MCLEs of the model parameters and quantification results are reported in Table 3.5. The parameter estimates indicate that both T_1 and T_2 are expressed and the

Table 3.4
Frequency table for each type of reads for gene ZNF238

type	r_{11}	r_{13}	r_{22}	r_{23}	r_{33}
counts	2	8	4	2	1313

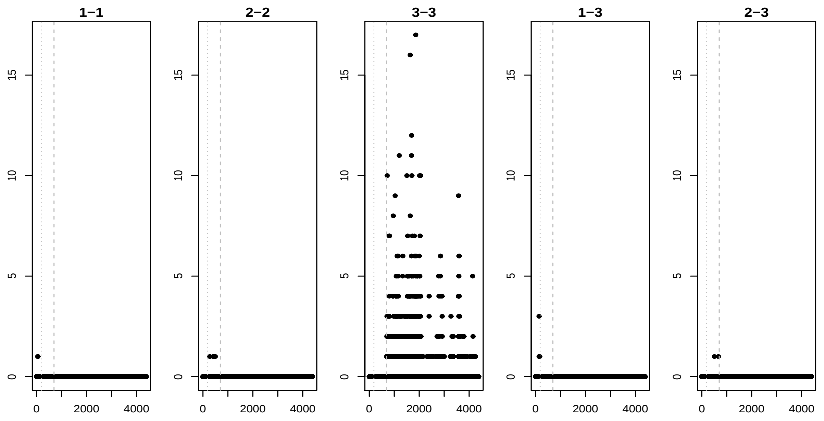

Figure 3.3. Base pair level counts for each type of reads in ZNF238

expression level of T_1 (0.741) is higher than that of T_2 (0.296). We also applied Cufflinks to quantify these two transcripts and the quantification results are also presented in Table 3.5. Cufflinks quantified the expression level of T_1 to be 11.653, whereas it quantified the expression level of T_2 to be 4.128e-05, which is almost zero. It appears that Cufflinks suggests that T_1 was expressed but not T_2. Therefore, for gene ZNF238, CPM-Seq and Cufflinks gave different quantification results.

Recall that $T_1 = e_1e_3$, $T_2 = e_2e_3$, and e_1, e_2, and e_3 are 187 bp, 507 bp, and 3691 bp long in lengths, respectively. It is clear e_3 is the longest among the three exons, followed by e_2 and then e_1, and e_3 is actually much longer than e_2 and e_1. From Table 3.4, e_3 received the majority of the reads mapped to gene ZNF238 (1313 out of 1329 reads). These reads are of type r_{33}. Because both T_1 and T_2 contain e_3, both transcripts can give rise to reads of type r_{33}, and they cannot be directly allocated to the transcripts. Reads of types r_{11} and r_{13} (10 in total) suggest the expression of T_1, whereas reads of type r_{22} and r_{23} (6 in total) suggest the expression of T_2. However, the counts of these types of reads are relatively small compared to the count of reads of type r_{33}, due to the short lengths of e_1 and e_2. We believe that this was the reason that Cufflinks was not able to separate the two transcripts and instead allocated all reads of type r_{33} to T_1. On the other hand, CPM-Seq was able to infer the expression levels of T_1 and T_2 using the convolution model that models the read count at each base pair. In other words, CPM-Seq successfully identified the two transcripts in this example.

Table 3.5
Fitting a real example

Transcript	$\hat{\lambda}$	$\hat{p}$	CPM-Seq	Cufflinks
T_1	$\hat{\lambda}_{11} = 0.040$ $\hat{\lambda}_{12} = 1.978$	$\hat{p}_{21} = 0.898$ $\hat{p}_{12} = 0.102$	0.741	11.653
T_2	$\hat{\lambda}_{21} = 0.040$ $\hat{\lambda}_{22} = 7.998$	$\hat{p}_{21} = 0.993$ $\hat{p}_{22} = 0.007$	0.296	≈ 0 (4.128e-05)

4. TRANSCRIPTS IDENTIFICATION

One of the advantages of the RNA-Seq technology is its ability to identify alternatively spliced events, because RNA-Seq technology offers single base pair resolution rather than intensities, compared to the traditional microarray technology. In some cases, if the sequenced fragments only cover an exon-exon junction that is unique in a particular transcript, direct inference about this transcript can be made, and existing statistical models, as discussed in previous chapters, can be used to normalize the transcript expression levels. However, a large number of fragments cannot be uniquely mapped to transcripts because of alternative splicing, an event that preserves and skips exons in different transcripts. As a result, when the transcripts are fragmented, multiple transcripts can give rise to identical reads. In other words, the labels of the transcripts that identical reads are generated from are missing. Therefore, the identification of transcripts is an indirect problem that requires the use of proper statistical models.

4.1 Introduction

A number of methods have been proposed for solving the transcript identification problem in the literature. These methods usually fall into one of the following two categories, which are graph-based assembly methods and model-based assembly methods, respectively. One typical graph-based assembly method is Cufflinks. First, Cufflinks defines compatibility between reads. Two reads are incompatible if they cannot be originated from the same transcript. Second, based on the starting position of the reads, Cufflinks defined a partial order P for all the reads. Third, Cufflinks finds a partition of P into chains, such that each chain represents a transcript and compatible reads are generated from the same chain. Then Cufflinks assembles a set of candidate transcripts based on graph theory. According to the Dilworth's theorem, Trapnell *et al.* showed that the number of transcripts that Cufflinks assembles is always equal to the number of incompatible reads. Transcripts for different genes are assembled independently, and then Cufflinks uses a probabilistic statement to model the generating scheme of each read. The probabilistic model involves the expression levels of the transcripts that can produce this read. The graph-based assembly method, such as Cufflinks, only uses the compatibility to assemble transcripts. They fail to use the information contained in the read counts data. On the other hand, model-based assembly methods use statistical models to quantify the expression levels of all possible candidate transcripts, and declare those highly expressed as identified transcripts. SLIDE is a typical example of model-based assembly methods [37]. SLIDE considers all possible candidate transcripts and fits a linear model to the bin counts. However, for some genes, the number of bins is less than the number of candidate transcripts. Thus, the linear regression framework may be non-identifiable. To address this issue, SLIDE proposed to impose a penalty on the transcript abundance parameters so that transcripts with low abundance will be shrunken to zero. After fitting the penalized regression, SLIDE declares those transcripts whose abundances are greater than zero as expressed transcripts.

Besides the two methods mentioned above, there are other models proposed in the literature to address the transcript identification problem. For example, IsoLasso [36] uses graph theory to assemble candidate transcripts and applies Lasso to regress read counts in segments against the transcript expression levels. However, most of these methods use improper observational units. As a result, the model itself is non-identifiable. To overcome these issues, we propose to use penalized convolution of Poisson mixture models to identify highly expressed transcripts.

4.2 Lasso

In practice, linear regression is one of the most popularly used statistical methods. In regression models, the response variable is assumed to be a linear function of the predictors. Let n and p denote the number of observations and the number of predictor variables, respectively. Traditionally, the linear model can only deal with the case where p is small and n is large. However, this is not in general the case in high dimensional problems. When the number of variables p is large, it is often desirable to assume that there exists a sparsity representation of the model. The identification of the sparse representation is usually done by variable selection procedures such as forward, backward, or best subset selection procedures.

Forward and Backward Selection

Forward selection method starts will the null model that includes none of the predictor variables. Based on the variables that have been selected in the current model, the variable that can most improve the model fitting is added to the model. The procedure stops if there is no significant improvement to the model fitting. Backward selection procedure works in a reverse order, by eliminating variables one at a time until all kept variables are significant. Forward and backward selection procedures are easy to understand, but are known to have inconsistent results. It is difficult to characterize the statistical properties of these procedures as well, such as their consistency in variable selection. These procedures are also known to have instability issues, that is a tiny change in the data can have a big impact on the variable selection results.

Best Subset Method

Best subset methods fit the linear regression model to all possible subsets of predictors and use information based criterion such as Akaike Information Criterion (AIC) to select the one that best fits the model. AIC score is defined as

$$-2l(\hat{\beta}|y) + 2 \cdot p, \tag{4.1}$$

where $l(\hat{\beta}|y)$ is the log-likelihood function and p is the number of parameters in the model. The term $2p$ can be viewed as a penalty on the number of parameters that are needed to fit the data. More parameters would fit the data better and would result in a bigger penalty. Thus the AIC achieves a balance between the likelihood and the size of the model. To compare different models, their AIC scores are compared and the model with the smallest AIC score is chosen. Bayesian information criterion (BIC) can be also used in this case.

Best subset methods also have their own weaknesses. When the number of variables becomes large, the number of possible subsets grows exponentially. Thus these methods usually do not work well when p is large. Nowadays, as is often the case that the number of the parameters of interest can be much larger than the number of observations, variable selection becomes extremely important. One such example is gene expression study, in which thousands of genes are simultaneously measured in only a few samples. The purpose is to identify a subset of significant genes that contribute to the sample differences. The large p small n problem can also come from single nucleotide polymorphism (SNP) study, in which millions of SNPs are included in the study and their associations with the only a few phenotypes are studied to determine a small number of SNPs that affect the phenotype. Based on the parsimonious assumption, Tibshirani proposed Least Absolute Shrinkage and Selection Operator (Lasso) that minimizes the residual sum of squares subject to a L_1 penalty on all the coefficients [44]. Lasso estimators can be solved computationally using convex optimization methods, or the entire regularization path can be computed efficiently using Least Angle Regression (Lars) algorithm [45]. These computational methods provide a feasible way to select a small number of variables that best explain the data.

4.2.1 Lasso Solution

Consider the usual linear regression setup. Each observation is related to p predictors through a linear relationship as

$$y = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2 I), \tag{4.2}$$

where $y \in \mathbb{R}^n$ denotes the vector of observed responses, $X \in \mathbb{R}^{n \times p}$, $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$ is a vector of i.i.d. random errors with mean 0 and variance σ^2, and $\beta \in \mathbb{R}^P$ is the vector of unknown coefficients. When $p \leq n$ and X is full column rank, the ordinary least squares solution exists, which can be written as

$$\hat{\beta}^{ols} = (X'X)^{-1}X'Y. \tag{4.3}$$

In this particular case, the ordinary least squares solution also corresponds to the maximum likelihood estimates of β under the normality assumption of ϵ. However, the ordinary least squares estimate cannot perform variable selection because the estimate of all coefficients will be nonzero. Additionally, when $p \geq n$, the ordinary least squares estimate of β does not exist. To address these problems, Tibshirani [44] proposed the Lasso estimator as follows:

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^P} \frac{1}{2} ||y - X\beta||_2^2 + \gamma ||\beta||_1, \tag{4.4}$$

where γ is a non-negative tuning parameter, $|| \cdot ||_2$ represents the L_2 norm, and $||\beta||_1$ stands for the L_1 norm of the coefficients vector β, which is the sum of the absolute values of the components in β. The scale of γ controls the penalty on each individual parameter of β. The model is assumed to be sparse, that is, only a small number of coefficients of β are nonzero, while the other variables are not related to the response. If γ is set to 0, the Lasso reduces to the ordinary least squares problem. On the other hand, a very large value of γ will completely shrink $\hat{\beta}$ to 0. In practice, a moderate level of γ needs to be determined so that variables with zero coefficients are shrunken to zero, while variables with large coefficients are kept. The shrinkage approach trades off bias for variance, and a parsimonious model can be obtained for easier interpretations. It can be shown that the Lasso framework is equivalent to the solution of the following optimization problem:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^P} \frac{1}{2}||y - X\beta||^2, \tag{4.5}$$

subject to $\sum_{i=1}^{p} |\beta_i| \leq t$. In equation (4.5), Lasso is framed into a constrained convex optimization problem. The two problems are equivalent in the sense that for any given value of γ, there exists a t, such that the two problems have identical solutions.

4.2.2 Variable Selection Consistency of Lasso

In this section, we review the variable selection consistency result of the Lasso. Zhao and Yu studied the necessary and sufficient conditions for Lasso to select the true model [46]. Based on their findings, a single condition, called the irrepresentable condition is proved to be sufficient and almost necessary for Lasso to achieve model selection consistency.

They first defined the sign consistency of $\hat{\beta}$ as follows. If there exists γ, such that, $\lim_{n\to\infty} P(\hat{\beta}(\gamma) =_s \beta) = 1$, where $\hat{\beta} =_s \beta$ means $sign(\hat{\beta}) = sign(\beta)$ for each component, then $\hat{\beta}$ is said to be sign consistent.

Let $\mathcal{T}_0$ be the set of variables with nonzero coefficients, and $\mathcal{T}_0^c$ be the set of variables with zero coefficients. Without loss of generality, we can write the vector of all coefficients in β as two parts, that is, $\beta = (\beta_{\mathcal{T}_0}, \beta_{\mathcal{T}_0^c})^T$, where $\beta_{\mathcal{T}_0}$ denotes the coefficients of variables that are in $\mathcal{T}_0$, and $\beta_{\mathcal{T}_0^c}$ denotes the coefficients of variables that are in $\mathcal{T}_0^c$. Suppose $|\mathcal{T}_0| = q$. Now let $X(1)$ and $X(2)$ represent the first q and last $p-q$ columns of the design matrix X, and let $C = \frac{1}{n}X^T X$. Furthermore, we can express C in a block-wise form as follows

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}. \tag{4.6}$$

Assume C_{11} is invertible, the irrepresentable condition holds if there exists a positive constant vector η such that

$$|C_{21}(C_{11})^{-1}sign(\beta_{\mathcal{T}_0})| \leq 1 - \eta, \tag{4.7}$$

where 1 is a vector of $p - q$ 1's and the equation holds element-wise. In other words, the variables that are not in the true model cannot be represented by variables that are in the true model.

Zhao and Yu further showed that under certain regularity conditions and strong irrepresentable condition, the Lasso solution is strongly sign consistent, that is $P(\hat{\beta} =_s \beta) = 1 - o(e^{-n^c})$, where $0 \leq c < 1$. This implies that if strong irrepresentable condition holds, then the probability that Lasso selects the true model approaches 1 at an exponential rate.

4.2.3 Estimation Consistency of Lasso

In this section, we consider the estimation consistency of Lasso estimate, when the irrepresentable condition is violated. In practice, it might be difficult to verify if the irrepresentable condition holds. Meinschausen and Yu have shown that when the irrepresentable condition does not hold, the Lasso solution will not be sign consistent [47].

To study the convergence property of the Lasso estimate when the irrepresentable condition does not hold, Meinschausen and Yu [47] demonstrated that the Lasso estimate can achieve L_2 consistency. Additionally, if a two-step hard thresholding procedure is used, the Lasso estimate can also achieve the sign consistency.

An estimate of β, denoted by $\hat{\beta}$, is L_2 consistent if

$$||\hat{\beta} - \beta||_2 \to 0 \text{ as } n \to \infty. \tag{4.8}$$

The L_2 consistency is an attractive property, because it shows that when the irrepresentable condition is violated, the Lasso estimate will not be identical in sign to the true variables, but Lasso will still manage to select the true variables together with a limited number of untrue variables and the estimated coefficients of these untrue variables will be small. In other words, it will at least do a good job and choose important variables with high probability and unimportant ones with only small coefficients. Meinschausen and Yu further showed that in order to achieve the sign consistency, hard-thresholding rule can be applied to the estimated coefficients.

4.2.4 Uniqueness of Lasso solution

For any y, X, and $\gamma \geq 0$, the Lasso estimate has the following properties:

(i). There is either a unique Lasso solution or an infinite number of solutions.

(ii). Every Lasso solution $\hat{\beta}$ gives the same fitted value $X\hat{\beta}$.

(iii). If the regularization parameter $\gamma > 0$, then every lasso solutions $\hat{\beta}(\gamma)$ has the same L_1 norm $||\hat{\beta}(\gamma)||_1$.

Besides these properties, Tibshirani [48] further showed that if each column of the design matrix $X \in \mathbb{R}^{n \times p}$ is drawn from a continuous distribution, then for any y, and γ, the lasso solution is unique. In order to show the uniqueness of the Lasso solution, Ryan first defined the equicorrelation set ε by

$$\varepsilon = \{i \in (1, \dots, p) : \ X_i^T(y - X\hat{\beta})| = \gamma\}, \tag{4.9}$$

and the equicorrelation sign s by

$$s = sign(X_\varepsilon^T(y - X\hat{\beta})). \tag{4.10}$$

For any y, X, and γ, if $null(X_\varepsilon) = 0$, or equivalently, if $rank(X_\varepsilon) = |\varepsilon|$ then the Lasso solution is unique, and is given by

$$\begin{cases} \hat{\beta}_{-\varepsilon} = 0 \\ \hat{\beta}_\varepsilon = (X_\varepsilon^T X_\varepsilon)^{-1}(X_\varepsilon^T y - \gamma s), \end{cases} \tag{4.11}$$

where the ε and s are the equicorrelation set and sign defined previously.

4.2.5 KKT Condition

As discussed before, Lasso regression is equivalent to an optimization problem with inequality constraints. Typically, when dealing with optimizations with equality and inequality constraints, Karush-Kuhn-Tucker (KKT) condition is used as the first order necessary condition for a solution to be optimal. In general, if the objective is to

$$\text{minimize } f_0 \tag{4.12}$$

$$\text{subject to } f_i(x) \leq 0, i = 1, \ldots, m, \tag{4.13}$$

$$h_i(x) = 0, i = 1, \ldots, p, \tag{4.14}$$

then according to the KKT condition, if x^*, λ^*, and ν^* are optimal, they must satisfy the following conditions, which are

$$f_i(x^*) \leq 0, \tag{4.15}$$

$$h_i(x^*) = 0, \tag{4.16}$$

$$\lambda_i^* \geq 0, \tag{4.17}$$

$$\lambda_i^* f_i(x^*) = 0, \tag{4.18}$$

$$\nabla f_0(x^*) + \sum_i \lambda_i^* \nabla f_i(x^*) + \sum_i \nu_i^* \nabla h_i(x^*) = 0. \tag{4.19}$$

For the Lasso problem, our objective function is

$$obj = \frac{1}{2}||y - X\beta||_2^2 + \gamma||\beta||_1. \tag{4.20}$$

Applying the KKT condition to the Lasso problem, we know that a minimizer $\hat{\beta}(\gamma)$ of the objective function in equation (4.20) has to satisfy either one of the following two conditions. In the first case, if the coefficient of ith variable $\hat{\beta}_i(\gamma) \neq 0$, then the ordinary first partial derivative of the objective function with respect to β_i at $\hat{\beta}(\gamma)$ has to be zero, that is,

$$\frac{\partial obj}{\partial \beta_i(\gamma)}|_{\beta=\hat{\beta}(\gamma)} = -X_i^T(y - X\beta) + \gamma \cdot sign(\beta_i)|_{\beta=\hat{\beta}(\gamma)} = 0. \tag{4.21}$$

In the second case, if $\hat{\beta}_i(\gamma) = 0$, the subdifferential at $\beta_i(\gamma)$ has to include the zero element, i.e.,

$$-X_i^T(y - X\beta) + \gamma \cdot e \text{ for some } e \in [-1, 1] = 0, \tag{4.22}$$

which is equivalent to $|X_i^T(y - X\beta)| \leq \gamma$.

4.2.6 Algorithm to Obtain Lasso Solution

Convex Optimizer

As is mentioned before, in regression settings, the Lasso problem is equivalent to a convex optimization problem, thus the usual convex optimizer can be used to solve the Lasso problem.

Coordinate Descent

Coordinate descent algorithm is a type of optimization algorithm that uses different coordinate direction cyclically to obtain the optimal solution. Tseng [49] established the converge of the coordinate descent algorithm for a general case where the objective function has the following special form,

$$f(\beta_1, \ldots, \beta_N) = f_0(\beta_1, \ldots, \beta_N) + \sum_{k=1}^{N} f_k(\beta_k), \tag{4.23}$$

where f_0 is differentiable and convex function. A key to the convergence result of using coordinate descent algorithm to solve $f(\beta_1, \ldots, \beta_N)$ in equation (4.23) is the separability of the penalty function $\sum\limits_{k=1}^{N} f_k(\beta_k)$, which is a sum function of each individual parameter. Tseng further showed that each parameter estimate generated by the coordinate descent method is a stationary point of the f.

For the Lasso problem, in order to derive the coordinate descent algorithm to obtain the coefficient estimates, we can write the objective function as

$$f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \gamma \sum_{j=1}^{p} |\beta_j|. \tag{4.24}$$

Then we can re-write it as

$$f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{k\neq j} x_{ik}\beta_k - x_{ij}\beta_j)^2 + \gamma \sum_{k=j} |\beta_k| + \gamma|\beta_j|. \tag{4.25}$$

We minimize $f(\beta)$ with respect to β_j, while keeping all the other β_k's fixed, for $k \neq j$. The updating rule for β_j can be written as

$$\hat{\beta}_j(\gamma) \leftarrow S\bigg(\sum_{i=1}^{n} x_{ij}(y_i - y_i^{(j)}), \gamma\bigg), \tag{4.26}$$

where $y_i^{(j)} = \sum_{k\neq j} x_{ik}\beta_k(\lambda)$ and $S(t, \gamma) = sign(t)(|t - \gamma|_+)$. The algorithm cycled through $j = 1, 2, \ldots, p$ until convergence criterion is met.

Lars Algorithm

Lars algorithm starts with $\gamma = \infty$ and with $\hat{\beta} = 0$ and then decrease the value of γ. At the kth iteration, under γ_k, Lars algorithm performs the following four steps:

(i). Compute the Lars solution by least squares, given γ_k.

(ii). Compute γ_{k+1}^{join}, when a variable outside the active set joins the active set.

(iii). Compute γ_{k+1}^{cross}, when a variable in the active set crosses zero.

(iv). Set $\gamma_{k+1} = \max(\gamma_{k+1}^{join}, \gamma_{k+1}^{cross})$. If $\gamma_{k+1} = \gamma_{k+1}^{join}$, the Lars algorithm adds a variable to the active set. However, if $\gamma_{k+1} = \gamma_{k+1}^{cross}$, Lars remove the crossing variable from the active set.

General Path-Following Algorithm

As is shown in the above section, Lars algorithm takes the advantage of the piecewise linear coefficient path property, and used it to obtain the entire solution path.

Rosset and Zhu [50] studied the general path solution for optimization problem. Borrowing the idea of Lars algorithm, they applied the path following algorithm to the general penalized likelihood function to obtain the solutions along the entire regularization path. In order to demonstrate how path-following algorithm works, we assume that a general setting for the penalized likelihood approach is to minimize the following objective function

$$obj(\beta) = -L(y, X\beta) + \gamma J(\beta). \tag{4.27}$$

Equivalently, it can be written as

$$\hat{\beta}(\gamma) = \arg\min_{\mathbb{R}^P} \Big\{ - L(\beta(\gamma)) + \gamma J(\beta(\gamma)) \Big\} \tag{4.28}$$

where $\gamma \geq 0$ is the penalty parameter on the coefficients. If $\gamma = 0$, there is no regularization and $obj(\beta)$ reduces to the negative of the original likelihood $L(y, X\beta)$. We can also see that in equation (4.27), if $\gamma \to \infty$, $\hat{\beta} \to \mathbf{0}$.

Taking the first order partial derivative of the objective function $obj(\beta)$ with respect to β, we get

$$f(\beta) = H(\beta(\gamma), \gamma) \equiv \frac{\partial obj(\beta)}{\partial \beta} = -\frac{\partial L(\beta(\gamma))}{\partial \beta} + \gamma \frac{\partial J(\beta)}{\partial \beta}. \tag{4.29}$$

For any $\hat{\beta}$ to be optimal, $f(\hat{\beta})$ has to be 0. Then we have

$$f'(\beta) = \frac{\partial H}{\partial \beta}\frac{\partial \beta}{\partial \gamma} + \frac{\partial H}{\partial \gamma} = 0. \tag{4.30}$$

We can solve $\frac{\partial \beta}{\partial \gamma}$ from equation (4.30) as

$$\frac{\partial \beta}{\partial \gamma} = -(\frac{\partial H}{\partial \beta})^{-1}\frac{\partial H}{\partial \gamma}. \tag{4.31}$$

From equation (4.29), we know that $\frac{\partial H}{\partial \beta} = -\frac{\partial^2 L(\beta(\gamma))}{\partial \beta^2} + \gamma\frac{\partial^2 J(\beta)}{\partial \beta^2}$. Thus plugging it into equation (4.31), we can get

$$\frac{\partial \beta}{\partial \gamma} = -\big(- \frac{\partial^2 L}{\partial \beta^2} + \gamma\frac{\partial^2 J}{\partial \beta^2}\big)^{-1}\frac{\partial H}{\partial \gamma}. \tag{4.32}$$

We can see that a necessary and sufficient condition for the solution path to be linear in γ, when both L and J are twice differentiable, is that equation (4.31) needs to be proportional to a constant vector in $\mathbb{R}^p$. Generally speaking, when $\frac{\partial \beta}{\partial \gamma}$ is not proportional to a constant vector, the solution path will not be piecewise linear. Thus, given the current regularization γ_k, it is difficult to determine the exact step length in the path algorithm to reach the next regularization γ_{k+1}, under which the active set changes. To obtain the coefficient estimate $\hat{\beta}(\gamma_{k+1})$ from $\hat{\beta}(\gamma_k)$, the predictor-corrector method will be used. The predictor-corrector method is used to approximate the corresponding change in β with the decrease in γ. The predictor-step estimates $\hat{\beta}^{k\star}$, which is the predicted value of $\hat{\beta}^{k+1}$. The corrector-step finds the exact solution of $\hat{\beta}^{k+1}$ that corresponds to γ_{k+1} using $\hat{\beta}^{k\star}$ as the initial values. At $k+1$th iteration, the path algorithm works as follows:

1. Determine step length: given current regularization γ_k, determine the next regularization as γ_{k+1}.

2. Predictor step: approximate corresponding change in β with the decrease in γ and denote the predicted value of β as $\hat{\beta}^{k\star}$.

3. Corrector step: calculate the exact $\hat{\beta}^{k+1}$ using $\hat{\beta}^{k\star}$ as the initial value.

4. Modify the active set according to KKT conditions given below, which either add or drop a variable from the active set.

$$\begin{cases} |\frac{\partial L(\hat{\beta}(\gamma))}{\partial \beta}| \le \gamma & \text{if } \hat{\beta}_j = 0, \\ \frac{\partial L(\hat{\beta}(\gamma))}{\partial \beta} = -sign(\hat{\beta}_j)\gamma & \text{if } \hat{\beta}_j \ne 0. \end{cases} \tag{4.33}$$

Adding a variable: for any variable in the non-active set $\mathcal{A}^c$, this variable should join the active set $\mathcal{A}$, if $|\frac{\partial L(\hat{\beta}(\gamma))}{\partial \beta}| \le \gamma$ stops to hold.

Deleting a variable: for any variable in the active set, if $\frac{\partial L(\hat{\beta}(\gamma))}{\partial \beta} = -sign(\hat{\beta}_j)\gamma$ stops to hold.

4.3 Other Penalty Functions and the Oracle Property

4.3.1 SCAD Penalty and Oracle Property

Fan and Li [51] introduced the concept of the model selection oracle property. Let $Q(\beta)$ be the penalized likelihood function $-L(\beta) + \sum_j p_{\gamma_n}(|\beta_j|)$. An estimator $\hat{\beta}$ is said to have the oracle property if:

(1). $P(\hat{\beta}_{\mathcal{T}_0^c} = 0) \to 1$ as $n \to \infty$, where $\mathcal{T}_0^c$ is the set of indices of the true zero variables;

(2). $\hat{\beta}_{\mathcal{T}_0}$ achieves an information bound mimicking the oracle estimator. In other words, the penalized method performs as well as the oracle estimator, which knows in advance that $\beta_{\mathcal{T}_0^c} = 0$. Besides, if the regularization parameter is properly chosen, the estimator $\hat{\beta}_{\mathcal{T}_0}$ is asymptotically normally distributed with covariance matrix I_1^{-1}, where I_1 is the fisher information knowing $\beta_{\mathcal{T}_0^c} = 0$.

They further established the conditions that an estimator can enjoy oracle property. Assume

$$\lim_{n\to\infty} \inf \lim_{\beta\to 0+} \inf p'_{\gamma_n}(\beta)/\gamma_n > 0. \tag{4.34}$$

Under certain regularity conditions, a local root-n consistent estimator $\hat{\beta}$ will have the oracle property if

$$\begin{cases} \gamma \to 0, \\ \sqrt{n}\gamma \to \infty \text{ as } n \to \infty. \end{cases} \tag{4.35}$$

For the Lasso estimator, it is shown that root-n consistency requires that $\gamma_n \to O_p(n^{-\frac{1}{2}})$. On the other hand, the oracle property requires that $\sqrt{n\gamma_n} \to \infty$. Thus, Lasso type of penalty is not able to satisfy both of these requirements. Therefore, Lasso does not have oracle property.

They proposed three good properties that any good estimator should have.

1. Unbiasedness: The resulting estimator should be nearly unbiased when the true unknown parameter is large to avoid unnecessary modeling biases.

2. Sparsity: The estimator should automatically set small coefficients to zero to reduce the model complexity.

3. Continuity: The estimator should be continuous in data to avoid instability in model predictions.

Fan and Li further proposed some conditions that a penalty needs to satisfy in order to have the above proposed properties. Let $p_\gamma(\theta) = \gamma\rho(\theta)$ denote the penalty term, where $\theta \in [0, \infty)$. In order for a penalty to satisfy the first conditions, $p'_\gamma(\theta)$ needs to be close to zero when θ is large. The second property requires that $p'_\gamma(0+) > 0$. The third property requires that the function $\theta + p'_\gamma(\theta)$ achieves its minimum at $\theta = 0$.

Based on these criteria, Fan *et al.* proposed the Smoothly Clipped Absolute Deviation (SCAD) Penalty [51]. The penalty function is defined as

$$p'_\gamma(\theta) = \gamma\Big\{I(\theta \leq \gamma) + \frac{(a\gamma - \theta)_+}{(a-1)\gamma} I(\theta > \gamma)\Big\}, \tag{4.36}$$

where $a > 2$ and $\theta > 0$. It can be verified that the SCAD penalty enjoy the above three properties.

4.3.2 Smooth Homotopy between L_0 and L_1 Penalty

Since the oracle property is a desirable property for model selection approaches, many penalties are designed to satisfy the oracle property. The L_0 penalty can recover the sparse variables, however, it is computationally infeasible when the dimension grows large. On the other hand, the L_1 penalty does not satisfy the conditions of oracle property, therefore, the using the L_1 penalty would not always recover the true model. Lv *et al.* proposed a smooth homotopy between L_0 and L_1 penalties as a unified approach as

$$\rho_a(\theta) = \frac{(a+1)\theta}{a+\theta} \text{ for } a \in (0, \infty) \text{ and } t \in [0, \infty). \tag{4.37}$$

We can see that

$$\begin{cases} \rho_0(\theta) = \lim_{a \to 0+} \rho_a(\theta) = I(\theta \neq 0), \\ \rho_\infty(\theta) = \lim_{a \to \infty} \rho_a(\theta) = t. \end{cases} \tag{4.38}$$

Thus, the family of penalty in equation (4.37) can be interpreted as a smooth homotopy between L_0 and L_1 penalties.

4.4 L_1 Penalized Convolution of Poisson Mixture Model

As is explained is the previous chapter that the transcript identification problem is challenging because of the following three reasons. First, the number of candidate transcripts grows exponentially with the increase of the number of exons. Second, because of the limitations of the current technology, fragment distribution is limited to certain ranges. The lack of the long reads makes the identification of long transcripts difficult. For example, the mean of the length of fragments from current Illumina's sequencing technology, such as Genome Analyzer or Hiseq 2000, is typically around 200 base pairs, and the standard deviation is usually around 25 base pairs. The relatively short reads would only span over a few exon-exon junctions, thus it is still not possible to map the reads directly to the transcripts. Third, even though junction reads are available, the label that each read is from is missing. Therefore, each read cannot be directly assigned to a particular transcript.

To address the transcripts identification problem, we proposed the penalized convolution of Poisson mixture models (penCPM-Seq) that incorporates a lasso penalty to the original composite likelihood. The objective function can be written as

$$\begin{aligned} Obj(\theta|\mathbf{y}) &= -\prod_{r \in \mathcal{R}} L^r(\theta|y^r) + \gamma \cdot \sum_{i=1}^{N} |\lambda_{i2}| \\ &= -\prod_{r \in \mathcal{R}} \prod_{m} \prod_{k=i_1}^{i_{N_r}} \left[\sum_{j_k=1}^{2} p_{k,j_k} Poi(\lambda_{k,j_k}) \right] + \gamma \cdot \sum_{i=1}^{N} |\lambda_{i2}|, \end{aligned}$$

and our goal is to

$$\min_{\theta} Obj(\theta|\mathbf{y}),$$

$$\text{subject to} \begin{cases} \lambda_{ij} > 0, \text{ for } 1 \leq i \leq N \text{ and } 1 \leq j \leq 2, \\ \lambda_{11} = \ldots = \lambda_{N1}, \\ p_{i1} + p_{i2} = 1 \text{ for } 1 \leq i \leq N, \end{cases}$$

where λ_{i1} for $1 \leq i \leq N$ is the intensity, which is used to model the background noises. The intensity λ_{i2} for $1 \leq i \leq N$ indicates the expression level of ith transcript. The value of γ controls the amount of regularization. Under different regularization values, different transcripts will be selected in the active set.

4.5 Penalized Convolution of Poisson Mixture Model

4.5.1 Low Dimensions

As discussed above, adding a Lasso penalty to the linear model makes it possible to obtain a sparse representation of the model. When the L_1 penalty is imposed on the intensity parameters in the convolution of Poisson mixture models, both the coordinate descent algorithm and the path following algorithm can be used to obtain the parameter estimates. Once the model parameters are estimated, KKT condition can be used to check if any variables should be added to or deleted from the active set.

We developed an EM-algorithm for the penalized convolution of Poisson mixture models. We kept on using the two-step data generating scheme that is introduced in the previous chapter. For a read of type $r \in \mathcal{R}$ and $m \in E^r$, suppose N_r transcripts can generate this type of read. We further assume that, for each transcript, X^r_{tm}

follows a two-component mixture of Poisson distribution with the probability mass function

$$f\left(X_{tm}^{r}=x|\lambda_t,p_t\right)=\sum_{i=1}^{2}p_{ti}Poi\left(x;\lambda_{ti}\right), \tag{4.39}$$

where $p_t=(p_{t1},p_{t2})'$ is the vector of mixing proportions satisfying $\sum_{i=1}^{2}p_{ti}=1$, $\lambda_t=(\lambda_{t1},\lambda_{t2})'$ is the vector of intensity rates of the two Poisson components, $Poi(x;\lambda_{ti})=(\lambda_{ti})^x\exp(-\lambda_{ti})/x!$, and x is any non-negative integer. Then

$$Y_m^r=X_{i_1m}^r+\ldots+X_{i_{N_r}m}^r \tag{4.40}$$

follows a 2^{N_r}-component mixture of Poisson distribution, and we index the components by $i_1j_{i_1}\ldots i_{N_r}j_{i_{N_r}}$, for $j_{i_k}\in\{1,2\}$ and $k\in\{1,\ldots,N_r\}$. We define membership indicator variables $Z_{m,(i_1j_{i_1}\ldots i_{N_r}j_{i_{N_r}})}^r$ such that

$$\begin{cases}Z_{m,(i_1j_{i_1}\ldots i_{N_r}j_{i_{N_r}})}^r=1\text{ if }Y_m^r\sim Poi\Big(\lambda_{i_1j_{i_1}}+\ldots+\lambda_{i_{N_r}j_{i_{N_r}}}\Big)\\ Z_{m,(i_1j_{i_1}\ldots i_{N_r}j_{i_{N_r}})}^r=0\text{ if }Y_m^r\not\sim Poi\Big(\lambda_{i_1j_{i_1}}+\ldots+\lambda_{i_{N_r}j_{i_{N_r}}}\Big).\end{cases} \tag{4.41}$$

Different from chapter 3, where we used conditional distribution to model the junction reads, in this chapter, for simplicity, we do not distinguish different types of reads. Instead, we treat the junction read types and non-junction read types in the same way. Let $z^r=\{z_{m,(i_1j_{i_1}\ldots i_{N_r}j_{i_{N_r}})}^r, m\in E^r\}$, for $r\in\mathcal{R}$. Let $\mathbf{z}=\{z^r:r\in\mathcal{R}\}$, which is the membership indicator of $\mathbf{y}$. With both $\mathbf{y}$ and $\mathbf{z}$, the penalized likelihood function for θ can be written as

$$l(\theta|\mathbf{y},\mathbf{z})=-\log(L(\theta^r|\mathbf{y},\mathbf{z}))+\gamma\sum_{t=1}^{N}\lambda_{t2} \tag{4.42}$$

$$=-\sum_{r\in\mathcal{R}}l^r(\theta^r|y^r,z^r)+\gamma\sum_{t=1}^{N}\lambda_{t2},$$

where for $r \in \mathcal{R}$,

$$\begin{aligned} l^r(\theta^r|y^r, z^r) &= p(y^r, z^r|\theta^r) \qquad (4.43)\\ &= \sum_{m\in E_r} \Bigg\{ \sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} (z^r_{m,i_1 j_{i_1} \ldots i_{N_r} j_{i_{N_r}}}) \cdot \Bigg[\log(p_{i_1 j_{i_1}} \\ &\ldots p_{i_{N_r} j_{i_{N_r}}}) + \log(Poi(\lambda_{i_1 j_{i_1}} + \ldots + \lambda_{i_{N_r} j_{i_{N_r}}})) \Bigg] \Bigg\}, \end{aligned}$$

$$\text{subject to} \begin{cases} \lambda_{ij} > 0, \text{ for } 1 \le i \le N \text{ and } 1 \le j \le 2, \\ \lambda_{11} = \ldots = \lambda_{N1}, \\ p_{i1} + p_{i2} = 1 \text{ for } 1 \le i \le N. \end{cases}$$

Suppose the current parameter estimate is $\hat{\theta}^{cur} = (\hat{\lambda}^{cur}, \hat{p}^{cur})'$. The E-step is to calculate the expected complete log-likelihood function

$$Q(\theta|\hat{\theta}^{cur}, \mathbf{y}) = E_{\mathbf{z}}\Big[- log(l(\theta|\hat{\theta}^{cur}, \mathbf{y}, \mathbf{z}))\Big], \qquad (4.44)$$

where the expectation is over the conditional distribution of $\mathbf{z}$ given $\hat{\lambda}^{cur}$, $\hat{p}^{cur}$, and $\mathbf{y}$. Notice that $Q(\theta|\hat{\theta}^{cur}, \mathbf{y})$ can also be written as $E[\sum_{r\in\mathcal{R}} l^r(\theta^r|y^r, \hat{\theta}^{cur}, z^r))]$.

In the above EM algorithm, the E-step is to compute

$$\begin{aligned} &E(z^r_{m,i_1 k_{i_1} \ldots i_{N_r} k_{i_{N_r}}} |\hat{\lambda}^{cur}, \hat{p}^{cur}, y^r_m) \qquad (4.45)\\ &= \frac{\hat{p}^{cur}_{i_1 k_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r} k_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1 k_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r} k_{i_{N_r}}})}{\sum_{j_{i_1}} \cdots \sum_{j_{i_{N_r}}} \Big[\hat{p}^{cur}_{i_1 j_{i_1}} \ldots \hat{p}^{cur}_{i_{N_r} j_{i_{N_r}}} Poi(\hat{\lambda}^{cur}_{i_1 j_{i_1}} + \ldots + \hat{\lambda}^{cur}_{i_{N_r} j_{i_{N_r}}}) \Big]}, \end{aligned}$$

The M-step is to maximize Q with respect to λ and p, and the resulting maximizers can be used to update $\hat{\theta}^{cur} = (\hat{\lambda}^{cur}, \hat{p}^{cur})'$. Notice that, in the EM algorithm, the parameters λ and p in $l(\theta|\mathbf{y}, \mathbf{z})$ can be maximized separately. First, we optimize Q with respect to λ. First, let

$$f_0 = \sum_{r\in\mathcal{R}} -l^r(\theta^r|y^r, z^r) + \gamma \sum_{t=1}^{N} \lambda_{t2} \qquad (4.46)$$

Our goal is to minimize f_0, subject to the following constraints

$$\begin{cases} f_1 = -\lambda_{12} < 0 \\ f_2 = -\lambda_{22} < 0 \\ \dots \\ f_N = -\lambda_{N2} < 0 \\ f_{N+1} = -\lambda_{11} < 0 \\ h_1 = \lambda_{11} - \lambda_{21} = 0 \\ h_2 = \lambda_{21} - \lambda_{31} = 0 \\ \dots \\ h_{N-1} = \lambda_{(N-1)1} - \lambda_{N1} = 0. \end{cases} \tag{4.47}$$

Using the KKT condition, necessary condition for $\lambda^\star$ of the penalized likelihood function has to satisfy the following conditions. to be optimal

$$\begin{cases} \nabla f_0(\lambda^\star) + \sum_{i=1}^{N+1} a_i \nabla f_i + \sum_{i=1}^{N-1} b_i \nabla h = 0 & \\ a_i \geq 0 & \\ a_i f_i(\lambda^\star) = 0 & \text{for } 1 \leq i \leq N+1 \\ f_i = -\lambda_{i2} < 0 & \text{for } 1 \leq i \leq N \\ f_{N+1} = -\lambda_{11} < 0 & \\ h_i = \lambda_{i1} - \lambda_{(i+1)1} & \text{for } 1 \leq i \leq N-1 \end{cases} \tag{4.48}$$

Thus we can see the gradient equation of the objective function becomes

$$\begin{cases} \frac{\partial f_0}{\partial \lambda_{11}} - a_{N+1} + b_1 & = 0, \\ \frac{\partial f_0}{\partial \lambda_{i1}} - a_i - b_{i-1} + b_i & = 0 \text{ for } 2 \leq i \leq N-1, \\ \frac{\partial f_0}{\partial \lambda_{N1}} - a_i - b_{i-1} & = 0, \\ \frac{\partial f_0}{\partial \lambda_{i2}} - a_i & = 0 \text{ for } 1 \leq i \leq N. \end{cases} \tag{4.49}$$

The optimization of Q with respect to p is straight forward, and the Lagrange multiplier can be used to solve the optimization problem with equality constraints.

Theoretically, it is possible to write out the gradient function $\frac{\partial f_0}{\partial \lambda}$ for each gene. However, it is impractical to write out gradient function and hessian matrix for each gene model. Thus, in the current EM algorithm for low dimensional case, numeric methods, such as finite differencing, are used to approximate the gradient function.

We applied coordinate descent algorithm and path-following algorithm to the penalized likelihood function to obtain the parameter estimates. It is worth pointing out that the likelihood function has multiple local optimum values. Thus we fed multiple starting values to mitigate this problem.

For the path solution, we can see from equation (4.30) that the path is no longer piecewise linear, because $\frac{\partial \lambda}{\partial \gamma}$ is no longer proportional to a constant vector. Thus we need to solve equation $\frac{\partial \lambda}{\partial \gamma} = 0$ to get the updates of λ. Runge Kutta method is used to solve the solutions of system of differential equations. However, the approximation becomes less precise when the step size becomes large.

4.5.2 High Dimensions

When the dimension becomes large, the number of candidate transcripts grows exponentially. The previously mentioned methods become less efficient to obtain the parameter estimates. Because of the complexity of the likelihood function, the coordinate descent algorithm requires a very long iteration to converge and it is easily trapped in local optimal mode. Another difficulty comes from evaluating the gradient function and hessian matrix. As mentioned before, although for a particular gene, exact gradient function and hessian matrix can be provided, it is generally not practical to do this for all the genes. Therefore, we resort to automatic differentiation model builder (ADMB), which is designed to solve complex likelihood functions with large number of parameters.

In general, for extremely complex likelihood function, if one cannot write down the gradient function explicitly, it could be very time consuming to evaluate the gradient function and hessian matrix. However, unlike the usual symbolic differentiation, automatic differentiation can work on extremely complex problems. Automatic differentiation utilizes the fact that every complex functions executes a sequence of elementary arithmetic operation, such as addition, subtraction, division, logarithm, etc. Thus by applying the chain rule recursively to these operations, the gradient function of the original function can be evaluated automatically and efficiently. The gradient evaluation is also more precise compared to numerical differentiation method because such approximation methods can introduce rounding errors. Another advantage of automatic differentiation is the speed it evaluates the gradient function, because no finite differencing are used when evaluating derivative.

With all these advantages, currently, ADMB is a standalone software that is based on the autodiff library in C++. For maximizing extremely complex likelihood function with large number of parameters, there are three ways that one can utilize the ADMB package. First, one can directly write code, compile code, and run program in ADMB user interface. Second, the ADMB compiler is able to produce a dynamic link library, which further can be called in R programs. Thirdly, the R2ADMB package in R contains a series of functions to call ADMB and run ADMB within R. Therefore, in the following simulation and real examples, we called the ADMB package within R, and collected the results so that they can be better summarized and plotted.

Using the ADMB software, the user needs to specify at least the following three core parts. First, users need to specify the data section, which describes the data structure used by the model. Any variables that do not require the evaluation of the derivative will be defined in the data section. Similar as the C++ languages, variables can be used only after they are defined. Once the program is compiled and run, the data that is stored in a separate .dat file will be read in and used in later steps. Second, parameter section describes the structure of the model parameters. The parameters that we want to infer about can be integer type or floating type, and they all need to be defined correspondingly. Additionally, the objective function value will also need to be declared. By default, the program performs minimization. Different from other optimization packages, such as "optim" in R, where the bounded parameters are more difficult to optimize, ADMB automatically handles the parameter bounds in the parameter section. One byproduct of using ADMB is that the standard error or likelihood profile for designated parameters is automatically generated. Initial values can be given in a separate .pin file. If there is no .pin file, the ADMB will either assign 0 or midpoint of the bounded interval. Third, the model procedure or the objective function is given in the procedure section. The lines in this section need to be written in C++ language and need to by coherent to C++ syntax. No gradient function needs to be provided because when the compiler compiles the code in procedure section, the gradient function will also be generated based the chain rule of derivative we described above.

4.5.3 Determine the Level of γ

In simple linear regression, one can use the AIC or BIC values to select the best model. The degrees of freedom in AIC or BIC is the number of predictors that need to be estimated. The degrees of freedom relates to the model complexities, and by choosing the model with the smallest AIC values, one can achieve a balance between the likelihood and the size of the model. However, in general, the degrees of freedom could be difficult to estimate.

Given a model, let $\hat{\mu}$ represent its fit. Assume that y is generated according to $y \sim (\mu, \sigma^2 I)$, where μ is the true mean vector and σ^2 is the common variance. It is shown by Efron [52] that the degrees of freedom of the model fitting is

$$df(\hat{\mu}) = \sum_{i=1}^{n} cov(\hat{\mu}_i, y_i)/\sigma^2. \tag{4.50}$$

Zou *et el.* [53] showed that the for the Lasso fit, the degree of freedom of $\hat{\mu_\gamma}(y)$ equal to the expectation of the effective set $\mathcal{A}_\gamma$, that is $df(\gamma) = E|\mathcal{B}_\gamma|$. The $\mathcal{A}_\gamma$ is the active set corresponding to γ, which is easily obtained in the Lars algorithm [45].

Once the degrees of freedom of the model fitting is estimated, it can be plugged into BIC to select the tuning parameters that minimizes BIC.

4.6 Simulation Example and Real Example

In order to show that the penCPM-Seq can identify transcripts with high expression levels, we need to know the true expression levels of transcripts. In simulation study, we can assign expression levels to transcripts, and treat them as the true expression levels. In real data application, however, the true expression levels of transcripts are usually not available. Instead, we use the annotated transcripts and their base level read counts as the evidence of the expression levels.

Both simulation study and real example is presented in this section to demonstrate how transcript selection can be performed. We carried out one simulation study and one real data application. The simulation study is done in R. In the first step, different expression levels are assigned to transcripts. In the second step, base level read counts are generated from each transcript. In the third step, the read counts are convoluted and this gives rise to the observed read counts data. In the fourth step, all candidate transcripts are considered, and the penalized CPM-Seq model is applied to identify the true transcripts. In the real data example, we first processed the exon annotation so that each exon is either uniquely included or skipped in a transcript. Then, all possible transcripts are considered, and penCPM-Seq model is applied to identify the true expression levels. Although we do not exactly know which transcripts are biologically meaningful and expressed, in some cases, we can reply on the annotation and the mapped reads, especially the junction reads to determine the expression levels of each transcripts.

Example 1

In this example, we created a hypothetical gene that contains three exons. Theoretically, there are $2^3 - 1 = 7$ possible transcripts. Each transcript is given a unique ID, and the transcript annotation is given in Figure 4.6 and Table 4.6. For example, transcripts T_1, T_2, and T_3 include exon e_1, e_2, and e_3, correspondingly. We set four transcripts (T_1, T_2, T_4, and T_7) to be highly expressed and the other three transcripts (T_3, T_5, and T_6) to be lowly expressed. Their true expression levels are summarized in Table 4.6. The Poisson intensities in Table 4.6 are treated as the true expression levels that the base pair read counts are subject to. For type r reads, we generated counts X^r_{tm} at each base pair from each expressed transcript that can give rise to type r reads. Then the read counts are convoluted as $Y^r_m = X^r_{1m} + \ldots + X^r_{N_r m}$, where N_r is the number of transcripts $\{T_{i_1} \ldots T_{i_{N_r}}\} \subset \mathcal{T}_g$ that can give rise to type r reads. For example, for type r_{11} reads, the convolution formula is given as $Y^{r_{11}}_m = X^{r_{11}}_{1m} + X^{r_{11}}_{4m} + X^{r_{11}}_{7m}$, whereas for type r_{33} reads, $Y^{r_{33}}_m = X^{r_{33}}_{7m}$. In this simulation study, we only generated type r_{11}, r_{22}, r_{33}, r_{12}, and r_{13} reads.

Table 4.1
Transcript ID and exon inclusion table

Transcript ID	e_1	e_2	e_3
T_1	1	0	0
T_2	0	1	0
T_3	0	0	1
T_4	1	1	0
T_5	1	0	1
T_6	0	1	1
T_7	1	1	1

In general, when we construct the candidate transcripts set, robust filtering rules need to be used. We should not only intend to construct a small number of transcripts that explain the observed read counts data. Rather, we should rule out impossible transcripts so that when we apply the penCPM-Seq model to the candidate transcripts, we do not miss any true ones.

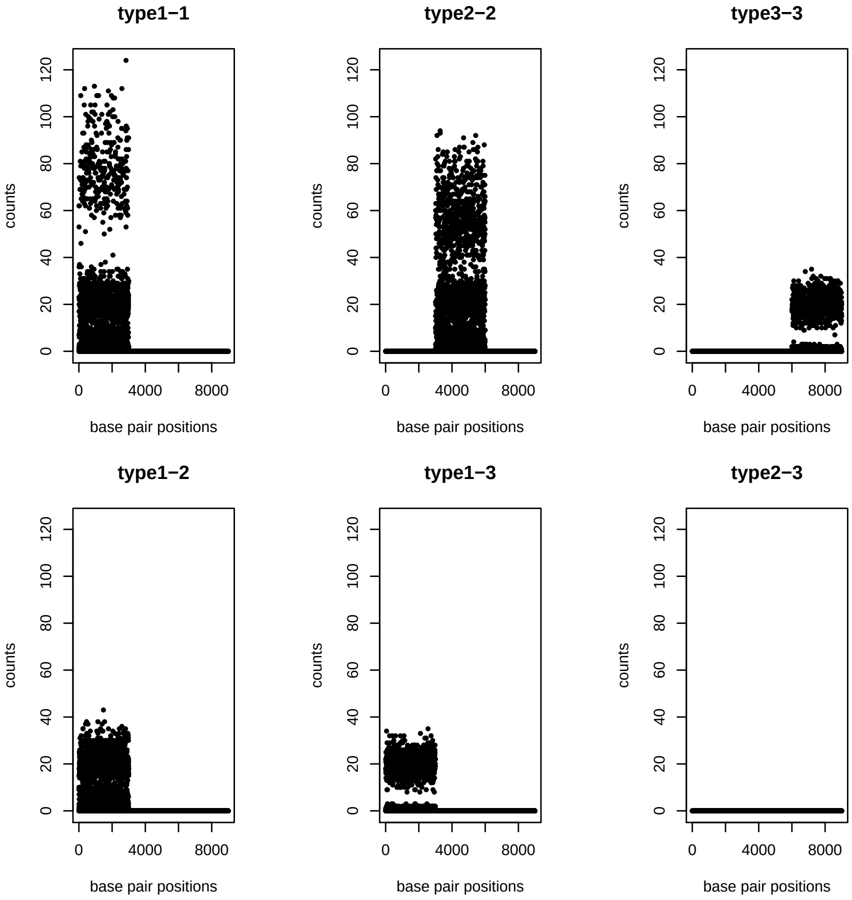

Figure 4.1. Base pair level counts for each type of reads of hypothetical gene. In each plot, the x-axis indicates base pair positions, and the y-axis represents the counts.

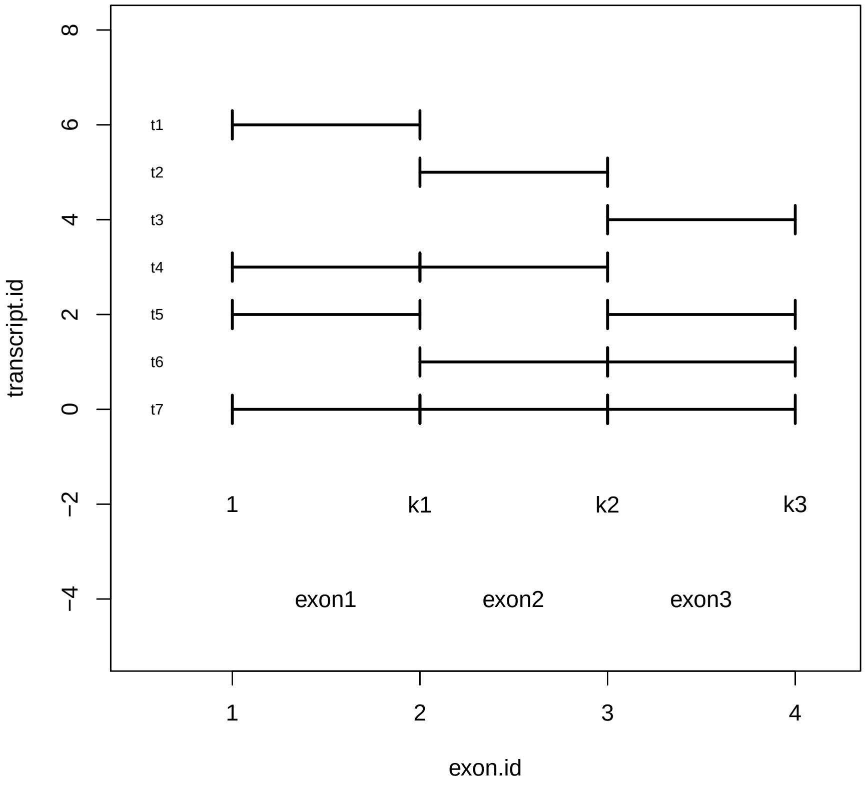

Figure 4.2. Transcripts ID for all possible candidate transcripts and their annotations

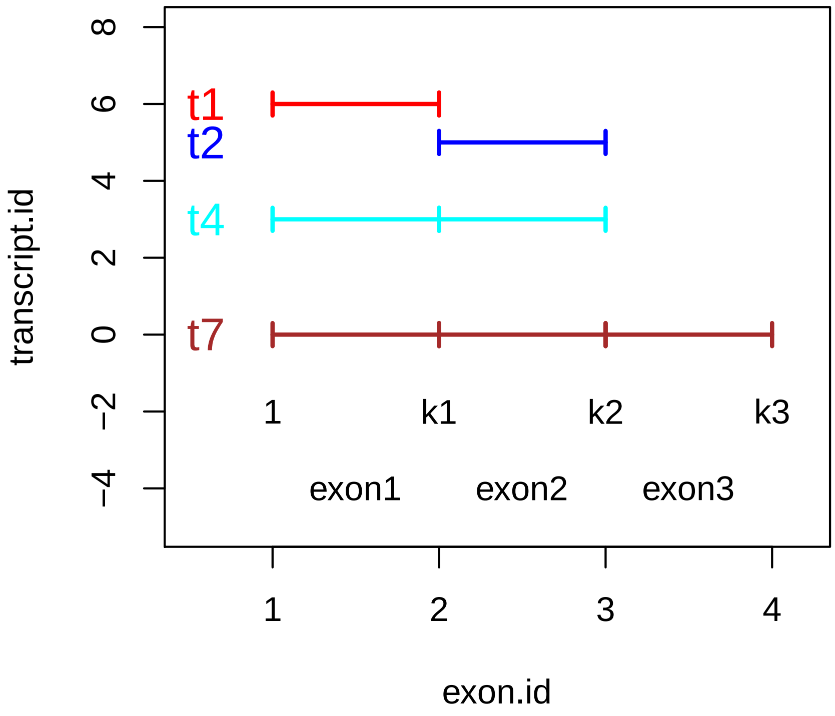

Figure 4.3. Transcripts ID for 4 expressed transcripts and their annotations

Table 4.2
True intensity and mixing proportions of 4 expressed transcripts

T_t	λ_{t1}	λ_{t2}	p_{t2}
$T_1 = e_1$	0.3	**70**	0.1
$T_2 = e_2$	0.3	**50**	0.2
$T_4 = e_1e_2$	0.3	**7**	0.2
$T_7 = e_1e_2e_3$	0.3	**20**	0.3

We applied the penCPM-Seq method to all 7 candidate transcripts, and computed the solution path of the parameters of interest using ADMB. The solution path is depicted in Figure 4.6. We can see from Figure 4.6 that when the penalty is large, none of the transcripts are selected. As the penalty term decreases, transcript T_7 join the active set first. Transcript T_7 is selected because it contains three exons and best explains counts observed on each exon. As the penalty further decreases, transcript T_4, T_2 and T_1 joins the active set sequentially. Eventually, as the penalty approaches 0, the penCPM-Seq model correctly selects the true transcripts and their estimated intensities are also very close to the true values.

A real example of gene According to human refseq hg18, gene MARCKSL1 contains two exons, which we denote as e'_1 and e'_2, and it has two annotated transcripts labeled as NM023009 and NR052852. Transcript NM023009 consists of exon e'_1 and entire of exon e'_2, and transcript NR052852 consists of exon e'_1 and part of exon e'_2. In order to make the transcripts either contain or skip an exon entirely, we split exon e'_2 into two sub-exons denoted as e_2 and e_3. We re-denote exon e'_1 as e_1. Therefore, exons e_1, e_2 and e_3 form a partition of the exonic region of gene MARCKSL1. The total length of gene MARCKSL1's exonic region is 1564, and the exonic base pairs are indexed as $1, \ldots, 1564$. Exons e_1, e_2, and e_3 contain base pairs $\{1, \ldots, 1270\}$, $\{1271, \ldots, 1495\}$, $\{1496, \ldots, 1564\}$, respectively. NM023009 consists of e_1, e_1, and e_3, and NR052852 consists of e_1 and e_3. We re-label the two transcripts NM023009

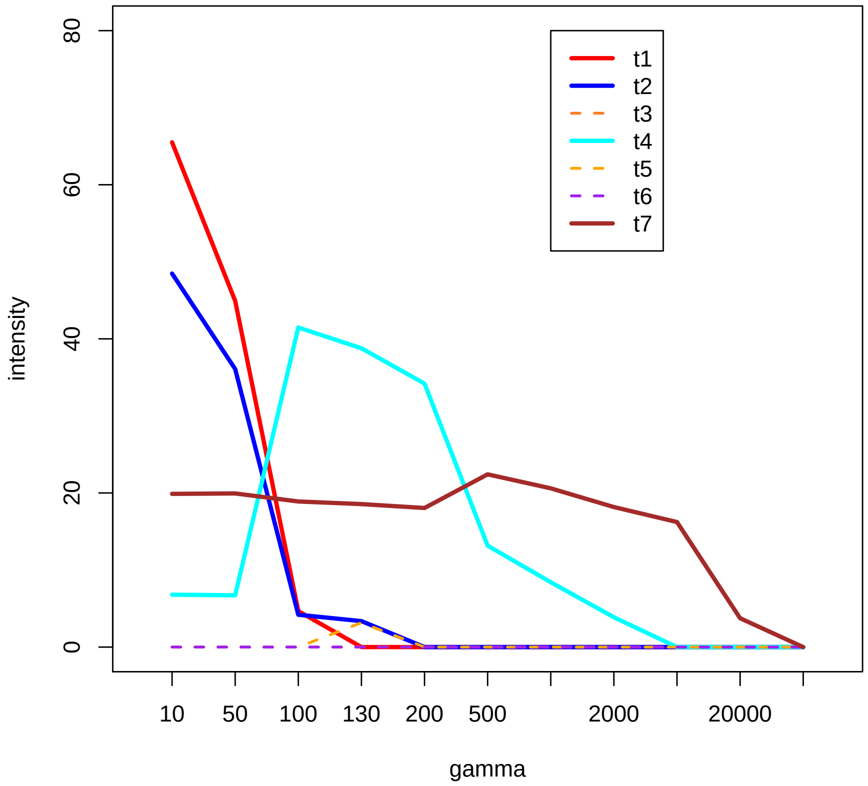

Figure 4.4. Solution path of the hypothetical gene. The x-axis indicates the level of regularization, and the y-axis represents intensity of expression levels.

and NR052852 as T_1 and T_2, respectively, and assume that they form the collection of candidate transcripts, that is, $\mathcal{T} = \{T_1, T_2\}$. The annotation of this gene is plotted in Figure 4.6.

For this particular gene, because the number of exons or the number of candidate transcripts is small, we did not filter out any transcripts based on the observed reads. Instead, we considered all 7 transcripts as our candidates, whose annotation are given in Figure 4.6 and Table 4.6.

We used the RNA-Seq data of the Human Brain Reference RNA (Brain) sample, which was originally generated by Wong's lab using the Illumina Genome Analyzer platform [43]. The data consists of one lane of eight millions 50 bp paired-end reads. The data set can be downloaded from NCBI Short Read Archive (SRA) at http://www.ncbi.nlm.nih.gov/sra under the accession numbers GS475204 and GSM475205 [43]. Tophat was used to map the reads to refseq hg18 [6].

For this particular gene, which consists of three exons, it is able to generate six types of reads, which are r_{11}, r_{12}, r_{13}, r_{22}, r_{23}, and r_{33}. The total numbers of each type of reads are summarized in Table 4.3. The base pair level read counts data for each type is depicted in Figure 4.6. We can see that exon e_1 receives a lot of reads because of its relative length. However, exon e_2 and e_3 receive little and no read counts data because of their relatively short length. Similarly, there are no type r_{13} or r_{33} read counts data, which will make the transcript identification problem difficult.

We applied penCPM-Seq to all seven candidate transcripts to select the true ones and the solution path is plotted in Figure 4.6. We can see that as the regularization decreases, variables join the active sets. Eventually, as the penalty gets smaller, penCPM-Seq model selects transcripts t_5 and t_7 as highly expressed transcripts, and transcripts t_1 and t_3 as lowly expressed.

Although we do not now truely expressed transcripts in this case, we do know that transcripts t_5 and t_7 are in the annotation. The reads of type r_{12} indicates the existence of the e_1e_2 exon-exon junction.

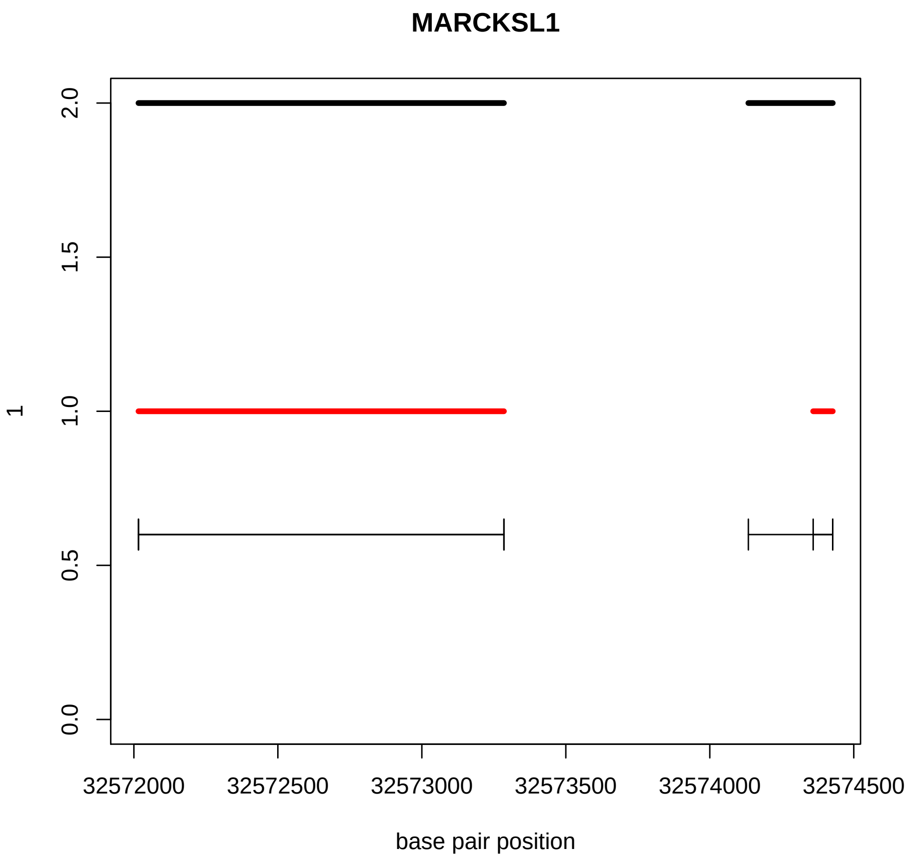

Figure 4.5. Annotation of gene MARCKSL1

Table 4.3
Frequency table for all read types for MARCKSL1

type	r_{11}	r_{12}	r_{13}	r_{22}	r_{23}	r_{33}
counts	1374	187	0	1	1	0

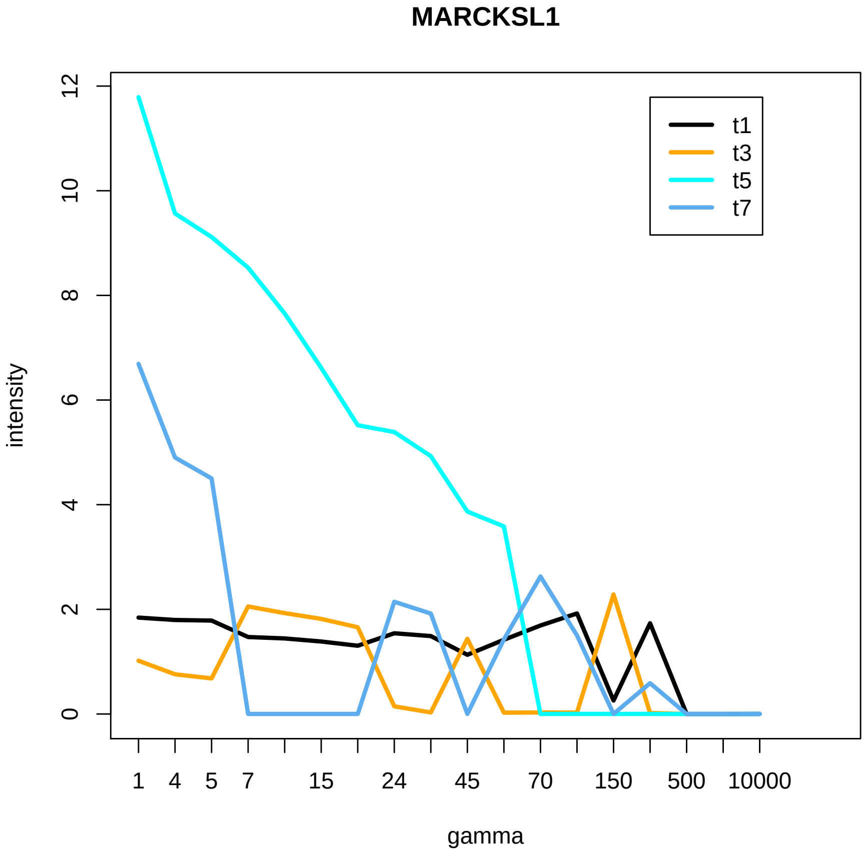

Figure 4.6. Path solution of gene MARCKSL1. Solid orange and blue lines represent the solution path of two annotated transcripts T_5 and T_7, correspondingly.

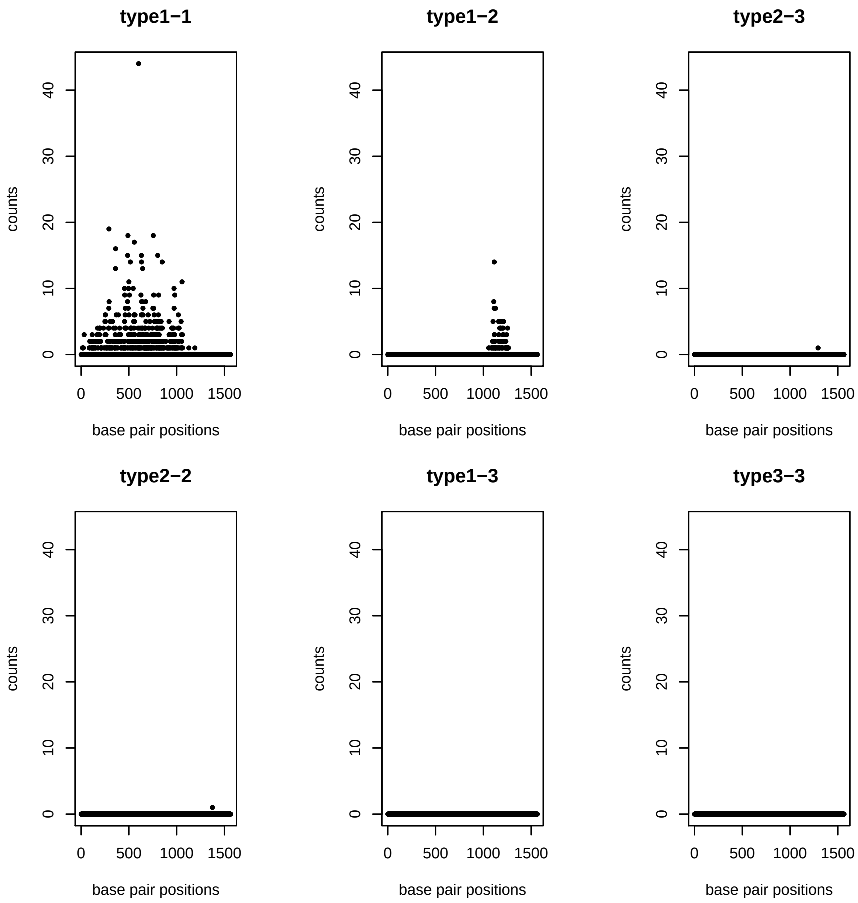

Figure 4.7. Base pair level counts for each type of reads of hypothetical gene. In each plot, the x-axis indicates base pair positions, and the y-axis represents the counts.

5. CONCLUSION

RNA-Seq is poised to replace microarray as the main workhorse for transcriptome studies in the near future. Despite all the advantages RNA-Seq has over microarray, some doubts remain about whether RNA-Seq is able to deliver its promises. One major concern is that RNA-Seq data demonstrates excessive variability, which is more difficult to decipher than that of microarray data, and the sources that contribute biases and variations to RNA-Seq data are more elusive than those of microarray. To make things worse, it is not immediately clear how to use statistical models to characterize and further normalize RNA-Seq data in a ROI [54]. For the transcript expression level quantification, it is an indirect inference problem to identify transcripts and quantify their expression levels using RNA-Seq data, and various types of observation units used in the literature such as exons, segments, and bins can make the problem non-identifiable. The lack of proper statistical tools for RNA-Seq data analysis will greatly hinder the potential of this promising new technology in practice.

In this thesis, we first reviewed the current RNA-Seq technology, and examined various sources of biases and variations that technology has.

Second, we proposed to use Poisson mixture models to characterize RNA-Seq data and quantify the expression levels. As discussed in the introduction, finite Poisson mixture models form a type of semi-parametric models that combines the strength of fully parametric models with the flexibility of fully nonparametric model, and are extremely suitable for modeling data with distributions of unknown shapes and high heterogeneity such as RNA-Seq data. Because the components of a mixture model are parametrically specified, it is straightforward to incorporate other information or structures into the model. For example, in this article, an autoregressive structure is incorporated into the component intensities to account for correlations between adjacent base pair reads counts. Different components in a finite Poisson mixture model correspond naturally to clusters in RNA-Seq reads count data, which may shed light on the data generating mechanism in RNA-Seq experiments. The application of the proposed methods to real RNA-Seq data analysis demonstrated that finite Poisson mixture models can adapt to individual transcripts via model selection and subsequently lead to more accurate and consistent measurements of transcript expression levels.

Third, following the framework of Poisson mixture model, we further proposed to use individual exonic base pairs as observation units and further proposed the convolution of mixture Poisson model to model the base level zero as well as non-zero read counts. The base pair units coupled with the CPM model, to a large degree, resolve the non-identifiability issue. Furthermore, we considered different types of reads and developed the EM algorithm for computing the parameter estimates. Both simulation study and real data application have demonstrated the effectiveness of CPM-Seq. CPM-Seq was shown to produce more accurate and consistent quantification results than Cufflinks.

Fourth, the quantification of a ROI or all transcripts that belong to the same gene is relatively easier because the genes and transcripts are well annotated. However, the identification of new transcripts requires proper statistical models that are not only identifiable but also computationally efficient. We proposed to add a lasso penalty on the intensities parameters in the convolution of Poisson mixture models to shrink transcripts with small effects to zero. The proposed method worked well in simulation and real examples.

There are several immediate directions to further enhance the Poisson mixture models framework for the sequencing data. An immediate direction is to incorporate various types of biases in RNA-Seq data that have been reported in the literature to better correct biases discussed in the first chapter. It is also of immediate interest to use finite Poisson mixture models for detecting differentially expressed genes.

The second direction is to incorporate the fragment length distribution into the CPM model. In the literature, the fragment length distribution is typically modeled by $N(\mu, \sigma^2)$, where μ and σ^2 are either known or can be estimated from reads mapped to genes with single exon. When considering reads of different types, we also need to include the possible lengths of the reads. For example, for read type r, instead of simply counting the reads of type r, we need to count the reads of type r and length l. Let X_{tm}^{rl} denote the number of reads of type r and length l at base pair m of transcript t. We assume that X_{tm}^{rl} follows the following mixture of Poisson distribution $f\left(X_{tm}^{rl} = x|\lambda_t, p_t\right) = \sum_{i=1}^{2} p_{ti} Poi\left(x; a_l, \lambda_{ti}\right)$, where a_l is the probability of obtaining a read of length l according to the fragment length distribution $N(\mu, \sigma^2)$. Subsequently, the convolution distribution for Y_m^{rl} needs to be updated.

The third direction is to use the fused Lasso penalty [55], so that the λ_{i2}'s, for $1 \leq i \leq N$, are shrunken toward the background noises. The fused Lasso penalty is given as $\sum_{i=1}^{N} |\lambda_{i2} - \lambda_{11}|$.

We believe that the Poisson mixture models, convolution of Poisson mixture models, the penalized convolution of Poisson mixture models, and their further developments have the potential to become indispensable statistical tools for RNA-Seq data analysis.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Z Wang, M Gerstein, and M Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10:57–63, 2009.

[2] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, 270(5235):467–470, October 1995.

[3] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature biotechnology*, 14(13):1675–1680, December 1996.

[4] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature Methods*, 5(7):621–628, 2008.

[5] Heng Li and Richard Durbin. Fast and accurate short read alignment with BurrowsWheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[6] Cole Trapnell, Lior Pachter, and Steven L. Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, May 2009.

[7] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

[8] CGrunenwald H., Baas B., Caruccio N., and Syed F. Rapid, high-throughput library preparation for next-generation sequencing. *Nature Methods*, 7(8), 2010.

[9] Lira Mamanova and Daniel J. Turner. Low-bias, strand-specific transcriptome Illumina sequencing by on-flowcell reverse transcription (FRT-seq). *Nat. Protocols*, 6(11):1736–1747, 2011.

[10] David R. Bentley, Shankar Balasubramanian, Harold P. Swerdlow, Geoffrey P. Smith, John Milton, Clive G. Brown, Kevin P. Hall, Dirk J. Evers, Colin L. Barnes, Helen R. Bignell, Jonathan M. Boutell, Jason Bryant, Richard J. Carter, R. Keira Cheetham, Anthony J. Cox, Darren J. Ellis, Michael R. Flatbush, Niall A. Gormley, Sean J. Humphray, Leslie J. Irving, Mirian S. Karbelashvili, Scott M. Kirk, Heng Li, Xiaohai Liu, Klaus S. Maisinger, Lisa J. Murray, Bojan Obradovic, Tobias Ost, Michael L. Parkinson, Mark R. Pratt, Isabelle M. J. Rasolonjatovo, Mark T. Reed, Roberto Rigatti, Chiara Rodighiero, Mark T. Ross, Andrea Sabot, Subramanian V. Sankar, Aylwyn Scally, Gary P. Schroth, Mark E. Smith, Vincent P. Smith, Anastassia Spiridou, Peta E. Torrance, Svilen S. Tzonev, Eric H. Vermaas, Klaudia Walter, Xiaolin Wu, Lu Zhang, Mohammed D. Alam, Carole Anastasi, Ify C. Aniebo, David M. D. Bailey, Iain R. Bancarz,

Saibal Banerjee, Selena G. Barbour, Primo A. Baybayan, Vincent A. Benoit, Kevin F. Benson, Claire Bevis, Phillip J. Black, Asha Boodhun, Joe S. Brennan, John A. Bridgham, Rob C. Brown, Andrew A. Brown, Dale H. Buermann, Abass A. Bundu, James C. Burrows, Nigel P. Carter, Nestor Castillo, Chiara, Simon Chang, R. Neil Cooley, Natasha R. Crake, Olubunmi O. Dada, Konstantinos D. Diakoumakos, Belen Dominguez-Fernandez, David J. Earnshaw, Ugonna C. Egbujor, David W. Elmore, Sergey S. Etchin, Mark R. Ewan, Milan Fedurco, Louise J. Fraser, Karin V. Fuentes Fajardo, W. Scott Furey, David George, Kimberley J. Gietzen, Colin P. Goddard, George S. Golda, Philip A. Granieri, David E. Green, David L. Gustafson, Nancy F. Hansen, Kevin Harnish, Christian D. Haudenschild, Narinder I. Heyer, Matthew M. Hims, Johnny T. Ho, Adrian M. Horgan, Katya Hoschler, Steve Hurwitz, Denis V. Ivanov, Maria Q. Johnson, Terena James, T. A. Huw Jones, Gyoung-Dong Kang, Tzvetana H. Kerelska, Alan D. Kersey, Irina Khrebtukova, Alex P. Kindwall, Zoya Kingsbury, Paula I. Kokko-Gonzales, Anil Kumar, Marc A. Laurent, Cynthia T. Lawley, Sarah E. Lee, Xavier Lee, Arnold K. Liao, Jennifer A. Loch, Mitch Lok, Shujun Luo, Radhika M. Mammen, John W. Martin, Patrick G. McCauley, Paul McNitt, Parul Mehta, Keith W. Moon, Joe W. Mullens, Taksina Newington, Zemin Ning, Bee Ling Ng, Sonia M. Novo, Michael J. O/'Neill, Mark A. Osborne, Andrew Osnowski, Omead Ostadan, Lambros L. Paraschos, Lea Pickering, Andrew C. Pike, Alger C. Pike, D. Chris Pinkard, Daniel P. Pliskin, Joe Podhasky, Victor J. Quijano, Come Raczy, Vicki H. Rae, Stephen R. Rawlings, Ana Chiva Rodriguez, Phyllida M. Roe, John Rogers, Maria C. Rogert Bacigalupo, Nikolai Romanov, Anthony Romieu, Rithy K. Roth, Natalie J. Rourke, Silke T. Ruediger, Eli Rusman, Raquel M. Sanches-Kuiper, Martin R. Schenker, Josefina M. Seoane, Richard J. Shaw, Mitch K. Shiver, Steven W. Short, Ning L. Sizto, Johannes P. Sluis, Melanie A. Smith, Jean Ernest Sohna Sohna, Eric J. Spence, Kim Stevens, Neil Sutton, Lukasz Szajkowski, Carolyn L. Tregidgo, Gerardo Turcatti, Stephanie vandeVondele, Yuli Verhovsky, Selene M. Virk, Suzanne Wakelin, Gregory C. Walcott, Jingwen Wang, Graham J. Worsley, Juying Yan, Ling Yau, Mike Zuerlein, Jane Rogers, James C. Mullikin, Matthew E. Hurles, Nick J. McCooke, John S. West, Frank L. Oaks, Peter L. Lundberg, David Klenerman, Richard Durbin, and Anthony J. Smith. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, 2008.

[11] James Bullard, Elizabeth Purdom, Kasper Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11(1):94, 2010.

[12] Yuval Benjamini and Terence P. Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 2012.

[13] Fan Li, Paul Ryvkin, Daniel M. Childress, Otto Valladares, Brian D. Gregory, and Li-San Wang. SAVoR: a server for sequencing annotation and visualization of RNA structures. *Nucleic Acids Research*, 40(W1):W59–W64, 2012.

[14] Lauren McIntyre, Kenneth Lopiano, Alison Morse, Victor Amin, Ann Oberg, Linda Young, and Sergey Nuzhdin. Rna-seq: technical variability and sampling. *BMC Genomics*, 12(1):293, 2011.

[15] ET Wang, R Sandberg, S Luo, I Khrebtukova, L Zhang, C Mayr, SF Kingsmore, GP Schroth, and CB Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456:470–476, 2008.

[16] Jun Li, Hui Jiang, and Wing Wong. Modeling non-uniformity in short-read rates in RNA-Seq data. *Genome Biology*, 11(5):R50, 2010.

[17] Wei Zheng, Lisa Chung, and Hongyu Zhao. Bias detection and correction in rna-sequencing data. *BMC Bioinformatics*, 12(1):290, 2011.

[18] Alicia Oshlack and Matthew J Wakefield. Transcript length bias in RNA-Seq data confounds systems biology. *Biology Direct*, 4(1):14, 2009.

[19] Adam Roberts, Cole Trapnell, Julie Donaghey, John Rinn, and Lior Pachter. Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, 12(3):R22, 2011.

[20] Sudeep Srivastava and Liang Chen. A two-parameter generalized poisson model to improve the analysis of rna-seq data. *Nucleic Acids Research*, 38(17):e170, 2010.

[21] Ming Hu, Yu Zhu, Jeremy M. Taylor, Jun S. Liu, and Zhaohui S. Qin. Using Poisson mixed-effects model to quantify transcript-level gene expression in RNA-Seq. *Bioinformatics (Oxford, England)*, 28(1):63–68, January 2012.

[22] Han Wu, Zhaohui Qin, and Yu Zhu. Pm-seq: Using finite poisson mixture models for rna-seq data analysis and transcript expression level quantification. *Statistics in Biosciences*, pages 1–17, 2012.

[23] Kasper D Hansen, Steven E Brenner, and Sandrine Dudoit.

[24] Daniel Aird, Michael Ross, Wei-Sheng Chen, Maxwell Danielsson, Timothy Fennell, Carsten Russ, David Jaffe, Chad Nusbaum, and Andreas Gnirke. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biology*, 12(2):R18, 2011.

[25] JC Marioni, CE Mason, SM Mane, M Stephens, and Y Gilad. RNA-Seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–17, 2008.

[26] Geoffrey Mclachlan and David Peel. *Finite Mixture Models.* Wiley Series in Probability and Statistics. Wiley-Interscience, 1 edition, 2000.

[27] Wing Hung Wong. Theory of partial likelihood. *The Annals of Statistics*, 14(1):pp. 88–123, 1986.

[28] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, CRC Monographs on Statistics and Applied Probability, New York, 1994.

[29] Alexandre X. Carvalho and Martin A. Tanner. Modelling nonlinear count time series with local mixtures of poisson autoregressions. *Comput. Stat. Data Anal.*, 51(11):5266–5294, 2007.

[30] Dimitris N. Politis and Joseph P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303+, December 1994.

[31] Lee T. Sam, Doron Lipson, Tal Raz, Xuhong Cao, John Thompson, Patrice M. Milos, Dan Robinson, Arul M. Chinnaiyan, Chandan Kumar-Sinha, and Christopher A. Maher. A comparison of single molecule and amplification based sequencing of cancer transcriptomes. *PLoS ONE*, 6(3):e17305, 2011.

[32] L. Shi, L. H. Reid, W. D. Jones, R. Shippy, J. A. Warrington, S. C. Baker, P. J. Collins, F. de Longueville, E. S. Kawasaki, K. Y. Lee, Y. Luo, Y. A. Sun, J. C. Willy, R. A. Setterquist, G. M. Fischer, W. Tong, Y. P. Dragan, D. J. Dix, F. W. Frueh, F. M. Goodsaid, D. Herman, R. V. Jensen, C. D. Johnson, E. K. Lobenhofer, R. K. Puri, and Scherf. The microarray quality control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol*, 24:1151–1161, 2006.

[33] Ann S Zweig, Donna Karolchik, Robert M Kuhn, David Haussler, and W James Kent.

[34] Hui Jiang and Wing Hung Wong. Statistical inferences for isoform expression in rna-seq. *Bioinformatics*, (8):10261032, 2009.

[35] Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J Van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

[36] Wei Li, Jianxing Feng, and Tao Jiang. Isolasso: A lasso regression approach to rna-seq based transcriptome assembly. *Journal of Computational Biology*, 18(11):1693–1707, November 2011.

[37] Jingyi Jessica Li, Ci-Ren Jiang, James B. Brown, Haiyan Huang, and Peter J. Bickel. Sparse linear modeling of next-generation mrna sequencing (rna-seq) data for isoform discovery and abundance estimation. *Proceedings of the National Academy of Sciences*, 108(50):19867–19872, 2011.

[38] M. K. Sakharkar, V. T. Chow, and P. Kangueane. Distributions of exons and introns in the human genome. *In Silico Biol*, 4(4):387–393, 2004.

[39] Henry Teicher. Identifiability of Finite Mixtures. *The Annals of Mathematical Statistics*, 34(4):1265–1269, 1963.

[40] Thasso Griebel, Benedikt Zacher, Paolo Ribeca, Emanuele Raineri, Vincent Lacroix, Roderic Guigó, and Michael Sammeth. Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic acids research*, 40(20):10073–10083, November 2012.

[41] Eric T. Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F. Kingsmore, Gary P. Schroth, and Christopher B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, November 2008.

[42] Hyunsoo Kim, Yingtao Bi, Sharmistha Pal, Ravi Gupta, and Ramana Davuluri. IsoformEx: Isoform level gene expression estimation using weighted non-negative least squares from mRNA-seq data. *BMC Bioinformatics*, 12(1):305+, 2011.

[43] Kin Fai F. Au, Hui Jiang, Lan Lin, Yi Xing, and Wing Hung H. Wong. Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic acids research*, 38(14):4570–4578, August 2010.

[44] R. Tibshirani. Regression shrinkage and selection via the lasso. 1994.

[45] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. 2002.

[46] Peng Zhao and Bin Yu. On Model Selection Consistency of Lasso. *J. Mach. Learn. Res.*, 7:2541–2563, December 2006.

[47] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246–270, February 2009.

[48] R. Tibshirani. Regression shrinkage and selection via the lasso, 1994.

[49] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009.

[50] Saharon Rosset and Ji Zhu. Piecewise Linear Regularized Solution Paths. 2004.

[51] Jianqing Fan and Runze Li. Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[52] Bradley Efron. The estimation of prediction error: Covariance penalties and cross-validation. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, pages 99–467, 2004.

[53] Hui Zou, Trevor Hastie, and Robert Tibshirani. On the "degrees of freedom" of the lasso. December 2007.

[54] H. Craig Mak. John Storey. *Nature Biotechnology*, 29(4):331–333, April 2011.

[55] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

VITA

VITA

Han Wu was born in Taiyuan. He came to study at the Department of Statistics at Purdue University in 2008 after his bachelor's degree in Mathematics at Zhejiang University. He has worked with Professor Yu Zhu on interesting projects in statistics and bioinformatics since 2010.