THE UNIVERSITY
of EDINBURGH

Structured Bayesian methods for splicing analysis in RNA-seq data

Yuanhua Huang

Doctor of Philosophy

Institute for Adaptive and Neural Computation

School of Informatics

University of Edinburgh

2018

Abstract

In most eukaryotes, alternative splicing is an important regulatory mechanism of gene expression that results in a single gene coding for multiple protein isoforms, thus largely increases the diversity of the proteome. RNA-seq is widely used for genome-wide splicing isoform quantification, and several effective and powerful methods have been developed for splicing analysis with RNA-seq data. However, it remains problematic for genes with low coverages or large number of isoforms. These difficulties may in principle be ameliorated by exploiting correlations encoded in the structured data sources.

This thesis contributes to developments of Bayesian methods for splicing analysis by leveraging additional information in multiple datasets with structured prior distributions. First, we developed DICEseq, the first isoform quantification method tailored to time-series RNA-seq experiments. DICEseq explicitly models the correlations between experiments at different time points to aid the quantification of isoforms across experiments. Numerical experiments on both simulated and real datasets show that DICEseq yields more accurate results than state-of-the-art methods, an advantage that can become considerable at low coverage levels. Furthermore, DICEseq permits to quantify the trade-off between temporal sampling of RNA and depth of sequencing, frequently an important choice when planning experiments.

Second, we developed BRIE (Bayesian Regression for Isoform Estimation), a Bayesian hierarchical model which resolves the difficulties in splicing analysis in single-cell RNA-seq (scRNA-seq) data by learning an informative prior distribution from sequence features. This method combines the quantification and imputation for splicing analysis via a Bayesian way, which is particularly useful in scRNA-seq data due to its extreme low coverages and high technical noises. We validated BRIE on several scRNA-seq data sets, showing that BRIE yields reproducible estimates of exon inclusion ratios in single cells. Third, we provided an effective tool by using Bayes factor to sensitively detect differential splicing between different single cells. When applying BRIE to a few real datasets, we found interesting heterogeneity patterns in splicing events across cell population, for example alternative exons in DNMT3B.

In summary, this thesis proposes structured Bayesian methods to integrate multiple datasets to improve splicing analysis and study its biological functions.

Acknowledgements

First and foremost, I would like to thank my enthusiastic supervisor Guido Sanguinitti for his continuous support. Guido is always ready to help, patiently guiding me into the right path in research from many aspects. With him, I found my passion in scientific research and had great opportunities to try a couple of exciting projects, which helped me gradually grow up into a researcher.

Similarly, I express my gratitude to my co-supervisors Jean Beggs and Douglas Armstrong for their contributions to my annual reviews, ensuring the smooth progress of my Ph.D. Jean's profound knowledge in molecular biology helped me quickly enter this interdisciplinary area with a clear big picture and a specific start point. From her, I also learned the importance of meticulous scientific attitudes in basic research.

In addition, I want to extend my appreciation to my collaborators, including Sander Granneman, David Barrass, Jane Reid, and Vahid Aslanzadeh. Their wonderful collaborative projects offered me valuable chances not only to closely study exciting biological questions, but also to develop problem-driven methodologies partially based on their experimental data. Particular thanks go to Sander, who gave me a lot of helpful advice on data analysis in my first project.

I am also highly appreciative to all members from Sanguinitti's lab, for creating a such friendly and supportive atmosphere in the group. A lot of fruitful discussions happened with them, e.g., Edward Wallace, Gabriele Schweikert and Andreas Kapourani, which were very helpful for planning and conducting my research projects. I was very lucky to have them and other nice friends around, so that I can have an enjoyable Ph.D. life. I also thank the University of Edinburgh for funding my Ph.D. study through a Principal Career Development scholarship and a Global Research Scholarship.

Finally, but by no means least, I would like to express my gratitude to my family: my wife Wei Zhu, my parents and sister for their love, encouragements and support. They are always my energy to stay positive and pursue my dream. I dedicate this thesis to them.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yuanhua Huang*)

Table of Contents

Chapter 1

Introduction

Curiosity and inherent desire of reasoning are primitive forces of human to explore the universe, including the mystery of life science, where people are fascinated by the common molecular mechanisms behind the diversity of species. Thanks to the fast technology development, we are entering an era with exponential growth of data, for example, we can easily generate genomic sequences on millions of individual people and thousands of species, or collect massive health related records from hospital and personal mobile devices. The big data brings the possibility for us to more thoroughly understand complex systems. However, data is not equal to knowledge, and scientific finding tends to be more difficult by mining vast amount of high-dimensional data than by conventional direct observations.

Therefore, there is high demand for automated methods to analyse this deluge of data, which machine learning can provide. More precisely, machine learning, a subfield of computer science, offers a set of computer algorithms that can automatically recognize patterns in data, and use these patterns to take actions for future data, such as predict category labels (Murphy, 2012). In recent years, machine learning has been increasingly applied into many fields, for example, object recognition in images or videos, prediction of stock market, analyses of disease risk in genetic mutations. These analyses are not results from a precisely programmed algorithm, but based on the patterns learned from the vast amount of training data. One exciting machine learning example is that by training on hundreds of thousands of clinical images with skin lesions, patterns are learned and can be used to accurately predict the skin disease type, including deadliest skin cancer, on a new image (Esteva et al., 2017).

The ability of machine learning in handling large-scale datasets and automatically detecting patterns is what modern biology needs. Biology, a discipline of natural sci-

ence that involves the study of life and living organisms, originally focused on organism level, now fast stretches into cellular and molecular level, especially with the developments in biochemistry after mid 20th century. In the last two decades, the development and application of sequencing technology accumulated a huge amount of data at molecular level, including proteins, RNAs and DNAs, and thus more heavily depends on computers to store, process and analyse such data. Consequently, computational biology became an interdisciplinary field between biology, computer science and statistics. This research field involves the development and application of data-driven methodology, including computer algorithm and mathematical modelling, for simulation, analysis, and pattern discovery in biological problems. Depending on the specific biological problem, computational biology could be further divided into a few sub-fields, for example, computational genomics which focuses on analysing genomic data, and computational cancer biology which focuses on modelling of cancer biology.

Statistical modelling of the dynamics of a cell or an organism at molecular level is very important to understand regulation mechanisms. As a very complex system, cellular dynamics involve multiple molecular layers, e.g. mRNA and protein, with each layer having specific difficulties and noises in measurement. However, many variables in a biological system are not independent, thus integrating multiple data sources by a sensible way is very important in effectively modelling biological systems. Bayesian methods, a type of probabilistic methods, combine the prior distribution of all parameters in a model and the probability to observe the given data from the model. By defining different structures of prior distribution of a model, Bayesian methods offer a hierarchical way to connect multiple variables with considering their dependency and uncertainty (Gelman et al., 2014). In other words, the Bayesian hierarchical model provides a unique way to integrate multiple data sources by specifying a structure of prior distribution to model a complex system, which is very useful for many cases with big data, including some biological problems.

In this thesis, I focus on modelling RNA splicing which is an important layer in regulation of gene expression and associated with important biological processes and even serious diseases, and I developed a couple of Bayesian methods to model RNA splicing in time-series experiments or single cell resolution. These new Bayesian methods make the measurements and analyses on splicing much more accurate and reliable when the sequencing coverages are very low in RNA-seq experiments.

Contributions

In this thesis, my main contribution is developing Bayesian methods to improve isoform quantification from RNA-seq data, especially with tailored structured prior distributions for experiments with very low coverages.

In Chapter 2, I will introduce detailed backgrounds in both molecular biology and machine learning. There, I will precisely define all biological concepts involved in this thesis, and present all bases of the machine learning methods that will be used in this thesis. In the biological background, I will describe gene expression and introduce an important process, alternative splicing during transcription, and explain its mechanism, regulation and biological functions. I will also briefly describe the experiment of RNA-seq, and show how it can be used in studying RNA splicing, and discuss its intrinsic limitations. In the machine learning background, I will start with introducing some basic concepts of machine learning, and move to probabilistic graphical models with a focus on mixture models, and present EM algorithm for estimations in mixture models with details. I also introduce Bayesian statistics and show how to use MCMC algorithms to sample in high-dimensional spaces for Bayesian inference. These techniques will be largely used in the development and inference of Bayesian methods for the RNA splicing modelling tasks.

In Chapter 3, I will describe a widely used statistical framework, a mixture model, for isoform quantification with RNA-seq data. Then I will introduce a few different inference algorithms, including EM algorithm, Gibbs sampler, Metropolis-Hastings sampler to estimate isoform fractions, and compare their performances in splicing analysis. In addition, I will compare the probabilistic model and two direct measurements on two-isoform splicing analysis and highlight the benefits of the probabilistic method in isoform quantification. As a case study, I will describe our work on quantifying splicing efficiency with the probabilistic method, which was conducted together with collaborators from the Beggs's lab (Barrass et al., 2015).

In Chapter 4, I will present a dynamic model for isoform quantification in time series RNA-seq data. I will first introduce the importance of time-series experiment in studying splicing kinetics, for example using 4tU labelled RNA-seq to study splicing efficiency in yeast. I then describe DICEseq, a Bayesian method, that uses a Gaussian process to model the temporal correlation between time points, which gives a joint prior distribution on multiple time points (Huang and Sanguinetti, 2016). Finally, I will show how DICEseq improves isoform quantification in time-series experiments,

especially for time points with low coverages, and its impact on experiment design for the trade-off between number of time points and sequencing depth.

In Chapter 5, I will describe the Bayesian regression for isoform estimate (BRIE) method for studying splicing events in single cells. BRIE combines a likelihood function of splicing events from RNA-seq data, and an informative prior that is automatically learned from surrounding genetic features (Huang and Sanguinetti, 2017). I will also show how BRIE remarkably improves the splicing analysis in single-cell RNA-seq data, particularly in handling drop-out and low coverages. Also, I will show our effective tool using Bayes factor for sensitively detecting differential splicing between cells. By using this method, I found an interesting set of splicing events with high heterogeneity across cell population, which is associated with special biological functions.

In Chapter 6, I will summarize this thesis and discuss potential future directions for related study, for example detecting differential splicing for time-series experiments, and presenting the variability of splicing in single cells.

Publication list

Here, I list the four publications related to this thesis during my doctoral study.

1. Aslanzadeh V., **Huang Y.**, Sanguinetti G., and Beggs J. "Effects of transcription rate on the co-transcriptionality, efficiency and fidelity of splicing in budding yeast." *Genome Research*, 2017, doi:10.1101/gr.225615.117.
2. **Huang Y.**, and Sanguinetti G. "BRIE: transcriptome-wide splicing quantification in single cells." *Genome Biology*, 2017, 18(1): 123.
3. **Huang Y.**, and Sanguinetti G. "Statistical modeling of isoform dynamics from RNA-seq time series data." *Bioinformatics*, 2016, 32(19): 2965-2972.
4. Barrass D.*, Reid J.*, **Huang Y.***, Hector R., Sanguinetti G., Granneman S., and Beggs J. "Transcriptome-wide RNA processing kinetics revealed using extremely short 4tU labeling." *Genome Biology*, 2015, 16(1): 282. (*Equal contribution)

In addition to the publication in journals, the two methodology works were presented in conferences. The work of DICEseq was presented as a highlight talk in the workshop of Machine Learning in Systems Biology (MLSB) 2016 during ECCB 2016

conference. The work of BRIE was presented in Ascona workshop 2017 for statistical challenges in single cell biology. It was also presented in ISMB/ECCB 2017, as a contribution talk in the MLSB workshop and a poster in HiTSeq workshop, where it was given the best poster award.

Chapter 2

Background

2.1 Biological background

It is estimated that there are more than 10 million living species on Earth today; each has its unique characteristics, but all have the ability to reproduce offspring that keeps the same specific information as the parent organism. This ability is called *heredity*, which is the central definition of life (Alberts et al., 2008). Although species are highly different between each other, for example most organisms are single cells whereas others are multicellular organisms, there are still a lot of universal features shared by all living organisms. First, the **cell** is the basic building-block of all life, as each cell contains all the heredity information. Even for multicellular species, a whole organism, say a human body consisting of over 10^{13} cells, is generated by cell division initially from a single cell.

After evolving and diversifying for over 3.5 billion years, all living cells still use the same form in storing the heredity information, i.e, the double stranded molecules of **DNA** that are long unbranched paired polymer chains with only four types of monomers (see Fig 2.1(a-b)). Each monomer in a single DNA strand, is called a nucleotide, which comprises two parts: a sugar with phosphate group and a *base* which may be either adenine (A), guanine (G), cytosine (C) or thymine (T). A sequence of these four types of nucleotides – A, G, C, T – is the form of heredity information storage, which is similar to a computer file, though the latter only uses two types of letters, i.e., 0s and 1s. The genome, namely the whole heredity information in a cell, usually contains multiple DNA double stranded chains, for example, there are two copies of 3 billion monomers across 23 DNA chains in human somatic cells.

Furthermore, the information flow, involving DNA, RNA and protein, within a cell

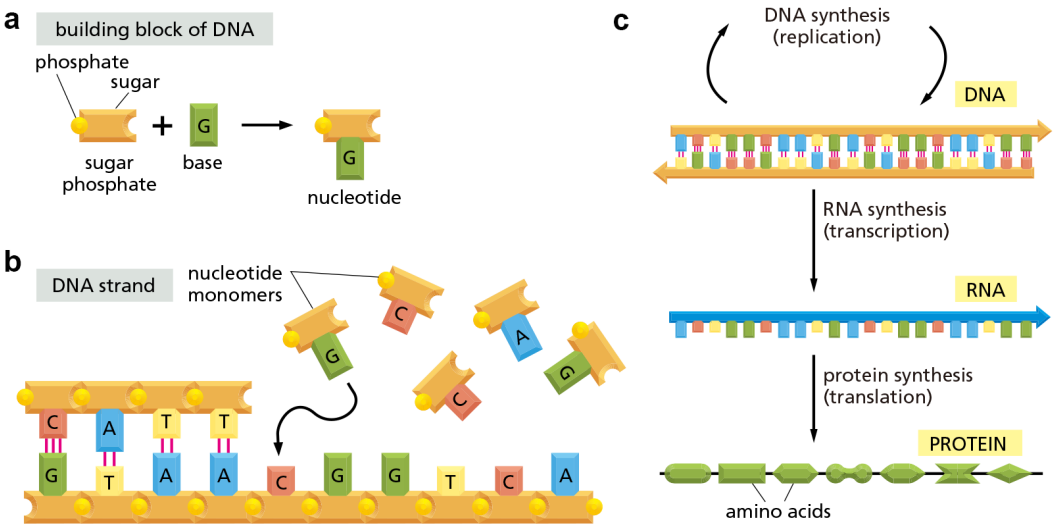

Figure 2.1: Cartoon for DNA and information flow from DNA to protein. (a) The monomer of DNA consists of a base and a sugar with phosphate group. (b) The DNA strand, a sequence of the four types of monomers (A,G,C,T), stores the information of heredity. The double DNA strands are always complementary, namely always with base pairs of A-T or C-G, so one strand could be a template for the polymerization of the other. (c) Information flow from DNA to protein. This is the main part of the central dogma of molecular biology in (Crick, 1970). The black arrows present the information transfer, i.e., DNA to DNA by DNA replication, DNA to RNA by transcription, and RNA to protein by translation. This figure is modified from (Alberts et al., 2008).

is the same for most species, which forms the central dogma of molecular biology (Crick, 1970). In order to perform a particular function, DNA must *express* its information into other molecular forms (see the cartoon in Fig 2.1(c)). This process starts from **transcription** (discussed in more detailed in the next subsection), during which a segment of DNA sequence is used as template for the synthesis of a shorter polymer **ribonucleic acid**, or **RNA**. Later, a more complex process, **translation**, will happen, where the transcribed RNA will be used to guide the synthesis of *proteins*, a form of *polypeptides*. Comparing to RNA, protein is a more direct player for most biological processes in the form of, say, catalysis as an enzyme. Thus, the RNA that is used for leading synthesis of coding protein during translation plays a role of delivering message from DNA to protein, thus it is also named messenger RNA (mRNA).

The most popular concept in genetics, the **gene**, was initially introduced to better understand the discrete function of the genome. However, the precise definition of

gene evolved a few times, initially from a discrete unit of heredity back in 1860s, to a DNA segment corresponding to a protein in 1960s, to more recently a DNA segment that contributes to phenotype/function in 2000s (Gerstein et al., 2007). Although even the latest definition of gene is not perfect when considering special cases of gene regulation, overlapping genes, RNA processing etc, the current definition of gene still provides good annotations for most protein coding genes. Namely, these genes are discrete units on the DNA chain with clear boundaries and do not overlap with other genes. Therefore, in this thesis, we will take this concept of gene and most of the analysis and conclusions are based on these common protein coding genes.

2.1.1 Gene expression

The function of a gene is normally performed by its product, rather than by the DNA template itself. Gene expression is the process where the genomic information of a gene is used to generate a functional product, mainly in the form of proteins (protein coding genes) and auxiliary RNAs (non-coding genes). Transcription is the first and very important step in expression for both protein coding/non-coding genes. By bringing a bridge between DNA information and biological functions, gene expression is an important indicator of cellular or tissue status. Consequently, its regulation is crucial to the development and survival of a cell and an organism.

In most conditions, the DNAs are extremely highly packed, for example the total DNAs in a human cell is approximately 2 meters if stretched end-to-end but actually is packed and stored in the nucleus, a separate compartment in the cell with the diameter of only about 6 μm. The DNA packaging involves a family of proteins, histones, and the complex between DNA and protein is called chromatin, due to its staining property. Two each of the four types of histones – H2A, H2B, H3, and H4 – form a histone octamer, which binds and wraps approximately 1.7 turns of DNA, or about 146 base pairs. This coiling combination between DNA and histone octamer is called nucleosome, which confers a 5- to 10-fold compaction of the genomic template. In addition, one H1 protein wraps another 20 base pairs, resulting in two full turns around the octamer. The histone H1 also provides a way to join two nucleosomes with a segment of DNA (linker DNA), and interacts with core histones to form condensed fibres that further contribute to a compaction on the order of over 50 fold. This structure consisting of hundreds of thousands of nucleosomes joined by linker DNAs gives the chromosome an appearance of a string of beads when viewed using an electron microscope.

An illustration of the packaging of DNA is shown in Fig 2.2(a).

As the DNA template is highly condensed with nucleosome packaging, the local modulation of DNA accessibility thereby provides an opportunity to regulate transcription (Bell et al., 2011). Probably by modulating the DNA accessibility and controlling protein binding to DNA, the chemical modifications on the histone octamer, i.e., histone modifications (Dong et al., 2012), and methylation of DNAs (Kapourani and Sanguinetti, 2016) have been shown highly predictive of gene expression.

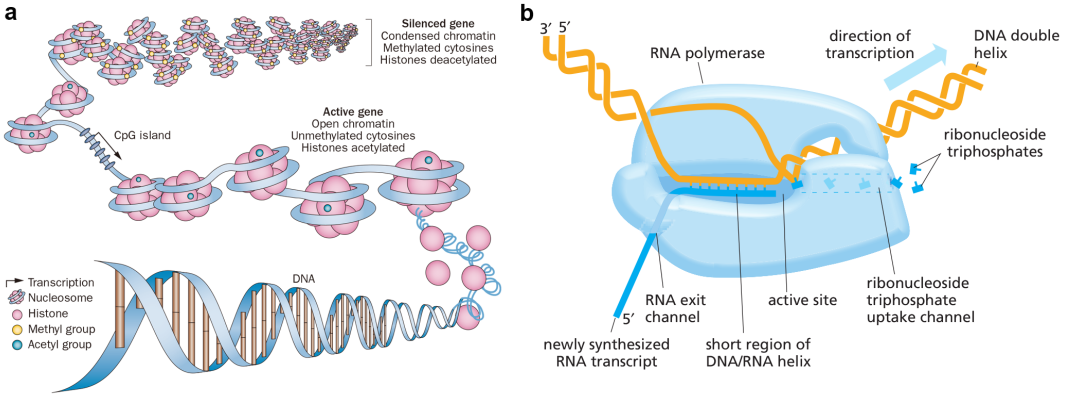

Figure 2.2: Illustration of DNA packaging and gene expression. (a) Chromatin, a complex of DNA and protein, is highly condensed with nucleosome. For active gene in terms of expression, chromatin is loosely open with depleted DNA methylation and enriched specific histone modifications. For silenced genes, chromatin is in the condensed form. Figure adapted from (Azad et al., 2013) and (b) RNA polymerase elongates along the template DAN strand, with the RNA synthesis. Figure adapted from (Alberts et al., 2008).

Transcription starts with the binding of RNA polymerase (polymerase II for protein coding genes), together with one or more general transcription factors (i.e., auxiliary proteins), to a segment of DNA sequence referred to as a *promoter*, which is near the transcription start site (TSS) of a gene. Then the RNA polymerase elongates along the template strand, with the synthesis of RNA at the same time, as shown in Fig 2.2(b). Interestingly, the elongation rate of the RNA polymerase is not static but actually highly dynamic throughout the transcription of a gene, and varies on a gene-by-gene basis (Jonkers and Lis, 2015). For example, the RNA polymerases usually pause and accumulate at very high levels near the promoter, 30-60 nucleotides downstream of the TSS. An increasing number of studies have shown that the RNA polymerase elongation rate affects co-transcriptional processes such as splicing (more details in next subsection),

termination and genome stability (Fong et al., 2014; Jonkers and Lis, 2015). Hence, the modulation of RNA polymerase elongation rate can regulate the outcome of gene expression.

2.1.2 Alternative splicing

There are several steps to precisely modulate the product and amount of gene expression, including transcription, RNA splicing, translation, and post-translational modification of a protein. As one of them, RNA splicing plays a particularly important role in controlling which distinct transcripts a gene is going to produce.

Pre-mRNA splicing The concept of gene is introduced because the biological products of DNA are discretely located on the DNA chain. In other words, the "functional" regions are interrupted by "junction" regions. Interestingly, a similar manner also exists within many genes, especially protein coding genes, namely, the DNA segments leading to protein product (coding regions) are also interrupted by the DNA segments not leading to functional product (non-coding regions). This process where non-coding sequence regions (introns) are removed and the coding sequence regions (exons) are joined together is precursor messenger RNA (pre-mRNA) splicing, one of the most interesting discoveries in molecular biology in the 20th century (Berget et al., 1977; Chow et al., 1977).

Splicing is precisely performed by the spliceosome, a complex which includes five small nuclear ribonucleoprotein particles (snRNPs) and a vast number of auxiliary protein cofactors (Wahl et al., 2009; Chen and Manley, 2009). Initially, the spliceosome accurately recognises the 5' and 3' splice sites (5'SS and 3'SS, the two boundary sites of the intron) and the branch point (BP, a site within intron for forming a circular RNA), and then finely catalyses the two-step splicing reactions. First, the upstream exon is excised, and the intron forms a circular RNA (lariat) by joining the 5'ss and branch point. Then, the intron in form of lariat is excised, and the upstream and downstream exons join together. A diagram in Fig 2.3 presents this process, and more details on spliceosome machinery can be found in (Wahl et al., 2009; Matera and Wang, 2014).

Early models of splicing envisioned a sequential step after transcription. However, the coupling between splicing machineries and transcription has been quickly observed by eletron micrograph (Beyer and Osheim, 1988), with introns being removed from nascent transcripts that are not terminated and released from DNA yet. An increasing

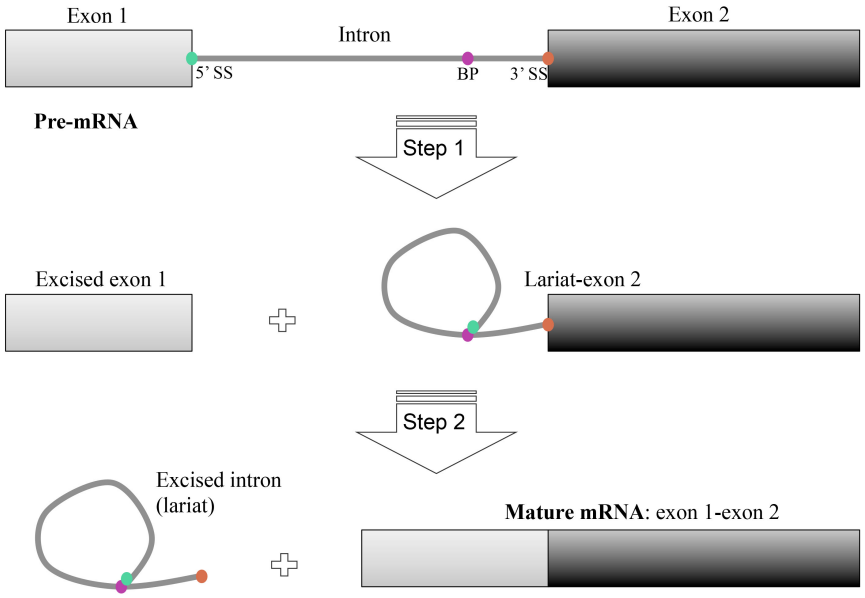

Figure 2.3: Processes of RNA splicing. Initially, the spliceosome recognises the 5' and 3' splice sites and branch point. Then the spliceosome catalyses the two-step reaction. First, the upstream exon is excised and the intron forms a lariat by joining the 5'ss and branch point. Then, the intron in form of lariat is excised, and the upstream and downstream exons are joint.

fraction of this co-transcriptional splicing has been found with the recent advances in high throughput sequencing (Tilgner et al., 2012), as well as new labelling techniques of nascent RNA (Churchman and Weissman, 2011; Windhager et al., 2012). Thus, this coupling between transcription and splicing requires an optimal rate along the transcription in order to regulate transcription and splicing simultaneously (Bentley, 2014; Jonkers and Lis, 2015).

Alternative splicing Soon after the first identification of intron, alternative splicing (i.e., different combination of exons) has been found in immunoglobulin μ gene that encodes both membrane-bound and secreted antibodies merely by distinct splicing (Alt et al., 1980; Early et al., 1980).

In most eukaryotes, a large number of genes have multiple splice sites, enabling alternative splicing. Consequently, a single gene could produce multiple protein isoforms through different combinations of exons, which significantly increases the proteome volume (Graveley, 2001; Nilsen and Graveley, 2010). The most common alternative splicing events include mutually exclusive exons, cassette exon, retained intron,

alternative 5' splice sites, alternative 3' splice sites, alternative promoters, and alternative poly-A sites (see Fig 2.4).

In human, around 95% of genes were found to undergo alternative splicing (Wang et al., 2008), with cassette exon (also known as exon-skipping event) as a major type. One of the most recognised cassette exon examples is the inclusion or exclusion of the extradomain-A exon (EDA) in Fibronectin (FN1) gene (Mardon et al., 1987; White et al., 2008). The EDA exon is skipped in liver, which results in a soluble FN1 isoform that is secreted into the plasma, whereas another isoform including EDA exon, e.g., in fibroblasts, is secreted to the extracellular matrix to regulate cell adhesion and migration.

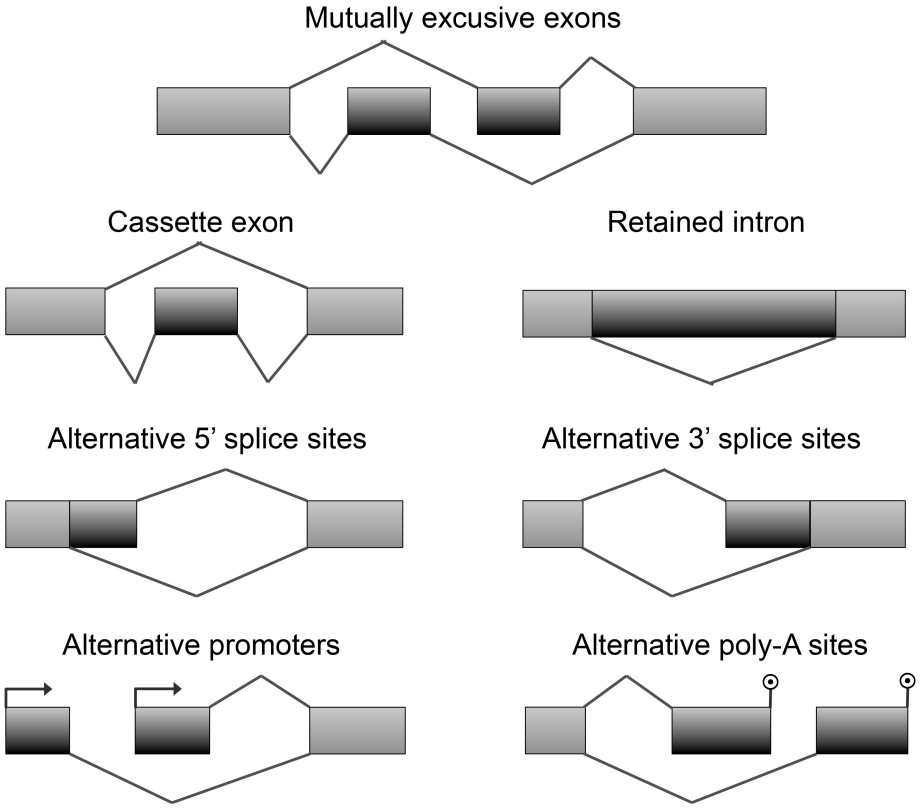

Figure 2.4: Seven common types of alternative splicing events. The rectangles denote the coding regions (exons), and the poly-lines above or below the rectangles denote two different splicing isoforms.

2.1.3 Function of splicing

Due to the existence of multiple splice sites, the spliceosome need to precisely recognise the specific splice site in a given condition. On the other hand, it also enables a

flexibility of choosing different splice sites when the condition changes. Therefore, complex and precise modulations are required for this process. RNA binding proteins are one of these regulation factors, which are required to read both local (*cis*-) or remote (*trans*-) regulatory sequence motifs accurately. False recognition of these regulatory sequences, maybe caused by genetic mutation, often causes diseases. In addition, the transcriptional machinery also has influence on splicing outcomes, as splicing is highly coupled with transcription (Tilgner et al., 2012). Increasing number of studies support the hypothesis that faster elongation gives lower "window of probability" of the exon inclusion due to the nature of co-transcriptional splicing (Bentley, 2014). Furthermore, epigenetic modifications, namely attachment of chemical groups to chromatin on either DNA or core histones, have also been found to regulate splicing patterns, for example by H3K36me3 (Luco et al., 2010). Possibly, this is because of the fact that RNA Polymerase II (Pol II) elongation rate and chromatin structure are intricately intertwined: epigenetic modifications modulate the accessibility of chromosome, and consequently affect the RNA Polymerase II (Pol II) elongation rate (Baralle and Giudice, 2017).

Comparing with the splicing mechanism, the functional consequences of splicing have been much less studied (Baralle and Giudice, 2017). Nevertheless, with the recent sequencing technology, increasingly studies have been conducted to explore the importance of splicing. Here, I do not aim to give a detailed introduction of its mechanisms in biological function, but introduce a few notable examples of splicing events and their role in controlling organ development, tissue homeostasis, and some mis-splicing related diseases.

Organismal complexity Recent sequencing experiments report similar number of genes for very different organisms, including Caenorhabditis elegans ($\sim$20,000 genes), D. melanogaster ($\sim$14,000 genes) and mammals ($\sim$20,000 genes). Mammals are believed to be more complex than nematodes or flies. The higher complexity could come from multiple layers, e.g., more complex gene regulatory networks, or bigger set of regulation elements. Very likely the higher complexity in alternative splicing also contributes to the organismal complexity. In human, the latest GENCODE release v26 (Harrow et al., 2012) has 19,817 protein coding genes, and 80,531 protein coding transcripts. Latest analysis (Nellore et al., 2016) on 21,504 human RNA-seq samples even further found 56,861 exon-exon junctions (18.6%) in at least 1000 samples that had not been annotated before.

Development It has been largely shown that alternative splicing plays a key role in cell differentiation, tissue identity, and organ development (Wang et al., 2008; Baralle and Giudice, 2017). Brain is one of the most comprehensively studied tissues where alternative splicing is one of main developmental regulators.

A transcriptome-wide comparison in alternative splicing between mouse embryonic and adult brain finds 387 splicing events that are significantly different between these two organs (Dillman et al., 2013). Among them, 123 differential splicing events (31%) during development happen in genes that do not have significant changes in total expression levels. As an example, the alternative inclusion of exon 21 in the gene *Nrxn1* has a regulatory function in nervous system, which is temporally and spatially controlled in the mouse brain mainly by the RNA binding protein *SLM2* (Iijima et al., 2011). When *SLM2* is knocked out, with observed deregulation of the alternative exon in *Nrxn1*, mice exhibit impaired synaptic plasticity and behavioural defects, but can be rescued by genetic correction via its alternative spliced region in *Nrxn1* gene (Traunmüller et al., 2016).

Mis-splicing related diseases Because alternative splicing plays an important role in regulating biological processes (Blencowe, 2006), its mis-splicing often causes serious diseases (Cáceres and Kornblihtt, 2002; Garcia-Blanco et al., 2004). There are multiple reasons that mis-splicing happens in a cell, including genetic mutation of regulatory sequences, and dysregulation of spliceosome or splicing factor (Scotti and Swanson, 2015).

In an example gene *LMNA*, mutations in different types of sequence elements can result in different pathological phenotypes. For example, 5' splice site mutation (the 5th nucleotide C mutates to G) on the 9th intron will result in intron retention, and cause a disease of limb girdle muscular dystrophy 1B (LGMD1B), while the same mutation on the 8th intron causes another disease of familial partial lipodystrophy type 2 (FPLD2) (Muchir et al., 2000; Morel et al., 2006; Scotti and Swanson, 2015).

Mis-splicing is also associated with cancers. For example, around half of active splicing events in ovarian and breast tissue change in tumours, and a large fraction of these mis-splicing events are related to the dysregulation of a single RNA binding protein FOX2 (Venables et al., 2009).

2.1.4 RNA-seq and splicing analysis

Since RNA plays a key role in transferring the genome to its biological products, it is always a fundamental task to identify transcripts and quantify their expressions. Currently, RNA-seq, a technique based on next generation sequencing (NGS), is the most widely used method to study gene expression at a genome wide scale. RNA-seq, similar to other NGS techniques, generates high-throughput short reads from all RNAs in a sample by massively parallel sequencing. This property thus provides a unique way to discover transcripts and quantify its expression simultaneously, whereas an earlier technology, micro-arrays, requires the knowledge of the sequence of the transcript to study. In the past decade, an increasing number of studies have used RNA-seq to discover transcripts (Trapnell et al., 2010), to investigate the marker genes in disease (Venables et al., 2009), to decipher genetic effects (Lappalainen et al., 2013), to study the mechanisms of RNA processing (Barrass et al., 2015), etc.

RNA-seq experiment RNA-seq experiments involve multiple steps to prepare a complementary DNA (cDNA) library before sending it into a sequencing machine, where each step has its own parameters. A nice illustration of the RNA-seq experiment is adapted from a review paper (Wang et al., 2009), and presented here in Fig 2.5.

First, a population of RNAs are isolated from a tissue or a group of cells of interest, and are mixed with deoxyribonuclease (DNase) to reduce the contamination of genomic DNA. Second, specific RNAs are selected or depleted depending on the purpose of the experiment. For example, 3' polyadenylated (poly(A)) tails are selected to only study mature mRNA, or ribosomal RNA (rRNA) is depleted as it occupies over 90% of the RNA in a cell. Third, cDNA is synthesized from the RNA via reverse transcription because DNA is more stable, and DNA sequencing technology is more mature. Strand information is lost after reverse transcription, but can be retained with chemical labelling. During this step, RNA, cDNA, or both will be fragmented, and fragment size will be selected (usually less than 500 base pairs, bp), for example, by removal of short sequence, or selection of a tight range of sequence lengths.

With cDNA library prepared, high-throughput sequencing reads either from one end (single-end sequencing) or both ends (paired-end sequencing) of the fragments will be produced from the sequencing machine, though the latter is more preferable for de novo transcript discovery or splicing analysis. The length of each end of reads is typically 30-200 bp, with the longer having higher mappability (i.e., the chance to map to a unique genome position) and transcript identification (Garber et al., 2011).

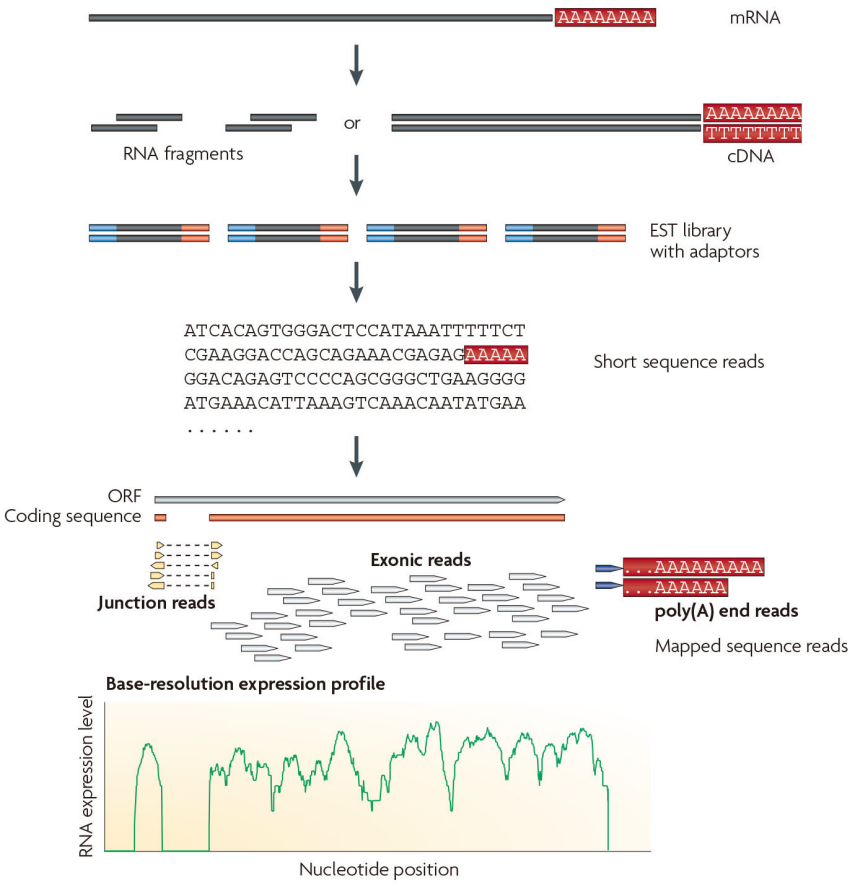

Figure 2.5: RNA-seq experiment pipeline. Usually, RNA-seq starts from cDNA preparation including RNA isolation, RNA selection, cDNA synthesis (i.e., reverse transcription) and fragmentation. Then the output sequencing reads are mapped back to either transcriptome or genome references before quantitative analysis. Figure is adapted from (Wang et al., 2009).

Another important factor of the experiment design is the sequencing depth or coverage or library size, which is the number of sequenced reads for a given sample. Given the nature of sampling strategy of RNA-seq, the deeper sequencing is expected and evidenced to give higher accuracy in identifying transcript and quantifying isoforms (Mortazavi et al., 2008). However, an optimal library size, namely number of reads just above the sufficient, is often a balance of the research purpose and the given budget, for example over 100 million reads for one sample are generated to study the novel splicing events, whereas as few as $< 100,000$ reads for one cell are generated in single cell RNA-seq experiment due to the large number of cells for study.

RNA-seq analysis pipeline The analysis of RNA-seq in identifying transcripts and quantifying gene expression requires multiple steps within an analysis pipeline. Depending on the availability of the reference genome and the reference transcriptome, there are three different analysis pipelines to construct the transcriptome from millions of raw RNA-seq reads (see the summary of these pipelines in Fig 2.6).

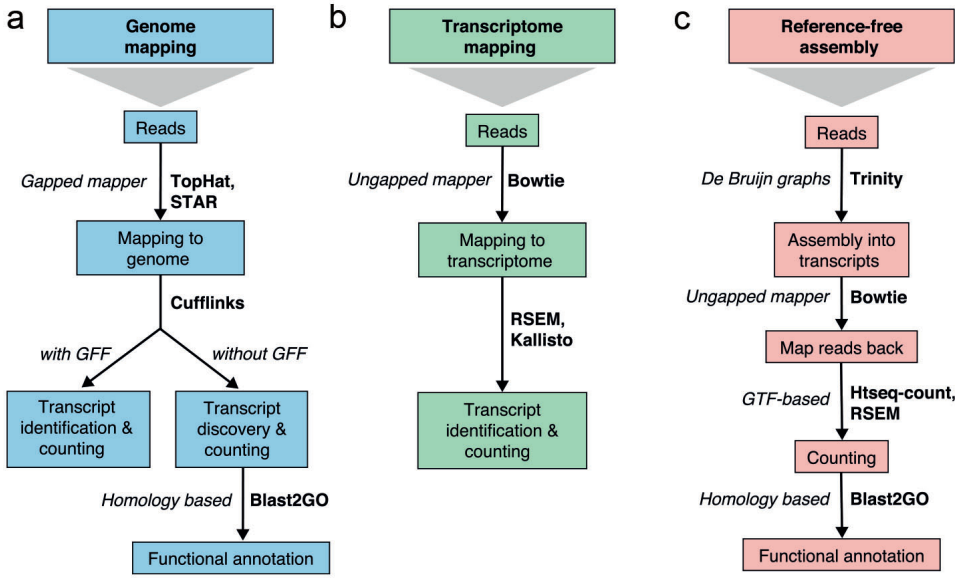

Figure 2.6: Three basic RNA-seq analysis strategies for reads mapping and isoform quantification. (a) An annotated genome is available and reads are mapped to the genome with a gapped mapper. Next (novel) transcript discovery and quantification can proceed with or without an annotation file. Novel transcripts are then functionally annotated. (b) If no novel transcript discovery is needed, reads can be mapped to the reference transcriptome using an ungapped aligner. Transcript identification and quantification can occur simultaneously. (c) When no genome is available, reads need to be assembled first into contigs or transcripts. For quantification, reads are mapped back to the novel reference transcriptome and further analysis proceeds as in (b) followed by the functional annotation of the novel transcripts as in (a). Representative software that can be used at each analysis step are indicated in bold text. Figure is adapted from (Conesa et al., 2016).

In most cases, RNA-seq reads are mapped to genome sequence reference (Fig 2.6(a)). Due to the existence of RNA splicing, many transcripts (mRNAs) cannot be continuously mapped to the genome. For the same reason, gaps often exist when mapping RNA-seq reads to the genome, and these reads are referred to as junction reads, as they

are mapped to the junction of two exons. Thus, one of the big challenges for mapping reads to genome is detecting the junctions. TopHat (Trapnell et al., 2009), one of the earliest sequence mapper accounting for gaps, uses a two-step strategy: mapping reads to the genome without gap first, and then checking the junctions between exons for the unmapped reads in the first step. Sometimes, the annotated junctions are used in the aligner to help the junction reads mapping. Recently, STAR (Dobin et al., 2013) and an enhanced version of TopHat, HISAT (Kim et al., 2015), can efficiently map the junction reads to the genome very accurately, even without the junction annotation. After mapping the reads to the genome, another big challenge is isoform quantification. Comparing to quantification at gene level only, the isoform quantification step is often needed for two reasons: first, many genes have multiple splicing isoforms, so the expression change of a specific transcript is not necessary to happen together with the change at gene level; second, the normalization of the counts to the length is more direct for transcripts than genes, as the latter may not have a precise length (isoforms have different lengths). In this step, the annotation file of the full transcripts (or the transcripts of interest) can be used to guide the direct isoform quantification, e.g., by Cufflinks (Trapnell et al., 2010) and MISO (Katz et al., 2010). Alternatively, isoform construction can be performed without an input annotation file in case it is incomplete, e.g., also by Cufflinks (Trapnell et al., 2010), so that the transcriptome can be constructed from the junction reads, which also leads to discovery of novel transcripts. In both situations, isoform quantification is difficult, because many reads mapped to the exons are shared by multiple isoforms, which direct method with counting strategy cannot use. Therefore, a statistical estimate of the fractions of these isoforms is needed to either maximize the likelihood or maximize a posteriori (for Bayesian methods). This statistical problem may be relieved if the sequencing coverage is high (i.e., more sequencing reads), so there are higher chances to see junction reads. However, in many cases, the coverages are relatively low due to the large scale of experiment samples, e.g., time-series experiments or single-cell experiments. Solving these statistical problems with Bayesian methods becomes the main contribution of this thesis (see Chapters 3-5).

Besides mapping the genome, a common alternative way is to map reads to the transcriptome sequence reference (Fig 2.6(b)), which avoid the situation of junction gap during alignment of reads to reference, e.g., by Bowtie (Langmead and Salzberg, 2012). This often requires a good annotation of transcriptome, thus it is mainly applied to a small group of well studied organisms, for example human and mouse, even

if neither annotation is perfect yet. Although the challenge in mapping gapped reads is avoided, mapping reads to transcriptome results in much higher chance to see reads mapped to multiple transcripts, which is also because some exons are shared by multiple transcripts. Thus, a similar statistical framework as MISO is also applied here, for example in BitSeq (Glaus et al., 2012) and RSEM (Li and Dewey, 2011). As the transcriptome is normally much smaller than the genome, the whole pipeline is generally faster than the one mapping to genome, especially the alignment part. Recently, researchers also tried to combine the transcriptome alignment and the isoform quantification, for example Sailfish (Patro et al., 2014) and Kallisto (Bray et al., 2016), which largely improve the computational efficiency, and allows to quantify thousands of transcriptome within a few hours on a normal desktop.

In some cases, both the genome reference and transcriptome reference are not available, or not very complete. Then the reference-free transcriptome assembly is an alternative way to identify and quantify transcripts (Fig 2.6(c)). First, the de Bruijn graphs are used to assemble the transcriptome as contigs, by using software such as Trinity (Grabherr et al., 2011). Then, with the assembled transcriptome, a similar pipeline to the transcriptome mapping can be applied to quantify the assembled transcripts.

Note, the statistical challenges in isoform quantification exist in all these three pipelines, as all scenarios have a big fraction of reads with ambiguous identity of isoforms, namely the reads that have the possibility to be originated from multiple isoforms. This challenge is particularly high when sequencing coverage is low.

Visualization Although the sequencing technology often outputs a large amount of data and requires systematic analysis, visualization of a few example genes in the genome is still very useful to better understand the story that the data tells. Visualization of RNA-seq data is generally similar to that of any other type of genomic sequencing data, and quite a few methods have already been developed to visualize it at different angles (Conesa et al., 2016). The most widely used tools are genome browsers, including the UCSC genome browser (Kent et al., 2002) and the Integrative Genomics Viewer (IGV) (Thorvaldsdóttir et al., 2013). The UCSC genome browser is a web based browser, includes a large list of species and their gene annotations. As an example, RNA-seq read densities around gene RPL28 in yeast from four time-series libraries are shown in Fig 2.7(a). As it is a web based tool, it is very convenient to visualize publicly available data sets without downloading to a local machine. Customized RNA-seq data can also be uploaded to the host, though files for sequencing data are

usually large (a few Gigabits). Another genome browser, IGV, is developed by Broad Institute. As local based software, IGV is very efficient and provides a good way to visualize multiple data sources, not only to the NGS data, but also for other data types e.g., micro-array or phenotypes data. An example of IGV is shown in Fig 2.7(c).

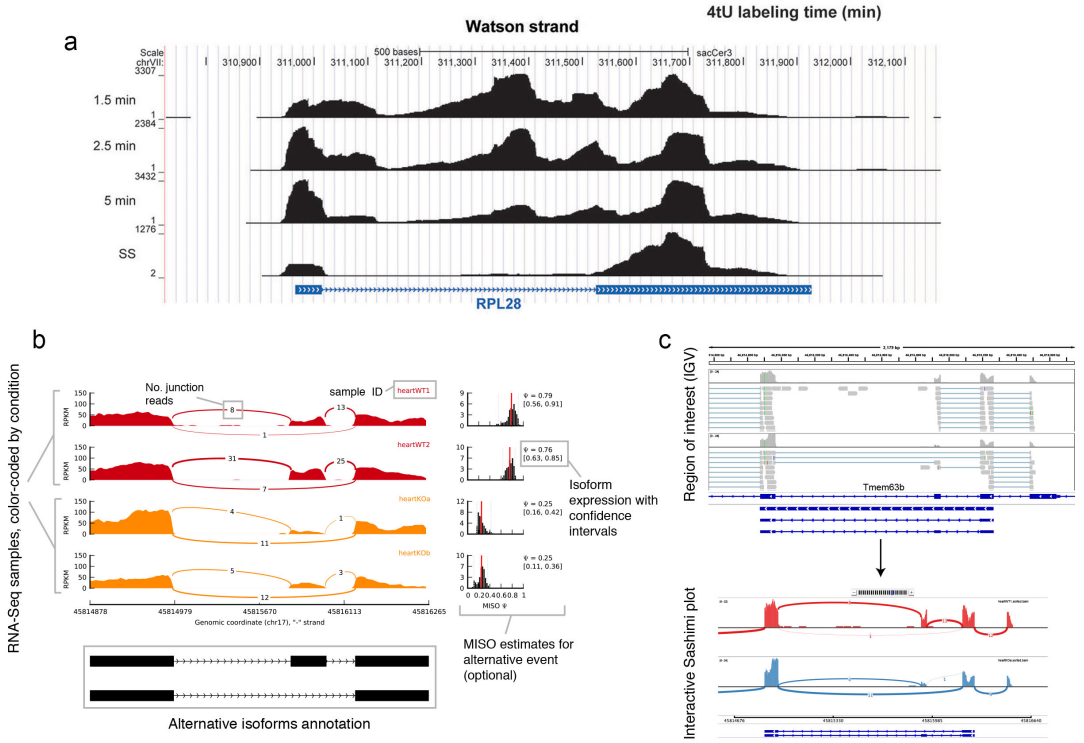

Figure 2.7: RNA-seq visualization with Genome browser or Sashimi plot. (a) Visualization of reads coverage for four RNA-seq experiments on UCSC Genome Browser for a specific genome region containing gene RPL28. The genome browser supports customized RNA-seq reads file and gene annotation. Usually, RNA-seq reads are mapped to the genome for visualization. (b) Sashimi plot (stand-alone) for alternatively spliced exon and flanking exons in four samples (colored by experimental condition). Right: optional isoform expression information produced by MISO. (c) Genomic region of interest in IGV along with two alignment tracks (top) from which a Sashimi plot is generated on the fly (bottom). Resulting Sashimi plot scales/resolution are set interactively by the user. Fig (a) is adapted from (Barrass et al., 2015), and Fig (b-c) are from (Katz et al., 2015).

Besides genome browser, a few other tools are specially designed for RNA-seq data, for example RNAseqViewer (Rogé and Zhang, 2013), which provides flexible

ways to display the read abundances on exons, transcripts and junctions, but usually slower than IGV. Another useful tool for visualizing splicing is Sashimi plot, as shown in Fig 2.7(b-c) which is adapted from the original paper (Katz et al., 2015). In Sashimi plot, the alignments of reads in exons are represented as read densities, and the splice junction reads are drawn as arcs connecting a pair of exons, where arc width is drawn proportional to the number of reads aligning to the junction. Thanks to its more intuitive and aesthetical way to display junction reads, Sashimi plot is preferred for presenting splicing events. This tool is available both as a standalone Python package, which gives publication standard figures for small scale splicing events, and also an IGV version, which can be very easily used to visualize splicing by browsing the genome.

Two direct ways of visualizing splicing are checking the existence of the exon-exon junction reads, and comparing the reads density mapping to each exon. The junction reads (see Fig 2.5) imply a splicing between the two exons, and much lower density of a certain exon than other exons means this exon is skipped. One example is shown in Fig 2.8. Mutually exclusive exon 3A and exon 3B in *SLC25A3* gene are presented for different tissues. In testes and liver, junction reads are presented between upstream exon and exon 3B, and no reads mapped to exon 3A, whereas, in skeletal muscle and heart, junction reads are shown between upstream exon and exon 3A, and very little reads mapped to exon 3B. This observation gives a binary measurement of splicing isoforms.

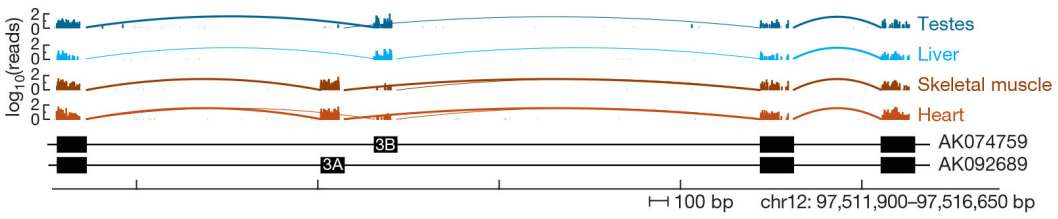

Figure 2.8: Splicing analysis with RNA-seq. RNA-seq reads mapped to *SLC25A3* gene, with density showing the number of mapped reads to exons, and bridge showing the existence of junction reads. Figure is adapted from (Wang et al., 2008)

These existing tools give great opportunities to explore results for individual genes of interest. However, due to the high complexity of the transcriptome, and the large scale of the data sources (including genomics, transcriptomics and epigenomics data)

on thousands of individuals, more efficient and effective display of these data sets is highly demanded in near future.

Challenges Though RNA-seq opens a new door for splicing analysis, this technology still has its intrinsic limitations, and brings challenges in quantification of fractions of splicing isoforms (rather than binary inclusion or exclusion), described as follows:

1. Sequencing reads are too short. At the current stage of the technology, sequenced reads in RNA-seq experiments are much shorter than almost all eukaryotic transcripts. Thus, most reads from an RNA-seq experiment cannot be unambiguously aligned to a specific splicing isoform. While in some cases a high level of coverage may obviate this problem, in many cases the number of reads that map to a single isoform is too low; when many isoforms are present, there may be no unambiguously assigned reads. In Chapter 3, I will show how statistical modelling can reduce this challenge.
2. Fragmentation and sequencing bias. Roughly, the assumption stands that sequenced reads are uniformly sampled from all transcriptome, however, when looking into local splicing events, the fragmentation and sequencing bias may bring a big fraction of false positives in detecting differential gene expression or splicing (Love et al., 2016). A few attempts have already been made to solve this challenge, for example by statistical modelling these biases from the empirical frequencies (Roberts et al., 2011; Love et al., 2016).
3. Transcriptome complexity. For different organisms, the complexity of transcriptome are different, for example, most human genes have more splicing isoforms than fission yeast. Therefore, genes with higher complexity require higher coverages and longer reads to confidently estimate splicing isoforms. Though the number of isoforms of a gene, or the overlap of these isoforms could be indicators of the complexity, a direct and accurate estimate of the complexity of the splicing structure is not available.
4. The balance between cost, coverage and samples. High coverage of sequencing is always in favour in order to get sufficient junction reads for splicing analysis. However, when it comes to time-series or single-cell experiments, another dimension also need to be considered, especially as the experiment budget is limited. In the Chapter 4 and 5, I will show how Bayesian methods could relieve this dilemma by sharing information between multiple samples.

2.2 Machine learning background

Machine learning is the study of data-driven methods to discover patterns in the data and help information processing. Depending on the problems to study, machine learning methods are usually divided into three categories: supervised learning, unsupervised learning and reinforcement learning. As reinforcement learning is not very relevant to this thesis, I will only briefly introduce the former two types of methods.

Supervised learning Supervised learning is also called predictive learning, aiming to learn a mapping f from inputs x to outputs y, given a labelled set of input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$. Here $\mathcal{D}$ is called the training set, with N data points, or training samples. The data outside of the training set is called unseen data, or test data if for evaluating the model. Given the trained model from the training set, the unknown label $\hat{y}$ of a new input $\hat{x}$ can be predicted as $f(\hat{x}|\mathcal{D})$.

The input x_i is a k-dimensional vector of numbers, for example the marks of mathematics and English exams of a student. These values are called features or attributes. In most cases, features are not independent from each other, and the structure can be complex, for example images, gene sequences, health records.

In principle, the output y_i can take any type of value. In most cases, it is either a categorical variable $y_i \in \{1, ..., C\}$, e.g., boys and girls, or a real-valued scalar, such as height. In the former case, the supervised learning is also called classification, and the model is called classifier. In the latter case with real-value scalar, it is known as regression.

One example of supervised learning problem is the prediction of gene expression from histone modifications e.g., H3K79me2 on promoter of a gene, from ENCODE project (ENCODE Project Consortium, 2012; Dong et al., 2012). For each gene, the output, gene expression, y_i can be measured by RNA-seq, and its inputs, histone modifications H3K79me2, x_i, can be measured by ChIP-seq, an NGS technology. Here, as a standard supervised learning problem, the task is learning a model from the training set to predict $\tilde{y}$ when given a test input data $\tilde{x}$. Fig 2.9 shows both a linear regression and a polynomial regression with degree 18 can be fitted into the training set of 20 genes, which can be used to predict the output for new data point. Alternatively, by setting a threshold, the continuous gene expression value can be collapsed into a binary value, i.e., expressed or unexpressed. Then this regression problem becomes a classification problem.

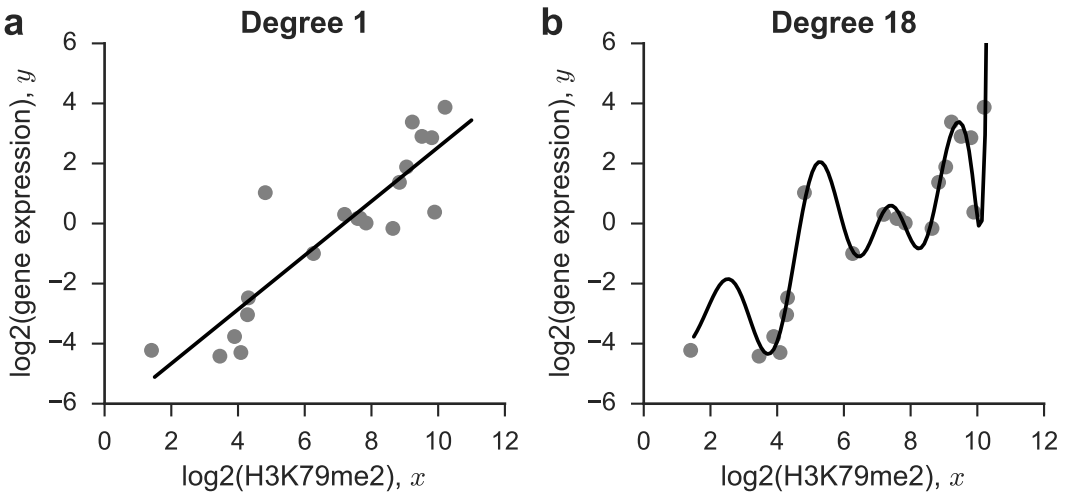

Figure 2.9: Regression on gene expression from histone modification H3K79me2. (a) linear regression model with the 1d data. (b) Same data with polynomial regression with degree 18.

Unsupervised learning Unsupervised learning is a descriptive approach, where only the input data is available, i.e., $\mathcal{D} = \{x_i\}_{i=1}^N$. In this type of task, the goal is to find some interesting structures or patterns in the input data, and therefore it is often called knowledge discovery. Different from supervised learning with knowing the desired output, unsupervised learning is more like a task of density estimation. Namely we want to estimate the parameters of a density function $p(x_i|\theta)$, which is different from the form of supervised learning $p(y_i|x_i, \theta)$ as a conditional density estimation. In addition, in supervised learning the conditional density on y_i is usually just one dimension; however, in the unsupervised learning the density on x_i is usually higher than one (even very high in some cases), which makes the task more complex.

One very widely studied task in unsupervised learning is clustering data into multiple groups. Let us look at the example of histone modification again, and this time we have two histone modifications, H3K79me2 and H3K27me3, on promoters of 200 genes, and we want to group these genes into different clusters by the distribution of the two histone modifications. Fig 2.10(a) shows that the original distribution of the two-dimensional data, and roughly it seems that there might be various ways to group these data into multiple clusters, and it is not clear how many. So the first task in clustering is determining the number of clusters, namely estimating the probability distribution $p(K|\mathcal{D})$ of the number K of clusters in the input data $\mathcal{D}$. This is difficult to choose, as we do not have direct evidence from the data. Here, we could manually set

$K = 2$ clusters, just as an example. Then the second task is to estimate the probability of which cluster a given data point belongs to. There are many heuristic methods to achieve this task, and we will discuss it in detail in latter sections. In this example, we used a K-means clustering algorithm (Murphy, 2012) to assign the 200 genes into two clusters in green and blue (Fig 2.10(b)).

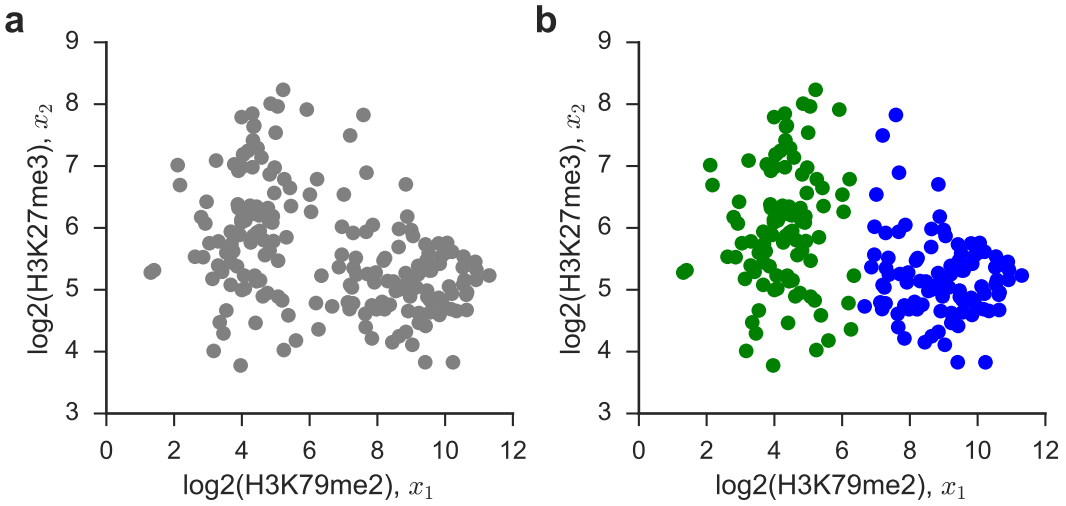

Figure 2.10: (a) The distribution of two histone modifications, H3K79me2 and H3K27me3, on 200 genes. (b) A possible clustering using K = 2 clusters by K-means. The colour green and blue denotes two different clusters.

Bias-variance tradeoff When fitting a model to the data, we need to be careful of over-fitting, especially for a highly flexible model. This problem comes from the fact that our training data set $\mathcal{D}$ is usually much smaller than the whole sample set. Thus, the model trained in the observed data set $\mathcal{D}$ may not suitable for the rest and unseen sample space. To understand this, let us look at the regression example in Fig 2.9, and we use a quadratic loss to evaluate the fitting of the model, namely the corresponding risk is the mean squared error (MSE). Now let us derive a very useful decomposition of the MSE. Assume the true function of the regression is $h(x)$, and the learned function is $y(x; \mathcal{D})$ on a given training set $\mathcal{D}$. For simplicity, let $h = h(x)$ denote the true function, $\hat{y} = y(x; \mathcal{D})$ denote a trained function from a specific data set, and $\bar{y} = \mathbb{E}_{\mathcal{D}}[y(x; \mathcal{D})]$

denote the expected function (as we vary $\mathcal{D}$). Then we have

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[[\hat{y}-h]^2\right] &= \mathbb{E}_{\mathcal{D}}\left[[(\hat{y}-\bar{y})+(\bar{y}-h)]^2\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[(\hat{y}-\bar{y})^2\right] + 2(\bar{y}-h)\mathbb{E}_{\mathcal{D}}\left[(\hat{y}-\bar{y})\right] + (\bar{y}-h)^2 \\ &= \mathbb{E}_{\mathcal{D}}\left[(\hat{y}-\bar{y})^2\right] + (\bar{y}-h)^2 \\ &= \texttt{var}\left[\hat{y}\right] + \texttt{bias}^2(\hat{y})\end{aligned} \tag{2.1}$$

In other words, the MSE can be decomposed into variance and bias, and this balance is called the bias-variance tradeoff (Bishop, 2006; Murphy, 2012). Intuitively, the bias is error from erroneous assumptions in the model. High bias can cause a model to miss the pattern in the data, or the relations between input and output (under-fitting). The variance is error from sensitivity to small fluctuations in the training set. High variance can cause over modelling the noise rather than the signal only (over-fitting). In other words, we want a model to fit the training data reasonably well, but not too much, especially not the random noise.

Let us look at the Fig 2.9 again on using regression to predict gene expression from histone modification, and now let us focus on the model complexity. Clearly, the polynomial regression with degree of 18 with a very "wiggly" curve fits most of the 20 training data points better than the linear regression (degree of 1). However, the true function is not likely to show such extreme oscillations which are probably caused by noise. Therefore, in the unseen data, for example $f(x = 2.5)$, the linear regression may gives better prediction than the polynomial regression with degree of 18.

2.2.1 Probabilistic graphical models

As briefly mentioned above, both supervised learning of $P(y|x, \mathcal{D})$ and unsupervised learning of $P(\boldsymbol{\theta}|\mathcal{D})$ can be treated from a probabilistic perspective, and in fact probabilities play a central role in modern machine learning (Bishop, 2006; Murphy, 2012). Though algebraic manipulations with probability theory can formulate and solve complicated probabilistic models, diagrammatic representations of probability distributions, called probabilistic graphical models (or simply graphical models), offer a unique way to augment the analysis, especially for models with complex structures.

There are two types of elements in a graph: *nodes* (i.e., vertices) and *links* (i.e., edges or arcs). In probabilistic graphical models, each node denotes a random variable (or set of random variables), and links between nodes represent the probabilistic relationships between these variables. With the graphical representation, the joint probability of all random variables can be decomposed into a product of factors defined

on the subset of nodes, which is a good way to express complex models compactly. There are two main categories of graphical models depending on the directionality of the links in the graphs: directed graphical models (also known as Bayesian networks) and undirected graphical models (also called Markov random fields). In this thesis, we will only use the directed graphical models, thus only introduce the directed graphical models here. For more detailed introduction for both types of graphical models, there are good books, e.g. Chapter 8 (Bishop, 2006), and good reviews, e.g., (Jordan et al., 2004).

Let us look at an example graphical model in Fig 2.11, where there are two graphical representations (left and right panels) for the same joint distribution. In the left panel, we see there are N observed variables $X = \{X_1, ..., X_N\}$ (observed variables are usually represented by nodes in shade), N unobserved variables $Z = \{Z_1, ..., Z_N\}$ (also known as latent variables), and two parameter variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The links in the graphical model show the *dependency* between variables for factorization, for example X_1 will be expressed with the dependency on Z_1 and $\boldsymbol{\beta}$; and Z_1 and Z_2 are conditionally independent given $\boldsymbol{\alpha}$. As the variables $\{X_1, ..., X_N\}$ or $\{Z_1, ..., Z_N\}$ are replications in the left panel and these replications are (conditionally) independent with each other, then a more compact representation can be achieved by using a *plate* in the right panel.

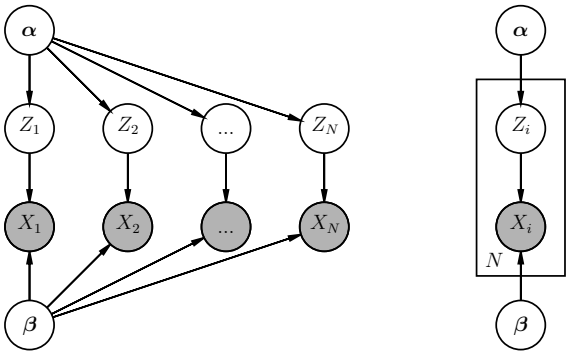

Figure 2.11: Example of probabilistic graphical model. Both panels show the same joint distribution, and the right panel is a shorthand for the left panel. In both panels, nodes in shade denote observed variables and the other nodes denote latent variables or parameters. The links show the probabilistic dependency between variables. This figure is adapted from (Airoldi, 2007).

In the probabilistic graphical models, one of the main quantities of interest is the *likelihood function*, namely the probability to measure the observed data given a set of

parameters. It can be computed using the structural hypotheses encoded in the graph, and the designed probability distributions for the nodes. In the example in Fig 2.11, we could express the likelihood as follows,

$$\begin{aligned} \mathcal{L}(X|\boldsymbol{\alpha},\boldsymbol{\beta}) \equiv P(X|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int_Z P(X,Z|\boldsymbol{\alpha},\boldsymbol{\beta})dZ \\ &= \int_Z \prod_{i=1}^{N} P(X_i|Z_i,\boldsymbol{\beta})P(Z_i|\boldsymbol{\alpha})dZ \end{aligned} \tag{2.2}$$

The *complete likelihood function*, namely the joint distribution of both observed and latent variables can be found in the integrand as $P(X, Z|\boldsymbol{\alpha}, \boldsymbol{\beta})$, which is also an important quantity in graphical models.

Given the graph structure and the observed data, we are interested in either *estimating* the parameters or *inferring* the distributions of latent variables. Both tasks are largely based on the manipulations on the likelihood functions or with adding specific prior distributions (more details in the following sections). For models in some special families, exact inference is available, for example by setting the derivative of the likelihood function to zeros and calculate the solution, as follows,

$$\frac{\partial \log \mathcal{L}(\mathcal{D}|\theta)}{\partial \theta} = 0 \tag{2.3}$$

where this estimate of the parameters achieves the maximum likelihood and thus is called maximum likelihood estimate (MLE or ML estimate).

However, in more cases, the likelihood function is intractable, for example the integral in Eq (2.2) cannot be solved in closed form. Therefore, we need to use approximation methods. There are three widely used stratagies in approximations for graphical models: Markov chain Monte Carlo (sampling-based), expectation-maximum (EM) and variational methods (optimization-based).

Markov chain Monte Carlo (MCMC) techniques are a family of sampling-based methods that exploit the property of Markov chains, which are effective to generate samples for high-dimensional distributions, and largely used in Bayesian inference for posterior distributions. They will be discussed with more details in Chapter 2.2.4.

The other two alternative methods aim to approximate the integral in Eq (2.2). Though EM algorithm and variational methods are very different from each other, they still can be viewed to share a common idea of finding a lower bound for the likelihood $\mathcal{L}(X|\boldsymbol{\alpha}, \boldsymbol{\beta})$, by using Jensen's inequality (in the third line in Eq (2.4)) and choose an

arbitrary distribution on the latent variable $q(Z)$, as follows,

$$\begin{aligned}\log \mathcal{L}(X|\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \log \int_Z P(X, Z|\boldsymbol{\alpha}, \boldsymbol{\beta}) dZ \\ &= \log \int_Z q(Z) P(X, Z|\boldsymbol{\alpha}, \boldsymbol{\beta})/q(Z) dZ \\ &\geq \int_Z q(Z) \log[P(X, Z|\boldsymbol{\alpha}, \boldsymbol{\beta})/q(Z)] dZ \\ &= \mathbb{E}_q[\log P(X, Z|\boldsymbol{\alpha}, \boldsymbol{\beta})] + H[q(Z)] \equiv L(q, \boldsymbol{\theta})\end{aligned} \tag{2.4}$$

where $H[q(Z)] = -\int q(Z) \log q(Z) dZ$ is the entropy of $q(Z)$, and $L(q, \boldsymbol{\theta})$ is the lower bound of the original likelihood. Here, we use the $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ to denote the parameter set.

In EM algorithm, the lower bound $L(q, \boldsymbol{\theta})$ is iteratively maximized with respect to $\boldsymbol{\theta}$ in M-step, and $q(Z)$ in E-step. Specifically, in the t-th step $q(Z)$ uses the posterior distribution of the latent variable in the previous step, as follows,

$$q^{(t)}(Z) = P(Z|X, \boldsymbol{\theta}^{(t-1)}) \tag{2.5}$$

In the next section, I will describe the full derivations of the EM algorithm for mixture models, especially for Gaussian mixture models.

For some complex models, the distribution $q^{(t)}(Z)$ is impossible to be described analytically. Therefore, a parametric approximation $\tilde{q} \equiv q_\Delta(Z)$ can be used to approximate the lower bound of the likelihood $L_\Delta(q, \boldsymbol{\theta})$. In each iteration, the variational parameters Δ will be optimized to minimize the Kullback-Leibler divergence between $q^{(t)}$ and $q_\Delta^{(t)}$, which is equivalent to maximize the low bound for the likelihood $L_\Delta(q, \boldsymbol{\theta})$. This is the basic idea of variational methods, which are very useful in complex models, especially for reducing the computation cost. However, in this thesis, I will mainly use the MCMC sampling methods and discuss performances of MCMC samplers and EM algorithms, therefore I stop further discussion of variational methods here.

2.2.2 Mixture models

In this section, let us look at an instance of probabilistic graphical models, mixture models, which will form the basis of the statistical modelling of RNA splicing from RNA-seq data in this thesis.

In mixture models, we assume that the observed data points come from multiple (usually simple) distributions. One simple example is that the overall joint distribution

of weights and heights in a class comprising boys and girls is a mixture distribution, as boys and girls have different distributions over these two variables. These multiple basis distributions in mixture models are also called multiple components. The probability to observe a sample $\boldsymbol{x}$ in the mixture model is the summary of the probability to see this data point over all components. Depending on the basis distributions, there are a variety of mixture models, for example Gaussian mixture models with Gaussian basis distribution, binomial mixture model with binomial basis distribution. RNA-seq reads distribution on a gene with splicing can be also modelled by a mixture model, with uniform distribution with distinctive gaps as the basis function (details in Chapter 3).

Without lack of generality, we use the introduction of a widely studied mixture model, multivariate Gaussian mixture models (GMM), to present the ideas of mixture models and to derive the EM algorithm for estimating the parameters. These ideas can be easily extended to other type of mixture models. Now let us start with formulating the probability distribution of Gaussian mixture model as follows,

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k) \tag{2.6}$$

where K is the number of different Gaussian distributions, and $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$ is the parameter set, mean and covariance matrix, of the kth Gaussian model. π_k is the coefficient of the kth component, denoting the probability that samples come from this component, and it also satisfies

$$\sum_{k=1}^{K} \pi_k = 1; \tag{2.7}$$

where $0 \leq \pi_k \leq 1, k \in \{1, ..., K\}$.

Given a set of observed data $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, we want to estimate all parameters of the mixture model $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\} = \{\pi_1, \boldsymbol{\mu}_1, \Sigma_1, ..., \pi_K, \boldsymbol{\mu}_K, \Sigma_K\}$ by MLE. Here, the log likelihood of the mixture Gaussian model could be expressed as follows,

$$\log \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log p(\mathcal{D}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \tag{2.8}$$

However, the derivation of the logarithm likelihood function with respect to the parameters is hard, as a summation over K components appears inside the logarithm in Eq. (2.8), thus the logarithm function no longer acts directly on the Gaussian. When setting the derivatives of the log likelihood to zero, the closed-form solution cannot be obtained any longer.

In order to better understand this model, auxiliary variables $\boldsymbol{z} = \{z_1, ..., z_N\}$ denoting the identity of each data point, namely the component that a certain sample origins. For each data point, z_i is a vector of K binary values, $z_i = \{z_{i1}, ..., z_{iK}\}$, and if the sample i comes from component j, then $z_{ij} = 1$ and $z_{ik} = 0, k \neq j$. As these variables z_i cannot be observed (missing data), they are often called latent variables. A graphical representation of the Gaussian mixture model is shown in Fig 2.12, where arrows show the dependency between variables.

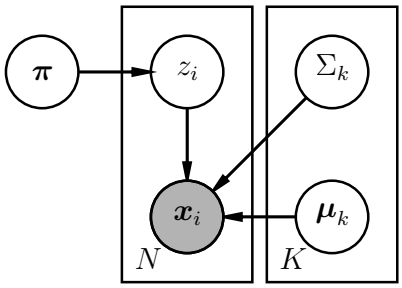

Figure 2.12: Graphical representation of Gaussian mixture model with latent variables. Each node denotes a variable (vector), and nodes with shade mean visible variable, otherwise latent variable or parameters. Here, a set of data points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ are observations and its corresponding latent variable z_i denotes the component from which it comes.

Therefore, we could derive the likelihood of the GMM by marginalizing $\boldsymbol{z}$ on the complete log likelihood $\log \mathcal{L}(\mathcal{D}, \boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$, and then get the same form as Eq (2.8), as follows,

$$\begin{aligned} \log p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) &= \sum_{i=1}^{N} \log \sum_{k=1}^{K} p(\boldsymbol{x}_i, z_{ik}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ &= \sum_{i=1}^{N} \log \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_i|\mu_k, \Sigma_k) p(z_{ik}|\pi_k) \end{aligned} \tag{2.9}$$

where the prior distribution of latent variable is $p(z_{ik} = 1) = \pi_k$. With known identity, the conditional distribution of $\boldsymbol{x}$ becomes a single Gaussian distribution $p(\boldsymbol{x}_i|z_{ik} = 1) = \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, or $p(\boldsymbol{x}_i|z_{ik} = 0) = 0$. With Bayes theorem (see Bayesian statistics in next section), we could obtain the posterior of the latent variable as follows,

$$\begin{aligned}\gamma(z_{ik}) = p(z_{ik}=1|\boldsymbol{x}_i) &= \frac{p(z_{ik}=1)p(\boldsymbol{x}_i|z_{ik}=1)}{\sum_{j=1}^{K} p(z_{ij}=1)p(\boldsymbol{x}_i|z_{ij}=1)} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j \mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}\end{aligned} \tag{2.10}$$

where the posterior probability $\gamma(z_{ik})$ denotes the responsibility that component k takes for data point $\boldsymbol{x}_i$.

It seems introducing latent variables makes the model more complex, however it actually simplifies the expression of derivations and inspires an iterative algorithm to reach the maximum likelihood.

Before going into the detailed algorithm, let us look at the conditions that must be satisfied at a maximum of the likelihood function. First the derivatives of log likelihood of $\log \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to the mean $\boldsymbol{\mu}_k, k \in \{1, ..., K\}$ of each Gaussian component needs to be zero, thus we have

$$\begin{aligned}0 &= \sum_{i=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j \mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma_k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) \\ &= \sum_{i=1}^{N} \gamma(z_{ik})\boldsymbol{\Sigma_k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\end{aligned} \tag{2.11}$$

By multiplying by $\boldsymbol{\Sigma_k}^{-1}$, which we assume is invertible, and rearranging the equation, then we have

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma(z_{ik})\boldsymbol{x}_i \tag{2.12}$$

where we define N_k as the effective number of points assigned to cluster k, as follows,

$$N_k = \sum_{i=1}^{N}\gamma(z_{ik}) \tag{2.13}$$

Similarly, we could set the derivative of the log likelihood with respect to $\boldsymbol{\Sigma_k}$ to zero, and have the conditions that $\boldsymbol{\Sigma_k}$ need to satisfy for the maximum likelihood, as follows,

$$\boldsymbol{\Sigma_k} = \frac{1}{N_k}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \tag{2.14}$$

Finally, we maximize the log likelihood with respect to coefficients $\boldsymbol{\pi_k}$. However, here we need to consider the constraint in Eq (2.7) that the mixing coefficients sum to one. Thus, we add a Lagrange multiplier and maximizing the following quantity

$$\log p(\mathcal{D}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right) \tag{2.15}$$

so $\boldsymbol{\pi}$ needs to satisfy

$$0 = \sum_{i=1}^{N} \frac{\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda \Rightarrow 0 = \sum_{i=1}^{N} \gamma(z_{ik}) + \lambda \pi_k \tag{2.16}$$

Here, we have K equations from Eq (2.16) and an additional equation (Eq (2.7)), thus it is sufficient to get a closed form solution for these $K+1$ variables. By summarizing both sides of the K equations in Eq (2.16), we have $\lambda = -N$, and by taking this back to Eq (2.16) we could further have

$$\pi_k = \frac{N_k}{N} \tag{2.17}$$

Note, Eq (2.12, 2.13, 2.14, 2.17) do not mean that we obtain a closed form of the maximum likelihood estimate, because we used the auxiliary latent variable $\gamma(z_{ik})$ that itself contains parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}$. However, this representation provides an iterative way to reach the maximum likelihood, which is called expectation maximization, or EM for short (Dempster et al., 1977; Meng and Van Dyk, 1997). Intuitively, EM is an iterative algorithm which alternates between inferring the missing values $\boldsymbol{z}$ given the parameters (E step), and then optimizing the parameters given the "filled in" data (M step). The pseudo code of the algorithm is presented in Algorithm 1.

Algorithm 1: EM algorithm for Gaussian mixture model

Initialize $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}$ and evaluate $\log \mathcal{L}(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$
while *not converged* **do**
  E step: Calculate $\gamma(z_{ik})$ with current parameters
  $\gamma(z_{ik}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$ and $N_k = \sum_{i=1}^{N} \gamma(z_{ik})$
  M step: Maximizing likelihood on parameters with current responsibilities
  $\boldsymbol{\mu}_k^{\texttt{new}} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(z_{ik}) \boldsymbol{x}_i$
  $\boldsymbol{\Sigma}_k^{\texttt{new}} = \frac{1}{N_k} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{\texttt{new}})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{\texttt{new}})^\top$
  $\pi_k^{\texttt{new}} = \frac{N_k}{N}$
  Update $\log \mathcal{L}(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$ and check convergence
return $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}, \boldsymbol{z}$

More specifically, EM algorithm starts from initializing all parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$, which can be set randomly or with a particular guess. By setting these parameters, the initial log likelihood can be evaluated. Then iterations will be repeated until the algorithm converges. Within each iteration, there are three steps. The first is the E

step to calculate the expectation of the complete likelihood with latent variables, and the posterior probability of the latent variable $\gamma(z_{ik})$ by using the current parameters; then the second is the M step to maximize the likelihood with respect to the parameters by using current latent variables $\gamma(z_{ik})$. In addition, with the updated parameters, we re-evaluate the log likelihood, which should be higher than the previous step. Finally, we check the convergence, for example by checking whether the improvement of the updated log likelihood from the previous one is smaller than a given threshold. If yes, then the iteration can be stopped and return the last parameters and latent variables.

Note: though the log likelihood increases during the iterations in the EM algorithm, this method can only guarantee an approximation of local maximum of likelihood, rather than the global maximum. Briefly, this is because in the M step, the maximization of the likelihood with respect to the parameters are based on their gradient, consequently, the EM algorithm can stop at any point with zero gradient, e.g., a local maximum.

Before the end of this subsection, let us look at an example on applying Gaussian mixture model to the two-dimensional histone modifications data that we showed in Fig 2.10. With GMM, Fig 2.13 shows the contour lines of the negative log likelihood of each single $\boldsymbol{x}$ with the estimated parameters. By using the latent variable, the 200 genes are grouped into two clusters: green and blue. Note: this clustering result is slightly different from the one by K-means in Fig 2.10, and the GMM latent variables give the posterior probability that a sample belongs to a cluster (denoted by the transparency of each dot), which is called soft clustering.

Although this section of mixture models mainly focuses on the widely used GMM, the ideas of introducing latent variables, using graphical representation and applying EM algorithm can be applied to many other mixture models with straightforward modifications, including the mixture modelling of RNA splicing in following chapters.

2.2.3 Bayesian statistics

Bayesian statistics, named after Thomas Bayes ($1701-1761$), is a sub-field of statistics, where the probability is used to quantify our uncertainty about variables, i.e., the degrees of belief. This interpretation is called the Bayesian interpretation of probability, which is different from the frequentist interpretation. Bayesian probability is fundamentally related to information rather than repeated trials (Murphy, 2012). For example, the Bayesian probability of 0.5 in the coin toss is just our belief that the

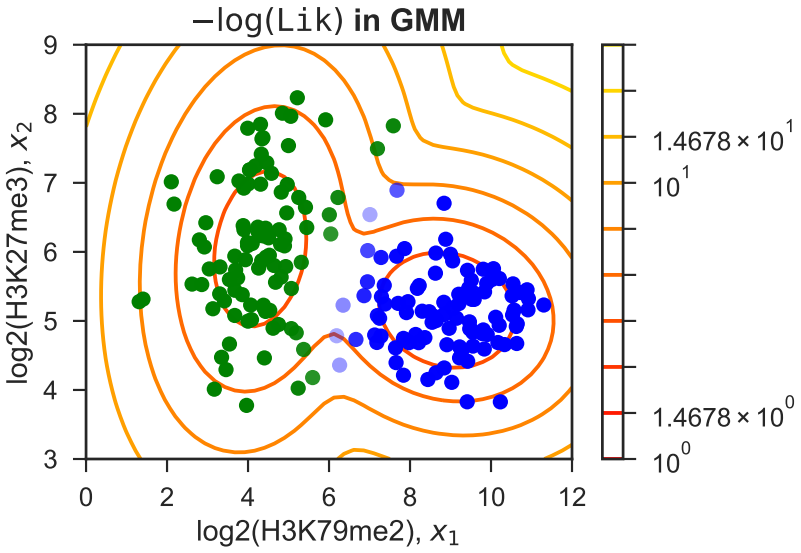

Figure 2.13: Application of Gaussian mixture model on two-dimensional histone modification data. The colour green and blue denotes two different Gaussian components, and the transparency of the data points denotes the probability that it comes from the according cluster. The contours denotes the negative log likelihood for a given $\boldsymbol{x}$ with estimated parameters.

coin is equally likely to land heads or tails on the next toss, while from the frequentist perspective, it is about the frequency of 50% to have coin toss in a large number of trials.

Bayes theorem One of the key ideas of Bayesian statistics is that the original belief (prior probability) can be revised in the light of relevant new information, as expressed in the posterior probability, using Bayes theorem or Bayes rule, as follows,

$$p(Y = y|X = x) = \frac{p(X = x, Y = y)}{p(X = x)} = \frac{p(X = x|Y = y)p(Y = y)}{\sum_{\hat{y}} p(X = x|Y = \hat{y})p(Y = \hat{y})} \tag{2.18}$$

where $p(Y = y)$ and $p(Y = y|X = x)$ are called prior and posterior probabilities, respectively. The conditional probability $p(X = x|Y = y)$ is sometimes called likelihood if Y is a set of parameters of a model. The normalization term, $p(X = x)$, is often called evidence.

Two distinctive properties in Bayesian statistics are the addition of a prior distribution, and that we care more about posterior than likelihood. In the example of Gaussian mixture model, Bayesian estimation will introduce a prior distribution on the parame-

ters and focus on the posterior, as follows,

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\mathcal{D}, \boldsymbol{\phi}) = \frac{p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\boldsymbol{\phi})}{p(\mathcal{D})} \tag{2.19}$$

where $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\boldsymbol{\phi})$ is the introduced prior distribution and parameters $\boldsymbol{\phi}$ of the prior distribution are called hyper-parameters. From the Bayesian view, we are interested in the whole posterior distribution, though in practice the estimates may take the values for maximum a posteriori (MAP estimate), or the mean of the posterior distribution. As these prior distributions may have specific form in terms of dependency on other variables, graphical representation is often used to describe Bayesian models. Indeed, parameters of prior distribution may themselves have prior distributions, leading to Bayesian hierarchical modelling, or may be interrelated, leading to general Bayesian networks.

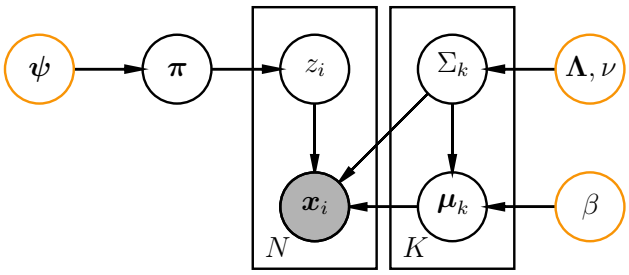

Figure 2.14: Graphical representation of Bayesian Gaussian mixture model. Each circle denotes a variable (vector): circle with shade means visible variable, otherwise latent variable or parameters. Orange circle means hyperparameters.

The example of the graphical representation of a Bayesian GMM is shown in Fig 2.14, where we applied a `Dirichlet` prior distribution to $\boldsymbol{\pi}$ with hyper-parameter $\boldsymbol{\psi}$, a `Wishart` prior distribution to the inverse Σ_k with hyper-parameters $\boldsymbol{\Lambda}, \nu$, and a Gaussian prior distribution to $\boldsymbol{\mu}_k$ with hyper-parameters β. Therefore, the posterior in Eq (2.19) could be extended as follows,

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\mathcal{D}, \boldsymbol{\phi}) = \frac{p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\boldsymbol{\Lambda}, \nu, \beta, \boldsymbol{\psi})}{p(\mathcal{D})} \tag{2.20}$$

where by using the designed dependency and prior distributions, the joint prior distri-

bution can be further expressed as follows,

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}|\boldsymbol{\Lambda}, \nu, \beta, \boldsymbol{\psi}) &= p(\boldsymbol{\pi}|\boldsymbol{\psi})p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{\Lambda}, \nu, \beta) \\ &= \texttt{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\psi}) \prod_{k=1}^{K} \texttt{Wishart}(\Sigma_k^{-1}|\boldsymbol{\Lambda}, \nu)\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{0}, \beta\Sigma_k) \end{aligned} \tag{2.21}$$

Benefits Bayesian inference, focusing on the posterior rather than the likelihood, provides a balance between the prior distribution and the likelihood that the data supports. Thus, this method brings two important benefits. First, it allows to integrate background knowledge by choosing a sensible prior distribution and hyper-parameters. This is particularly useful for fitting a model on a small data set, because over-fitting often happens when fitting a very flexible model to a small data set. As the small training set cannot represent the full distribution well, fitting a flexible model tightly to the small training set can cause a high variance in prediction on new data (when training set changes). However, if we have a sensible prior distribution, even with a very broad prior distribution on the parameters when our belief is not very strong, the overfitting to the small data set will be relieved thanks to the balance provided by the prior knowledge.

Secondly, if there are multiple data sources, e.g., by multiple experiments, Bayesian methods can provide a way to integrate these data sources by a special structure in the prior distribution, usually represented by hierarchical model. For example, if we have multiple data sources of Gaussian mixture models, and all data sets share the same frequencies of components $\boldsymbol{\pi}$, then we could build a hierarchical model for these multiple data sources jointly with a shared hyper-parameters on $\boldsymbol{\pi}$. This integration of multiple data sources can be very useful for transferring information from one data set to another, especially if the sizes of these data sets are not even.

Here, you may argue that when the Bayesian method reduces the variance in fitting the model, it will increase the bias, because the bias and the variance are always a tradeoff. Yes, Bayesian methods usually indeed increase the bias in fitting a model, but they usually reduce the variance in a larger scale, thus reducing the overall errors in the model. Actually, we usually choose sensible prior distributions: strong prior distributions are only employed if we have a very confident domain knowledge for the specific cases, otherwise broad prior distributions are preferred. More importantly, another distinctive and important property of Bayesian methods is that the MAP is a consistent estimate to MLE if sample size is large enough. This is simply because the likelihood term depends exponentially on the sample size N, and the prior stays

constant. Thus, when we get more and more data, the MAP estimate converges towards the MLE which only uses the likelihood term.

Choice of prior The choice of a proper prior distribution for Bayesian statistics is not an easy task. Sometimes, if we do not have a strong belief about what the parameter θ should be, it is common to use an uninformative or non-informative prior, and to let the data speak for itself.

When the posterior $p(\theta|\mathcal{D})$ and the prior $p(\theta)$ have the same form, we say that the prior is a conjugate distribution for the corresponding likelihood. For example, if the likelihood function is a Gaussian distribution, then by adding a Gaussian prior over the mean will return Gaussian distribution for the posterior. This means that the Gaussian distribution is a conjugate prior for the Gaussian likelihood, namely the Gaussian family is conjugate to itself (or self-conjugate). Conjugate priors are often in favour because of their algebraic convenience, which gives a closed-form expression for the posterior. Conjugate priors are often easier to interpret than other priors, and may give intuitions.

Besides, domain knowledge inspired priors can also be very useful in Bayesian statistics. In the splicing modelling problem, I will introduce different structured priors in differently designed experiments, and show the benefits by adding them.

2.2.4 Markov chain Monte Carlo

Monte Carlo In general, computing important statistics or general functions of random variables can be difficult because it may requires integrations. One simple but powerful approximation is to generate S samples, $x_1, ..., x_S$, from a given probability distribution $p(X)$. Then we could approximate the function $f(X)$ by using the empirical distribution of $\{f(x_s)\}_{s=1}^{S}$. This sampling strategy is called a **Monte Carlo** approximation, named after a city in Europe known for its plush gambling casinos.

The Monte Carlo method can be used to approximate the expected value of any function of a random variable. We simply draw a set of samples following its distribution, and then compute the arithmetic mean of the function applied to the samples. This can be written as follows:

$$\mathbb{E}[f(X)] = \int f(x)p(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} f(x_s) \tag{2.22}$$

where $x_s \sim p(X)$. This is called Monte Carlo integration. By changing the function

$f()$, we can have many important statistics, for example, the mean $\mathbb{E}[X]$ if $f(x) = x$, and the variance $\texttt{var}[X]$ if $f(x) = (x - \bar{x})^2$.

This sampling strategy is particularly useful in Bayesian statistics, as there are often many integrations over a large number of unknown parameters in the hierarchical models, for the purposes of normalisation, marginalisation and expectation on a certain distribution. Furthermore, if we have obtained sufficient samples, then we could have an approximation of the whole distribution, rather than a point estimate of the parameters either by ML or MAP estimate.

For a computer, a uniform distribution can be easily sampled, for example by linear congruential generator, which is a simple but effective algorithm for generating pseudorandom numbers. Given the samples of uniform distributions, sampling for some standard distributions, such as Gaussian distribution, is also straightforward by inverse transform sampling, namely by taking its inverse cumulative distribution function (CDF) on the uniformly distributed samples. However, in many more cases, the direct expression of the probability function is hard, for example when the posterior distribution has a normalization term, or when the dimension of a distribution is very high; thus all these distributions are very hard to sample. Here, we introduce the Markov chain Monte Carlo (MCMC) method, a family of sampling algorithms exploiting the properties of Markov chain, to solve these challenges. There are also many detailed introductions of MCMC methods, for example (Murphy, 2012; Andrieu et al., 2003).

Markov chain The MCMC algorithm is based on the mechanism of Markov chain. Here, we introduce Markov chains on finite states, where a variable $x^{(n)}$ in any step can only take S discrete values, i.e., $x^{(n)} \in X = \{1, 2, ..., S\}$. The stochastic process $\{x^{(n)}, n \geq 0\}$ is a Markov chain if it satisfies,

$$p(x^{(n)}|x^{(n-1)}, ..., x^{(1)}) = p(x^{(n)}|x^{(n-1)}) = P^{(n)}(x^{(n)}|x^{(n-1)}) \tag{2.23}$$

where $P^{(n)}$ is the transition matrix for the nth step with $P^{(n)}_{i,j} = P^{(n)}(j|i)$ denoting the probability of jumping from state i to state j, and it satisfies $\sum_{x^{(n)}} P^{(n)}(x^{(n)}|x^{(n-1)}) = 1$. We also assume the transition matrix is invariant for all steps, i.e., the chain is *homogeneous*, and the transition matrix can be denoted by P. Therefore, with fixed transition matrix, the transition of the nth step in the Markov chain only depends on values of its previous step.

A very useful property in Markov chain is that it may have *stationary distribution* (i.e., invariant distribution) on the finite states. For a homogeneous Markov chain with

transition probabilities P, the stationary distribution $\boldsymbol{\pi} = \{\pi_1, ..., \pi_S\}$ will satisfy,

$$\boldsymbol{\pi} = \boldsymbol{\pi} P \tag{2.24}$$

This means that the stationary distribution $\boldsymbol{\pi}$ is able to keep the same after a jump via the transition matrix. Namely, once a Markov chain enters the stationary distribution, it will never leave.

Before we discuss the existence of stationary distribution, let us look at two important properties of Markov chain: *reducibility* and *periodicity*. The *irreducible* Markov chain means that we can get from any state to any other state. Formally, it means for any state i and j there exists an integer $n_{ij} \geq 0$, such that

$$P(x_{n_{ij}} = j | x_0 = i) = P_{i,j}^{n_{ij}} > 0$$

Then, let us look at periodicity. A state i has period k if any return to state i must occur in multiples of k time steps. If $k = 1$, the state will be called *aperiodic* state. Further, if all states are aperiodic, we call the Markov chain aperiodic. For an irreducible and aperiodic Markov chain, the existence of the stationary distribution can be guaranteed by the following theorem.

Theorem 1 (Markov chain theorem) *If a homogeneous Markov chain has a transition matrix P, and this Markov chain is irreducible and aperiodic, then $\lim_{n\to\infty} P_{ij}^n$ exists and does not depend on i, which can be written as $\lim_{n\to\infty} P_{ij}^n = \pi(j)$.*

This theorem means from a random initial state distribution $\boldsymbol{\pi}^{(0)}$, after a certain number transitions, say $\tilde{n}$, the Markov chain will enter the stationary distribution, thus we have

$$\boldsymbol{\pi}^{(\tilde{n}+m)} = \boldsymbol{\pi}^{(\tilde{n})} \Leftrightarrow \boldsymbol{\pi}^{(0)} P^{\tilde{n}+m} = \boldsymbol{\pi}^{(0)} P^{\tilde{n}} \Leftrightarrow P^{\tilde{n}+m} = P^{\tilde{n}}, m > 0 \tag{2.25}$$

In other words, we have the stationary distribution as follows,

$$\boldsymbol{\pi} = \{P_{i,j}^n, j = 1, ..., S\} \text{ for any i and } n \geq \tilde{n} \tag{2.26}$$

This means that the Markov chain can converge to the stationary distribution within a certain steps, which plays a fundamental role in MCMC simulation. Therefore, from a random initial state distribution $\boldsymbol{\pi}^{(0)}$, we transit the state along the Markov chain with the transition matrix P. Then after convergence from $\tilde{n}$th step, the samples $\{x^{(\tilde{n})}, x^{(\tilde{n}+1)}, x^{(\tilde{n}+2)}...\}$ will follow the same stationary distribution $\boldsymbol{\pi}$, and can be used

to perform downstream analysis, although these samples are not independent. Very importantly, this property in convergence also stands in continuous state space, which makes the MCMC algorithm also applicable for sampling in continuous distribution space.

Metropolis-Hastings sampling Given a distribution $p(x)$, we want to efficiently generate samples that follow this distribution. Then one question is whether we could make up a transition matrix P so that it has a stationary distribution that we want to sample. This wonderful idea was proposed by Metropolis in 1953 to study the state for substances consisting of interacting individual molecules (Metropolis et al., 1953). This method first introduced the Markov chain into Monte Carlo sampling, and inspired a series of MCMC methods, boosting the development of the simulation technology. Therefore, the Metropolis algorithm is ranked as top 10 algorithm during 20th century (Dongarra and Sullivan, 2000).

Here, we introduce the Metropolis-Hastings algorithm (Hastings, 1970), a variant of the Metropolis algorithm, for generating samples from a high dimensional distribution. From the above Markov chain theorem, we see that the convergence of a Markov chain is determined by the transition matrix, so the challenge for sampling from a Markov chain is the design of a suitable transition matrix. In order to design a transition matrix, let us look at a sufficient (but not necessary) condition for the stationary distribution.

Theorem 2 (Detailed balance condition) *For an irreducible and aperiodic Markov chain with a transition matrix P, if a distribution $\boldsymbol{\pi}$ satisfies*

$$\boldsymbol{\pi}(i)P_{ij} = \boldsymbol{\pi}(j)P_{ji}, \textit{ for all } i, j \tag{2.27}$$

then $\boldsymbol{\pi}$ is the stationary distribution.

This detailed balance condition is easy to understand intuitively. The probability lost from state i to state j is exactly equal to the probability supplement jumping back from state j to i, therefore the probability of each state is stable, and $\boldsymbol{\pi}$ is the stationary distribution of the Markov chain.

Assume we have a Markov chain with transition matrix Q, and use $Q(i, j)$ to denote the probability from state i to j. Usually, the detailed balance condition is not satisfied

$$\boldsymbol{\pi}(i)Q(i, j) \neq \boldsymbol{\pi}(j)Q(j, i)$$

Thus, the target distribution $p(x)$ is probably not the stationary distribution of this Markov chain. However, we could introduce an additional distribution to meet the detailed balance condition,

$$\boldsymbol{\pi}(i)Q(i,j)\alpha(i,j) = \boldsymbol{\pi}(j)Q(j,i)\alpha(j,i) \tag{2.28}$$

One easy option for the additional distribution could be

$$\alpha(i,j) = \boldsymbol{\pi}(j)Q(j,i); \alpha(j,i) = \boldsymbol{\pi}(i)Q(i,j) \tag{2.29}$$

Thus, the detailed balanced condition is satisfied. Namely, if taking Q' as a new transition matrix with $Q'(i,j) = Q(i,j)\alpha(i,j)$, then $p(x)$ will be the stationary distribution of the new transition matrix Q'.

The adding of the distribution $\alpha(i,j)$ can be achieved by a separate step of accepting or rejecting the original transition. In other words, in the original Markov chain, the state i can jump to state j with probability of $Q(i,j)$; now we add a probability of $\alpha(i,j)$ to accept this jump. Consequently, the modified Markov chain will have the transition matrix Q' with $Q'(i,j) = Q(i,j)\alpha(i,j)$. This is the original Metropolis algorithm.

However, the acceptance ratio $\alpha(i,j) = \boldsymbol{\pi}(j)Q(j,i)$ can be very small, especially in continuous space with high dimensions. Thus, the Markov chain will stay in the same state for a very long time, and explore the whole space very slowly. Alternatively, we could amplify the acceptance ratio of $\alpha(i,j)$ and $\alpha(j,i)$ with the same scale factor $K > 0$, as long as the amplified acceptance ratios are not greater than 1, as follows,

$$\alpha(i,j) = K\boldsymbol{\pi}(j)Q(j,i) \leq 1; \alpha(j,i) = K\boldsymbol{\pi}(i)Q(i,j) \leq 1 \tag{2.30}$$

Therefore, we have the $K = \min\{[\boldsymbol{\pi}(j)Q(j,i)]^{-1}, [\boldsymbol{\pi}(i)Q(i,j)]^{-1}\}$, and the acceptance ratio will be amplified to

$$\alpha(i,j) = \texttt{min}\left\{\frac{\boldsymbol{\pi}(j)Q(j,i)}{\boldsymbol{\pi}(i)Q(i,j)}, 1\right\} \tag{2.31}$$

Clearly, the detailed balance condition is still satisfied, thus $\boldsymbol{\pi}(x)$ is still the stationary distribution of the modified Markov chain. This above strategy is the famous Metropolis-Hastings algorithm (Hastings, 1970), which is summarized in the following Algorithm 2.

Note again, the distribution can be continuous, and we only need to modify the transition matrix as $Q'(x_n, x_{n+1}) = Q(x_n, x_{n+1})\alpha(x_n, x_{n+1})$, where x does not need to be univariate, but can be any dimension.

Algorithm 2: Metropolis-Hastings algorithm

Initialize: x_1
for $n = 1$ *to* N **do**
  Sample: $\mu \sim U(0, 1)$
  Sample: $x^* \sim Q(x^*|x_n)$
  if $\mu < \alpha(x_n, x^*) = \texttt{min}\left\{\frac{p(x^*)Q(x_n|x^*)}{p(x_n)Q(x^*|x_n)}, 1\right\}$ **then**
    $x_{n+1} \leftarrow x^*$
  else
    $x_{n+1} \leftarrow x_n$

Gibbs Sampling In the MH sampler, the acceptance ratio has been amplified comparing to the original Metropolis sampler, however it still can be low, especially in complex models. The Gibbs sampler, a special case of the MH sampler, can boost the acceptance to 100%, while the cost is that the transition is always along a single dimension (or in some cases, multiple dimensions for a group of variables) in turn, and requires the conditional distribution, $p(x_i|\boldsymbol{x}_{i-})$ for variable x_i, given the values of all other variables $\boldsymbol{x}_{i-} = \{x_1, ..., x_{i-1}, x_{i+1}, ..., x_n\}$, to be available in a closed form.

First, we will see that the detailed balance still stands in the Gibbs sampling, with jumping along a single dimension following its conditional distribution. This can be proved easily as follows,

$$\begin{aligned} p(x_i = t_1, \boldsymbol{x}_{i-})p(x_i = t_2|\boldsymbol{x}_{i-}) &= p(x_i = t_1|\boldsymbol{x}_{i-})p(\boldsymbol{x}_{i-})p(x_i = t_2|\boldsymbol{x}_{i-}) \\ &= p(x_i = t_2|\boldsymbol{x}_{i-})p(\boldsymbol{x}_{i-})p(x_i = t_1|\boldsymbol{x}_{i-}) \\ &= p(x_i = t_2, \boldsymbol{x}_{i-})p(x_i = t_1|\boldsymbol{x}_{i-}) \end{aligned} \tag{2.32}$$

which means the stationary distribution can still be achieved.

The procedures of the Gibbs sampler can be precisely described in the following Algorithm 3, simply with repeating sampling along the conditional distribution in turn.

The Gibbs sampling reduces the task of sampling from a joint distribution in a high dimensional space, to sampling from a sequence of univariate conditional distributions. However, the requirement of the closed-form expression on the conditional distributions is not easy to meet, especially in Bayesian methods with special structured prior distributions. Therefore, many Bayesian models choose conjugate priors (see more in Chapter 2.2.3), which have the same form as the posterior distribution, and thus achieve a closed-form conditional distribution in the Gibbs sampling. For

example in the mixture model for splicing isoform quantification, `Dirichlet` distribution is a conjugate prior to multinomial likelihood distribution, and thus the Gibbs sampler is applicable for inference (see Chapter 3.2.3).

Algorithm 3: Gibbs sampling

1 **Initialize:** $\{x_i : i = 1, ...n\}$
2 **for** $t = 1$ *to* T **do**
3 **for** $i = 1$ *to* N **do**
4 **Sample:** $x_i^{(t+1)} \sim p(x_i|x_1^{(t+1)}, ..., x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, ..., x_n^{(t)})$

Convergence diagnosis Based on the MCMC sampling (either the MH sampler or the Gibbs sampler), we can generate a sequence of samples $\{x_0, x_1, ...\}$. After the Markov chain converges to the stationary distribution, the samples will follow the target distribution $p(x)$, which can be used to represent the distribution. The period before the convergence is called the burn-in period. However, the convergence diagnosis in MCMC chain is still an open question. In other words, it is still hard to decide when to safely stop sampling. Here, I only introduce two widely used strategies in deciding the length of the sampling; a more comprehensive comparison can be found in the review (Cowles and Carlin, 1996).

There are two types of strategies: by a single MCMC chain or by multiple MCMC chains. One widely used single chain strategy is Geweke's method (Geweke, 1991). Geweke's convergence diagnostic (Z score) is calculated by taking the difference of the mean of the samples in the initial region $A = \{x_1, ..., x_{n_A}\}$ and that of the last region $B = \{x_{n-n_B+1}, ..., x_n\}$, and dividing by the asymptotic standard error of the two regions, as follows,

$$Z = \frac{\bar{A} - \bar{B}}{\sqrt{\texttt{var}(A) + \texttt{var}(B)}} \tag{2.33}$$

where $\bar{A}$ and $\bar{B}$ are the mean of their corresponding sequences. The ratio of the initial region n_A/n and the ratio of the last region n_B/n are usually fixed. Geweke suggested the lengths of the two regions as $n_A = 0.1n$ and $n_B = 0.5n$, respectively. For a univariate diagnosis, $|Z| \leq 2$ is often taken as the threshold for passing the convergence, and the last say 75% samples can be used to represent the distribution.

For multiple-chain convergence diagnosis, Gelman and Rubin's method (Gelman and Rubin, 1992) is the most popular one. This method requires two steps, including first selecting $m \geq 2$ start points that are over-dispersed relative to the target density,

and second running the m MCMC chains with desired number of iterations, say $2n$. Then the last n iterations are often used to represent the target distribution. Here, convergence is monitored by comparing the between-chain variance and the within-chain variance, with following $\hat{R}$ statistic,

$$\sqrt{\hat{R}} = \sqrt{\Big(\frac{n-1}{n} + \frac{m+1}{mn}\frac{B}{W}\Big)\frac{df}{df-2}} \tag{2.34}$$

where B and W are the between-chain and within-chain variances, respectively, and df is the degree of the freedom of the approximating $p(x)$ density. The idea behind this method is that in the beginning B will be much larger than W as these chain starting points are over-dispersed, while after convergence, the above "shrink factors" for all quantities of interest will become near 1.

Though these methods are not perfect and may still fail in detecting convergence, in practice they both work reasonably well. As the single-chain strategy costs less computation (only requiring one burn-in period rather than multiple), we use Geweke's method in this thesis for convergence diagnosis.

Chapter 3

Mixture modelling for isoform quantification

In chapter 2.1, we described how RNA-seq technology can be used to quantify gene expression levels, and also provided an indirect way to quantify alternative splicing isoforms. In this chapter, I will introduce a commonly used probabilistic modelling framework with a mixture model to estimate splicing isoform fractions by using RNA-seq data, and show a case study on RNA splicing efficiency in yeast, a collaborative work with Prof. Jean Beggs's lab.

There are usually two different focuses on splicing analysis: transcript level and splicing level. At transcript level, we want to quantify the expression of all transcript isoforms, which usually involves many shared exons, for example the 8 transcript isoforms from human gene *SLC25A3* (see Fig 3.1) share 14 distinct exons. At a splicing level, we only focus on an alternative exon and its neighbour flanking exons. In the same example gene *SLC25A3*, the mutual exclusive exons splicing event usually only involves 4 exons (exons in green and orange and the flanking exons in Fig 3.1). Note, the upstream exon can be different in different transcripts, but we could use the shared part of this exon as the upstream exon. Thus, each isoform of a splicing event can be part of one or multiple transcripts, and one transcript may be involved in multiple splicing events. Quantification of transcript isoforms is much more complex than inclusion or exclusion of a particular exon, due to the existence of large number of transcripts and high overlaps between each other. However, one benefit is that the quantification at transcript level theoretically contains all information for quantification at the splicing events level. Given the quantification at the transcript level (i.e., relative expression of all transcripts), the fraction of splicing isoforms can be calculated by directly summa-

rizing transcripts containing each splicing isoform.

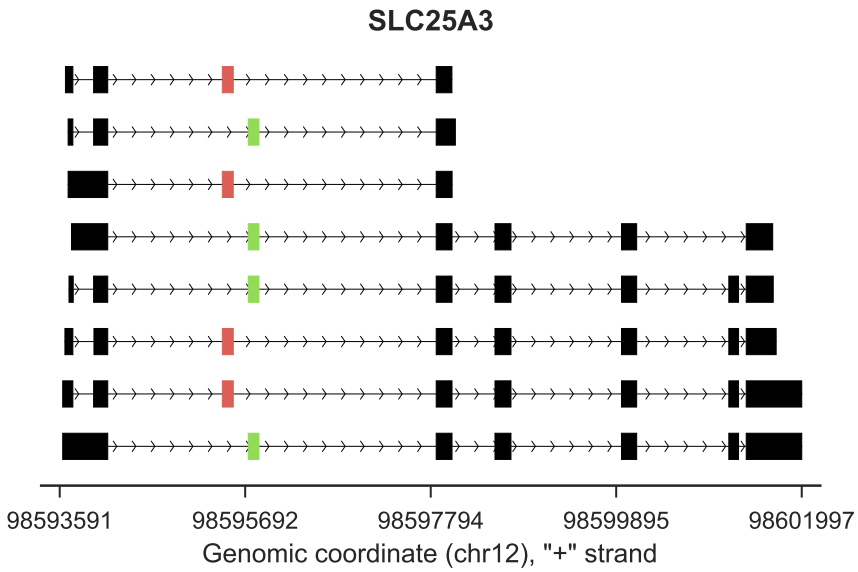

Figure 3.1: Transcript isoforms and splicing event isoforms in human gene *SLC25A3*. At transcript level, there are 8 protein coding transcript isoforms, while at splicing level, there are only two isoforms in the mutual exclusive exons event (the selected 3rd exon either in green or orange).

Many biological studies focus on splicing level for particular splicing events, not only because of its computational simplicity, but also due to its easier interpretation of the analysis and the biological mechanisms. There, junction reads (see definition in Chapter 2.1.4) are often used to directly count the inclusion or exclusion of a specific alternative exon. In Fig 3.2, in the example exon-skipping event, the ratio between inclusion and exclusion of the alternative exon can be directly counted by the junction reads (i.e., reads overhanging two exons) exon1-exon2 versus exon1-exon3, or the coverages of exon2 versus exon1, or other similar direct counting strategies.

However, the direct counting strategy, especially based on the junction reads, wastes many other reads that mapped to the shared exons, but these ambiguous reads are not uninformative. Also, this counting strategy cannot give a confidence of the estimate. More importantly, the direct counting method usually does not work for transcript-level quantification, as many genes have multiple transcripts and multiple splicing events, for example *SLC25A3*. In the next section, we will introduce a commonly used probabilistic framework to quantify isoforms at the transcript level, which is also applicable to the splicing level.

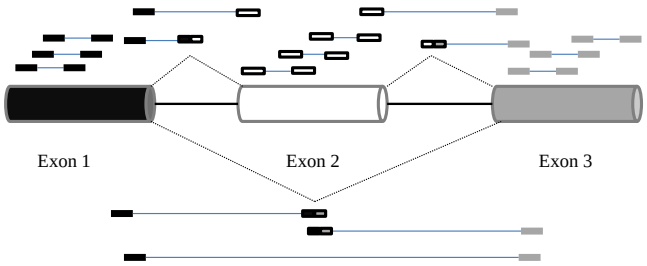

Figure 3.2: Paired-end RNA-seq reads aligned to genome reference for an example exon-skipping event. Alternative splicing events usually involves two isoforms and 3 or 4 exons, and can possibly be measured by direct counting.

3.1 Statistical framework

For isoform quantification at transcript level, we aim to estimate the fractions of each isoform from a given gene. More specifically, assuming a gene g has K transcript isoforms, we want to estimate the fraction array $\Psi = \{\psi_1, ..., \psi_K\}$ of each isoforms with a constraint of $\sum_{k=1}^{K} \psi_k = 1$. Usually, we assume the read counts at gene level are unambiguous (otherwise, we could treat the overlapped genes as a single super gene), thus, the expected read counts for each transcript isoform can be easily obtained by multiplying the gene level counts by the isoform fractions Ψ and its according weights e.g., (effective) transcript length.

On each transcript isoform, we simply assume that the RNA-seq reads are uniformly generated. This assumption is slightly different from the reality where biases exist in the reads distribution, for example, GC bias, positional bias and sequence bias (Roberts et al., 2011). However, commonly used strategies for correcting these biases are based on a separate step from the core statistical model, for example a pre-step by defining a specific reads distribution along the gene body with empirically learned bias weights, e.g., in Cufflinks (Trapnell et al., 2010) and Kallisto (Bray et al., 2016), or on a post-step by regressing out these biases, e.g., in Sailfish (Patro et al., 2014) and GAM correction (Zheng et al., 2011). Without loss of generality, we use a gapped uniform distribution (i..e., uniform distribution with gaps of intron) for mapped reads positions here, and leave the discussion of sequencing bias to a separate section.

As the length of RNA-seq reads (30-200bp) is usually much shorter than most transcripts, and constitutive exons are shared by several transcript isoforms, those reads aligned to the shared exons are a mixture of reads from several isoforms. Therefore, isoform quantification could be treated as a mixture model, i.e., the overall reads distri-

bution is a mixture of multiple differently gapped uniform or biased distributions from all isoforms.

In an example gene in Figure 3.3 (top panel), there are 3 isoforms, and each of the four exons is shared by at least two isoforms. Due to the different combinations of exons, the gaps and the lengths of the isoforms are different. Consequently, the reads distribution of each isoform along the gene structure is different between each other (see the reads probability density in the middle panel of Figure 3.3). If the fractions of the three isoforms are $\Psi = [0.3, 0.1, 0.6]$, the theoretical reads density with uniform noise is shown in Figure 3.3 bottom panel. Then, the aim of the isoform quantification from RNA-seq data is to estimate the isoform fractions Ψ from the reads with density such as the Figure 3.3 bottom panel. Though the real reads density is usually much more uneven, the idea of using a mixture model for this problem remains the same.

3.1.1 Mixture model framework

Here, we formally introduce the probabilistic mixture model for estimating isoform fractions, which has been used in many methods to quantify splicing isoforms, including single gene based methods MISO (Katz et al., 2010) and Cufflinks (Trapnell et al., 2010), and full transcriptome based methods BitSeq (Glaus et al., 2012) and Kallisto (Bray et al., 2016). As mentioned in Chapter 2.1.4, the gene based methods try to map reads to a genome reference and allocate the precisely aligned reads to a specific transcript or splicing isoform, while the transcriptome based methods try to map reads to the transcriptome reference and assign the multi-mapped reads to a specific transcript. Theoretically, these two methods are equivalent, while the former has the advantages of the emphasis on the relative fractions and the potential to discover new splicing variants. Therefore, in this thesis I focus on the single gene based framework. In other words, I only look at the data for each gene individually, for example N reads $R_{1:N}$ mapped to a gene with K isoforms. The likelihood to observe a set of reads $R_{1:N}$ given a certain isoform fraction vector Ψ can be written as follows,

$$\mathcal{L}(\Psi) = P(R_{1:N}|\Psi) = \prod_{n=1}^{N} \sum_{I_n=k}^{K} P(R_n|I_n)P(I_n|\Psi) \tag{3.1}$$

As a mixture model, each read R_n (a data point) has its *identity* $I_n \in \{1, \ldots, K\}$, i.e. the specific isoform it originated from, but, unless the read is aligned to isoform specific region, e.g., a junction, we will not know its identity. The conditional distribution of $I_n|\Psi$ is assumed to be Multinomial, $(I_n|\Psi) \sim \texttt{Multinomial}(\Psi \times \boldsymbol{w})$ where $\boldsymbol{w}$ is a

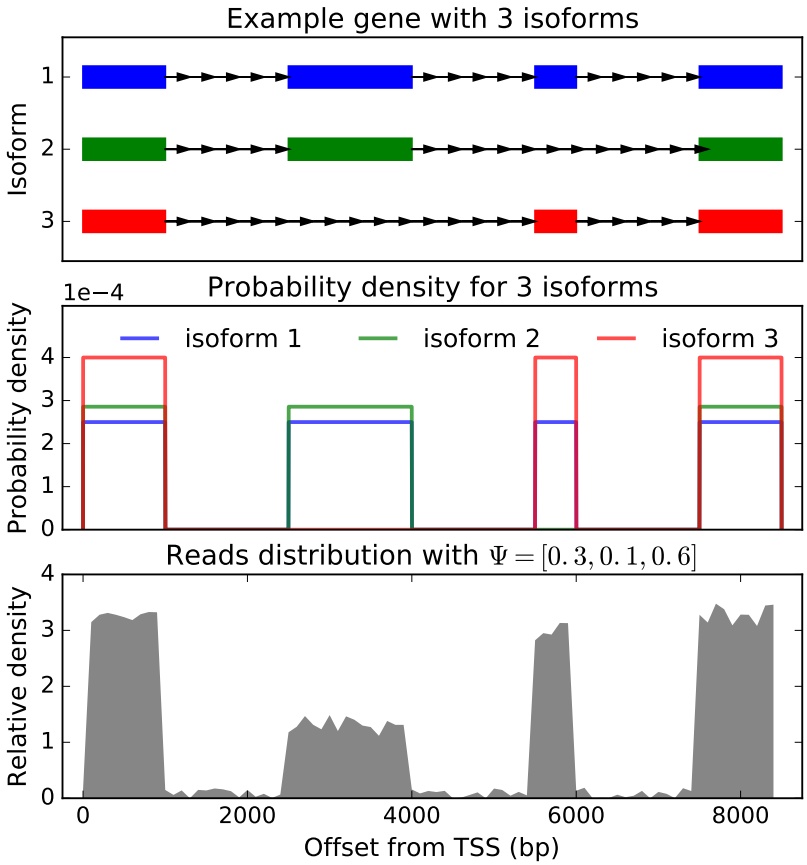

Figure 3.3: An example gene with 3 isoforms and four shared exons (Top panel). For a single read, the probability density of the read comes from a certain position in an isoform (Middle panel). Given isoform fractions as $\Psi = [0.3, 0.1, 0.6]$, the total reads density of the mixture of the three isoforms (Bottom panel). Here, the reads are assumed to be extremely short and uniformly distributed.

weight vector adjusting the isoform proportion by the effective length of each isoform, as follows,

$$\phi_k = \psi_k \times w_k = \frac{\psi_k \times l_k^{(eff)}}{\sum_{k=1}^{K} \psi_k \times l_k^{(eff)}}, k \in [1, ..., K] \tag{3.2}$$

where $l_k^{(eff)}$ is the effective length of isoform k, and it is often calculated from the isoform length l_k and the average fragment or read length $l^{(r)}$ as, $l_k^{(eff)} = l_k - l^{(r)} + 1$ (or additionally with bias correction). Then this weighted isoform fraction is equivalent to the reads frequency of each isoform, and is denoted as Φ, with $\sum_{k=1}^{K} \phi_k = 1$.

The term $P(R_n|I_n)$ presents the probability of observing a certain read from a specific isoform I_n. Given the aligned position of this read, the term $P(R_n|I_n)$ is usually

precisely defined with fixed parameters $\boldsymbol{\vartheta}$ (more details in next section). Importantly, this term automatically adjusts for the different informativeness of different reads: for example, junction reads will generally have a reduced number of possible isoforms (in extreme cases, only one), and as such will carry considerably more information through a reduced-entropy term $P(R_n|I_n)$. In this way, while the approach uses all sequenced reads for inference, the architectural information of the transcript is still retained and automatically used.

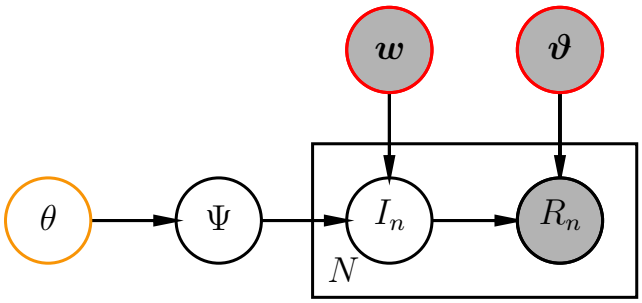

Figure 3.4: A graphical representation of the mixture model for isoform quantification. The isoform fraction, Ψ, is the parameter to estimate, from observing a set of reads $R_{1:N}$. As each read may come from any of the K isoforms, a latent variable I_n is introduced to denote the identity, i.e., the isoform the read comes from. θ is a hyperparameter for Ψ, and is only used in a Bayesian method. $\boldsymbol{w}$ and $\boldsymbol{\vartheta}$ are fixed coefficients for I_n and R_n, respectively. $P(I_n|\Psi, \boldsymbol{w})$ is a multinomial distribution, and $P(R_n|I_n, \boldsymbol{\vartheta})$ is a gapped uniform distribution, or with bias modelling, encoded in $\boldsymbol{\vartheta}$. Nodes in shade are observed variables, otherwise are unobserved. Nodes with yellow circle are hyper-parameters, with red circle are fixed coefficients.

Besides the perspective as a frequentist focusing on the likelihood, we could view it in a Bayesian manner, namely we have an initial belief of the Ψ, termed as prior distribution, and then we use the observed data (RNA-seq reads) to adjust our initial belief with Bayes theorem. Then the posterior of Ψ can be described as follows,

$$\begin{aligned} P(\Psi|R_{1:N}) &\propto P(\Psi|\theta) \times P(R_{1:N}|\Psi) \\ &\propto P(\Psi|\theta) \times \prod_{n=1}^{N} \sum_{I_n=k}^{K} P(R_n|I_n) P(I_n|\Psi) \end{aligned} \tag{3.3}$$

where a common choice of the prior distribution is a $\texttt{Dirichlet}(\Psi|\theta)$, which is a conjugate distribution of a multinomial distribution.

3.1.2 Probability of isoform specific reads

The term $P(R_n|I_n = k)$, as the probability of isoform specific reads, is one of the most important elements of the probabilistic model. In a simplistic case, it is assumed that reads are uniformly distributed along the gene with gaps of intron, and thus can be simply expressed as constants as follows,

$$P(R_n|I_n = k) = \begin{cases} 0 & \text{if } R_n \text{ does not match isoform } k \\ 1/l_k^{(eff)} & \text{if } R_n \text{ matches isoform } k \end{cases} \tag{3.4}$$

Compared to this simplified version, this term of the probability of isoform specific reads could take more information into account, for example the fragment length, the sequencing and alignment quality, and the sequence and positional biases. Here, I describe the full computation of this term based on paired-end reads. For single-end reads, the fragment length will be replaced by the read length and only one read will appear in the formula.

For a given isoform $I_n = k$, the probability of matching a pair of reads $r_n^{(1,2)}$ to this isoform is determined by the probability of the reads being sequenced at a specific position p with a specific fragment length l_f and the probability of the correct alignment encoded in the `mapq` tag, as follows,

$$P(R_n|I_n) = P(l_f|I_n)P(p|I_n, l_f)P(R_n|\texttt{mapq}) \tag{3.5}$$

The fragment length distribution is assumed to follow a Gaussian distribution, $P(l_f|I = k) = \mathcal{N}(l_f|\mu, \sigma)$, with its parameters (mean μ and variance σ^2) estimated from the read pairs that map to the given isoform, or estimated from single-transcript genes, or customized by users from experiment design. In some species, most reads can be very well mapped to a single position in the genome, and we could simply use uniquely mapped reads. However, in some other species, such as yeast that contains many paralogs, there are higher chances to align a read to multiple positions. In the latter case, it is better to keep this big fraction of multiply aligned reads, therefore the alignment quality score, measured by the `mapq` tag from the alignment report (Li et al., 2009), should be taken into account, as $P(R_n|\texttt{mapq}) = 1 - 10^{-\texttt{MAPQ}/10}$, and we take the better score for the reads pair, i.e., $\texttt{MAPQ} = \max(\texttt{MAPQ}^{(1)}, \texttt{MAPQ}^{(2)})$.

For the remaining term $P(p|I_n, l_f)$, we could simply take the gapped uniform distribution in Eq (3.4), or encode the sequencing bias into it. A popular strategy of how to encode the bias correction into this term will be discussed in the next section.

3.1.3 Methods for sequencing bias correction

There are quite a few sequencing bias correction methods (Roberts et al., 2011; Zheng et al., 2011; Love et al., 2016). Here, we only introduce one of the most popular methods, which was proposed by Roberts et al. (Roberts et al., 2011) for correcting sequencing bias, including fragmentation bias, positional bias (i.e., start position along transcript) and sequence bias (e.g., GC bias). All details can be found in the original paper, and the usage and the performance of the bias correction in my work can be found in Chapter 4 or our published paper (Huang and Sanguinetti, 2016).

Under this model, we have the joint probability to see a read mapping to a position p, as follows,

$$b_k(p) = b_k^{s,5}(e_5)b_k^{s,3}(e_3)b_k^{p,5}(e_5)b_k^{p,5}(e_3) \tag{3.6}$$

where $b_k^{s,5}(e_5)$ and $b_k^{s,3}(e_3)$ denote the sequence specific biases for the 5' and 3' ends of the fragment, respectively, and $b_k^{p,5}(e_5)$ and $b_k^{p,3}$ are the corresponding positional specific biases.

A variable length Markov model was applied to correct the sequence biases (see supplementary Figure 2 of (Roberts et al., 2011)). This model is based on 21 bases from 8 bases before and 12 bases after the read starting position, among which 6 bases have no parents node; 5 bases have one parent node and 10 bases have two parent nodes. Taken the 5' end of the fragment as a case, the sequence bias is described as follows:

$$b_k^{s,5}(e_5) = \prod_{n=1}^{21} \frac{\phi_{n,\pi_n}^{5,B}}{\phi_{n,\pi_n}^{5,U}} \tag{3.7}$$

where ϕ_{n,π_n}^5 is the probability of the base on the nth position and its parents π_n; $\phi_{n,\pi_n}^{5,B}$ and $\phi_{n,\pi_n}^{5,U}$ refer to the bias model and uniform model, respectively. There are $4 \times 6 + 4^2 \times 5 + 4^3 \times 10 = 744$ parameters, which are estimated empirically. The parameters for the 5' end and 3' end of fragment are estimated separately.

Besides the sequence specific bias, an additional model was used to correct the positional bias. For the case of the 5' end of the fragment, the bias is

$$b_k^{p,5}(e_5) = \frac{\omega_{l_k,e_5}^B}{\omega_{l_k,e_5}^U} \tag{3.8}$$

where $\omega_{l,p}$ is the probability for the starting position on p within mRNA length l. The probabilities are modelled by taking 20 bins of relative position, and mRNAs are divided into five groups by their length. Therefore, there are $20 \times 5 = 100$ parameters, which are again estimated empirically.

The read's position could be assumed to come from a uniform distribution, or could be explicitly modelled on sequence and position biases. In both cases, we could describe the probability as follows,

$$P(p|I_n = c, l_f) = \frac{b_c(p)}{\sum_{j=1}^{l_k - l_f + 1} b_c(j)} \tag{3.9}$$

where $b_c(p)$ is relative weight of a position p. For uniform distribution, $b_c(p) \equiv 1$, so that $P(p|I_n, l_f) = 1/(l_k - l_f + 1)$. For the biased distribution, this modelled probability could be used to correct the position and sequence biases.

3.2 Inference methods

In the above section, we precisely defined the likelihood and the posterior distribution of the isoforms' fraction vector Ψ to observe a set of aligned reads $R_{1:N}$. Here, we will show three commonly used algorithms to infer the parameter Ψ: the EM algorithm that estimates Ψ by reaching the maximum likelihood; two MCMC sampling algorithms, Metropolis-Hastings (MH) sampler and Gibbs sampler that sample the whole posterior distribution.

3.2.1 EM algorithm for isoform inference

The Expectation maximization (EM) algorithm is widely used to solve mixture models, such as Gaussian mixture model as discussed in Chapter 2.2.2. As a problem of mixture of components (isoforms), the EM algorithm is also capable of approximating the isoform fractions Ψ with a maximum likelihood estimation. In the standard EM algorithm (see more details in Chapter 2.2.2), there are two main steps in each iteration: E-step to calculate the expectation of the likelihood function given the parameters in a previous step, and M-step to maximize the given likelihood function with regard to the parameters. Similar to the Gaussian mixture model, we also introduce the isoform responsibility $\gamma(I_{n,k}) = P(I_n = k|R_n, \Psi)$, namely given the parameters and the read information, the expectation of the read coming from isoform k. The EM algorithm on this problem was initially implemented by Nicolae *et al.* (Nicolae et al., 2011), and I introduce it here as Algorithm 4 with minor modifications to be more easily understood.

More specifically, this EM algorithm starts with random initialization of Ψ, followed by the iteration of alternating E-step and M-step. In the E-step, the isoform

responsibility $\gamma(I_{n,k})$ will be calculated with the Ψ fitted in the M-step, and then in the M-step, the Ψ will be optimized with the updated $\gamma(I_{n,k})$. Finally, convergence can be detected by checking whether the change of Ψ or the change of the likelihood is smaller than a threshold.

Algorithm 4: EM algorithm for isoform inference

Initialize Initialize Ψ randomly
while *not converged* **do**
  E step: Calculate responsibility of isoform k on read R_n
  $\gamma(I_{n,k}) = \frac{P(R_n|I_n=k)P(I_n=k|\Psi)}{\sum_{k=1}^{K} P(R_n|I_n=k)P(I_n=k|\Psi)}$ for all n and k
  M step: update Ψ to maximize the likelihood
  $\psi_k^{\texttt{new}} \times w_k = \frac{\sum_{n=1}^{N}\gamma(I_{n,k})}{\sum_{k=1}^{K}\sum_{n=1}^{N}\gamma(I_{n,k})} \Rightarrow \Psi^{\texttt{new}}$
  Update likelihood and check convergence
return Ψ

Here, $P(R_n|I_n)$ is fixed with predefined coefficient $\boldsymbol{\vartheta}$, thus is independent from Ψ, and $P(I_n|\Psi, \boldsymbol{w}) \sim$ Multinomial($\Psi \times \boldsymbol{w}$) with fixed coefficient $\boldsymbol{w}$ denoting the effective lengths, as shown in Eq (3.2).

Thanks to the fast speed in computation, the EM algorithm is also widely used in isoform quantification on a transcriptome basis, e.g, in RSEM (Li and Dewey, 2011), eXpress (Roberts and Pachter, 2013), and a faster version is also used in Sailfish (Patro et al., 2014) and Kallisto (Bray et al., 2016).

3.2.2 MH sampler for isoform inference

In addition, Markov chain Monte Carlo (MCMC) inference is able to obtain the full distribution of Ψ, especially for the posterior distribution. The Metropolis-Hastings (MH) sampler is a flexible algorithm, and can handle a variety of prior distributions (see details in Chapter 1.2.4). Here, I use the MH algorithm to sample the posterior of $p(\Psi|R_{1:N})$ with `Dirichlet` prior distribution and fixed hyper-parameters, as described in Eq (3.3). For efficient sampling, the most important part in the MH algorithm is to design a good proposal distribution, to ensure an overall 30-50% acceptance ratio. Here, we use a `Logistic-Normal` distribution $f_{LN}(\Psi|\boldsymbol{\mu}, \Sigma)$ (Atchison and Shen, 1980) as the proposal distribution, as follow,

$$f_{LN}(\Psi|\boldsymbol{\mu}, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}}(\prod_{k=1}^{K}\psi_k)^{-1}\texttt{exp}\{-\frac{1}{2}(\boldsymbol{\varphi}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{\varphi}-\boldsymbol{\mu})\} \tag{3.10}$$

where the $\boldsymbol{\varphi}$ is a reverse `softmax` transformation of Ψ, and has a fixed value $\varphi_K = 0$, as follows,

$$\varphi_i = \log \frac{\psi_i}{1 - \sum_{k=1}^{K-1} \psi_k} \Leftrightarrow \psi_i = \frac{e^{\varphi_i}}{1 + \sum_{k=1}^{K-1} e^{\varphi_k}}, i \in \{1, ..., K-1\} \tag{3.11}$$

Here, $\psi_K = 1 - \sum_{k=1}^{K-1} \psi_k$, and we take the first $K-1$ values, so that $\boldsymbol{\varphi}$ has the same dimension as $\boldsymbol{\mu}$ and Σ. Therefore, the transition function of the proposal distribution is described as follows,

$$Q(\Psi_{t+1}|\Psi_t) = f_{LN}(\Psi_{t+1}|\texttt{softmax}(\Psi_t), \Sigma) = f_{LN}(\Psi_{t+1}|\boldsymbol{\varphi}_t, \Sigma) \tag{3.12}$$

Given a fixed covariance matrix Σ for the proposal distribution, the sampling scheme can be described as the following Algorithm 5.

Note, in this algorithm, we did not sample the latent variable I_n, because given a sampled Ψ, the distribution $p(I_n|\Psi, R_{1:N})$ can be calculated analytically. Therefore, we can integrate out and collapse I_n. This is very useful, as the dimension of I_n is equal to the number of reads, which can be very large. By collapsing it, the dimension for sampling is equal to the dimension of Ψ, which is much smaller, usually smaller than 20.

Algorithm 5: MH sampler for isoform inference

Initialize: Ψ_1 randomly
for $m = 1$ *to* M **do**
  Sample: $\mu \sim U(0, 1)$
  Sample: $\Psi^* \sim Q(\Psi^*|\Psi_m)$
  if $\mu < \alpha(\Psi_m, \Psi^*) = \texttt{min}\left\{\frac{p(\Psi^*|R_{1:N})Q(\Psi_m|\Psi^*)}{p(\Psi_m|R_{1:N})Q(\Psi^*|\Psi_m)}, 1\right\}$ **then**
    $\Psi_{m+1} \leftarrow \Psi^*$
  else
    $\Psi_{m+1} \leftarrow \Psi_m$
return $\Psi_{1:M}$

Within this algorithm framework, there still are two remaining questions to solve. First, how to choose a sensible covariance matrix Σ for the step size in the transformation proposal? Simply, we could try a set of candidate Σ and choose the one closest to the 30-50% acceptance ratio, or we could use an adaptive way to set this covariance matrix. Namely, we set a reasonable large transformation step, and run a short

chain, say 100 steps and calculate the covariance Σ' of these 100 steps, and update the covariance for the proposal according to the empirical covariance with a scaling rule of $\Sigma = 2.38/\sqrt{K-1} \times \Sigma'$ (Roberts and Rosenthal, 2009). Usually, this adaptive strategy will also give a 30-50% acceptance ratio. The second question is convergence diagnosis, i.e., how to set a proper length M of the MCMC chain, so that the MCMC chain converges to the stationary distribution. As discussed in Chapter 2.2.4, the convergence diagnosis is still an open question in MCMC sampling. Here, I use a single-chain strategy, Geweke's method (Geweke, 1991) to diagnose the convergence. Specifically, I run a single MCMC chain, and use the first 10% and the last 50% samples (A and B, respectively) from the chain to calculate the Z score as follows,

$$Z = \frac{\bar{A} - \bar{B}}{\sqrt{\mathtt{var}(A) + \mathtt{var}(B)}} \tag{3.13}$$

If $|Z| \leq 2$ for all $\psi_k, k \in \{1, ..., K-1\}$, then the chain is treated as converged, otherwise 100 more iterations are added until the criterion is passed.

3.2.3 Gibbs sampler for isoform inference

The Gibbs sampler is a special type of MCMC sampler, or a special case of Metropolis-Hastings algorithm, where the proposal is always along a single dimension, ensuring 100% acceptance ratio. This complete acceptance is the greatest advantage comparing to the MH sampler. However, the Gibbs sampler requires the conditional distribution, $p(\theta_i|D, \theta_{i-})$ for parameter θ_i given all other parameters θ_{i-}, to be written in a closed form. This requirement is not easy to meet, especially in Bayesian methods with a special structured prior distribution. Conjugate priors, which have the same form as the posterior distribution, are often used to achieve a closed form of the conditional distribution in the Gibbs sampling. In the mixture model for isoform quantification, the `Dirichlet` distribution is a conjugate distribution to multinomial distribution, and can be used in a Gibbs sampler. Here, we use an uninformative prior as $\mathtt{Dir}(\boldsymbol{\tau})$, where $\tau_1 = \tau_2 = ... = \tau_K$.

In the Gibbs sampler, the read identities $I_{1:N}$ cannot be collapsed, otherwise the closed form for Ψ is not available. Alternatively, we can marginalize Ψ, and still apply a collapsed Gibbs sampler to generate samples from the posterior probability distribution over $I_{1:N}$. As a multinomial distribution, $p(I_n = k|R_n, \boldsymbol{\tau}, I_{n-})$ can be written as follows,

$$p(I_n = k|R_n, \boldsymbol{\tau}, I_{n-}) = \frac{P(R_n|I_n = k)(C_{k,n-} + \tau_k)}{\sum_{j=1}^{K} P(R_n|I_n = j)(C_{j,n-} + \tau_j)} \tag{3.14}$$

where $C_{k,n-}$ denotes the number of reads except read R_n belonging to isoform k, i.e., $C_{k,n-} = \sum_{i \neq n} I_{i,k}$. In addition, $P(R_n|I_n = k)$ is fixed in a pre-step. Therefore, we can sample I_n following the above distribution. As a Gibbs sampling, we could sample I_n from $n = 1$ to $n = N$ in order, or we could adaptively produce more samples on those reads with I_n converging more slowly. In Algorithm 6, we describe the simpler version, and collect a Ψ after each full updating from 1 to N reads, as follows,

$$\psi_k = \frac{\sum_{i=1}^{N} I_{i,k}/w_k}{\sum_{j=1}^{K}\sum_{i=1}^{N} I_{i,j}/w_j} \tag{3.15}$$

This algorithm was initially used in BitSeq (Glaus et al., 2012) for isoform quantification at a transcriptome level. Here, we modified it to quantify transcript isoforms at the gene level in Algorithm 6.

Similar to the MH sampler, convergence diagnosis is also needed in the Gibbs sampling, and similar strategies, including Geweke's method (Geweke, 1991) can be applied here.

Algorithm 6: Gibbs sampler for isoform inference

1 **Initialize:** Ψ_1 and $I_{1:N}^{(1)}$ randomly
2 $C_k = \sum_{n=1}^{N} \mathbb{I}(I_n^{(1)} = k)$ for $k \in \{1, K\}$
3 **for** $m = 1$ *to* M **do**
4 **for** $n = 1$ *to* N **do**
5 $C_{k,n-} = C_k - \mathbb{I}(I_n^{(m-1)} = k)$ for $k \in \{1, K\}$
6 **Sample**: $I_n^{(m)}|I_1^{(m)}, ..., I_{n-1}^{(m)}, I_{n+1}^{(m-1)}, ..., I_N^{(m-1)}$ as Eq (3.14)
7 $C_k = C_{k,n-} + \mathbb{I}(I_n^{(m)} = k)$ for $k \in \{1, K\}$
8 **Calculate** Ψ as Eq (3.15) and save as Ψ_m
9 **return** $\Psi_{1:M}$

In this algorithm, the sampling of $I_n^{(m)}$ is based on $I_1^{(m)}, ..., I_{n-1}^{(m)}, I_{n+1}^{(m-1)}, ..., I_N^{(m-1)}$, from which we could calculate $C_{k,n-}, k \in \{1, K\}$ with a computational complexity of $O(1)$ (given K is small) as presented in the above algorithm. However, lines 5 and 7 in the for loop substantially increase the computing time in our implementation with Python. Alternatively, we could approximate it with $I_n^{(m)}|I_1^{(m-1)}, ..., I_{n-1}^{(m-1)}, I_{n+1}^{(m-1)}, ..., I_N^{(m-1)}$, which is equivalent to $I_n^{(m)}|\Psi_{m-1}$. This strategy allows the calculation of $C_{k,n-}$ in a matrix format, which is a much faster way than employing a for loop in Python. Since both strategies give a similarly accurate estimate, the faster version will be used in the next section for comparison.

3.3 Performance of probabilistic method

In the above sections, we explicitly defined a probabilistic method for isoform quantification, and introduced a few inference algorithms. Here, we will first look at how the inference algorithms work for an example gene, and investigate the benefits of the probabilistic method in splicing measurement compared with direct methods. In the end, I will briefly show the remaining challenges in isoform quantification.

3.3.1 Performance of inference algorithms

In order to closely check how these inference algorithms work, it is good to view each iteration in isoform qualification. Here, I conduct a simulation on the 3-isoform example gene (see Fig 3.3), with 500 theoretical reads (uniformly distributed, no bias, extremely short) as a toy example, and perform the EM algorithm, MH sampler and Gibbs sampler on the estimate task.

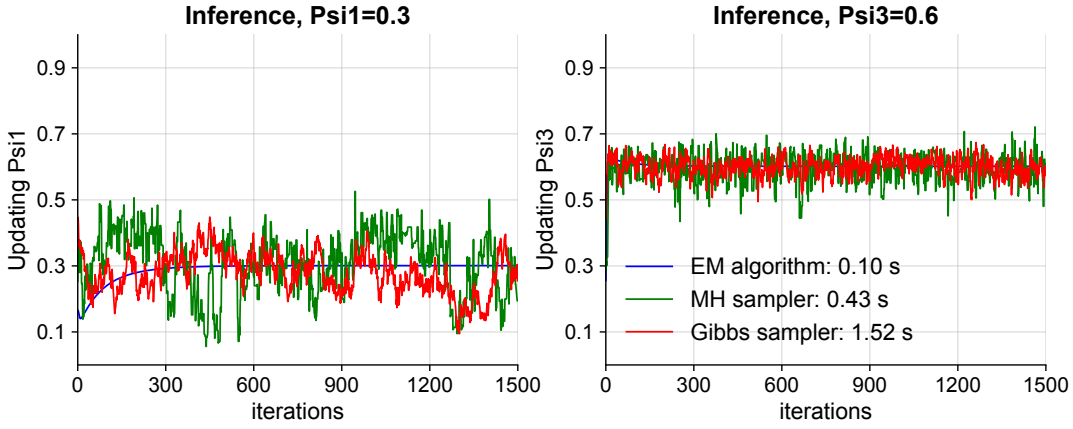

Figure 3.5: Iterations of three inference algorithms for the example case in Fig 3.3. There are 1,500 iterations for the three inference algorithms: EM algorithm, MH sampler and Gibbs sampler. The left panel shows the updating of ψ_1 with the truth of 0.3, and the right panel shows the updating of ψ_3 with the truth of 0.6. The total running times for the whole iterations is shown in the legend in the right panel. Note: in the Gibbs sampler, each iteration contains updating the identities of all reads.

In the simulation, 1,500 iterations have been recorded in each of the three algorithms, and the updating trajectories are presented in Fig 3.5 for ψ_1 in left panel and ψ_3 in right panel. As can be seen in the figure, the EM algorithm fast approaches the

true value of $\psi_1 = 0.3$ within 50 iterations, and also $\psi_3 = 0.6$ within 20 iterations, which means the EM algorithm is a very efficient and accurate inference method for isoform quantification in this case. However, the EM algorithm can only return a point estimate of Ψ, but cannot give a confidence of the estimate nor the whole distribution. Then, we tested the MCMC sampling methods to obtain the set of samples of the posterior distribution. Here, we can see that both the MH sampler and Gibbs sampler could effectively sample the distribution of Ψ, and even within the same gene, the variance of ψ_1 is much larger than ψ_3, which the EM algorithm cannot observe. In addition, both the MH sampler and Gibbs sampler converge reasonable quickly, with around 100 iterations for ψ_1 and around 20 iterations for ψ_3. It takes 0.43 seconds for the MH sampler to run 1,500 iterations, slightly slower than the EM algorithm with 0.10 second for the same number of iterations. Compared with MH sampler, Gibbs sampler has the same computational complexity of $O(N)$ in each iteration. However, in our implementation, Gibbs sampler is clearly slower than MH sampler (1.52 vs 0.43 seconds), probably because the sampling of N reads' identities $I_{1:N}$ in Gibbs is implemented via a for loop but the calculation of N reads in MH sampler is via a matrix format. In Python, the matrix computing is usually much faster than the equivalent for loop. With an optimal implementation, both Gibbs sampler and MH sampler may give similar computing speed.

In summary, all three inference methods can obtain accurate estimate in the above example case. The EM algorithm is the most efficient, particularly when considering the actual required length is much shorter than the 1,500 iterations. If the whole distribution is needed, both MH and Gibbs samplers are good choices, while the MH sampler might be a more efficient option in practice. Therefore, in the following analysis, we only use the MH sampler for the probabilistic method.

3.3.2 Benefit of probabilistic method and challenges

In the above section, we saw the accuracy of the probabilistic method in an example case. Now, I will compare the probabilistic method with two direct methods in estimating mRNA fraction from the mixture of pre-mRNAs and mature mRNAs (see the illustration of RNA splicing in Fig 2.3). Before moving into the simulation experiment, let us define the two direct methods to quantify the mRNA fraction. Note, the quantification of pre-mRNA and mature mRNA is equivalent to the quantification of two isoforms in a gene.

The simplest direct method for calculating splicing ratios is to use only reads that are unambiguously assigned to either mature or pre-mRNA. Such reads cover the exon1-exon2 junction for mature mRNA and the intron-exon2 (or exon1-intron) boundary for pre-mRNA. Denoting the number of junction reads as N_m and the number of intron-exon2 boundary reads as N_p, the mature mRNA fraction can be estimated as:

$$\psi_m = \frac{N_m}{N_m + N_p} \tag{3.16}$$

Tilgner *et al.* (Tilgner et al., 2012) used a similar measurement to study the subcellular splicing completion with deep RNA-seq data. This direct method has several theoretical advantages: it provides an unbiased estimate of the splicing ratio, and does not require any normalization/ bias correction procedure. Nevertheless, it can only be effective for highly covered genes due to the requirement of having a sufficient amount of boundary/ junction reads.

In addition to using the junction reads only, the normalized reads counts (RPK, reads per kilo-base pairs) on introns and exons were used in the work of Windhager and colleagues (Windhager et al., 2012), which could be described with minor modifications as follows,

$$\psi_m = 1 - \frac{N_{in}/(l_{in} + l_r)}{N_{ex}/(l_{ex} - l_r + 1)} \tag{3.17}$$

where l_r is the read length; l_{in} and l_{ex} are the total lengths of intron and exons, respectively; N_{in} and N_{ex} are the corresponding numbers of reads mapped to intron (including partially) or exons. Note, slight modifications may be needed if upstream or downstream exons are shorter than the read length. This approach does not suffer from the low coverage issues of the junction/ boundary approach. However, normalization issues are more problematic, as the sequencing biases exist and may even give a negative value to this score.

In order to compare these two direct measurements to the probabilistic method, I conducted 5 simulation experiments with 187 intron-containing genes in yeast. The ground truth of the mRNA fraction in these 5 experiments linearly ranges from 0.1 to 0.9. Based on the yeast genome annotated by Ensembl R64-1-1.77, we simulated single-end reads with length of 76 bp, and average depth of RPK=400 with Spanki (Sturgill et al., 2013), a published RNA-seq reads simulator. Then the simulated reads were aligned to the yeast genome with HISAT (Kim et al., 2015).

The comparison results are shown in Fig 3.6, which illustrates that both direct methods and the probabilistic method with the MH sampler are reasonably accurate.

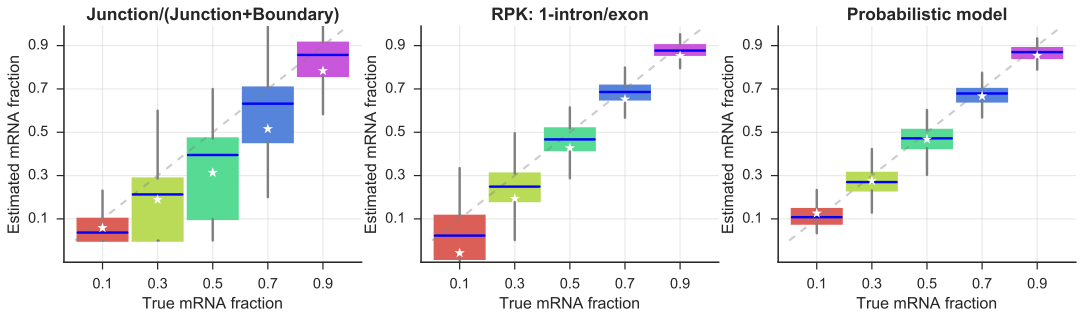

Figure 3.6: The comparison between two direct methods and a probabilistic method for estimating the spliced mRNA fractions. The single-end reads are simulated on 187 intron-containing genes in yeast. The reads have length of 76 bp and depth of RPK=400. The ground truth of the 5 experiments linearly ranges from 0.1 to 0.9. Each boxplot is the distribution of the estimate of the 187 genes, with the star for the mean and the short blue line for the median.

However, the direct method with junction and boundary reads suffers from a higher variability due to the insufficient reads. In addition, this method introduces a substantial bias, precisely lower mRNA fraction, for a set of genes in yeast with short exon 1 (much shorter than read length), thus there are more positions for intron-exon2 boundary than exon1-exon2 junction. The other direct method with read coverage (RPK) on exon and intron reduces the variation, but still suffers a lot from its poor normalization when mRNA fraction is low. Compared with the direct methods, the probabilistic method largely reduced the variation and bias at all levels of mRNA fraction. Namely the probabilistic method can give a much more accurate estimate than any direct counting method.

Challenges In the above simulation, the probabilistic method shows large improvements in accuracy with less variance and less bias in estimating splicing isoforms, compared to the two direct methods. However, even with the probabilistic method, there exist substantial challenges when the sequencing coverage is very low or the number of transcripts is very high.

In order to present these challenges, I further performed an experiment on 7,627 human genes with 3 to 11 transcript isoforms at 7 different coverages ranging from RPK of 25 to 1,600. The simulation is also performed by using the simulator Spanki (Sturgill et al., 2013), and the human gene annotation from GENCODE with release 22. Here, we introduce a metric, Gene Summarized Squared Error $gErr$, to measure

the difficulties in transcript isoform quantification, as follows,

$$gErr = \sum_{k=1}^{K} (\psi_k^* - \psi_k)^2 \tag{3.18}$$

where ψ_k^* is the estimate of ψ_k. The two squared errors are identical for the two transcripts in the two-isoform genes, thus this metric may be not fair for the two-isoform genes, and I only look at genes with 3 to 11 isoforms.

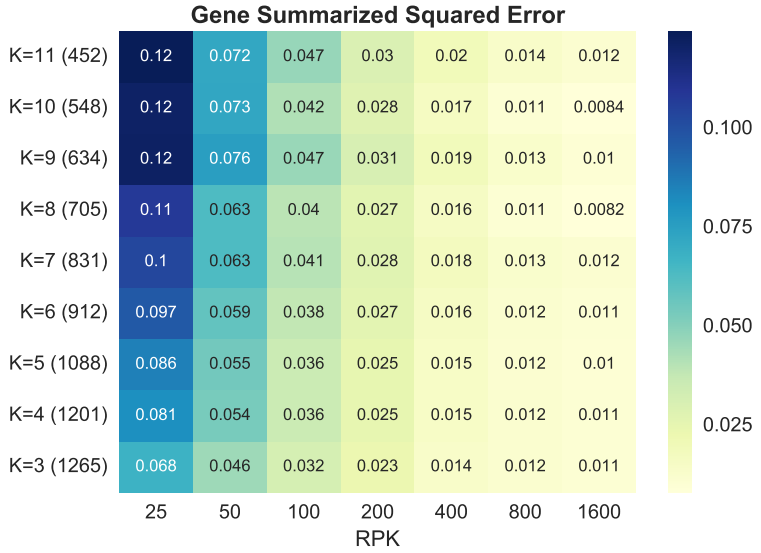

Figure 3.7: The challenges in isoform quantification from RNA-seq data with low coverages and large number of isoforms. Each pixel denotes an averaged Gene Summarized Squared Error for a set of genes with the same number of transcripts and coverage level. The y-axis is the number of transcript isoforms and the number of genes with the according transcript number is shown in the brackets.

Fig 3.7 presents the averaged Gene Summarized Squared Error for a set of genes with the same number of isoforms at the same coverage levels. The $gErr = 0.02$ roughly means the total error in the estimate at gene level is 0.141 (square root calculated as a single transcript), which can be a reasonable good threshold. Then, we can see that when $RPK \geq 400$, the averaged $gErr \leq 0.02$ for all isoform number from 3 to 11, but we still see the Gene Summarized Squared Error goes up when the number of isoforms increases. More strikingly, the averaged $gErr \geq 0.03$ when the coverages $RPK \leq 100$, and can even reach $gErr = 0.12$ when $RPK \leq 25$ and number of isoforms more than 8.

These tough challenges can be normally reduced by very deep sequencing for standard RNA-seq experiments, but usually is hard for large number of RNA-seq libraries, e.g., time-series or single-cell RNA-seq, especially with limited budgets. Thus, these challenges widely exist and the demands of solutions are increasingly high, which motivate the main research in this thesis. In chapters 4 and 5, I will introduce two structured Bayesian methods to integrate multiple data sources to ameliorate these challenges.

3.4 A case study on RNA splicing efficiency

In the above sections, we see that the probabilistic model can accurately estimate fractions of splicing isoforms with a reasonable coverage and transcript numbers. Here, I introduce a case study on mRNA processing and splicing by measuring nascent RNA with 4tU labelling. This is a collaborative work with Prof. Jean Beggs's lab and Dr. Sander Granneman's lab both in the University of Edinburgh, and the full paper has been published and is available (Barrass et al., 2015). Here, I only focus on the contribution I made for this work, including the quantification of mRNA proportions in the nascent RNA, the measurement of the splicing efficiency, and the exploration of features associated with splicing speed.

3.4.1 4tU labelling for RNA splicing

The RNA levels detected in cells at steady state are the consequence of multiple dynamic processes within the cell. For the intron-containing genes, the level of mature transcripts is influenced by splicing, as well as by synthesis and decay. Splicing of pre-mRNAs (precursors of messenger RNAs) occurs in the nucleus, often co-transcriptionally (see Chapter 2.1.2). The spliced mRNA is exported to the cytoplasm where it can be translated, whereas the excised intron, which has a branched, lariat structure, is rapidly debranched and degraded. Measurements of in vivo RNA processing rates and efficiencies depend on the ability to estimate the levels of the unprocessed precursors and processing intermediates in cell extracts; however, this is challenging because they are highly transient and present in low abundance in wild-type cells at steady state.

Nascent RNA labelling with uridine analogs like 4-thiouridine (4sU) and 4-thiouracil (4tU) provides a way of proportionally enriching classes of nascent RNA that are dif-

ficult to detect in wild-type cells at steady state. Labelling with 4sU to the newly synthesized transcripts has been used to measure RNA synthesis, decay and splicing rates in human and yeast; however, the shortest labelling time was 3 min, by which time a substantial fraction of the newly transcribed RNA was already spliced or degraded (Schulz et al., 2013). Therefore, to be able to measure RNA processing rates with higher accuracy and resolution transcriptome-wide, the Beggs's lab have developed an extremely short (as little as 60 s) 4tU RNA labeling protocol and combined it with high-throughput RNA sequencing (RNA-seq).

Our original paper demonstrated that this method (4tU-seq) readily detects low abundance and labile transcripts in wild-type cells that are normally detected only in cells that are defective in RNA degradation. Therefore, it opens a door to measure relative pre-mRNA splicing kinetics transcriptome-wide (in this work, we focus on budding yeast), and we further investigated the features associated with splicing efficiency.

3.4.2 Estimate of mRNA proportion and splicing speed

As shown in the simulation experiment in Fig 3.6, the probabilistic method gives a much more accurate estimate of mature mRNA with much lower variance and less bias, compared with direct measurements. Here, we applied this probabilistic method to measure the mRNA fraction in the 4tU-labelled nascent RNA.

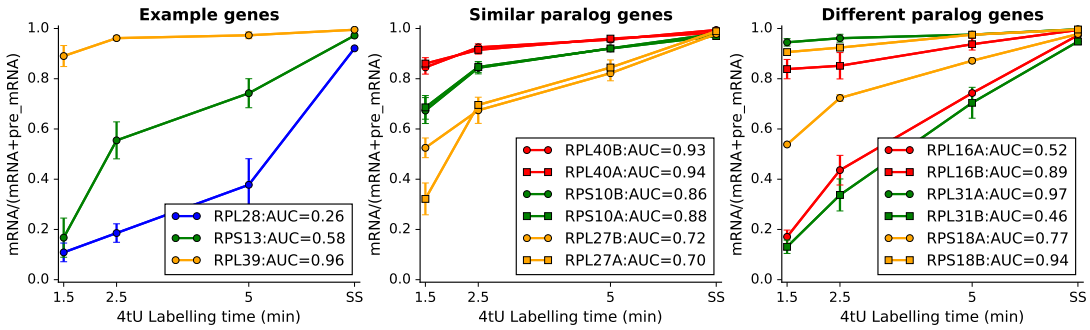

Figure 3.8: Example genes with estimated mRNA fractions. (Left panel) Three well-studied example gene with RT-PCR validation. (middle panel) Three pairs of paralogs with very similar splicing dynamic profiles. (right panel) Three pairs of paralogs with very different splicing dynamic profiles. Paralogs are genes related by duplication within a genome, and corresponding exons between paralogs usually have very similar sequence whereas introns are more different.

Besides the more accurate quantification, another major benefit of using a probabilistic model for estimating the abundance of precursor and mature mRNA lies in the possibility of obtaining posterior confidence intervals (CI) on the splicing ratios, which can then be used to filter noise in a principled way. In the analysis, we only retained genes with $95\%CI < 0.3$, filtering out genes for which reliable estimation was not possible (primarily due to low sequence coverage). This filtering improved the correlation between replicates from 0.757 to 0.864 (see Additional file 1: Tables S3, S4 and S7 in the original paper (Barrass et al., 2015)).

In this 4tU-seq analysis, 187 intron-containing transcripts were selected that had a fragments per kilobase per million reads score (FPKM) of > 10, that did not encode snoRNAs in the introns, and that only contained a single intron (to simplify the data analyses). Following posterior CI filtering, data for 82 ribosomal protein (RP) and 35 non-RP intron-containing genes were retained for the splicing ratio analysis (Figure A.1 in Appendix A). This figure shows the mean splicing ratio of the three replicates at different time points for the 35 non-RP and 82 RP intron-containing genes, in which transcripts are ranked by speed of splicing (see definition in next section) from fastest (top) to slowest (bottom).

Three well-expressed ribosomal protein genes *RPL28*, *RPS13* and *RPL39* are also shown in Fig 3.8(left panel). To validate the 4tU-labelled RNA-seq data, 4tU-RT-qPCR was performed on these three example genes (Fig. 6a, b in original paper), which shows a good agreement with the splicing dynamic profiles.

3.4.3 Associated features with splicing speed

In order to define the splicing speed from the splicing dynamics at 1.5 min, 2.5 min and 5.0 min, we introduce the Area Under the Curve (AUC) to measure the splicing speed, as follows,

$$AUC = [(\psi_{1.5} + \psi_{2.5})/(2.5 - 1.5)/2 + (\psi_{2.5} + \psi_{5.0})/(5.0 - 2.5)/2]/\psi_{SS} \quad (3.19)$$

where the index of ψ is the time at minutes, except SS means steady state, and the ψ means the mRNA fraction at each time point. This calculation is equivalent to the AUC of the curves as shown in Fig 3.8, and the AUC scores for these example genes are also shown in the legends.

Our results show that different introns were spliced at different rates based on area under the curve (AUC) calculations (see Figure A.1). Various features of introns could

impact their speed of splicing. These include the strength of sequences motif at 5' splice site (5'ss), 3' splice site (3'ss) and branch point (BP), as well as the secondary structure of the intron. We looked for correlations between different transcript features and the relative speed of splicing. Analysing the fastest-splicing and slowest-splicing thirds of transcripts, we noticed there was a marked difference in the behaviour of intronic RP transcripts compared to that of non-RP intronic genes: for the intronic RP transcripts, a highly significant difference (`Wilcoxon`'s test $p < 3 \times 10^{-4}$) was found only with regard to the normalized secondary structure scores of RP introns, with the major contribution coming from the 5ss to BP region ($p < 1 \times 10^{-4}$; see Fig 3.9(left panel)). In the case of the non-RP transcripts, those that were spliced faster generally had less secondary structure at the 3ss and a shorter exon 2 (Fig 3.9(right panel)). All the feature comparisons are shown in Figure A.2 (RP transcripts) and Figure A.3 (non-RP transcripts). The failure to see a significant effect of 5'ss, 3'ss and BP sequences in RP transcripts was likely due to the high similarity of these features.

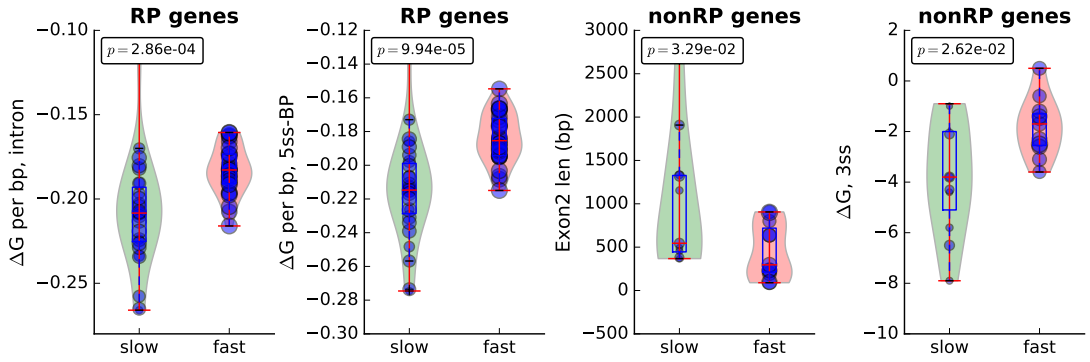

Figure 3.9: Features associated with splicing speed for RP genes or nonRP genes. Left panel: comparison of secondary structure scores (ΔG; y-axis) of the whole intron or 5'SS-BP for the fastest-splicing and slowest-splicing thirds of 82 ribosomal protein (RP) intron-containing genes (x-axis). The violin plots show the distribution of the features, and the blue dots represent individual RP genes, with dot size corresponding to the splicing speed. The p-value was obtained using Wilcoxon's test. Right panel: comparison of exon 2 length and secondary structure at the 3' splice site (3'ss) (y-axis) for 35 non-ribosomal protein (non-RP) intron-containing genes to splicing speed (x-axis), with the same violin plot format. Note, ΔG is the predicted energy of secondary structure: lower ΔG means more complex structure

The effect of intron secondary structure should be evident for paralogous RP genes

that share highly related or identical exon sequences but have different intron sequences. Indeed, we found that some paralogs, such as RPL27A/B, RPS10A/B and RPL40A/B, that have introns with similar predicted secondary structure stabilities (ΔG values), had similar splicing speeds (Fig 3.8, middle panel), whereas paralogs with introns that have different predicted ΔG values, such as RPL31A/B and RPS18A/B, had different splicing speeds (Fig 3.8, right panel).

Because the predicted secondary structure within RP introns was significantly correlated with the splicing speed, we further explored the intron sequences and found that several short k-mers were significantly correlated with splicing speed (Fig 3.10). Generally, fast splicing introns were enriched for adenosines, while slower splicing introns had a higher density of uridines. Surprisingly, the proportion of "A" and "U" in the introns was highly negatively correlated (Pearsons correlation coefficient< -0.75). Furthermore, we used the above features (all listed in Additional file 1: Tables S8 and S9 in the original paper) to predict the splicing speed by a random forest regression model. Fig 3.11 shows that the splicing speeds for RP pre-mRNAs could be well predicted by these features (Pearsons R=0.58 between observed and predicted splicing speed).

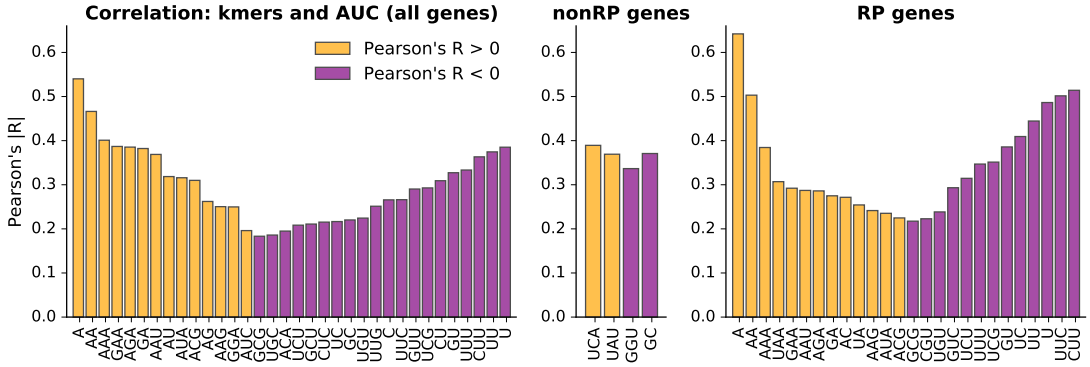

Figure 3.10: Correlated k-mers for splicing speed. Pearson's correlations between splicing speed and sequence patterns show the significantly correlated features ($p < 0.05$) to splicing speeds for all 117 intron-containing genes (left panel), 35 non-RP intron-containing genes (middle panel), and 82 RP intron-containing genes (right panel). The features are the occurrence of the specific k-mers in the intron. Yellow represents positive correlation with splicing speed and purple represents negative correlation.

Overall, the most significant features we have been able to identify that distinguish intron splicing rates for RP pre-mRNAs in budding yeast are the predicted secondary

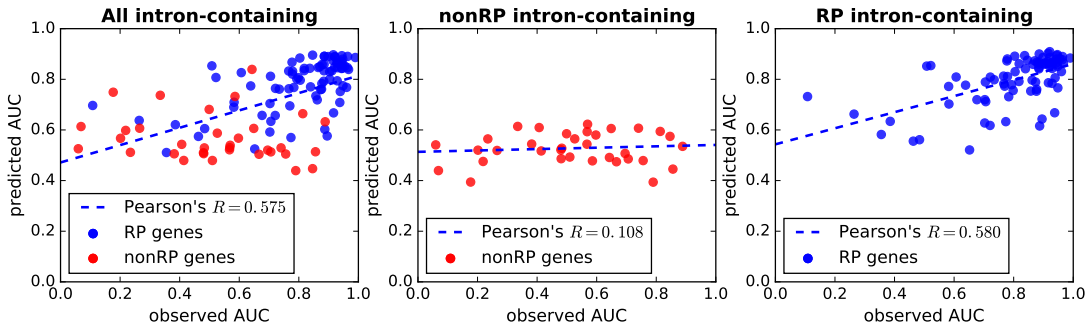

Figure 3.11: Prediction of splicing speed with sequence related features. Scatterplot of observed and predicted splicing speeds from the associated features. The features include secondary structures, splice site scores, intron length, exon length k-mers etc. The predictions are obtained by random forest regression with automatic feature selection. The splicing speed can be predicted reasonably well for RP genes while the prediction is poor for nonRP genes.

structure in the region between the 5'ss and the BP, and A or U density. In our data, slower splicing was associated with greater predicted secondary structure stability and U-richness in the intron. The effect of secondary structure was observed to be strongest for the set of highly expressed RP transcripts whose introns were mostly longer than average in budding yeast. Together, our work indicates that efficient splicing of RP pre-mRNA transcripts requires an optimal amount of secondary structure between the 5ss and the BP, with either too much or too little being detrimental. Furthermore, our results suggest that in the endogenous context, RP transcripts that splice slower are more often victims of too much structure rather than too little.

3.5 Discussion

In this chapter, I introduced a widely used probabilistic framework with a mixture model to quantify splicing isoforms, and compared three inference algorithms: EM algorithm, MH sampler and Gibbs sampler. Our simulation analysis showed that the EM algorithm is the most efficient method, but it can only return a point estimate. Two MCMC sampling methods, MH sampler and Gibbs sampler, could return the whole posterior distribution, however, the Gibbs sampler suffers from its high computational costs. Alternatively, I showed that with a `Logistic-Normal` proposal distribution, the MH sampler could accurately sample the posterior distribution, with a much faster speed compared with the Gibbs sampler.

Then, I compared the performance of the probabilistic method (with MH sampler for estimate) with two direct measurements on estimating the mRNA fractions from the mixture of pre-mRNA and mRNA. The simulation study on 187 yeast intron-containing genes shows the probabilistic method gives much more accurate estimate and lower bias at all true value settings compared to the two direct counting strategies. Thanks to the accurate estimate in the splicing analysis, the probabilistic method has been used in a collaborative work on RNA splicing efficiency and the discovery of its associated features (Barrass et al., 2015). In this work, I used the above probabilistic method to measure the mRNA fraction in nascent RNA at 1.5 min, 2.5 min, 5.0 min and steady state. Then based on the estimates of the mRNA fractions across multiple time points, we introduced a metric to measure the RNA splicing efficiency. Furthermore, we found a set of features highly associated with splicing efficiency, especially the secondary structure in introns for the 87 RP genes, and secondary structure around the 3' splice site for the 35 non-RP genes. Together, by using a set of sequence related features, e.g., short k-mers, we found the splicing efficiency can be predicted reasonably well by a random forest regression model.

In addition to the above case study, the probabilistic method can be also applied into other splicing related work. For example, another collaborative work also from the Beggs's lab (Aslanzadeh et al., 2017) introduces mutations on the elongation rate of Pol-II to make the transcription rate either faster or slower than the wild type. By introducing the mutations of elongation rate, we could study the effects of transcription on splicing, including its efficiency and fidelity. Actually, by measuring the fractions of the annotated splicing events and novel splicing events, we found that both the fast and slow mutations increase the splicing errors on recognising 5'ss and (or) 3'ss. In other words, the wild type transcription rate is probably around the optimal rate to ensure splicing processes correctly. In addition, with a similar analysis pipeline in the section 3.4, we also found a set of sequence related features to the splicing fidelity, for example the intron length, the motif strength of the 3' splice site.

Even though the probabilistic method has been successfully applied to a few real studies on RNA splicing, the isoform quantification can still be very challenging when the coverages are very low and the numbers of transcripts are very high. As shown in Chapter 3.3.2, when the coverages are lower than $RPK = 100$, the estimate will suffer from high variance and low confidence. In spite of the deep sequencing, there are still many chances to end up with low coverages for a set of genes. For example, in the time-series RNA-seq experiments, there are often genes with low expression in

a certain time point, and the coverages are usually not high due to the limited budget for large number of time points. In addition, in single-cell RNA-seq experiments, the coverages can be even much lower, as there are often thousands of cells that are sequenced together. Therefore, the demands for a solution to these tough challenges are increasingly high. In the next two chapters, I will introduce two Bayesian methods with a structured prior distribution to improve the analysis in splicing with time-series RNA-seq data or single-cell RNA-seq data, respectively.

Chapter 4

Modelling splicing in time-series RNA-seq data

4.1 Introduction

As shown in chapter 2.1, alternative splicing is an important post-transcriptional mechanism of regulation of gene expression, plays a vital role in many biological processes, and greatly increases the diversity of the proteome. RNA-seq became an effective tool for studying gene expression and splicing after its invention. However, its short read length and sequencing bias make the direct quantification of splicing isoforms very difficult. In Chapter 3, I showed that probabilistic methods with mixture models have been applied to solve this problem and achieved great improvements in accuracy. This statistical framework has been employed and implemented in quite a few publicly available methods, including IsoEM (Nicolae et al., 2011), Cufflinks (Trapnell et al., 2010), MISO (Katz et al., 2010), and BitSeq (Glaus et al., 2012). More recently, a couple of methods have tried to break the computational bottleneck in transcriptome quantification for thousands of samples, for example Sailfish (Patro et al., 2014) and Kallisto (Bray et al., 2016).

Most of these computational methods can quantify isoform proportions accurately in many cases (Kanitz et al., 2015). However, for all methods isoform quantification at low coverages remains challenging (see Chapter 3.3.2). A natural approach in these cases is to exploit additional information, for example exploiting correlations across different experiments arising out of structured experimental designs such as time series or dosage response experiments. Time series RNA-seq designs, in particular, are becoming increasingly popular as an effective tool to investigate the dynamics of gene

expression in a range of systems (Bar-Joseph et al., 2012; Tuomela et al., 2012; Zhang et al., 2014; Honkela et al., 2015). On example is the case study in Chapter 3.4 where we used time-series RNA-seq with 4tU labelling to nascent RNAs to study splicing kinetics and efficiency (Barrass et al., 2015). However, there is still a lack of methods that can exploit structured experimental designs in order to improve isoform estimation. This methodological gap also negatively affects the ability to design effective experiments: for example, it is difficult to understand whether resources should be invested in gathering more time points, or in sequencing at a deeper level on a more limited number of samples.

In this chapter, I will present a new methodology from our recent publication (Huang and Sanguinetti, 2016), DICEseq (Dynamic Isoform spliCing Estimator via sequencing data) to jointly estimate the dynamics of isoform proportions from RNA-seq experiments with structured experimental designs. DICEseq is a Bayesian method based on a mixture model whose mixing proportions represent isoform fractions, as in (Katz et al., 2010; Glaus et al., 2012); however, DICEseq incorporates the correlations induced by the structured design by coupling the isoform proportions in different samples through a latent Gaussian process (GP) prior. By doing so, DICEseq effectively transfers information between samples, borrowing strength which can aid in identifying the isoform proportions. Our results show that DICEseq consistently improves in accuracy and reproducibility over the state of the art. This improvement can be very significant for a large fraction of genes: on one real data set, the correlation between estimates from replicate data sets increased by over 10% across one third of the genes with low expression as a result of taking temporal information into account. Furthermore, simulation studies indicate that DICEseq can be an important tool in experimental design, enabling an effective trade-off of resources between sequencing depth and sample numbers. DICEseq therefore offers an effective way to maximise information extraction from complex high-throughput data sets.

4.2 Methodology

Here, I will describe the details of the DICEseq method, which is an extension of the probabilistic method with mixture model described in Chapter 3. This method models multiple time points jointly rather than individually. The correlations between time points are encoded by a Gaussian process prior distribution. Before entering the full model, let us first look at the Gaussian process in the next subsection.

4.2.1 Gaussian processes

Gaussian processes (GPs) are a generalisation of the multivariate normal distribution to infinite-dimensional random functions. The key property of a GP is that all of its finite dimensional marginals are multivariate normals; in other words, evaluating a random function drawn from a GP at a finite set of points yields a normally distributed random vector. A GP over a suitable input space $\mathcal{T}$ is uniquely specified by a mean function $m\colon \mathcal{T} \to \mathbb{R}$ and a covariance function $k\colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}$, which models how correlations between function outputs depend on the inputs. In this chapter, we will identify the input space $\mathcal{T}$ with the time axis, and use as a covariance function the *squared exponential* (or radial basis function, RBF) covariance

$$k(t_1, t_2) = \theta_1 \texttt{exp}(-\frac{1}{2\theta_2}(t_1 - t_2)^2). \tag{4.1}$$

The covariance function depends on two hyper-parameters, the prior variance θ_1 and the (squared) correlation length scale θ_2.

The fundamental property of GPs relates the abstract function space view of GPs reported above with the explicit parametric form of their finite dimensional marginals. Let f denote a random function sampled from a GP, $(t_1, \ldots, t_N)$ denote a set of input (time) points and $\mathbf{f} = (f(t_1), \ldots, f(t_N))$ the vector obtained by evaluating the function f over the input points. Then, we have that

$$f \sim \mathcal{GP}(m, k) \leftrightarrow \mathbf{f} \sim \mathcal{N}(\boldsymbol{m}, K) \tag{4.2}$$

where $\boldsymbol{m}$ and K are obtained by evaluating the mean and covariance functions over the set of points $(t_1, \ldots, t_N)$ (and pairs thereof). The fundamental property (4.2) is key to the success of GPs as a practical tool for Bayesian inference: given observations of the function values $\mathbf{y}$, it is in principle straightforward to obtain posterior predictions of the function values everywhere by applying Bayes' theorem

$$p(f(t_{new}|\mathbf{y})) \propto \int d\mathbf{f} p(\mathbf{f}, f(t_{new})) p(\mathbf{y}|\mathbf{f}) \tag{4.3}$$

If the observation noise model $p(\mathbf{y}|\mathbf{f})$ is Gaussian, then the integral in (4.3) is analytically computable. Notice that equation (4.3) provides a way of predicting the latent function *at all time points*, not just the observation points. In the following, we describe an algorithm to approximate the computation of (4.3) for multinomial observations. For a thorough review of GPs and their use in modern machine learning, we refer the reader to the excellent book (Rasmussen and Williams, 2006).

4.2.2 Posterior of splicing dynamics with GP prior

Before looking at the full posterior distribution on multiple time points, I will briefly review here the mixture modelling framework for isoform identification at a single time point, as described in Chapter 3.1. Here, we will use the model on a per gene basis again. Assume, we have N reads $R_{1:N}$ aligned to a gene with C isoforms. Each read R_n has its *identity* $I_n \in \{1, \ldots, C\}$, i.e. which specific isoform it originated from, but, unless the read is aligned to an isoform specific region, e.g., a junction, we will not know its identity. The proportion of each specific isoform within the pool of total mRNA is defined by the vector Ψ, whose entries must be positive and sum to 1. We can then define the likelihood of isoform proportions Ψ as mixture model as follows

$$P(R_{1:N}|\Psi) = \prod_{n=1}^{N} \sum_{I_n=1}^{C} P(R_n|I_n)P(I_n|\Psi). \tag{4.4}$$

The conditional distribution of $I_n|\Psi$ is assumed to be Multinomial, $(I_n|\Psi) \sim \texttt{Multinomial}(\Psi * w)$ where w is a weight vector adjusting the isoform proportion by the effective length of each isoform. The term $P(R_n|I_n)$ encodes the probability of observing a certain read coming from a specific isoform I_n, and can include the bias correction in this term (see more details in Chapter 3.2.2 and 3.2.3).

Extending the time independent model to time series RNA-seq experiments involves a choice on how to model temporal correlations between the values of Ψ at different time points; we will use a flexible non-parametric prior in the form of a Gaussian process for this (see above section).

Given a set of RNA-seq reads $\boldsymbol{R} = [R^{(1)}_{1:N_1}, ..., R^{(T)}_{1:N_T}]$ for T time points that are aligned to a gene with C isoforms, the posterior of the splicing dynamics for the isoform proportions $\boldsymbol{\Psi} = [\Psi^{(1)}_{1:C}, ..., \Psi^{(T)}_{1:C}]$ is as follows,

$$\begin{aligned} P(\boldsymbol{\Psi}|\boldsymbol{\Theta}, \boldsymbol{R}) &\propto P(\boldsymbol{\Theta})P(\boldsymbol{\Psi}|\boldsymbol{\Theta}) \times \prod_{t=1}^{T} P(R^{(t)}_{1:N_t}|\Psi^{(t)}) \\ &\propto P(\boldsymbol{\Theta})P(\boldsymbol{\Psi}|\boldsymbol{\Theta}) \times \prod_{t=1}^{T} \prod_{n=1}^{N_t} \sum_{I^{(t)}_n=1}^{C} P(R^{(t)}_n|I^{(t)}_n)P(I^{(t)}_n|\Psi^{(t)}) \end{aligned} \tag{4.5}$$

where Ψ is assumed as a $\texttt{Softmax}$ function of latent variable Y, i.e., $\psi_c = e^{y_c} / \sum_{i=1}^{C} e^{y_i}$, and $y_C = 0$ to make the correspondence. Also $Y_c = [y_c^{(1)}, ..., y_c^{(T)}]$ follows a Gaussian process with its isoform specific hyperparameters $\boldsymbol{\theta}_c$ and mean $\boldsymbol{m}_c$. By introducing the GP prior here, the joint analysis of time series RNA-seq data becomes possible, as shown in a cartoon in Figure 4.1.

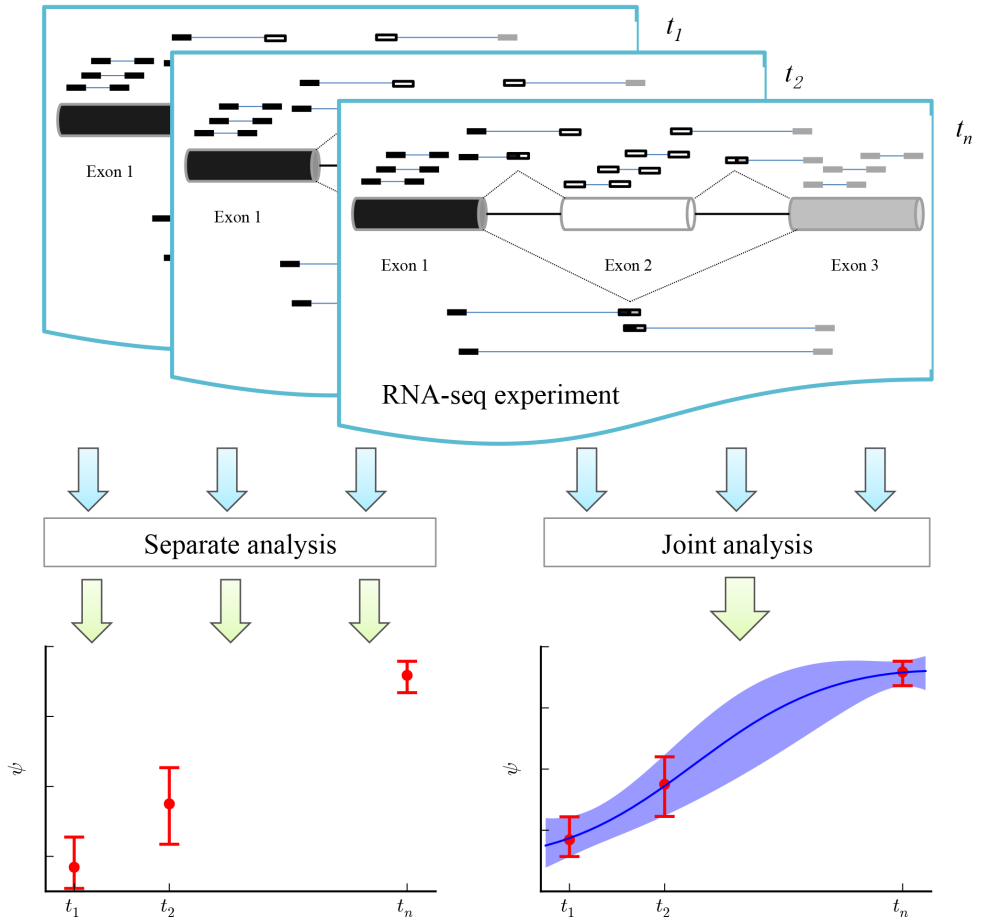

Figure 4.1: A cartoon comparison between separate and joint analysis of time-series RNA-seq experiments. In the example gene, there are two isoforms with one alternative exon (the white one), and many paired-end reads are aligned to the genome for isoform quantification. The "separate analysis" estimates the isoform proportions for three time points independently, but the "joint analysis" estimate them together with a joint Gaussian process prior.

We assume in the following that the prior GP has zero mean, but this can be adjusted in a straightforward way to a more informative prior. Hyperparameters can also be sampled, however this leads to a much more complex inference problem since latent function values and hyperparameters are strongly correlated. We therefore fix $\theta_{c,1} = 3.0$, so that the 95% prior confidence intervals of ψ at an independent time point goes from 0.03 to 0.97, and set the second hyperparameter θ_2 empirically to account for approximately 20-40% of the duration of the experiment. A sensitivity analysis to θ_2 is provided in Supplementary Table B.1 and Supplementary Figure B.3 (in Appendix B, same below). Inference of θ_2 can also be achieved by a straightforward extension

of Algorithm 7 (see Algorithm 9 in Appendix B). However, this comes with a large additional computational cost, and in our experiments does not lead to improvements in accuracy; this is probably due to the fact that the typical RNA-seq time series is too short to carry enough information about the value of hyperparameters.

Having defined the posterior of the splicing dynamics, we introduce a Metropolis-Hasting sampler in Algorithm 7, which is a Markov chain Monte Carlo (MCMC) method, to infer the posterior of the splicing dynamics.

Algorithm 7: Metropolis-Hastings sampler for posterior of latent $\boldsymbol{Y}$

Data: $T, \boldsymbol{R}, \boldsymbol{\Theta}, \lambda$

Initialize: $\boldsymbol{Y}^{(0)}$; $\boldsymbol{\Psi}^{(0)} = \texttt{Softmax}(\boldsymbol{Y}^{(0)})$; $\boldsymbol{K} = \texttt{GPcov}(\Theta, T)$

for $i = 0$ *to* H **do**

  Sample: $\mu \sim U(0, 1)$

  Sample: $\boldsymbol{Y}^* \sim Q_y(\boldsymbol{Y}^* | \boldsymbol{Y}^{(i)}, \lambda \boldsymbol{K})$; $\boldsymbol{\Psi}^* = \texttt{Softmax}(\boldsymbol{Y}^*)$

  if $\mu < \texttt{min}\left\{ \dfrac{P(\boldsymbol{\Psi}^* | \boldsymbol{R}) \times Q_y(\boldsymbol{Y}^{(i)} | \boldsymbol{Y}^*, \lambda \boldsymbol{K})}{P(\boldsymbol{\Psi}^{(i)} | \boldsymbol{R}) \times Q_y(\boldsymbol{Y}^* | \boldsymbol{Y}^{(i)}, \lambda \boldsymbol{K})}, 1 \right\}$ **then**

    $\boldsymbol{Y}^{(i+1)} \leftarrow \boldsymbol{Y}^*$; $\boldsymbol{\Psi}^{(i+1)} \leftarrow \boldsymbol{\Psi}^*$

  else

    $\boldsymbol{Y}^{(i+1)} \leftarrow \boldsymbol{Y}^{(i)}$; $\boldsymbol{\Psi}^{(i+1)} \leftarrow \boldsymbol{\Psi}^{(i)}$

Here, the proposal distribution Q_y for Y_c is a multivariate Gaussian distribution, whose mean is the last accepted $Y_c^{(i)}$, and the covariance matrix is defined by the fixed hyper-parameters $\boldsymbol{\theta}_c$ and the times T, but adjusted to the data itself, including the empirical variance of y, the number of isoforms, and number of time points, to ensure the 30-50% acceptance ratio. Namely, $\hat{K}_c = \lambda K_c; \lambda = (5\sigma_y^2)/(CT\theta_{c,1})$, and the proposal distribution is $\mathcal{N}(Y_c^{(i)}, \hat{K}_c)$. Notice that, in contrast to the MISO algorithm Katz et al. (2010), our sampler directly collapses the read identity variables, leading to considerable speedups when the number of isoforms is not too high, i.e., less than 10.

For each gene, the initial MCMC chain contains 1000 iterations. Then Geweke's diagnostic Z score (Geweke, 1991) is applied to check the convergence of $\boldsymbol{Y}$, using the first 10% and the last 50% iteration of the sampled chain. If $|Z| > 2$, then 100 more iterations will be added until the criterion is passed.

4.2.3 Simulation and data processing

Simulated reads in fastq format were generated from Spanki v0.5.0 (Sturgill et al., 2013). It is based on the human gene annotation and genome sequences which were

downloaded from GENCODE with release 22. In addition to exclusively keeping protein coding genes, we further removed those genes that only have one isoform (for these the problem is trivial) or overlap with others. Note that overlapping genes on the same strand can also be accommodated by considering an extended isoform identification problem, whereby the identity of the read also includes the gene to which it belongs; this however requires a modification of the annotation file and was not considered for the purposes of illustrating our algorithm. Consequently, 90,759 isoforms from 11,426 genes were included for simulation. We randomly generated isoform ratios for each gene at 8 time points, with an assumption of either Gaussian process or first-order dynamics (i.e., $y(t) = a\exp(-c \times t) + b$). Then the randomly generated isoform ratios were multiplied with the fixed library reads-per-kilobase (RPK, ranging from 50 to 1,600), to further define the number of isoform specific reads for the Spanki simulator.

4tU-seq data sets are available from the Gene Expression Omnibus (GEO; accession number GSE70378). The yeast gene annotation and genome sequences were downloaded from Ensembl with version R64-1-1, and all 309 intron-containing genes were included for analysis.

Circadian RNA-seq and microarray data sets from mouse liver were downloaded from GEO: GSE54652. The gene annotation and genome sequences were downloaded from GENCODE with release M6. Based on the annotation, we included 55,440 isoforms from 10,553 multiple-isoform, non-overlap, protein-coding genes. Processed microarray data (Zhang et al., 2014), which are based on Affymetrix MoGene 1.0 ST, were employed for validation of the isoform estimate from RNA-seq. The microarray probe ids were mapped to GENCODE ids by Ensembl BioMart, leaving 30534 isoforms from 9755 genes for study.

All above RNA-seq data sets were downloaded in fastq format, and first aligned to corresponding Genome sequences above via HISAT 0.1.6-beta (Kim et al., 2015), in paired-end mode with default setting.

4.3 Results

4.3.1 Methods comparison using simulated reads

In order to assess the performance of DICEseq, we compared it with three commonly used methods in their latest version: IsoEM v1.1.4 (Nicolae et al., 2011), MISO v0.5.3

(Katz et al., 2010), and Cufflinks v2.2.1 (Trapnell et al., 2010). We also report results for a variant of DICEseq which ignores temporal correlations (DICE-sepa). Notice that DICE-sepa is essentially the same as MISO as a model, only differing in the estimation procedure and prior (collapsed MH sampler and softmax of a Gaussian). Simulated reads for 11,426 human protein coding genes, accounting for a total of 90,759 distinct isoforms, were generated by Spanki v0.5.0 (Sturgill et al., 2013) with coverage from RPK of 50 to 1600 for 8 time points. We initially induced a temporal correlation between isoform proportions at different time points by enforcing the assumption of Gaussian process dynamics. All methods used paired-end reads, with the exception of MISO, which provided better performance in these experiments using single-end reads (see Figure B.2 and Table B.3 in Appendix B). We focus here on comparing the accuracy of the various methods; for a comparison of computational performance see Figure B.1 in Appendix B.

We first studied the accuracy of each method at different coverage levels. We report average accuracy by computing the mean absolute error (MAE) between inferred isoform ratios and the truth from all the 90,759 isoforms of the 11,426 genes and 8 time points. Figure 4.2A shows that all methods return accurate estimates, and that the errors generally decrease with the increase of coverages. As expected, DICEseq is able to exploit effectively the temporal information, providing a significantly lower mean absolute error than the other methods, an advantage which is particularly marked at lower coverage. In a real RNA-seq time series experiment, many genes are likely to have relatively low coverage in at least one time point (see next sections for our real data experiments), therefore the improved performance of DICEseq is likely to be important in quantifying isoforms for a substantial fraction of genes.

A second, often very important, metric is the confidence intervals associated with the predictions. These can be useful when deciding e.g. which genes to include in downstream analyses as in (Barrass et al., 2015). We examined the average size of the confidence intervals for the three Bayesian methods DICEseq, MISO and Cufflinks as we vary the simulated coverage levels. As expected, confidence intervals shrink as we increase coverage for all three methods. However, DICEseq clearly is able to provide more confident predictions at all coverage levels (Figure 4.2B). DICEseq is particularly strong at lower coverage; this is important, as often the confidence of an estimate is used to select genes which are further analysed (Barrass et al., 2015).

Thirdly, we investigate the influence of isoform number on the quality of the estimate at a specified coverage level. By selecting the genes with a specific number of

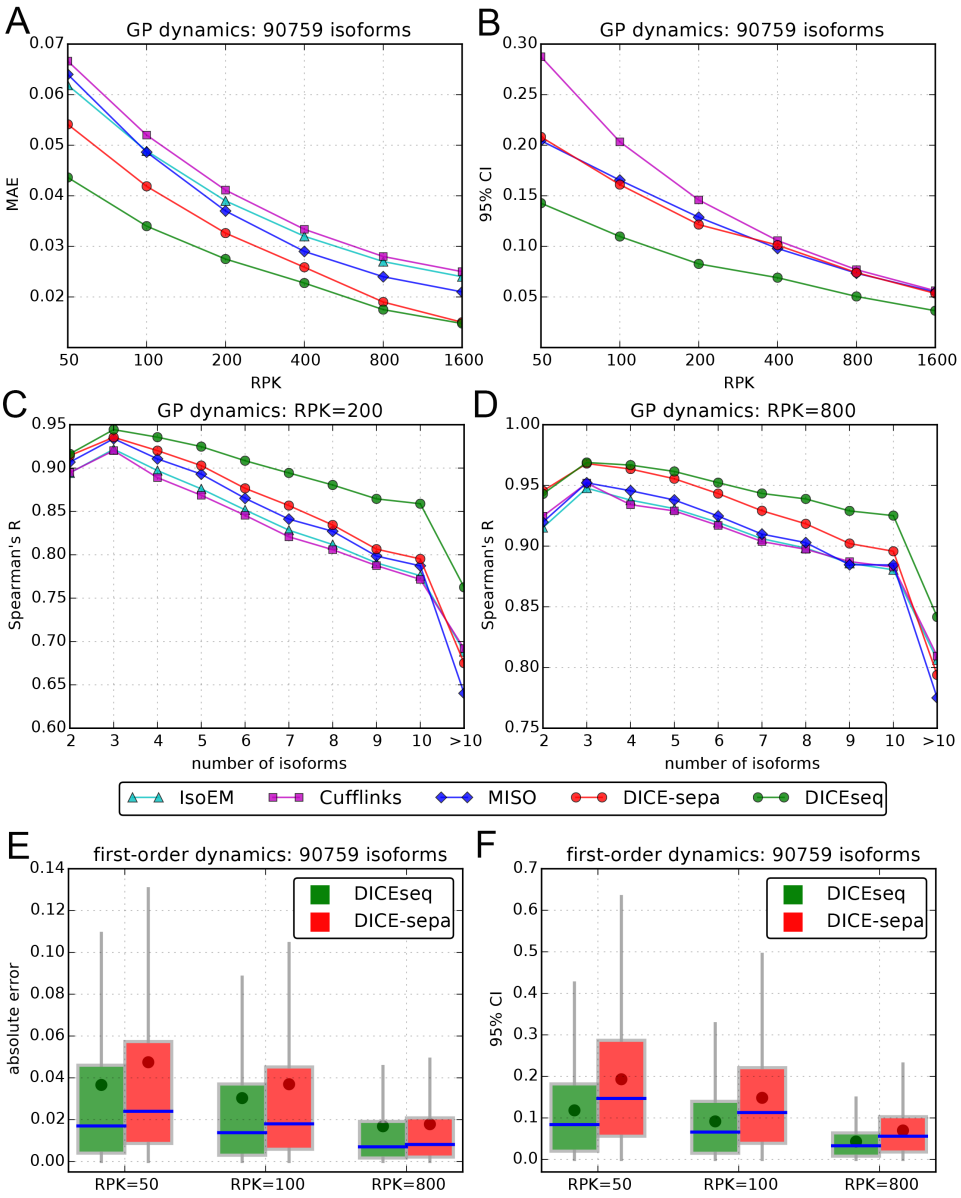

Figure 4.2: Comparison of accuracy between methods using simulated reads. **(a)** Mean absolute error between estimated isoform proportion and the truth. **(b)** 95% confidence interval of the estimates. **(c)** Influence of the number of isoforms on the estimates when RPK=200. **(d)** Influence of the number of isoforms on the estimates when RPK=800. The simulation is based on GP dynamics assumption for **(a-d)**. **(e)** Boxplot of absolute error between estimated isoform proportion and the truth. **(f)** Boxplot of 95% confidence interval of the estimates. The round dot is the mean. The simulation is based on first-order dynamics assumption for **(e-f)**.

isoform, Figure 4.2C (RPK=200) and 2D (RPK=800) both show that the rank correlation (Spearman's correlation) coefficient between the estimated isoform proportions and the truth generally decreases as the number of isoform increases. This is expected, because the presence of more isoforms reduces the number of uniquely assignable reads. Once again, we see that including temporal information can yield significantly improved estimates, with DICEseq yielding an improvement in rank correlation of more than five percentage points for genes with many isoforms ($>$8).

Finally, we investigate the robustness of DICEseq to model mismatch. To do so, we generated time series data where the isoform proportions vary according to a first-order dynamical system (rather than a Gaussian process, see simulation subsection), a commonly used modelling hypothesis (Eser et al., 2016). Figure 4.2E-F clearly shows that incorporating temporal information yields a considerable improvement, even under model mismatch. This improvement is particularly marked at low coverages. Notice that the mean accuracy (represented by a dot in the box plots) is very similar to the one obtained under the GP assumption (Figure 4.2A). Additional simulations varying hyper-parameters were also performed (see Appendix B), and Table B.1 again shows robustness to mis-specification of the hyper-parameters.

In summary, the results of these simulation studies show that DICEseq can provide accurate reconstruction of isoform proportions, and can successfully leverage temporal information to provide more accurate and confident predictions at low coverage and for higher numbers of isoforms.

4.3.2 Design of time-series RNA-seq experiments

Incorporating temporal information in the analysis of time series experiments is desirable in principle, because it provides experimentalists with a further direction for experimental design. Intuitively, resources can be invested in either improving the accuracy of each time point (by sequencing deeper), or by collecting more time points. This is an important trade-off, and it can only be achieved if the data is analysed jointly. To address these questions, we compared DICEseq versus DICE-sepa as we vary coverage levels and number of time points, by simulating reads as in the previous section (under GP assumption). In Figure 4.3A, we clearly see again that with the coverage increasing, all MAE decrease. In the joint model, the MAE largely decreases when more time points (i.e., 8) are used, especially for the case with low coverage.

These results highlight the importance of the analysis method for experimental

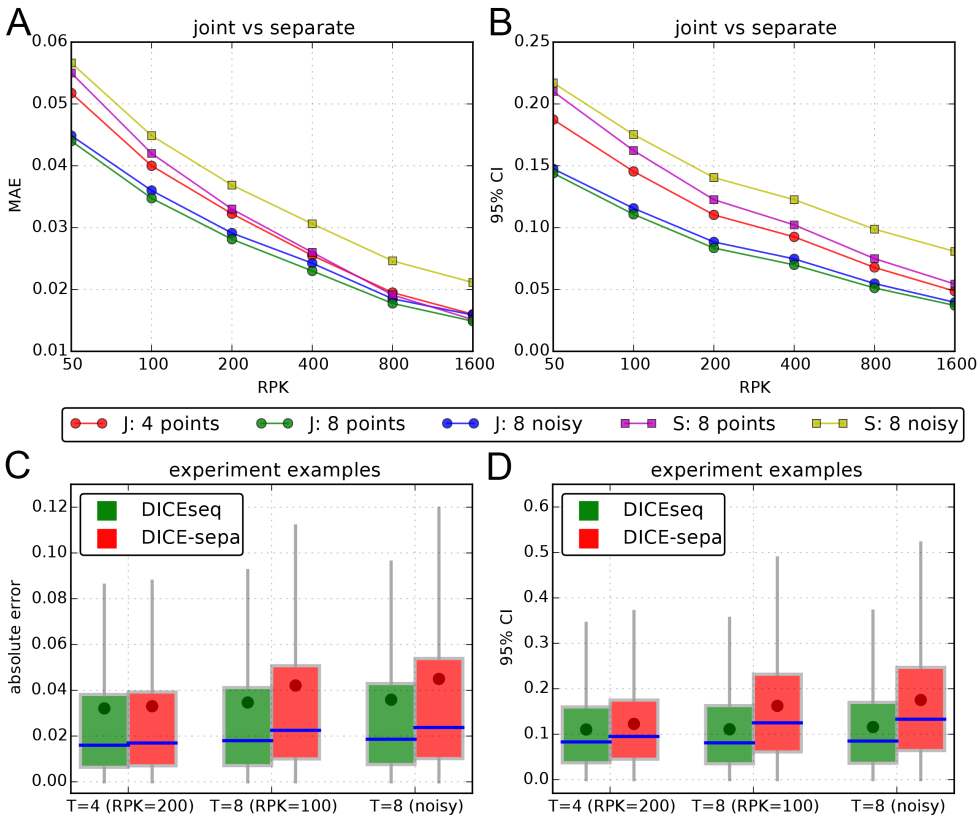

Figure 4.3: Comparison between experiment design on time points and coverages. **(a)** Mean absolute error between estimated isoform proportion and the truth for different experiments. "S" and "J" means DICEseq separate and joint mode, respectively, and the "noisy" means the RPK=25 at the 5th point for all **(a-d)**. All simulation here is based on a GP dynamics assumption. **(b)** 95% confidence interval of the estimates. **(c)** Boxplot of absolute error between estimated isoform proportion and the truth for three example experiments. The round dot is the mean. "T=4" and "T=8" indicate 4 and 8 time points. The "noisy" example was also conducted at RPK=100. **(d)** Boxplot of 95% confidence interval of the estimates.

design: while with the non-temporal model DICE-sepa increasing coverage is the only way to improve accuracy, methods that incorporate temporal information can benefit both from an increase in coverage and an increase in sampling frequency. Broadly speaking, we see that a doubling of the sampling frequency is roughly equivalent to a doubling of the sequencing depth, with the obvious advantage that a finer temporal information is provided. Figure 4.3C and 4.3D show an example of this trade-off: 4 time points and higher coverage of $RPK = 200$ give indistinguishable results for the

joint model to 8 time points and lower coverage of $RPK = 100$ (first two pairs in Figure 4.3C/D).

Another potential advantage of incorporating temporal information is to improve robustness of the estimation against noise/ low coverage at some time points. This aspect is particularly important as of course coverage level for a particular gene is largely determined by the gene's expression level, therefore genes with a large dynamic range of expressions during the time series will necessarily have some time points with low coverage. To simulate this situation, we generated time series with very low coverage ($RPK = 25$, termed "noisy") in the 5th time point. From the "noisy" case in Figure 4.3, we could see that the joint model dramatically reduces the variation compared to the separated model. Thus, incorporating time information in the joint model leads to a more robust estimation, facilitating isoform estimation for genes with dynamic expression levels and providing a possibility to combine low coverage with high coverage time points for time series libraries.

4.3.3 RNA splicing dynamics with 4tU-seq data

Recently, biotin labelling combined with RNA-seq has become an important tool to study the kinetics of RNA transcription and splicing with high temporal resolution (Windhager et al., 2012; Veloso et al., 2014; Fuchs et al., 2014). These experiments naturally produce RNA-seq data sets with high temporal resolution; furthermore, at very early time points, labelled RNA may be of low abundance, resulting in high uncertainty estimates. Here we use a recent data set with high temporal resolution to probe the suitability of DICEseq as an analysis tool for biotin labelled RNA-seq; the data was produced by our collaborators in the Beggs and Granneman labs at the Wellcome Trust Centre for Cell Biology in Edinburgh (Barrass et al., 2015). The data set consists of approximately 50M mapped reads; roughly 50% of genes have a coverage of RPK$<$120 in at least one time point.

To assess accuracy of our method, we compare the correlation between two replicates for 309 intron-containing genes at 1.5, 2.5 and 5.0 minutes. Figure 4.4A shows that IsoEM, Cufflinks and MISO all result in a good correlation between replicates, with Pearson's correlation coefficient varying between 0.83 and 0.85; DICEseq further improves with a Pearson's correlation coefficient of 0.896, outperforming by between 4 and 6 percentage points existing methods (all p values $< 10^{-5}$ under the Fisher r-to-z transform test (Diedenhofen and Musch, 2015)). The improvement is particularly

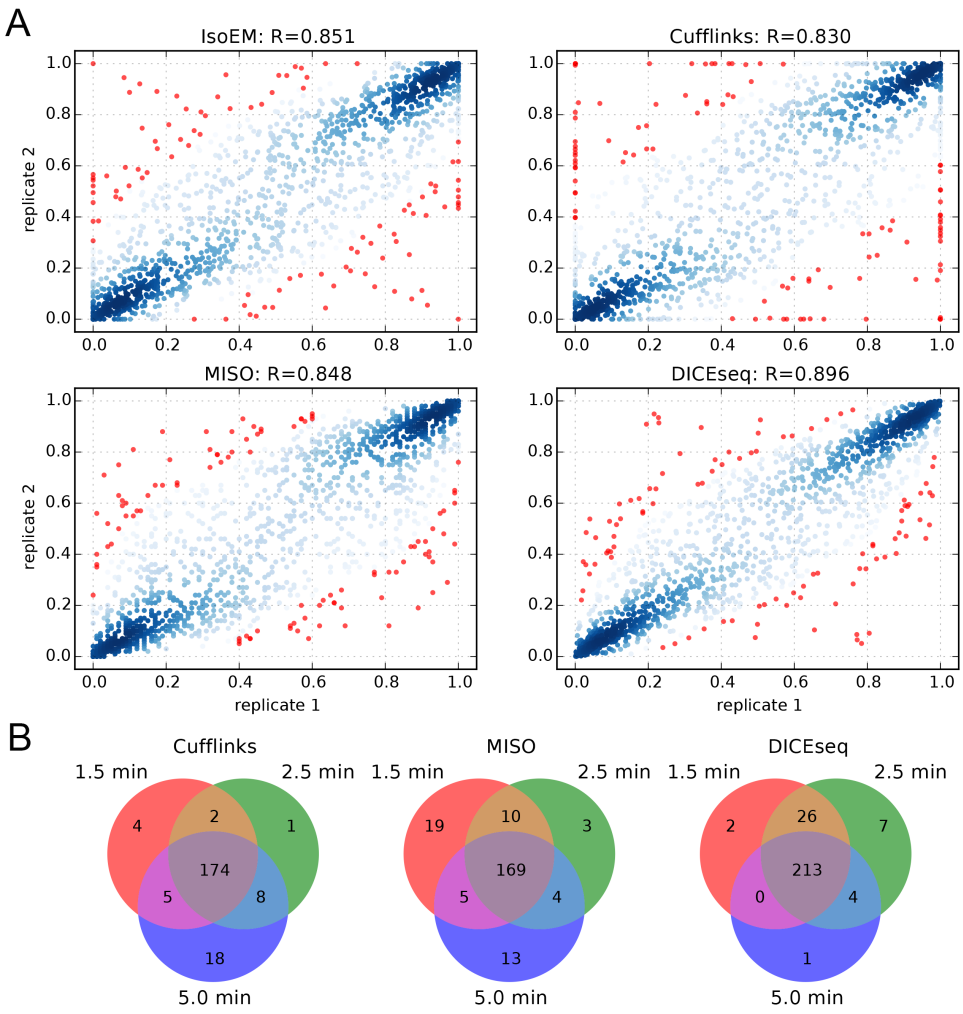

Figure 4.4: Analysis of time series 4tU-seq data. **(a)** The Pearson's correlation between two replicates. **(b)** The number of genes whose 95% confidence interval < 0.3.

marked if we consider the lowest expressed genes (see Table 4.1 on page 87): on the lower third of the expression range, DICEseq still obtains a Pearson correlation of 0.860, while the other methods achieve much lower correlations, ranging from 0.657 (Cufflinks) to 0.775 (IsoEM). This is remarkable since, as there are only three time points, the improvement obtained by taking temporal information into account could be expected to be limited. Notice in particular that, while IsoEM and particularly Cufflinks sometimes give deterministic estimates in one replicate but not on the other (red points on the boundaries of the square in Figure 4.4A), this problem does not occur with DICEseq, presumably due to the stronger regularisation enforced by the temporal correlations.

To further explore the usefulness of DICEseq, we consider the confidence intervals reported by the various methods. Isoform quantification methods are often used as an initial step in kinetic analyses of individual transcripts; in order to reduce false positives, genes with unreliable isoform estimates (as determined by thresholding on the confidence intervals) are discarded. When quantifying isoforms in isolation, some genes are then discarded just because one of the time points have lower expression level. Therefore, we computed the number of transcripts that pass a frequently used threshold (95%CI$<$0.3) for further analysis (Barrass et al., 2015). Figure 4.4B illustrates the results, showing that at all time points around 20% more genes are retained using a joint analysis, compared to methods that analyse data points in isolation.

To summarise, our results on a real yeast kinetic data set confirm that DICEseq yields significantly more reproducible and confident results than existing state-of-the-art methods, highlighting the value of incorporating temporal information in the analysis of time series real data. Note, this experiment is performed on yeast, whose genes only have two isoforms (with two exons), but our method is also applicable to genes with more isoforms.

4.3.4 Circadian dynamics of alternative splicing

As a second real-data example, we turned to a recent data set investigating circadian control of gene expression in mouse. Due to the day-night oscillations, many biological processes, including gene expression, show circadian rhythms. Recently, Zhang *et al* (Zhang et al., 2014) systematically studied circadian gene expression for 12 mouse tissues using high-temporal resolution microarrays and RNA-seq, and found that 43% protein coding genes oscillate in at least one of the 12 tissues; here we focus on data from liver. The RNA-seq here has a comparably low time resolution, as eight time points were collected over a period of 48 hours; we expect therefore that the advantages of incorporating time information may be less pronounced in this scenario. In total, there are between 67M and 105M uniquely mapped reads in each experiment; on average of 8 time points, 50% of genes have all isoforms with RPK$<$70; 75% of genes have all isoforms with RPK$<$400. Here, the RNA-seq reads are paired-end and the read length is 101 bp.

To assess the performance of the various methods, we used the microarray data set to validate the isoform estimates from RNA-seq. Unfortunately, only about one hundred microarray probes map to a unique annotated isoform (out of 30,534 annotated

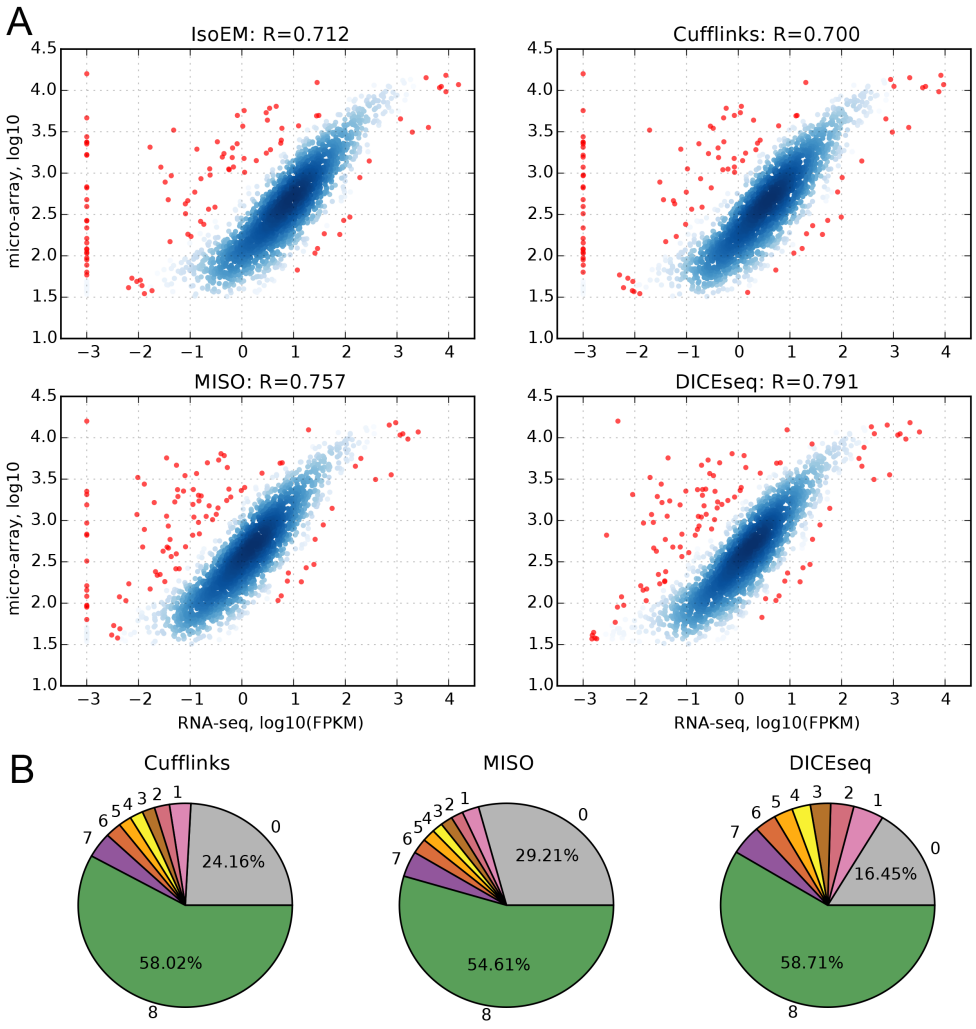

Figure 4.5: Analysis of circadian time series data. **(a)** The Pearson's correlation between the measurement of RNA-seq and microarray. **(b)** The proportion of genes whose 95% confidence interval $<$0.3 in a certain number of time points (index on the external side of the circle).

isoforms which map to at least one microarray probe); in other words, most microarray probes map to multiple isoforms within a gene. Thus, we used the estimated isoform proportions, together with the total numbers of reads mapped to each gene, to quantify the gene expression level (as FPKM), and then compared the resulting estimate from RNA-seq with the microarray measurement with Pearson's correlation coefficient. In Figure 4.5A, we see that the estimates obtained from RNA-seq using all methods have a high correlation with the direct measurements from the microarrays. Still, DICE-seq shows a significantly improved correlation from 0.757 to 0.791 (p value $< 10^{-5}$, Fisher r-to-z transform test); in particular, very low expressed isoforms (outliers in the

left end of the plot) show a much better quantification with DICEseq than with the other methods, probably due to the sharing of the temporal information, which is also evidenced by the middle third genes in Table 4.1.

We further measured 95% confidence intervals (CI) of all the 55,440 isoforms at the 8 time points, and quantified the fraction of isoform quantifications that pass the threshold 95%CI$<$0.3. In Figure 4.5B we see that all Bayesian methods (Cufflinks, MISO and DICEseq) give confident estimates for between 50 and 60 % of isoforms at all time points. Once again, DICEseq estimates are more confident, thanks to the value of temporal information sharing at low coverages, even though the advantage is more modest in this data set.

To summarise, our results in this low-frequency RNA-seq time series data set show that even in this case DICEseq produces quantitatively better estimates of isoform ratios, even though the value of sharing temporal information is more limited here due to the weaker correlations between time points.

Table 4.1: Robust performance of DICEseq in lower or medium coverage. "All" means all annotated genes; "1/3 low" and "1/3 mid" respectively mean lowest and medium 1/3 genes in coverage. The scores are Pearson's correlation coefficients between two replicates (4tU-seq) or two techniques (circadian).

	IsoEM	Cufflinks	MISO	DICEseq
4tU-seq, all	0.851	0.830	0.848	0.896
4tU-seq, 1/3 low	0.775	0.657	0.757	0.860
circadian, all	0.712	0.700	0.757	0.791
circadian, 1/3 mid	0.336	0.296	0.408	0.513

4.4 Discussion

In recent years, RNA-seq technology has been widely used for the analysis of dynamical biological processes, resulting in a remarkable increase of biological studies adopting RNA-seq within a time series experimental design. In this chapter, we presented DICEseq, the first method to jointly estimate the dynamics of the splicing isoform proportions from time series RNA-seq data. A comparison of DICEseq to a selection of popular state-of-the-art methods shows that DICEseq has excellent accuracy and good computational performance; in particular, DICEseq can effectively pool information

across multiple time points to improve isoform quantification at low coverages, giving more accurate and confident estimates in both simulated and real data sets. Our analysis also points to the importance of coverage versus temporal sampling trade-offs in designing dynamic RNA-seq experiments; while our analysis focussed on time series experiments, we expect similar considerations to hold for other structured designs, such as dose response experiments. In this light, the use of methods which can capture structural information, such as DICEseq, may lead to a rethink of biological experimental designs for a broad class of experiments. A similar trade-off between the number of time points and the number of replicates in high-throughput experiments has also been studied (Sefer et al., 2016), where the authors suggested that under a reasonable noise level, the temporal correlations allow dense sampling to determine the dynamic profile more accurately when compared to replicate sampling.

Methodologically, DICEseq builds on a fertile line of research using GPs to model transcriptional dynamics. GPs have been used to study the dynamical behaviour of gene expression in various contexts, from transcriptional regulation (Lawrence et al., 2006) to identifying the time intervals of differential expression with time series microarray data (Stegle et al., 2010), and to detect the differential dynamic profiles of gene expression during time course (Äijö et al., 2014). Very recently, GPs have been used to study the pseudo-time trajectory in single cell biology, including identifying gene-specific branching dynamics (Boukouvalas et al., 2017) and modelling transcriptional cell fates (Lönnberg et al., 2017). Here, we showed for the first time that GPs can be successfully used to model the temporal correlation for splicing at the real time scale. It is also very interesting to investigate the temporal structure of splicing at a pseudo-time scale in single cells.

Furthermore, the Gaussian process prior, which is based on a general regression, could be extended to more general dynamic splicing modelling, e.g., a first-order linear dynamic system for RNA splicing kinetics, and an oscillatory system for circadian changes. All of these could be incorporated in a straightforward way as parametric mean functions in a GP framework. However, it would also be of interest to explicitly model the noise correlations they induce. More generally, DICEseq could provide a flexible Bayesian framework for explaining RNA-seq data from other observations, and aid studies attempting to link splicing with other genetic and epigenetic factors, as we will see in Chapter 5.

Chapter 5

Splicing quantification in single-cell RNA-seq data

5.1 Introduction

All RNA-seq experiments we discussed in earlier chapters are based on libraries with thousands or millions of cells, which is also known as bulk RNA-seq. Therefore, the quantification of the transcriptome is actually an average value of a population of cells. However, the transcriptome can be very different from cell to cell, which is termed as *heterogeneity* in gene expression. Single-cell RNA sequencing (scRNA-seq) is a new technology, combining single-cell techniques and high-throughput sequencing, and allows for transcriptome-wide analyses of individual cells, investigating the stochasticity of transcription and its importance in cellular diversity. Ground-breaking applications of scRNA-seq include the ability to discover novel cell types (Grün et al., 2015), to study transcriptome stochasticity in response to external signals (Shalek et al., 2014), to enhance cancer research by dissecting tumour heterogeneity (Patel et al., 2014), to mention but a few.

Different from bulk RNA-seq, scRNA-seq requires an additional step of isolation and lysis of single cells, followed by the conversion of their RNA into cDNA, and the amplification of cDNA to generate high-throughput sequencing libraries (see the work flow in Fig 5.1). Due to the minute amounts of starting material, this process results in substantial technical variation (Ziegenhain et al., 2017). Those RNAs with low copies may be lost before amplification, for example when converting RNA into cDNA. Therefore, no matter how deep the sequencing, the missing RNAs from the beginning cannot be sequenced. These are called drop-out events, namely expressed

genes that cannot be detected by sequencing. The drop-out effects can give very misleading results in splicing analysis; if one or multiple transcript isoforms are lost in the beginning, then the estimated fractions for these isoforms will be zeros by conventional quantification methods. Actually, the fraction of drop-out is high; Fig 5.2 shows that over 30% genes expressed in bulk RNA-seq cannot be detected in scRNA-seq with SMARTer protocol. The median drop-out rate varies from 25% to 75% across different protocols (Ziegenhain et al., 2017).

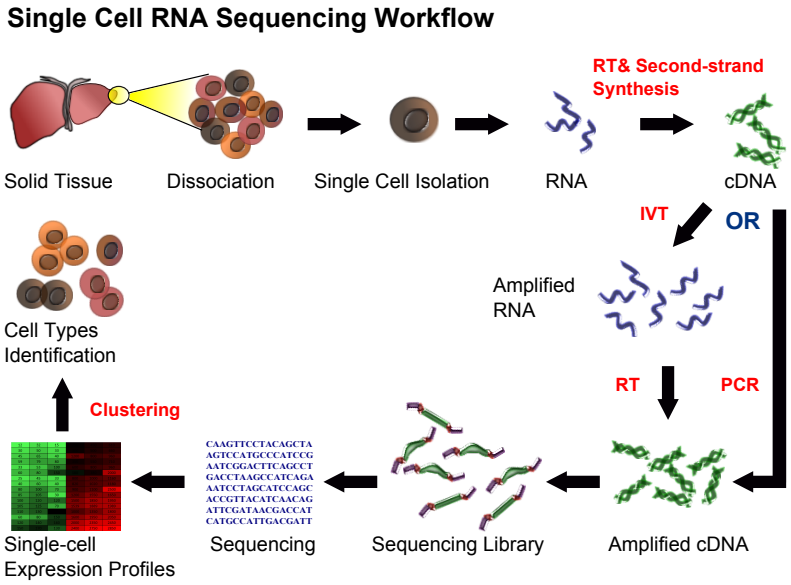

Figure 5.1: Single-cell RNA sequencing work flow (adapted from Wikipedia).

In addition, transcript coverage can be very uneven in many protocols, for example CEL-seq only captures the 3' end of the transcripts. Alternatively, SMART-seq and SMART-seq2 protocols can provide full length sequencing (i.e., reads covering full transcript rather than 3' end only), which offer chances for splicing analysis, in spite of sequence bias. Importantly, due to the large number of cells understudy, sequencing depth is usually very shallow, and can be as low as 0.1 million reads per cell for example in a population of 1,000 cells.

These properties of scRNA-seq data make analysis very difficult, and statistical challenges remain at multiple levels. At the lower level, it is very challenging to model the technical noises, especially drop-out events. A couple of methods have been developed for accounting for this, for example BASiCS (Vallejos et al., 2015) and MAST (Finak et al., 2015). At the higher level, even with perfect scRNA-seq data, it is still very hard to determine the cell states or cell developmental fates, as the dimension

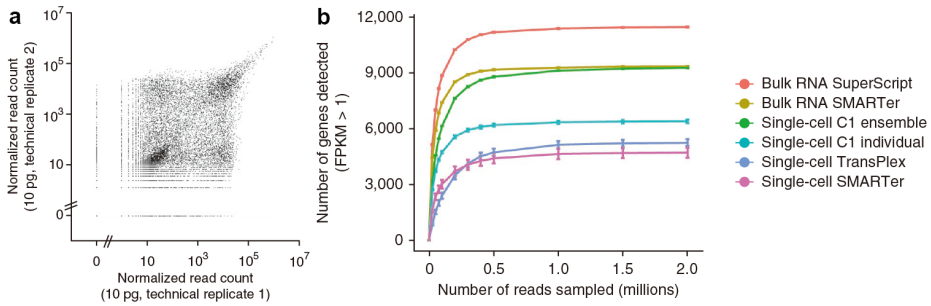

Figure 5.2: Technical variation in single-cell RNA-seq. (a) Mimic correlation between two technical replicates of scRNA-seq by dilution of total *A. thaliana* RNA to 10 pg, a similar amount of RNA as a single cell. This figure is adapted from (Brennecke et al., 2013). (b) The number of human genes detected in single-cell or bulk RNA-seq protocols. Note, Single-cell C1 ensemble means pooling sequencing reads from all single cells. This figure is adapted from (Wu et al., 2014).

of the transcriptome is very high. Quite a few methods have been developed to solve this problem, for example by ordering cells along one or more trajectories (referred to as pseudo-time trajectory) (Trapnell et al., 2014), by clustering cells into populations by a selected gene set, e.g., from a pathway (Fan et al., 2016), and by regressing out confounding factors, e.g., cell cycle (Buettner et al., 2015).

However, advances in scRNA-seq have been limited to the exploration of variability between single cells at the gene level, and we know very little about the global variability of RNA splicing between individual cells. Bulk RNA-seq splicing quantification algorithms cannot be easily adapted to the single cell case due to the high technical noise and extremely low coverages (Brennecke et al., 2013). This considerably limits the usefulness of scRNA-seq to investigate questions about RNA processing and splicing at the single cell level.

In earlier chapters, I have shown that splicing analysis has been revolutionised by the advent of (bulk) RNA-seq techniques, and probabilistic methods with mixture models largely improved splicing analysis. However, low coverage still brings a challenge even for probabilistic methods. Recent work has shown that improved predictions at lower coverage can be achieved by incorporating informative prior distributions within probabilistic splicing quantification algorithms, leveraging either aspects of the experimental design, such as time series (Huang and Sanguinetti, 2016), or auxiliary data sets such as measurements of Pol-II localisation (Liu et al., 2016). Such auxiliary data

are not normally available for scRNA-seq data. Nevertheless, recent studies have also demonstrated that splicing (in bulk cells) can be accurately predicted from sequence-derived features (Xiong et al., 2015). This suggests that overall patterns of read distribution may be associated with specific sequence words, so that one may be able to construct informative prior distributions that may be learned directly from data.

Here we introduce the Bayesian Regression for Isoform Estimation (BRIE) method, a statistical model that achieves extremely high sensitivity at low coverage by the use of informative priors learned directly from sequence features via a (latent) regression model. The regression model on sequence features couples the task of splicing quantification across different genes, allowing a statistical transfer of information from well-covered genes to lower-covered genes, achieving considerable robustness to noise in low coverage.

5.2 Methodology

5.2.1 BRIE model for isoform estimate

Here, we formally define the BRIE statistical model. We consider exon inclusion / exclusion as two different isoforms. We start by reviewing the mixture modelling framework for isoform quantification, introduced in MISO (Katz et al., 2010), and also presented in Chapter 3. The likelihood of isoform proportions ψ_k for observing N_k reads $R_{k,1:N_k}$ in splicing event k, can be defined as follows

$$P(R_{k,1:N_k}|\Psi_k) = \prod_{n=1}^{N_k} \sum_{I_{kn}=1}^{2} P(R_{kn}|I_{kn})P(I_{kn}|\psi_k) \tag{5.1}$$

where the latent variable I_{kn} denotes read identity in event k, i.e., the isoform from which read n came. For bulk RNA-seq methods like MISO (Katz et al., 2010) or DICEseq (Huang and Sanguinetti, 2016), the conditional distribution of the read identity $I_{kn}|\psi_k$ is assumed to be a Multinomial distribution, and the prior distribution over ψ_k is taken to be an uninformative uniform distribution (suitably adjusted to reflect the potentially different isoform lengths). The pre-computed term $P(R_{kn}|I_{kn})$ encodes the probability of observing a certain read coming from a specific isoform I_{kn}. Bulk methods then proceed usually by adopting a Markov-chain Monte Carlo strategy to sample from the posterior distribution of the ψ_k variables.

BRIE enhances the mixture model approach by combining it with a Bayesian regression module to automatically learn an informative prior distribution by considering

sequence features. First, we use a `logit` transformation of ψ_k, i..e, $y_k = \texttt{logit}(\psi_k)$. We then model the transformed exon inclusion ratio y_k as a linear function of a set of m covariates $X \in \mathbb{R}^m$ (here the covariates are the sequence features described in the following section): $y_k = W^\top X + \epsilon_k$, where W is a vector of weights shared by all samples and ϵ_k follows zero-mean Gaussian distribution. All exon skipping events are independently modelled with shared W parameters.

Here, we use a conjugate Gaussian prior for the weights, i.e., $W \sim \mathcal{N}(0, \Lambda^{-1})$, with a common choice of $\Lambda = \lambda \mathbf{I}$, for a positive scalar parameter λ. Thus, the graphical representation of the full model is shown in Fig 5.3, and the full posterior is as follows (omitting the cell index for simplicity),

$$P(W, \sigma, \mathbf{\Psi}|\mathbf{X}, \mathbf{R}) \propto P(W|\lambda) \prod_{k=1}^{K} \left\{ P(\Psi_k|X_k, W, \sigma) \prod_{n=1}^{N_k} \sum_{I_n^k=1}^{2} P(R_n^k|I_n^k) P(I_n^k|\Psi_k) \right\} \quad (5.2)$$

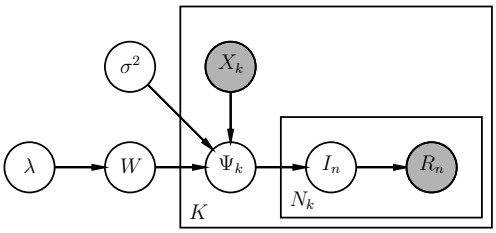

Figure 5.3: Graphical representation of the BRIE model, which combines Bayesian regression to learn an informative prior and a mixture model of RNA-seq reads (likelihood). The left part is the Bayesian regression that uses a set of features X_k to predict the ratios Ψ_k for K splicing events. The right part is a mixture model giving the likelihood of observed RNA-seq reads given the splicing ratio. For each splicing event, the observation of N_k reads $\mathbf{R} = (R_1, ..., R_N)$ are considered as N_k conditionally independent events, which depend on the originating isoform I_n, whose probability depends on the splicing ratio Ψ_k. Shaded nodes represent observed variables.

Furthermore, Figure 5.4 presents a schematic illustration of BRIE (see Methods for precise definitions and details of the estimation procedure). The bottom part of the figure represents the standard mixture model approach to isoform estimation, where reads are associated to a latent, multinomially distributed isoform identity variable. This module takes as input the scRNA-seq data (aligned reads) and forms the likelihood of our Bayesian model. The multinomial identity variables are then assigned an informative prior in the form of a regression model (top half of Figure 5.4), where the

prior probability of inclusion ratios is regressed against sequence-derived features (see next sections). Crucially, the regression parameters are shared across all genes and can be learned across multiple single cells, thus regularising the task and enabling robust predictions in the face of very low coverage. While the class of regression models we employ is different from the neural networks of (Xiong et al., 2015), they still provide a highly accurate supervised learning predictor of splicing on bulk RNA-seq data sets. Fig 5.5 shows that the Bayesian regression approach of BRIE can achieve a Pearson R in excess of 0.8 on test sets, validating our choice of model within BRIE.

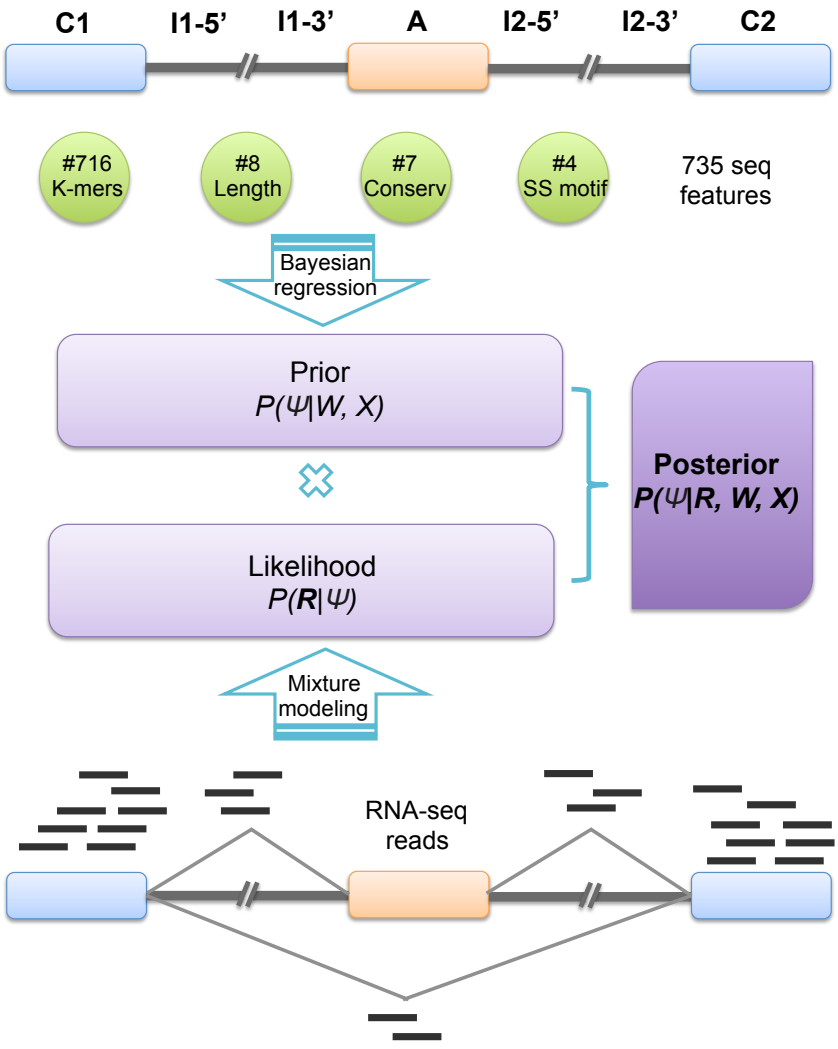

Figure 5.4: A cartoon of the BRIE method for isoform estimation. BRIE combines a likelihood computed from RNA-seq data (bottom part) and an informative prior distribution learned from 735 sequence-derived features (top).

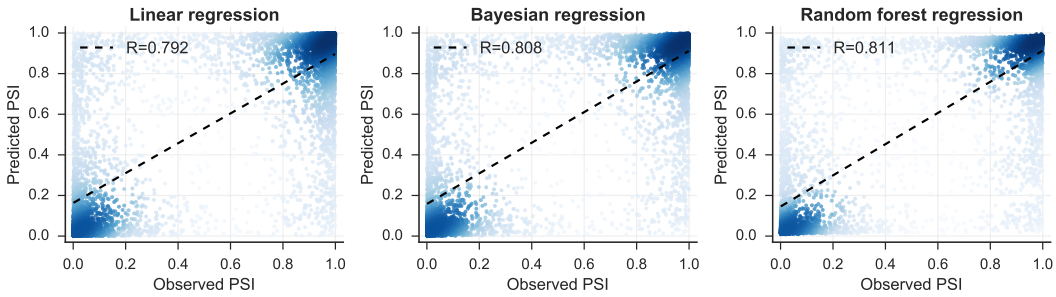

Figure 5.5: Sequence features are predictive for exon-skipping events. 735 sequence features are used to predict inclusion ratios of 6,922 out of 11,478 skipping exon (95%CI< 0.3) on human K562 cell line, with ordinary linear regression (left panel), Bayesian regression (middle panel), and random forest regression (right panel). The color density is according to the density of genes.

5.2.2 Inference in BRIE

As shown above, BRIE involves the whole set of exon-skipping events, thus there are thousands of parameters to infer jointly, which can lead to very high computational costs which are not easily distributed. Therefore, we introduce an approximate method to alternately learn ψ and W. Also, to alleviate computational burdens, there is an option to merge reads from all cells to learn parameters. For simplicity, we set λ empirically, using the value $\lambda = 0.1$ which gave the best predictive performance on tests on ENCODE data. Then, we collapse W and σ by taking their expected value in Bayesian regression given a set of ψ, i.e., $W = (\mathbf{X}^\top\mathbf{X} + \sigma^2\Lambda)^{-1}\mathbf{X}^\top\mathbf{Y}$ and $\sigma = \texttt{std}(\mathbf{Y} - W^\top\mathbf{X})$. At a single exon-skipping event level, we used an adaptive Metropolis-Hastings sampler to sample Ψ, where a univariate Gaussian distribution is used for proposal with adaptive variance, i.e., $\eta = 2.38 * \texttt{std}(y^{(1:m)})$. At this step, we could run short parallel MCMC chains on multiple events to alleviate computational costs, for example $h = 50$ steps if the total iteration is $n * h = 1000$. Pseudocode to sample from the (approximate) posterior distribution of Ψ is given in Algorithm 8. Also, this model supports fixed W and σ, which can be learned from other data sets, e.g. bulk RNA-seq; then the line 3 and 5 will be turned off in Algorithm 8. The convergence of the sampling is diagnosed by using the Geweke diagnostic Z score; in our experiments 1000 burn-in steps appeared to be sufficient in all cases.

BRIE then outputs an approximate posterior distribution on the ψ values as well as the learned regression weights. BRIE offers functionality to visualise both such posterior distributions as histograms (Fig 5.10c) and learned weights as heatmaps (Fig

C.9 in Appendix C for 19 sequence related features).

Algorithm 8: Approximation of $\mathbf{\Psi}, W, \sigma$

Data: $\mathbf{X}, \mathbf{R}, \Lambda$; optional: W and σ

Result: $\mathbf{\Psi}, W, \sigma$

initialization $\mathbf{Y}^{(0)} = \mathbf{0}; \sigma = 1.0; \eta = 1.0$

for $i \leftarrow 0$ ***to*** n **do**

  $W^{(i)} = (\mathbf{X}^\top \mathbf{X} + \sigma^2 \Lambda)^{-1} \mathbf{X}^\top \mathbf{Y}^{(i*h)}$; $\bar{\mathbf{Y}} = W^{(i)\top} \mathbf{X}$; $\sigma = \texttt{std}(\mathbf{Y}^{(i*h)} - \bar{\mathbf{Y}})$

  for $k \leftarrow 1$ ***to*** K **do**

    if $i * h > 10$ **then**

      $\eta = 2.38 * \texttt{std}(y_k^{(0:i*h)})$

    for $j \leftarrow i * h$ ***to*** $(i+1) * h$ **do**

      Sample: $\mu \sim U(0,1)$; $y_k^* \sim Q_y(y_k^* | y_k^{(j)}, \eta)$

      Calculate: $P(y_k^* | R) = \mathcal{N}(y_k^* | \bar{y_k}, \sigma) P(R | y_k^*)$

      if $\mu < \texttt{min}\Big\{ \dfrac{P(y_k^* | R) \times Q_y(y_k^{(j)} | y_k^*, \eta)}{P(y_k^{(j)} | R) \times Q_y(y_k^* | y_k^{(j)}, \eta)}, 1 \Big\}$ **then**

        $y_k^{(j+1)} \leftarrow y_k^*$; $\Psi_k^{(j+1)} \leftarrow \texttt{logistic}(y_k^*)$

      else

        $y_k^{(j+1)} \leftarrow y_k^{(j)}$; $\Psi_k^{(j+1)} \leftarrow \texttt{logistic}(y_k^{(j)})$

return $W^{(0:n)}$, $\mathbf{\Psi}^{(0:n*h)}$

In terms of computational efficiency, on a small server (48 CPUs and 64GB memory) and by using 20 CPUs, BRIE could finish a transcriptome-wide splicing quantification for a human cell (11,478 events) in 5 minutes, and for a mouse cell (4,549 events) in 2 minutes. This running time is a linear function of the number of cells (learning separate priors for different cells), and can be reduced by using more CPUs.

5.2.3 Detection of differential splicing using Bayes factors

The Bayes factor (Kass and Raftery, 1995) is a posterior odds in favour of a hypothesis relative to another, and is also able to detect whether splicing in two cells or conditions are different or not.

To detect differential splicing between two cells (or cell groups), A and B, $\delta = \Psi_A - \Psi_B$, we introduce a null hypothesis (H_0) as $\delta \approx 0$, and the alternative hypothesis (H_1) as $\delta \not\approx 0$. Here, D is the data used to sample the posterior of Ψ in two cells. Then, the Bayes factor in favour of the alternative hypothesis on observing data D is defined as follows,

$$\texttt{BF} = \frac{P(H_1|D)}{P(H_0|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_0)P(H_0)} \tag{5.3}$$

As usual, we assume that both hypotheses have the same prior, i.e., $P(H_1) = P(H_0)$, and we can clearly see that $P(D|H_0) = P(D|\delta \approx 0, H_1)$. Therefore, by taking the Savage-Dickey density ratio (Verdinelli and Wasserman, 1995), we could simplify the calculation of `BF` as follows,

$$\texttt{BF} = \frac{P(D|H_1)}{P(D|\delta \approx 0, H_1)} = \frac{P(\delta \approx 0|H_1)}{P(\delta \approx 0|D, H_1)} = \frac{P(-\epsilon < \delta < \epsilon|H_1)}{P(-\epsilon < \delta < \epsilon|D, H_1)} \tag{5.4}$$

where ϵ can be set as 0.05.

As BRIE samples Ψ_A and Ψ_B following their posteriors, the distribution of $P(\delta|D, H_1)$ is easily approximated by empirically re-sampling $\Psi_A - \Psi_B$. With a set of re-sampled $\delta_{1:M}$, we take the proportion of $|\delta_i| < \epsilon$ as the posterior probability $P(-\epsilon < \delta < \epsilon|D, H_1)$. Similarly, we could sample a set of $\hat{\Psi}_A$ and $\hat{\Psi}_B$ following their prior distributions, and use the same procedure to approximate the prior probability $P(-\epsilon < \delta < \epsilon|H_1)$. In the case of comparing two cell groups, one can multiply the individual likelihoods (with shared ψ values); this however is equivalent to pooling reads across different cells, and will lose the quantification of cell-to-cell heterogeneity.

5.2.4 Exon-skipping events and sequence features

Gene annotations were downloaded from GENCODE human release H22 and mouse release M6. 24,957 and 9,343 exon-skipping events were extracted from protein coding genes on human and mouse, respectively. In order to ensure high quality of the splicing events, we applied 6 constraints following two recent studies (Curado et al., 2015; Xiong et al., 2015) for filtering:

1) located on chromosome 1-22 (1-19 for mouse) and X
2) not overlapped by any other AS-exon
3) surrounding introns are no shorter than 100bp
4) length of alternative exon regions between 50 and 450bp
5) with a minimum distance of 500bp from TSS or TTS
6) surrounded by AG-GT, i.e., AG-AS.exon-GT

Consequently, 11,478 and 4,549 exon-skipping events from human and mouse respectively were finally used for this study.

Following Xiong *et al.* (Xiong et al., 2015), we extract predictive sequence features from the following 7 genomic regions for each exon-skipping event (see cartoon in Figure 5.4a): C1 (constitutive exon 1), I1-5ss (300nt downstream from the 5' splice site

of intron 1), I1-3ss (300nt upstream from the 3' splice site of intron1), A (alternative exon), I2-5ss (300nt downstream from the 5' splice site of intron 2), I2-3ss (300nt upstream from the 3' splice site of intron 2), C2 (constitutive exon 2).

From these 7 regions, four types of splicing regulatory features are defined. First, 8 length related features are included, i.e., log length of C1, A, C2, I1, I2, and the ratio of the log length of A/I1, A/I2 and I1/I2. Second, the motif strengths of the 4 splice sites, i.e., I1-5'ss, I1-3'ss, I2-5'ss and I2-3'ss, were calculated from mapping each sequence to its averaged position weight matrix. Here, we considered -4nt upstream to +6nt downstream around 5'ss (11nt in total), and from -16nt to 4nt for 3'ss. Third, we also include evolutionary conservation scores for each of the 7 genomic regions, which were calculated by phastCons (Pollard et al., 2010), and are available at the UCSC genome browser. We used the phastCons files in bigWig format with version hg38 for human and mm10 for mouse, where 99 and 59 vertebrate genomes were mapped to the human and mouse genome, respectively. Then the mean conservation scores for the above 7 regions were extracted by using `bigWigSummary` command-line utility. Lastly, 716 short sequences were extracted from the 7 regions, including 1-2mers for I1-5ss and I2-3ss (20 sequences each), and 1-3mers for C1, I1-3ss, I2-5ss and C2 (84 sequences each), and 1-4mers for A (340 sequences). In total, 735 splicing regulatory features were used to predict the exon inclusion ratio in Bayesian regression.

5.2.5 RNA-seq data and preprocessing

Bulk RNA-seq libraries for the K562 cell line were produced by the ENCODE project (ENCODE Project Consortium, 2012), downloaded from Gene Expression Omnibus (GEO: GSE26284); these were used to validate the prediction performance of the splicing regulatory features on bulk RNA-seq (Fig 5.5).

Two single cell RNA-seq data sets were used to validate the BRIE model. The first data set is from a benchmark study (Wu et al., 2014), consisting of 96 single cell RNA-seq libraries from the HCT116 cell line (GEO: GSE51254). These single-cell RNA-seq libraries were prepared with SMART-seq protocol, and have paired-end reads with read length of 125bp. By using a barcode, 48 cells were sequenced per lane, resulting in an average 2.2 million reads per cell. From the same study, two bulk RNA-seq libraries, each with 31.2M reads generated from 1 million HCT116 cells, were also used for comparison. Only reads mapping to alternatively skipped exons and their flanking regions (as described in the previous subsection) were considered.

In order to study differential splicing across different cell types, scRNA-seq data produced by the SMART-seq2 protocol from mouse embryo at embryonic day 6.5 and day 7.75 (Scialdone et al., 2016) were used. From each of the two groups, 20 individual cells were used, which can be accessed at Array Express (E-MTAB-4079).

All above RNA-seq reads were aligned to the relevant genome reference by HISAT 0.1.6-beta with known splicing junctions.

5.2.6 Simulation experiments design

There were three simulations conducted to assess BRIE's performance in quantifying isoform with low coverages, detecting differential splicing, and imputing splicing in drop-out cases. All synthetic reads were generated by the Spanki simulator (Sturgill et al., 2013), while we provide Python wraps to easily run the simulations, which is publicly available in the BRIE GitHub repository.

Simulation for very low coverages We assessed the robustness of BRIE at very low coverage on 11,478 human exon-skipping events. We assume that the ψ value follows a `logitNormal` distribution with mean $\mu = 0$ and $\sigma = 3$, i.e., $\texttt{logit}(\psi) \sim \mathcal{N}(0, 3.0)$, as presented in Figure C.1 in Appendix C, which is similar to that in the ENCODE K562 cell line. Then we set all splicing events at the same sequencing coverage, by fixing its RPK, i.e., reads per kilo-base in each experiment. Finally, five different coverage levels are used, including $RPK = 25$ (very low, but comparable to an average covered gene in a scRNA-seq experiment), $RPK = 50, RPK = 100, RPK = 200$ and $RPK = 400$.

For the purpose of generating a feature to learn an informative prior, we added Gaussian noise to the output ψ values from the Spanki simulator in its `logit` format, and ensured a Pearson's correlation coefficient of 0.8 between the feature and the truth, as shown in Fig C.1. This correlation is similar to that achieved by supervised learning in a human data set (see Fig 5.5). By contrast, five uniformly-distributed random features were used to learn a Null prior (i.e., random prior), which is named as BRIE.Null.

Simulation for imputation We mimicked the drop-out situation on 11,478 human exon-skipping events, and studied the imputation of BRIE in drop-out cases. We looked at one bulk RNA-seq library and 96 single-cell libraries of HCT116 cell lines (Wu et al., 2014), and only focus on the splicing events that are expressed in the bulk cells ($FPKM > 0$). We define the drop-out events as those splicing events that are

expressed in the bulk cells ($FPKM > 0$) but not in a given single cell ($FPKM = 0$). We further define the drop-out rate of a single cell as the fraction of drop-out events in this cell, and the drop-out probability of a skipping event as the fraction of its drop-out in 96 cells. Both distributions of the drop-out rates and the drop-out probabilities were shown in Fig C.2.

Given an expression profile (e.g., FPKM or TPM) Z from a bulk library and a profile of drop-out probability calculated from a group of single cells (e.g., the 96 cells here), we simulated the RPK for each isoform (or transcript) as follows. For each isoform k, we generate a binary variable I_k, i.e., either 0 or 1, following a binomial distribution with mean as its corresponding drop-out probability. Then each isoform expression level for the simulated single cell is $\alpha I_k Z_k$, where coefficient α is included to ensure a given number of total reads. If one wants a different overall drop-out rate, but keep the similarity of the drop-out probability profile, an intercept will be added to the drop-out probability in its `logit` space. In the simulation of drop-out, the 735 sequence features from real data are used to learn an informative prior. We take the mean of the learned prior as the imputed ψ for those drop-out events.

Simulation for differential splicing We tested the power of BRIE in detecting differential splicing events on 400 random mouse exon-skipping events with length of the exon triplet ranging from 300bp to 800bp. Eight categories of ψ from 0.1 to 0.9 except 0.5 were equally distributed to the 400 splicing events, and opposite ψ values were assigned to two conditions, e.g., ψ=0.1 in condition 1 and ψ=0.9 in condition 2. Then, the prior is set by the same procedure as the first simulation. Note, there are no drop-out considerations.

5.3 Results

The architecture of BRIE effectively enables it simultaneously to trade-off two tasks: in the absence of data (drop-out genes), the informative prior provides a way of imputing missing data, while for highly covered genes the likelihood term dominates, returning a mixture-model quantification. For intermediate levels of coverage, BRIE uses Bayes's theorem to trade off imputation and quantification. Now, let us look at the performance of BRIE in simulated data sets and real scRNA-seq experiments.

5.3.1 Benchmarking BRIE on simulated data

To assess the improvement in isoform quantification afforded by BRIE's informative prior, we simulated RNA-seq reads for 11,478 human exon-skipping events, and a correlated feature to learn prior (see details in above section for simulation experiments design, and Fig C.1). As we are interested in quantifying the effects of an informative prior, we compare BRIE with similar methods developed for bulk RNA-seq: MISO v0.5.3 (Katz et al., 2010), one of the first and still very widely used probabilistic methods, DICE-seq v0.2.6 (Huang and Sanguinetti, 2016), a modification of MISO using informative priors (for multiple time points). For completeness, we also compare with Kallisto (Bray et al., 2016), which was recently proposed as one of the most computationally efficient and robust quantification tools. To simulate the effect of the regression prior, we introduced an auxiliary variable with correlation 0.8 with the desired inclusion ratios (the correlation value was chosen to match the empirical performance of BRIE's regression prior on bulk RNA-seq data in Fig 5.5). We also consider the case when BRIE's auxiliary variable is uncorrelated with the inclusion ratio (denoted as BRIE.Null) as a control. Thanks to the informative prior, BRIE can also provide an imputation for drop-out transcripts (see below), which other methods cannot; in order to maintain the simulation fair, we did not include results on drop-out genes.

In the simulation, we set different coverage levels, RPK (reads per kilo-base) ranging from 25 to 400. Figure 5.6 clearly shows that the use of an informative prior can bring very substantial performance improvements at low coverage. At the lowest RPK level, BRIE achieves a gain of almost 20% in correlation between estimates and ground truth. Furthermore, this accuracy level is essentially maintained by BRIE at all coverage values. Interestingly, BRIE.Null can still achieve comparable accuracy to other existing methods at all coverage values; therefore, even in cases where an informative prior could not be effectively learned, BRIE's results would not be worse than using a state-of-the-art bulk RNA-seq method.

5.3.2 Imputation of drop-out in simulation

The informative prior learned by BRIE can also be used to impute isoform usage when there is a drop-out, i.e., no reads sequenced for an expressed isoform. In sc-RNAseq experiments, drop-out widely occurs (Brennecke et al., 2013), though it is sometimes hard to exactly detect, except for spike-in RNAs. Here, we could coarsely define its upper bound, by counting exon-skipping events expressed in bulk cells but not in a

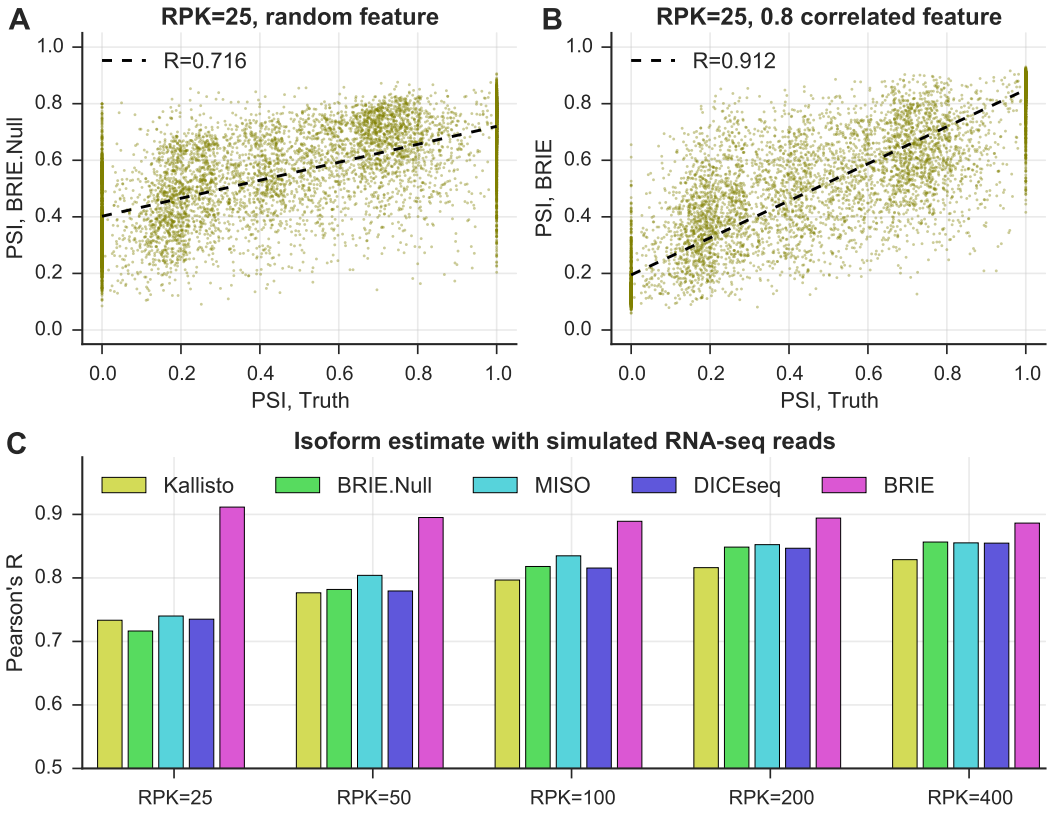

Figure 5.6: BRIE improves isoform estimates by using an informative prior on simulated data. (A-B) At very low coverage RPK=25, the scatter plot between the estimate of exon inclusion ration by BRIE and the simulation truth. (A) BRIE.Null uses five random uniform-distributed features to learn prior. (B) BRIE uses one correlated feature with Pearson's R=0.8 to the truth to learn informative prior. (C) Pearson's R between truth and estimate by BRIE, BRIE.Null and 3 other methods in different coverages.

given single cell. In Fig 5.7, we see that after removing drop-out events, the correlation of expression level between a single cell and bulk cells is dramatically higher for these splicing events.

As BRIE can transfer information from highly expressed genes to lowly expressed genes across multiple cells, we investigated the performance of BRIE in imputing the isoform usage if drop-out happens. Therefore, the expression profile from a bulk RNA-seq library and the drop-out probability profile estimated from 96 HCT116 human cell scRNA-seq libraries (Wu et al., 2014) were used to perform the simulation (see Fig C.2 and simulation details in Methods). Fig 5.8 shows that BRIE can produce a good imputation of the isoform usage simply by taking the mean of the informative prior

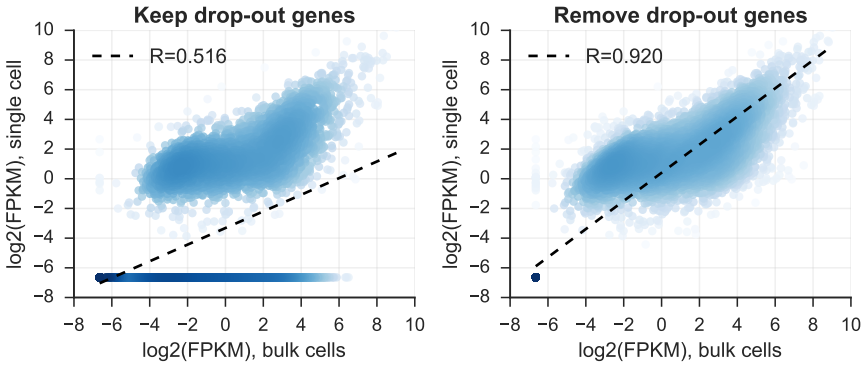

Figure 5.7: Effects of drop-out in single-cell RNA-seq. Scatter plot of log2(FPKM) of exon-triplets in human HCT116 cell line between bulk cells and a single cell. The drop-out genes are defined as thoese with $FPKM > 0$ in bulk cells and $FPKM = 0$ in the single cell. Left panel: keep the drop-out genes; Right panel: remove drop-out genes. The Pearson's correlation coefficient increases from 0.516 to 0.920 after removing drop-out genes.

learned from sequence features of the expressed genes (Pearson's R: 0.6∼0.7).

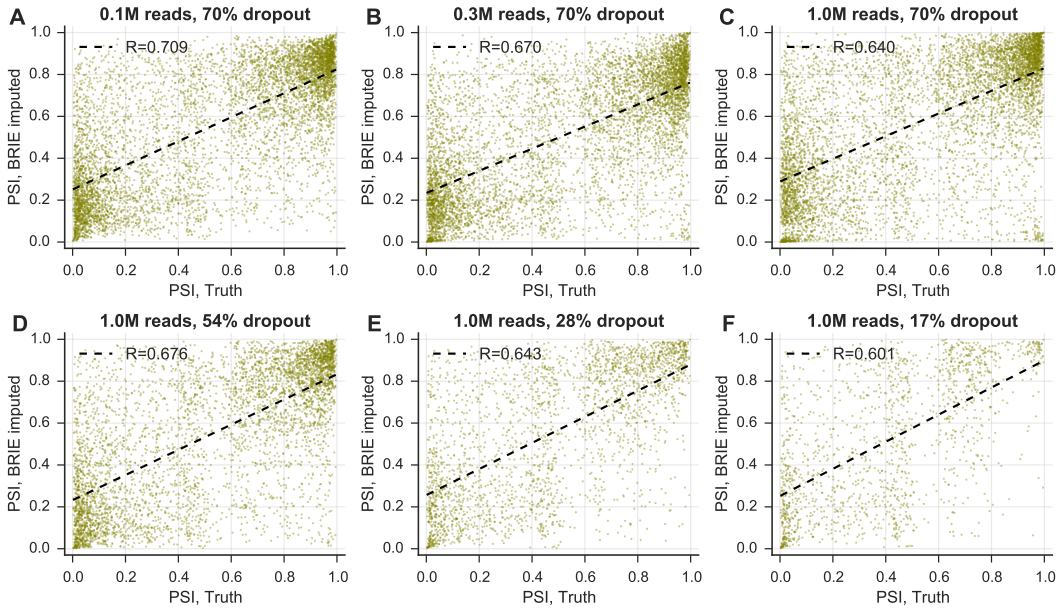

Figure 5.8: Imputation of exon inclusion ratio for drop-out genes with simulated data. Following the expression profile of bulk cells and the drop-out probability of 96 single cells, the RNA-seq reads library is generated with a given number of total reads and an overall drop-out rate. The true inclusion ratio ψ is the value in bulk RNA-seq and also the input for the simulator. The features are the same 735 genetic features for real data set.

5.3.3 Robust splicing estimates on real data

To assess BRIE's performance on real scRNA-seq data, we used 96 scRNA-seq libraries from individual HCT116 human cells from the benchmark scRNA-seq study of Wu et al (Wu et al., 2014) (see details in above section on data processing). Importantly, a bulk RNA-seq data set in the same conditions was also obtained from one million cells. To better explore performance on real data, we expand the set of competing methods to include Cufflinks v2.2.1 (Trapnell et al., 2010), RSEM v1.3.0 and the recently proposed single-cell quantification method Census (in Monocle v2.2.0) based on Cufflinks FPKM (Qiu et al., 2017). Figure 5.9 shows the results: BRIE clearly outperforms all other methods by a large margin, both in terms of correlation between estimates from different single cells (Fig 5.9f), and in terms of correlations between estimates from individual single-cells and bulk (Fig 5.9c). Example scatter plots for both comparisons are given in Fig 5.9e/b, clearly showing very consistent predictions. Notably, the performance of other methods was strongly degraded by the inability to handle the large drop-out rates (see Fig 5.9a/d for DICE-seq, where many estimates of splicing are centred around the uninformative prior value of 0.5). The high correlation between bulk and scRNA-seq predictions is particularly remarkable, as the analysis of the two data sets is not done with a shared prior. Similarly high correlations were found between splicing estimates obtained by BRIE in single cells and estimates from bulk RNA-seq obtained by other methods (Fig C.3).

These statistical advantages are reflected in a more effective and confident quantification: considering genes with quantified uncertainty smaller than 0.3 (a threshold adopted e.g. in (Barrass et al., 2015) to select for downstream analysis), Figure C.4 shows that BRIE retained 10.9% out of 11,478 genes on average from each single cell, as compared with 3.1% and 5.6% for MISO and DICEseq, respectively.

5.3.4 Differential splicing analyses with high sensitivity

BRIE can also be used for differential splicing detection across different data sets. To do so, we compute the evidence ratio (Bayes factor, BF) between a model where the two data sets are treated as replicates (null hypothesis) and an alternative model where the two data sets are treated as separate. We use the Savage-Dickey density-ratio approach and relax it in order to obtain more robust estimates (see above section on Bayes factor). Notice that there are several ways in which differential comparisons could be performed: we could compare groups of cells or individual cells, and we

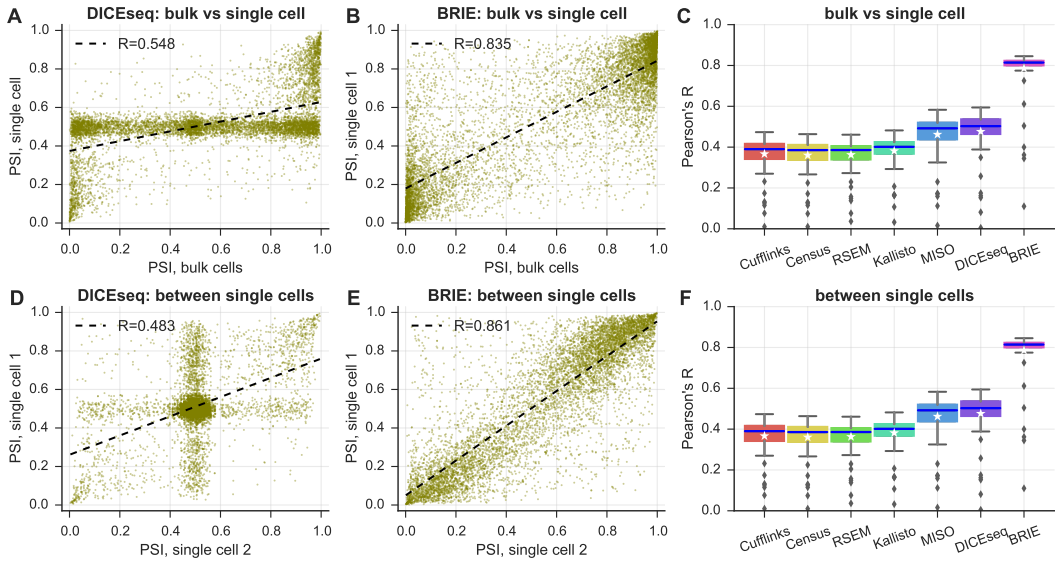

Figure 5.9: BRIE improves splicing estimates by using sequence features. (A-C) Pearson's correlation between between bulk and single cells on exon inclusion ratio ψ in HCT116 cells. Scatter plot of ψ estimates by DICEseq (A), or estimated by BRIE (B). Box-plot for all methods (C) in 96 cells. (D-F) Pearson's correlation between single cell pairs. Scatter plot of ψ estimates by DICEseq (D), or estimated by BRIE (E). Box-plot for all methods (F) in 4,608 cell pairs.

could share the learning of the prior across conditions, or learn separately. All of these options are supported in the BRIE software.

To benchmark the effectiveness of this strategy, we again turned to a simulation study, investigating the ability of BRIE to detect differential splicing as we vary coverage and the extent of the differential effect (see above section for simulation design). This benchmarking is important, as the informative prior might be expected to impede differential quantification. In practice, we see that, for substantial effect sizes ($\Delta\psi = 0.6$), we can detect a substantial fraction of differentially spliced genes already at RPK 50, further improving when the effect size is 0.8 (Fig C.5a in Appendix C). We also use the simulation study to explore the effect of different library size on our differential comparisons. We do this by fixing one of the comparison cells to an RPK level. The results shown in Fig C.5 b-c demonstrate that BRIE is robust to normalisation issues; this is not surprising, since relative quantification algorithms normally combine normalisation with estimation (see (Qiu et al., 2017) for a discussion of this topic in the scRNA-seq context).

We then moved to investigate the effectiveness of BRIE to detect differential splic-

ing in real cells. To estimate a background level of differential splicing between identical cells, we considered again the 96 single cell HCT116 libraries from Wu et al (Wu et al., 2014), and compared all possible pairs of cells. Figure 5.10a shows the fraction of genes called as differentially spliced at different BF thresholds in this control experiment; as we can see, this number is always very small, and around 1% at the normally recommended threshold of BF=10. This level of background calling could be partly attributed to intrinsic stochasticity or to residual physiological variability that was not controlled for in the experiment, such as cell cycle phase. As an additional comparison, we considered two bulk RNA-seq methods for differential splicing, MISO and the recently proposed rMATS (Shen et al., 2014). Both methods could only call a negligible number of events, far fewer than the expected number of false positives, confirming that bulk methods are not suitable for scRNA-seq splicing analysis.

We then considered a mouse early development scRNA-seq data set (Scialdone et al., 2016), and compared the single cell transcriptomic profiles from cells from mouse embryos at 6.5 and 7.75 days. We compared both the profiles of individual cells at the same and different time points; the results are summarised in Figure 5.10b. Comparing individual cells at 6.5 days yielded approximately 1% of events called as significantly differential (BF$\geq$ 10) at 6.5 days. Comparing this result with our investigation of HCT116 cells suggests that murine cells at 6.5 days are still similar to a homogeneous population, from the splicing point of view. The percentage nearly doubled at 7.75 days, suggesting that differential splicing becomes more widespread at this later stage of differentiation. A similar fraction of exon skipping events were differentially called between cells at 7.75 days and cells at 6.5 days. To define a group of differentiation-associated skipping events, we considered events that we called as differential in at least 10% of 7.75 vs 6.5 comparisons. The resulting 159 events were highly enriched for organelle and intracellular part GO terms ($p < 0.01$, protein-coding genes as background) (see Supplementary Table S1 and S2 in original paper (Huang and Sanguinetti, 2017)). Figure 5.10c shows the example of DNMT3B, a regulator of DNA methylation maintenance, which is known to undergo functionally relevant alternative splicing (Duymich et al., 2016). DNMT3B exhibited differential splicing between 7.75 days and 6.5 days in 153 out of 400 comparisons between individual single cells, clearly highlighting the strong differential inclusion effect. Four more example events, all of which have shown differential splicing in more than 100 pairs of comparisons, are presented in Figure C.6 in Appendix C.

We also directly compared the two groups of cells within a single test (7.75 vs

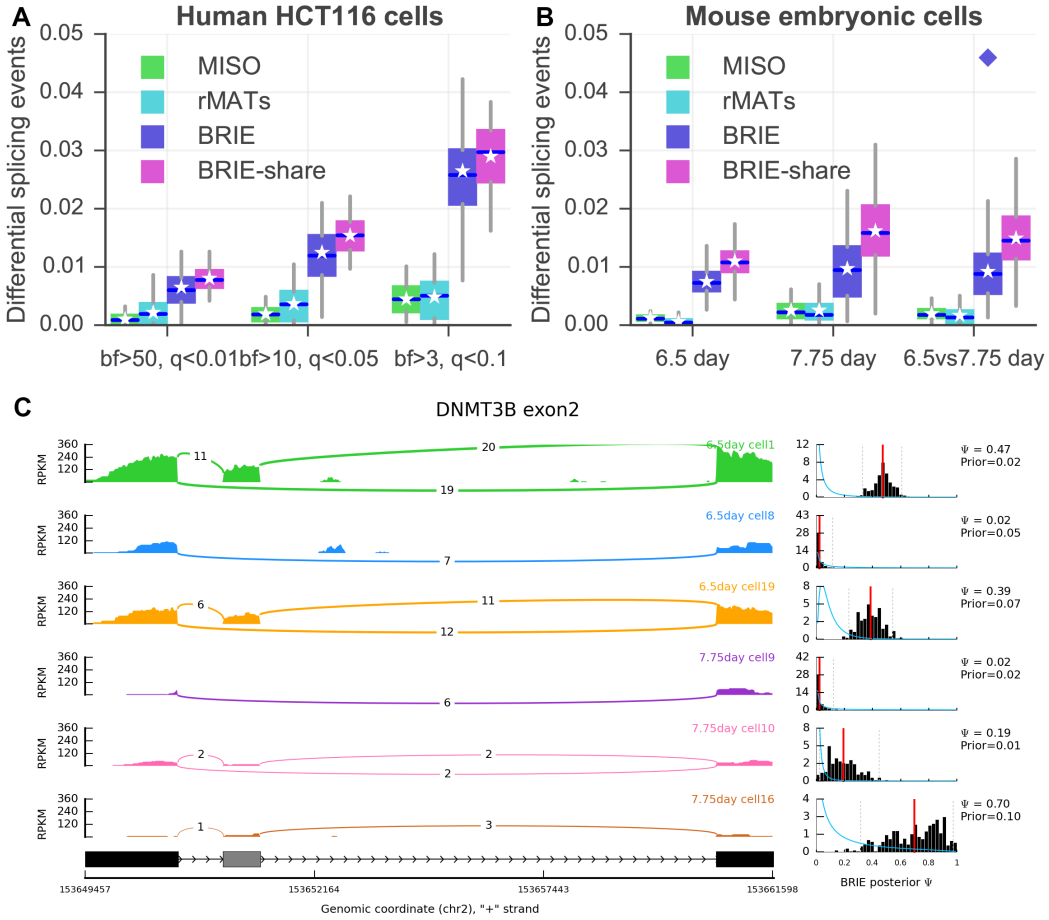

Figure 5.10: Detection of differential splicing between cells. (A) Percentage of differential splicing events between human HCT116 cells, detected by MISO, rMATS, BRIE and its mode with shared weights (i.e., BRIE.share) with different thresholds. MISO and BRIE use Bayes factor (bf) and rMATS uses false discovery rate (q value). (B) Percentage of differential splicing events between mouse early embryonic cells at 6.5 days or 7.75 days. The threshold is $bf > 10$ for MISO and BRIE, and $q < 0.05$ for rMATS. Diamond indicates pooling reads of 20 cells in each group. (C) An example exon-skipping event in DNMT3B in 3 mouse cells at 6.5 days and 3 cells at 7.75 days. The left panel is sashimi plot of the reads density and the number of junction reads. The right panel is the prior distribution in blue curve and a histogram of the posterior distribution in black, both learned by BRIE. For the histogram, the red line is the mean and the dash lines are the 95% confidence interval.

6.5); this can be easily achieved by assuming a shared splicing ratio ψ across all cells in a condition. Mathematically, this is equivalent to multiplying the likelihood terms

associated to each cell, in practice pooling the reads from different cells. While this achieves higher power (see the diamond dot in Fig 5.10b), it loses the considerable amount of cell-to-cell heterogeneity highlighted by the single-cell analysis. It would be interesting to explore a more refined way of partial pooling within the hierarchical model (Glaus et al., 2012), or to combine BRIE with scRNA-seq clustering approaches which can identify more homogeneous groups of cells (Grün et al., 2015).

5.4 Discussion

Our results demonstrate that BRIE can provide a reliable and reproducible method to quantify splicing levels within single cells. Alternative splicing is a major mechanism of regulation of the transcriptome, and splicing analyses within bulk studies have revealed important associations of splicing with disease. Therefore, the ability to quantify alternative splicing in individual cells would considerably expand the relevance of scRNA-seq technology to investigate variations in RNA processing, and its relevance to diseases, for example leukemia and multiple sclerosis. In these cell type specific diseases, splicing aberrance are found to be associated with disease occurrence. We believe the usage of a data-driven informative prior is essential for this task: directly using bulk RNA-seq methods on scRNA-seq is not a viable route due to the limitations of the technology, an observation that was made earlier (Grün and van Oudenaarden, 2015) that our results confirm. Recent work (Welch et al., 2016) has addressed the issue of *detection* of alternative splicing across a population of single cells, but as far as we are aware BRIE is the first method to be able to *quantify* splicing in individual single cells, and to detect differential splicing between individual cells from scRNA-seq data. We notice that, since BRIE focusses on estimating splicing ratios, it is relatively immune to normalisation issues, since it is essentially a relative quantification method (see (Qiu et al., 2017) for a compelling demonstration of this property of relative quantification methods).

BRIE provides a flexible framework for modelling and, while sequence features are particularly appealing due to their ease of usage and availability, additional side information, such as DNA methylation and chromatin accessibility, could easily be incorporated. In other words, all available features that are predictive of splicing can be included into this model. Importantly, BRIE is not specific to single-cell RNA-seq technology, and can be of use in any situation where standard quantification is hampered by low coverage.

BRIE's use of an informative prior enables a smooth trade-off between imputation (at extremely low coverages) and quantification. While this can be a highly effective strategy, it comes at the cost of biasing results at low coverage. In particular, when used with an informative prior learned across several cells, this may lead to underestimating splicing heterogeneity at low coverage. BRIE's probabilistic formulation however brings considerable advantages; in particular, BRIE can be easily combined with other probabilistic modelling strategies aimed at removing confounders such as cell-cycle stage (Buettner et al., 2015), or at estimating pseudo-time (Campbell and Yau, 2016).

BRIE cannot be deployed on all scRNA-seq protocols, as it assumes that sequenced reads can be distributed along whole transcripts. Naturally, protocols such as CEL-seq or STRT-seq that bias reads towards the ends of the transcript cannot provide information about exon skipping events that may be very far from the ends of a transcript. We believe that the availability of splicing quantification approaches such as BRIE can therefore be an important consideration in experimental design, particularly at a time when single-cell omic technologies are about to start being more routinely employed.

Chapter 6

Summary and future research

Alternative splicing of eukaryotic transcripts is a regulated mechanism that enables a single gene to produce multiple splicing isoforms, thus generating vast protein diversity from a limited number of genes. Regulation of specific splicing events involves many important biological processes, and aberrant splicing is often found to be associated with serious diseases. The advent of RNA-seq technologies has revolutionized the study of mRNA splicing, and provided a powerful stimulus for the development of computational biology methods (Katz et al., 2010; Bray et al., 2016; Love et al., 2016). Recent years have seen a more wide-spread use of RNA-seq technology for the analysis of dynamical biological processes, and stochasticity of transcriptome in cell populations, resulting in adding another dimension of RNA-seq data for analysis of these biological processes. The increase of RNA libraries often causes reduced coverage, in both time-series and single-cell RNA-seq experiments, particularly the latter. The very low coverage in RNA-seq data brings a tough challenge in splicing isoform quantification, as a sufficient data size is required to obtain a confident estimate of splicing isoforms. In order to ameliorate this challenge caused by low coverages, this thesis contributes to the development of novel structured Bayesian methods for splicing analysis with time-series and single-cell RNA-seq data.

In Chapter 3, I introduced a mixture model for isoform quantification with RNA-seq data, which is widely used in many existing methods. Though the framework is very similar to many methods, the processing of the data, the modelling of the sequencing and positional bias, and the inference strategies are different. I briefly presented a widely used strategy of modelling biases in RNA-seq data, and also provided a comparison of different inference algorithms including EM algorithm, Gibbs sampler, and Metropolis-Hastings sampler. I showed that these algorithms give similar accuracy in

splicing estimate, but a different performance in speed. Specifically, EM algorithm is much more efficient either of the MCMC samplers. However, EM algorithm can only obtain a point estimate of splicing isoforms, whereas both MCMC samplers can give the whole distribution. Though my implementation of Gibbs sampler was found to be slower than MH sampler, both algorithms theoretically have the same computational complexity. Furthermore, we showed that the probabilistic method has much higher accuracy and less bias compared with direct counting methods in analysing two-isoform splicing. However, even with the probabilistic method, isoform quantification is still very challenging when the number of isoforms is large and particularly when the coverages are very low. Therefore, additional information needs to be integrated for improving splicing analysis.

In Chapter 4, I presented DICEseq (Huang and Sanguinetti, 2016), a novel isoform quantification method tailored to correlated RNA-seq experiments. DICEseq explicitly models the temporal correlations between time-series RNA-seq experiments to aid the quantification of isoforms across experiments. Gaussian process prior has been used in this Bayesian method to model the temporal structure of splicing dynamics. Simulated and real data sets have shown that DICEseq significantly improves the accuracy and confidence in isoform quantification compared with the state-of-the-art methods, by taking into account the additional information encoded in the temporal correlation. Furthermore, by jointly analysing the time-series data, DICEseq offers a new balance in the trade-off between sequencing coverages and temporal resolutions, especially emphasising the possibility of probing more time points (and less coverages) with the similar budget and achieving the similar quantification accuracy at each time point.

In Chapter 5, I described BRIE (Huang and Sanguinetti, 2017), the first method to effectively quantify splicing in single cells and to detect differential splicing between individual cells from scRNA-seq data. BRIE combines the imputation and quantification in a Bayesian way, where the sequence features are used to automatically learn an informative prior distribution for each splicing event. The results on multiple simulated and real scRNA-seq data sets demonstrate that BRIE can provide a reliable and reproducible method to quantify splicing levels within single cells. Particularly, BRIE remarkably improves in splicing analysis for drop-out events and at very low coverages, which are the greatest challenges in analysing scRNA-seq data. In addition, the BRIE package provides an effective tool using Bayes factor for sensitively detecting differential splicing between cells. When applying this method to real scRNA-seq data, we identified a set of splicing events with high variability across cells, and found

that they are enriched in interesting biological processes with Gene Ontology analysis. Together, these results show that BRIE broadens the analysis of cellular heterogeneity from gene expression level to RNA processing level, and therefore brings new chances for deciphering more biological insights.

There are many open directions for future research. Low coverage is a fundamental challenge in isoform quantification, which is a common feature of experiments involving a large number of RNA libraries as discussed above. This thesis has attempted to mitigate this problem by exploiting additional information from either temporal correlation in time-series or shared regulation pattern from sequence features in single cells. However, it is still hard to determine how deep the sequencing coverage is needed to provide statistically confident analysis in splicing. This question is particularly hard, due to a lack of a good measurement of complexity for isoform quantification, even though we could intuitively think that genes with 20 isoforms are more complex in terms of quantification than genes with 2 isoforms.

One idea of measuring the complexity of isoform quantification could be calculating the Cramér Rao lower bound (CR bound) of the variance of Ψ estimation by taking its Fisher Information matrix, derived from Eq(3.1). Assume we have an unbiased estimate $\hat{\Psi}$ of Ψ, i.e., $\mathbb{E}(\hat{\Psi}) = \Psi^*$. Then the CR bound tells us that $\texttt{Var}(\hat{\psi}_k) \geq [M^{-1}(\Psi^*)]_{kk}$, where M is the Fisher information matrix, described as follows,

$$[M(\Psi^*)]_{k1,k2} = -\mathbb{E}\left[\frac{\partial \log P(\boldsymbol{R}|\Psi)}{\partial\psi_{k1}\partial\psi_{k2}}\bigg|_{\Psi=\Psi^*}\right] \tag{6.1}$$

where the reads set $\boldsymbol{R}$ contains only one read (following a specific distribution), therefore it is the variance per read.

In Fig 6.1, the CR bound shows a very good prediction of the estimation variance compared to a few popular methods. Therefore, it could be used to define the complexity of isoform quantification for each gene. Applying this measure may help better design the RNA-seq experiments, especially single-cell RNA-seq experiments, as sequencing depth is very hard to determine, when considering the number of cells to study. Note, that this is an ongoing work with Prof. Edo Airoldi's group in Harvard University.

Another open question is how we should use the correlation between time-series experiments, to improve the robustness in detecting differential splicing. Biological (or technical) variance is a confounder when detecting differential splicing, and may cause false positives. Therefore multiple biological replicates are important for modelling the biological variance in detecting differential splicing (and gene expression)

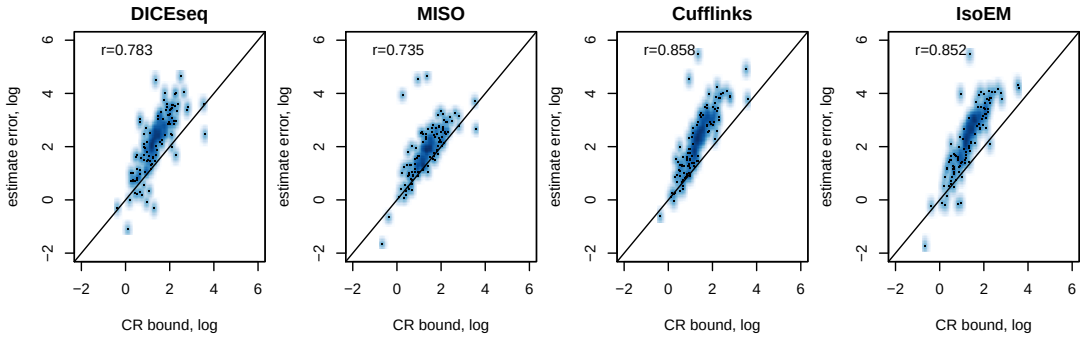

Figure 6.1: Complexity by CR bound vs estimation error by DICEseq, MISO, Cufflinks and IsoEM. Paired-end RNA-seq reads with read length of 50 and mean fragment length of 100 were simulated by Spanki. 200 random human genes with 2 to 11 isoforms were used. Here, both CR bound calculation and reads simulation are under the assumption of equal fractions of isoforms. Pearson correlation (r) is shown on each plot.

(Shen et al., 2014; Liu et al., 2013). However, in time-series experiments, biological replicates are often limited, as investigating more time points rather than replicates is more informative for analysing dynamics (Sefer et al., 2016). Therefore, how to use the temporal structure to estimate or improve the estimation of biological variance is important for robust analysis yet is very challenging. Topa and Honkela (Topa and Honkela, 2016) made an attempt to detect the time dependency of splicing with a Gaussian process model in an additional step following MCMC samples of isoform estimate from BitSeq (Glaus et al., 2012). However, it did not provide a way to model the biological variance from the temporal structure. One potential idea could be developing a Bayesian hierarchical model assuming the same biological variance at all time points. Though this assumption of sharing biological variance is probably not true, it is still believed to be more robust than without consideration of biological variance in detecting differential splicing and its dynamic profiles.

Furthermore, there are more open questions in splicing analysis at the single cell level. In Chapter 5, I presented our method BRIE to accurately quantify splicing isoform at single cell level. However, the biological insights largely remain undiscovered. The first question is whether splicing can work as markers to classify subtypes of cells. If so, to what degree can it contribute to the clustering of cells, especially when integrating with expression at gene level. One very recent study from the Yeo lab (Song et al., 2017) uses very deep scRNA-seq to study stochasticity in splicing in single cells. In Figure 1(F-G) in their paper, the independent component analysis shows that only

the splicing values rather than gene expression for the non-DE gene can distinguish three cell types. This result suggests that splicing analysis has the ability to provide a unique layer to identify cell types. Based on this promising analysis in splicing, more studies on published scRNA-seq data or in future experiments have good chances to decipher the regulation of splicing on cell types. However, in more general scRNA-seq experiments, sequencing coverages are $0.1\sim1$ million reads per cell, which are much lower than this study with around 20 million reads per cells. Therefore, I hope methods like BRIE will improve the analysis in this problem.

Second, in the above study (Song et al., 2017), it has been reported that around 80% genes have a uni-modal splicing patten, i.e., only inclusion or only exclusion of an alternative splicing in one cell population, and around the remaining 20% genes have bimodal splicing pattern, i.e., either inclusion or exclusion in a single cell (see Figure 2 in the original paper). This is a very interesting discovery. However, the drop-out events may be a confounder here, as the drop-out of one of the two RNA isoforms in the beginning of the experiment protocol will also lead to the bimodal distribution. On the other hand, if the bimodal distribution is the major distribution of alternative splicing in a cell population, then the detection of differential splicing becomes much easier. Consequently, detecting the bimodal distribution from other patterns, e.g., multi-modal distribution may be another task to solve, especially considering the uncertainty in splicing estimation.

Third, as splicing adds another layer for analysing scRNA-seq data, many questions asked at a gene expression level can be asked again at the splicing level, but the methodology may be slightly different, as splicing values are self-normalized and based on an indirect estimation. Some example questions could be 1) how to leverage splicing analysis for identifying cell development or evolution trajectory; 2) whether it is possible to use splicing patterns to regress out confounders, e.g., cell cycles, in defining cell types; 3) how to effectively visualize the splicing patterns in multiple cell subtypes.

Overall, the structured Bayesian methods developed in this thesis not only mitigate the challenges in splicing analysis with very low coverages, but also show examples of how Bayesian methods could be used to integrate multiple data sets and consider many factors together to improve the overall analysis. Therefore, Bayesian methods are very likely to continue their valuable contributions in addressing open questions in splicing analysis as discussed above or in a more general area of computational biology.

Appendix A

Supplementary Figures for Chapter 3

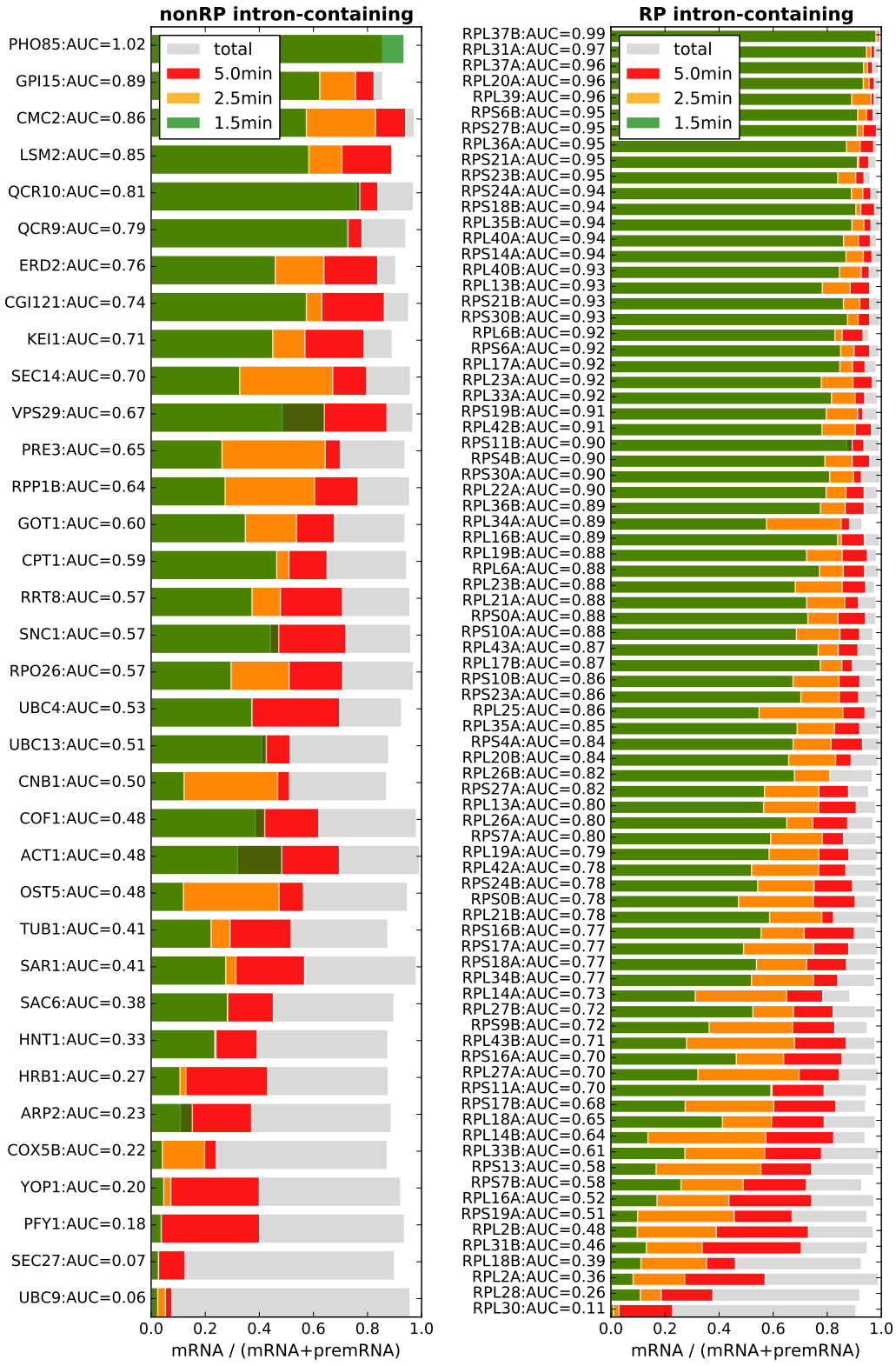

Figure A.1: The splicing speed and the mRNA proportions at 1.5 min, 2.5 min, 5.0 min and steady state for 35 non-ribosomal protein (RP) intron-containing genes (left panel) and 82 RP intron-containing genes (right panel). The proportion of mRNA is estimated from RNA-seq data using the static version of DICE-seq, and the area under the curve (AUC) score denotes the splicing speed.

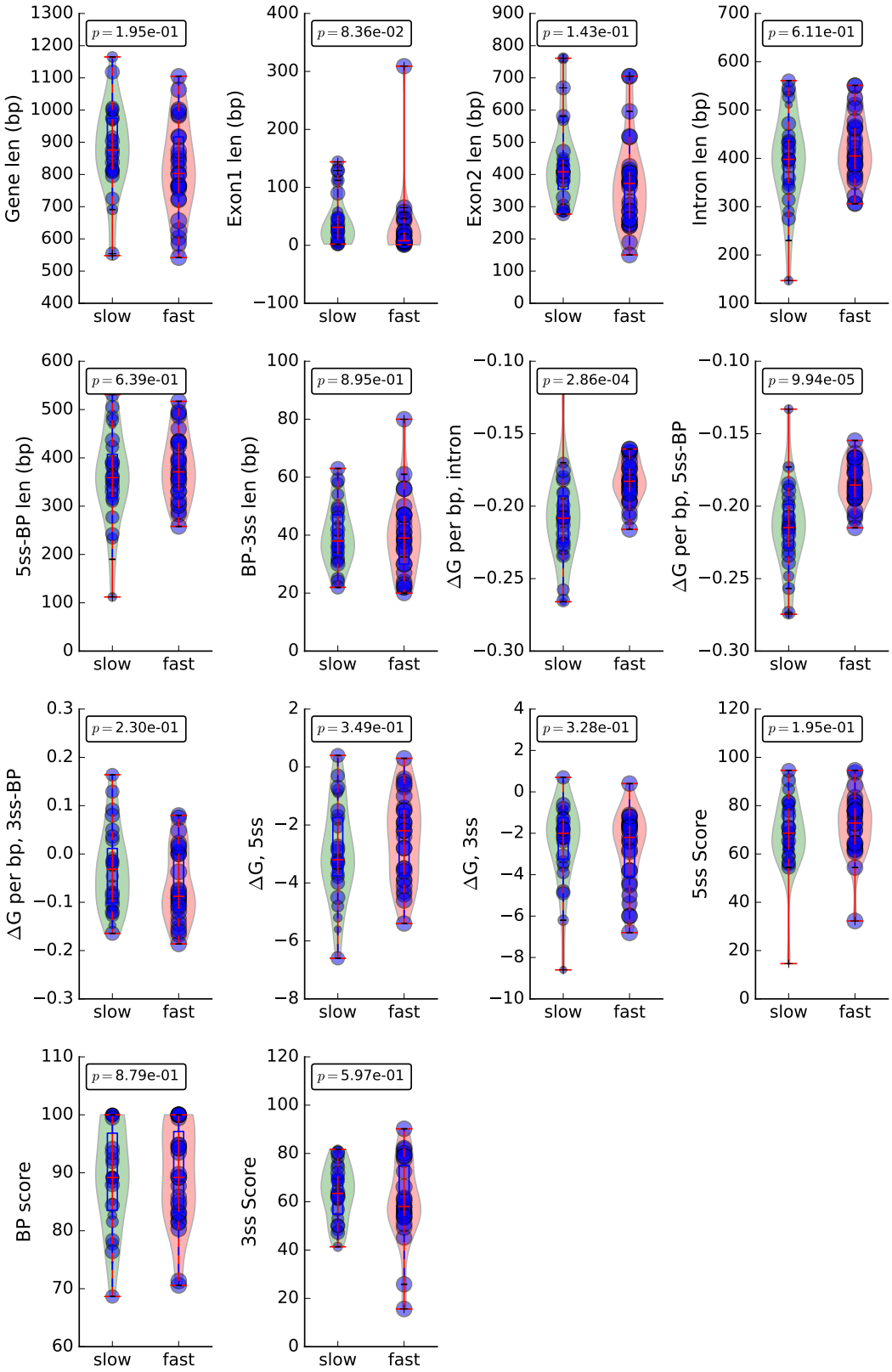

Figure A.2: Violin plots of 11 features for the 1/3 fastest and 1/3 slowest RP intron-containing genes. The splicing speed is measured by AUC (see Methods). The red horizontal line is the median of the feature, and the red vertical solid line ends at the quartiles of the feature. The dots in the violin box are the samples of each feature, whose sizes are corresponding to its splicing speed.

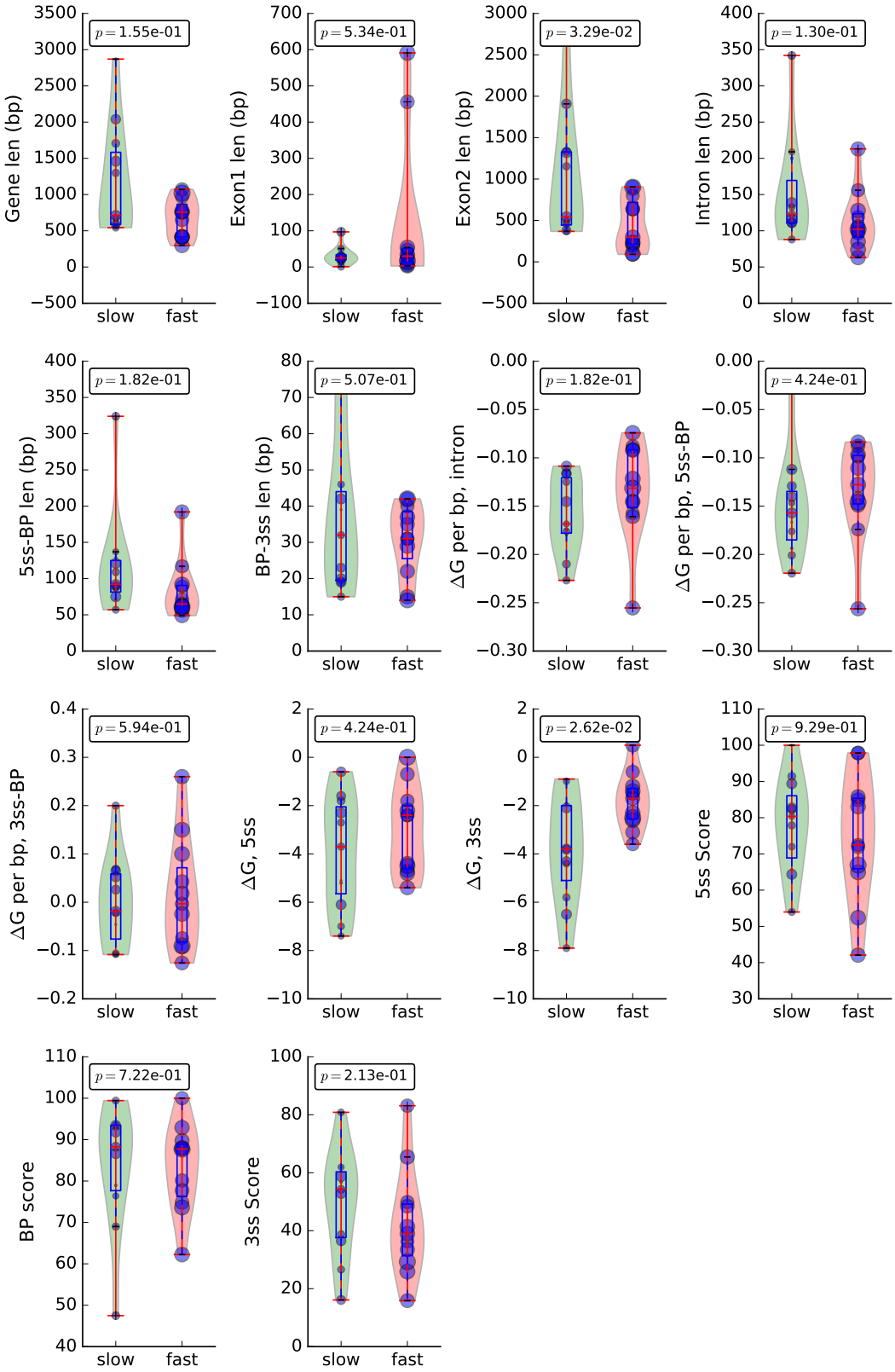

Figure A.3: Violin plots of 11 features for the 1/3 fastest and 1/3 slowest non-RP intron-containing genes. The splicing speed is measured by AUC (see Methods). The red horizontal line is the median of the feature, and the red vertical solid line ends at the quartiles of the feature. The dots in the violin box are the samples of each feature, whose sizes are corresponding to its splicing speed.

Appendix B

Supplementary materials for Chapter 4

This is a supplementary file for Chapter 4 "Gaussian process prior for time-series RNA-seq data". It contains the analysis of computation performance, inference of hyperparameter, supplementary algorithm S1 (Algorithm 9), supplementary figures S1-S3 and supplementary tables S1-S3. Note, the labels of the figures in the appendix is slightly different from the way I mensioned it in the Chapter 4, for example Figure B.1 here means Figure S1, and similar to other number and tables.

Computation performance We first investigate the speed of these methods (IsoEM, Cufflinks, MISO, DICEseq, and its separate mode DICE-sepa) at different coverages. Here, 4 out 64 CPU cores were parallel used for Cufflinks, MISO, DICEseq, and DICE-sepa. IsoEM was used as default in multiple cores setting. From supplementary Figure S1, we see that IsoEM is the fastest method, followed by Cufflinks. Though MISO, DICEseq and DICE-sepa are much slower than IsoEM and Cufflinks, they still finish the 8 time points within a reasonable period. Also, we noticed that the running time for both DICEseq and DICE-sepa increases much slower along the coverage than other methods. Therefore, we further tested the running time for different number of genes in the annotation file on a single ENCODE library. And we found that the running time for both MISO and DICEseq is almost linearly correlated with the number of genes for estimate. Notably, if there are only a fewer hundreds genes for study (e.g., total $\sim$300 intron-containing genes in yeast), DICEseq could finish the job within 10 minutes for a single time point.

Inference of hyperparameter In the main paper, we fixed the hyperparameters when inferring isoform propotions Ψ. Here, we introduce a Metropolis-Hasting sampler in Algorithm S1 for inferring hyperparameter Θ_2 simultaneously.

Algorithm 9: A Metropolis-Hastings sampler for posterior of $\boldsymbol{Y}$ and $\boldsymbol{\Theta}$

Data: $T, \boldsymbol{R}$

Initialize: $\boldsymbol{\Theta}^{(0)}, \boldsymbol{Y}^{(0)}$; $\boldsymbol{\Psi}^{(0)} = \texttt{Softmax}(\boldsymbol{Y}^{(0)})$

for $i = 0$ *to* H **do**

  Sample: $\mu \sim U(0, 1)$

  Sample: $\boldsymbol{\Theta}^* \sim Q_\theta(\boldsymbol{\Theta}^* | \boldsymbol{\Theta}^{(i)})$

  Sample: $\boldsymbol{Y}^* \sim Q_y(\boldsymbol{Y}^* | \boldsymbol{Y}^{(i)}, \boldsymbol{\Theta}^*)$; $\boldsymbol{\Psi}^* = \texttt{Softmax}(\boldsymbol{Y}^*)$

  if $\mu < \texttt{min}\Big\{ \dfrac{P(\boldsymbol{\Psi}^*, \boldsymbol{\Theta}^* | \boldsymbol{R}) Q_\theta(\boldsymbol{\Theta}^{(i)} | \boldsymbol{\Theta}^*) Q_y(\boldsymbol{Y}^{(i)} | \boldsymbol{Y}^*, \boldsymbol{\Theta}^*)}{P(\boldsymbol{\Psi}^{(i)}, \boldsymbol{\Theta}^{(i)} | \boldsymbol{R}) Q_\theta(\boldsymbol{\Theta}^* | \boldsymbol{\Theta}^{(i)}) Q_y(\boldsymbol{Y}^* | \boldsymbol{Y}^{(i)}, \boldsymbol{\Theta}^{(i)})}, 1 \Big\}$ **then**

    $\boldsymbol{Y}^{(i+1)} \leftarrow \boldsymbol{Y}^*$; $\boldsymbol{\Theta}^{(i+1)} \leftarrow \boldsymbol{\Theta}^*$; $\boldsymbol{\Psi}^{(i+1)} \leftarrow \boldsymbol{\Psi}^*$

  else

    $\boldsymbol{Y}^{(i+1)} \leftarrow \boldsymbol{Y}^{(i)}$; $\boldsymbol{\Theta}^{(i+1)} \leftarrow \boldsymbol{\Theta}^{(i)}$; $\boldsymbol{\Psi}^{(i+1)} \leftarrow \boldsymbol{\Psi}^{(i)}$

Here, we fix the $\theta_1 = 3.0$ as it matches the 95% confidence interval well. The only difference between Algorithm 1 in the main text is that we propose a new $\theta^*_{c,2}$ via its proposal distribution Q_θ, a truncated Gaussian distribution. Its mean is the previously accepted $\theta^{(i)}_{c,2}$, and its variance and boundaries are predefined, namely $\mathcal{N}_{[0.01:100]}(\theta^{(i)}_{c,2}, 1.0)$. In addition, the newly sampled $\theta^*_{c,2}$ is used in proposing Y^*_c. Finally, the same strategy on convergence diagnostics in Algorithm 1 is applied here for Θ_2.

Note, the computation of hyperparameters is much slower than that with fixed ones. In our tests, no significant improvement has been observed by sampling the hyperparameters. Thus, we highly suggest fixing the hyperparameters from knowledge of experiment design. In addition, we provide example splicing dynamics with different θ_2 on one 4-isoform gene in Figure S3. Furthermore, the Table S1 shows that the best estimates are achieved by setting the correct θ_2, but also evidences that the model is robust to mis-specification of θ_2. In the 4tU-seq and circadian studies, we set the length-scale of θ_2 as 90% and 7.5%, respectively, due to their different gap of experiments.

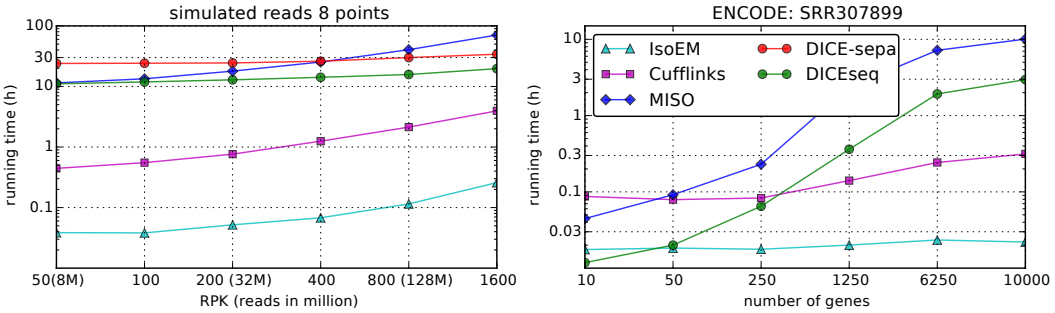

Figure B.1: Comparison of the running time between methods. Running time on simulated libraries with different coverages (left panel). Running time on an ENCODE library for different number of genes (right panel).

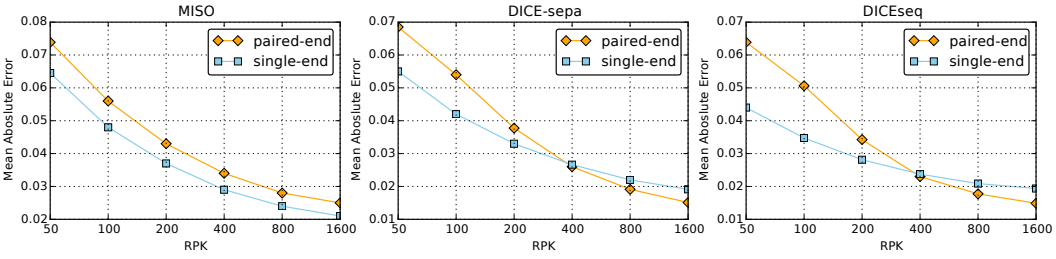

Figure B.2: Comparison between using single-end and paired-end reads for MISO (left panel), DICE-sepa (middle panel) and DICEseq (right panel) on simulated reads, by measuring the mean absolute error between the estimates and the truth.

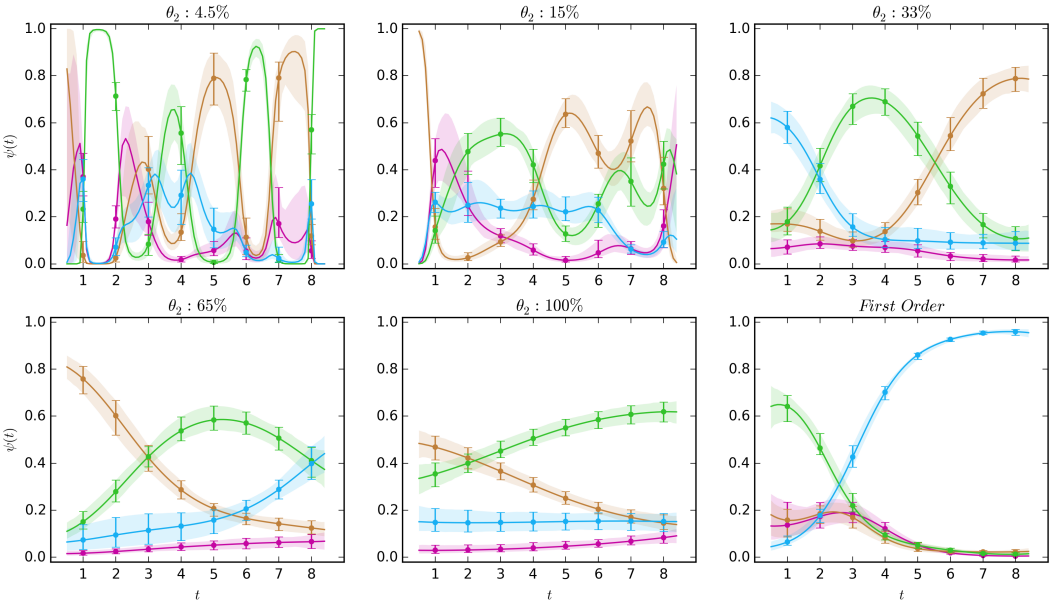

Figure B.3: Simulation examples of isoform dynamics on human gene ENSG00000100207 with 4 isoforms. There are five dynamic systems following Gaussian process with different changing speeds, denoted by the scale of θ_2, and one first-order dynamic system.

Table B.1: Mean absolute errors between inferred and true isoform proportions with different fixed θ_2 (rows) in different situations (column). Here, the length-scale of θ_2 is used as relative values. Seven situations were simulated: 6 Gaussian process dynamics with different θ_2, and a First Order dynamics. The quantification is also performed in 7 different ways: 6 joint measurements with different θ_2, and one separate measurement (column "sepa"). A total of 1,527 isoforms from 200 random human genes were used in the simulation, with 8 time points and coverage of RPK=400. The bold is the diagonal of a matched setting pair, with all reaching the best performance.

Truth \ Fixed	sepa	4.5%	15%	33%	65%	100%	150%
4.5%	0.028	**0.028**	0.033	0.066	0.084	0.091	0.096
15%	0.028	0.029	**0.028**	0.041	0.059	0.067	0.075
33%	0.029	0.029	0.027	**0.026**	0.031	0.038	0.047
65%	0.028	0.028	0.027	0.024	**0.023**	0.025	0.029
100%	0.029	0.029	0.026	0.024	0.023	**0.023**	0.024
150%	0.028	0.028	0.026	0.023	0.022	0.022	**0.021**
First order	0.029	0.029	0.026	0.024	0.024	0.025	0.029

Table B.2: Comparison between uniform and biased distribution of reads. In the 4tU-seq experiment, the scores are the Pearson's correlation coefficient between two replicates. In the circadian experiment, the scores are the Spearman's correlation coefficient between the measurements of RNA-seq and microarray. MISO does not support bias correction, and IsoEM failed to return results when correcting bias for replicate 2 in 4tU-seq experiment.

	IsoEM	Cufflinks	MISO	DICEseq
4tU-seq, unif	0.851	0.830	0.843	0.892
4tU-seq, bias	X	0.839	X	0.897
circadian, unif	0.802	0.794	0.804	0.809
circadian, bias	0.803	0.796	X	0.807

Table B.3: Comparison between using single-end (SE) and paired-end (PE) reads in circadian experiment. The second and third rows are the the Spearman's correlation coefficient between the measurement of RNA-seq and microarray. The fourth and fifth rows are the number of genes which passed the threshold of 95%$<$0.3 at all 8 time points.

	MISO	DICEseq
Spearmans' R, SE	0.805	0.807
Spearmans' R, PE	0.804	0.809
N(95%$<$0.3), SE	34213	35424
N(95%$<$0.3), PE	30275	32551

Appendix C

Supplementary Figures for Chapter 5

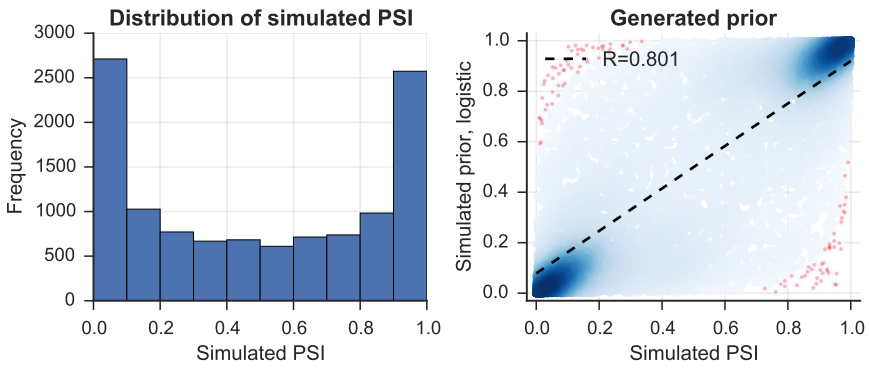

Figure C.1: The settings for simulation of exon inclusion ratio and corresponding prior. Left panel: the distribution of simulated inclusion ratio ψ, which follows a logit-normal distibution with mean $\mu = 0$ and variance $\theta^2 = 3^2$. Right panel: the correlation between the generated feature (for learning prior) and the input truth for simulation.

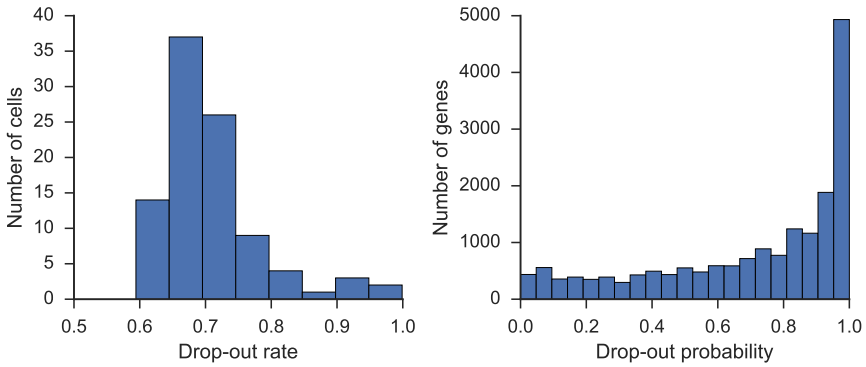

Figure C.2: The distribution of drop-out rate and drop-out probability of exon-triplets. Left panel: drop-out rate distribution across 96 human HCT116 cells. Drop-out rate calculate by the fraction of expressed exon-triplets in bulk cells that have no reads in single cells. Right panel: drop-out probability of each gene, which is calculated by the frequency of its drop-out in 96 cells.

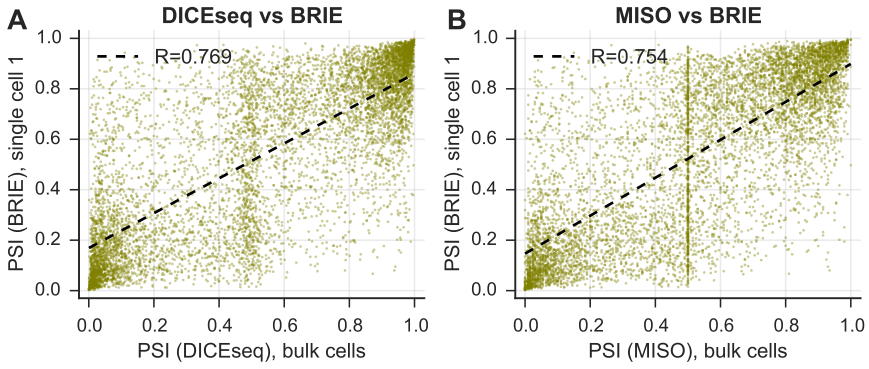

Figure C.3: Scatter plots of exon inclusion ratio estimates from HCT116 cells. (A) Bulk cells by DICE-seq and single cell by BRIE. (B) Bulk cells by MISO and single cell by BRIE.

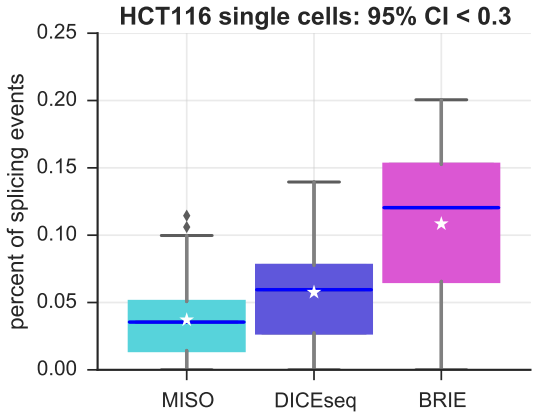

Figure C.4: Box plot of percentage of splicing events that have 95% confidence interval $<$0.3 in 96 HCT116 cells. Three methods are used: MISO, DICEseq, BRIE.

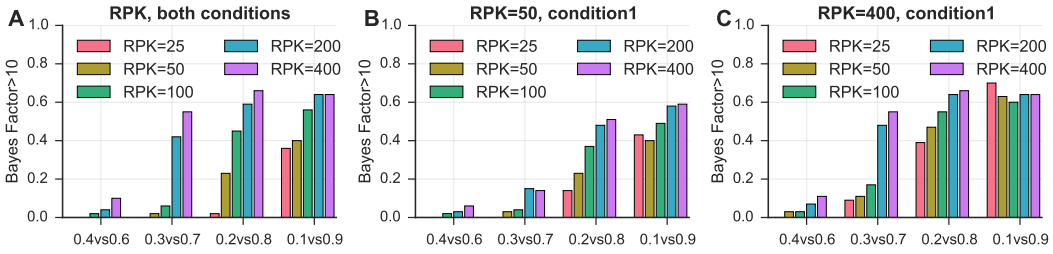

Figure C.5: Percentage of genes with $BayesFactor > 10$ by comparing two conditions in different coverages and ψ values. (A) Coverages for both conditions are the same, ranges from RPK=25 to RPK=400. (B) RPK fixed as 50 in condition 1. (C) RPK fixed as 400 in condition 1.

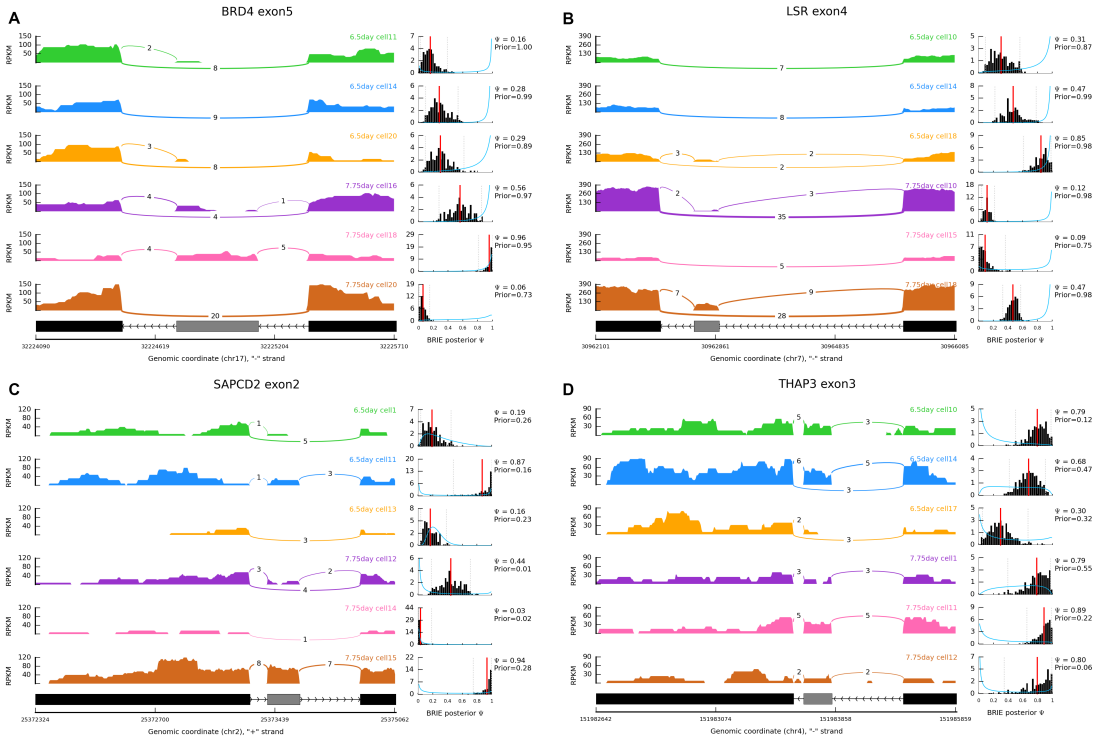

Figure C.6: Four more example events with high splicing variation between 6.5 days and 7.75 days. The format of the figure is the same as Fig 5.10c

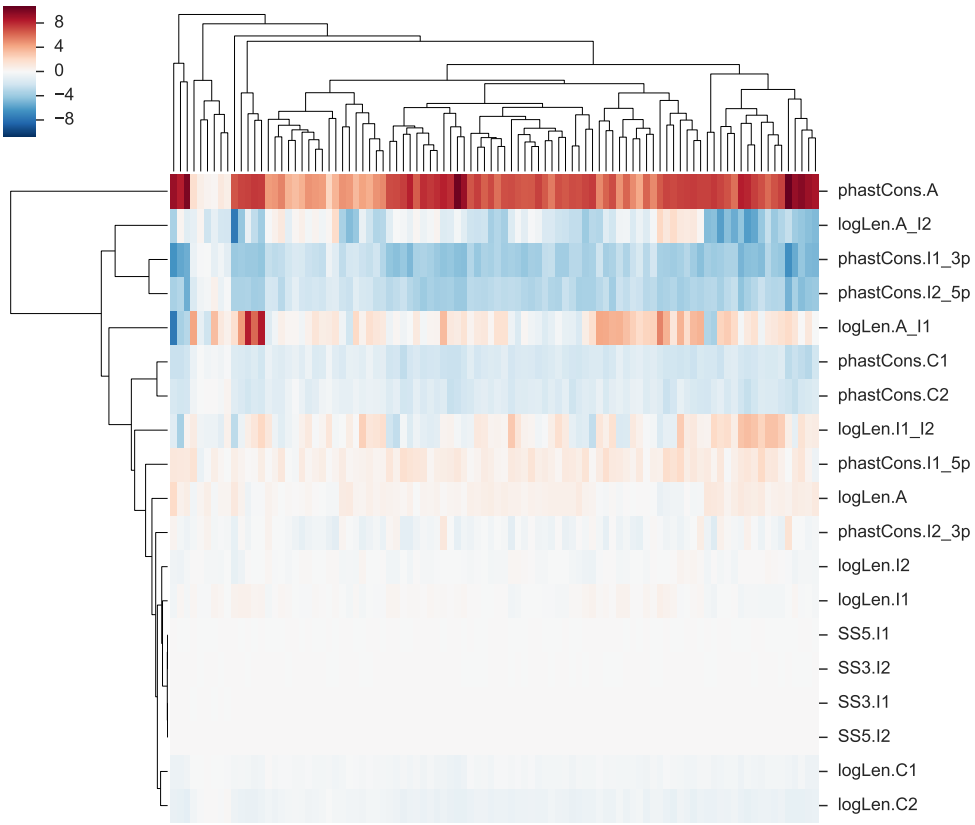

Figure C.7: Heatmap of weights of 19 sequence related features, learned by BRIE in 96 cells. The x-axis is 96 cells, and the y-axis is 19 features.

Bibliography

Äijö, T., Butty, V., Chen, Z., Salo, V., Tripathi, S., Burge, C. B., Lahesmaa, R., and Lähdesmäki, H. (2014). Methods for time series analysis of RNA-seq data with application to human Th17 cell differentiation. *Bioinformatics*, 30(12):i113–i120.

Airoldi, E. M. (2007). Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2008). *Molecular Biology of the Cell*. Garland Science, New York, Fifth edition.

Alt, F. W., Bothwell, A. L., Knapp, M., Siden, E., Mather, E., Koshland, M., and Baltimore, D. (1980). Synthesis of secreted and membrane-bound immunoglobulin mu heavy chains is directed by mRNAs that differ at their 3 ends. *Cell*, 20(2):293–301.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43.

Aslanzadeh, V., Huang, Y., Sanguinetti, G., and Beggs, J. D. (2017). Transcription rate strongly affects splicing fidelity and cotranscriptionality in budding yeast. *Genome research*.

Atchison, J. and Shen, S. M. (1980). Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272.

Azad, N., Zahnow, C. A., Rudin, C. M., and Baylin, S. B. (2013). The future of epigenetic therapy in solid tumours – lessons from the past. *Nature reviews Clinical oncology*, 10(5):256–266.

Bar-Joseph, Z., Gitter, A., and Simon, I. (2012). Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564.

Baralle, F. E. and Giudice, J. (2017). Alternative splicing as a regulator of development and tissue identity. *Nat Rev Mol Cell Biol*, 18(7):437–451.

Barrass, J. D., Reid, J. E., Huang, Y., Hector, R. D., Sanguinetti, G., Beggs, J. D., and Granneman, S. (2015). Transcriptome-wide RNA processing kinetics revealed using extremely short 4tU labeling. *Genome biology*, 16(1):282.

Bell, O., Tiwari, V. K., Thomä, N. H., and Schübeler, D. (2011). Determinants and dynamics of genome accessibility. *Nature Reviews Genetics*, 12(8):554–564.

Bentley, D. L. (2014). Coupling mRNA processing with transcription in time and space. *Nature Reviews Genetics*, 15(3):163–175.

Berget, S. M., Moore, C., and Sharp, P. A. (1977). Spliced segments at the 5 terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, 74(8):3171–3175.

Beyer, A. L. and Osheim, Y. N. (1988). Splice site selection, rate of splicing, and alternative splicing on nascent transcripts. *Genes & development*, 2(6):754–765.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blencowe, B. J. (2006). Alternative splicing: new insights from global analyses. *Cell*, 126(1):37–47.

Boukouvalas, A., Hensman, J., and Rattray, M. (2017). BGP: Branched Gaussian processes for identifying gene-specific branching dynamics in single cell data. *bioRxiv*, page 166868.

Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*, 34(5):525–527.

Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A. A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S. A., Marioni, J. C., et al. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, 10(11):1093–1095.

Buettner, F., Natarajan, K. N., Casale, F. P., Proserpio, V., Scialdone, A., Theis, F. J., Teichmann, S. A., Marioni, J. C., and Stegle, O. (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160.

Cáceres, J. F. and Kornblihtt, A. R. (2002). Alternative splicing: multiple control mechanisms and involvement in human disease. *TRENDS in Genetics*, 18(4):186–193.

Campbell, K. and Yau, C. (2016). Ouija: Incorporating prior knowledge in single-cell trajectory learning using bayesian nonlinear factor analysis. *bioRxiv*, page 060442.

Chen, M. and Manley, J. L. (2009). Mechanisms of alternative splicing regulation: insights from molecular and genomics approaches. *Nat Rev Mol Cell Biol*, 10(11):741–754.

Chow, L. T., Gelinas, R. E., Broker, T. R., and Roberts, R. J. (1977). An amazing sequence arrangement at the 5 ends of adenovirus 2 messenger RNA. *Cell*, 12(1):1–8.

Churchman, L. S. and Weissman, J. S. (2011). Nascent transcript sequencing visualizes transcription at nucleotide resolution. *Nature*, 469(7330):368–373.

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., et al. (2016). A survey of best practices for RNA-seq data analysis. *Genome biology*, 17(1):13.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

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

Curado, J., Iannone, C., Tilgner, H., Valcárcel, J., and Guigó, R. (2015). Promoter-like epigenetic signatures in exons displaying cell type-specific splicing. *Genome Biology*, 16(1):1–16.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Diedenhofen, B. and Musch, J. (2015). Cocor: A comprehensive solution for the statistical comparison of correlations. *PloS one*, 10(4):e0121945.

Dillman, A. A., Hauser, D. N., Gibbs, J. R., Nalls, M. A., McCoy, M. K., Rudenko, I. N., Galter, D., and Cookson, M. R. (2013). mRNA expression, splicing and editing in the embryonic and adult mouse cerebral cortex. *Nature neuroscience*, 16(4):499–506.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21.

Dong, X., Greven, M. C., Kundaje, A., Djebali, S., Brown, J. B., Cheng, C., Gingeras, T. R., Gerstein, M., Guigó, R., Birney, E., et al. (2012). Modeling gene expression using chromatin features in various cellular contexts. *Genome biology*, 13(9):R53.

Dongarra, J. and Sullivan, F. (2000). Guest editors introduction: The top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23.

Duymich, C. E., Charlet, J., Yang, X., Jones, P. A., and Liang, G. (2016). DNMT3B isoforms without catalytic activity stimulate gene body methylation as accessory proteins in somatic cells. *Nature Communications*, 7.

Early, P., Rogers, J., Davis, M., Calame, K., Bond, M., Wall, R., and Hood, L. (1980). Two mRNAs can be produced from a single immunoglobulin μ gene by alternative RNA processing pathways. *Cell*, 20(2):313–319.

ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.

Eser, P., Wachutka, L., Maier, K. C., Demel, C., Boroni, M., Iyer, S., Cramer, P., and Gagneur, J. (2016). Determinants of RNA metabolism in the Schizosaccharomyces pombe genome. *Molecular Systems Biology*, 12(2).

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118.

Fan, J., Salathia, N., Liu, R., Kaeser, G. E., Yung, Y. C., Herman, J. L., Kaper, F., Fan, J.-B., Zhang, K., Chun, J., et al. (2016). Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nature methods*, 13(3):241.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., et al. (2015). MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome biology*, 16(1):278.

Fong, N., Kim, H., Zhou, Y., Ji, X., Qiu, J., Saldi, T., Diener, K., Jones, K., Fu, X.-D., and Bentley, D. L. (2014). Pre-mRNA splicing is facilitated by an optimal RNA polymerase II elongation rate. *Genes & development*, 28(23):2663–2676.

Fuchs, G., Voichek, Y., Benjamin, S., Gilad, S., Amit, I., and Oren, M. (2014). 4sUDRB-seq: measuring genomewide transcriptional elongation rates and initiation frequencies within cells. *Genome Biology*, 15(5):R69.

Garber, M., Grabherr, M. G., Guttman, M., and Trapnell, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature methods*, 8(6):469–477.

Garcia-Blanco, M. A., Baraniak, A. P., and Lasda, E. L. (2004). Alternative splicing in disease and therapy. *Nature biotechnology*, 22(5):535–546.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472.

Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., Emanuelsson, O., Zhang, Z. D., Weissman, S., and Snyder, M. (2007). What is a gene, post-ENCODE? History and updated definition. *Genome Research*, 17(6):669–681.

Geweke, J. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, volume 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, USA.

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

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., et al. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature biotechnology*, 29(7):644–652.

Graveley, B. R. (2001). Alternative splicing: increasing diversity in the proteomic world. *TRENDS in Genetics*, 17(2):100–107.

Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., and van Oudenaarden, A. (2015). Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251–255.

Grün, D. and van Oudenaarden, A. (2015). Design and analysis of single-cell sequencing experiments. *Cell*, 163(4):799–810.

Harrow, J., Frankish, A., Gonzalez, J. M., Tapanari, E., Diekhans, M., Kokocinski, F., Aken, B. L., Barrell, D., Zadissa, A., Searle, S., et al. (2012). GENCODE: the reference human genome annotation for The ENCODE Project. *Genome research*, 22(9):1760–1774.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Honkela, A., Peltonen, J., Topa, H., Charapitsa, I., Matarese, F., Grote, K., Stunnenberg, H. G., Reid, G., Lawrence, N. D., and Rattray, M. (2015). Genome-wide modeling of transcription kinetics reveals patterns of RNA production delays. *Proceedings of the National Academy of Sciences*, 112(42):13115–13120.

Huang, Y. and Sanguinetti, G. (2016). Statistical modeling of isoform splicing dynamics from RNA-seq time series data. *Bioinformatics*, 32(19):2965–2972.

Huang, Y. and Sanguinetti, G. (2017). BRIE: transcriptome-wide splicing quantification in single cells. *Genome Biology*, 18(1):123.

Iijima, T., Wu, K., Witte, H., Hanno-Iijima, Y., Glatter, T., Richard, S., and Scheiffele, P. (2011). SAM68 regulates neuronal activity-dependent alternative splicing of neurexin-1. *Cell*, 147(7):1601–1614.

Jonkers, I. and Lis, J. T. (2015). Getting up to speed with transcription elongation by rna polymerase ii. *Nature reviews Molecular cell biology*, 16(3):167–177.

Jordan, M. I. et al. (2004). Graphical models. *Statistical Science*, 19(1):140–155.

Kanitz, A., Gypas, F., Gruber, A. J., Gruber, A. R., Martin, G., and Zavolan, M. (2015). Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data. *Genome Biology*, 16(1):1–26.

Kapourani, C.-A. and Sanguinetti, G. (2016). Higher order methylation features for clustering and prediction in epigenomic studies. *Bioinformatics*, 32(17):i405–i412.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Katz, Y., Wang, E. T., Airoldi, E. M., and Burge, C. B. (2010). Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nature methods*, 7(12):1009–1015.

Katz, Y., Wang, E. T., Silterra, J., Schwartz, S., Wong, B., Thorvaldsdóttir, H., Robinson, J. T., Mesirov, J. P., Airoldi, E. M., and Burge, C. B. (2015). Quantitative visualization of alternative exon expression from RNA-seq data. *Bioinformatics*, 31(14):2400–2402.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002). The human genome browser at UCSC. *Genome research*, 12(6):996–1006.

Kim, D., Langmead, B., and Salzberg, S. L. (2015). HISAT: a fast spliced aligner with low memory requirements. *Nature methods*, 12(4):357–360.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357–359.

Lappalainen, T., Sammeth, M., Friedländer, M. R., ACt Hoen, P., Monlong, J., Rivas, M. A., Gonzalez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P. G., et al. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511.

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2006). Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 785–792.

Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

Liu, P., Sanalkumar, R., Bresnick, E. H., Keleş, S., and Dewey, C. N. (2016). Integrative analysis with ChIP-seq advances the limits of transcript quantification from RNA-seq. *Genome Research*, 26(8):1124–1133.

Liu, Y., Zhou, J., and White, K. P. (2013). RNA-seq differential expression studies: more sequence or more replication? *Bioinformatics*, 30(3):301–304.

Lönnberg, T., Svensson, V., James, K. R., Fernandez-Ruiz, D., Sebina, I., Montandon, R., Soon, M. S., Fogg, L. G., Nair, A. S., Liligeto, U., et al. (2017). Single-cell rna-seq and computational analysis using temporal mixture modelling resolves th1/tfh fate bifurcation in malaria. *Science immunology*, 2(9).

Love, M. I., Hogenesch, J. B., and Irizarry, R. A. (2016). Modeling of RNA-seq fragment sequence bias reduces systematic errors in transcript abundance estimation. *Nature Biotechnology*, 34(12):1287–1291.

Luco, R. F., Pan, Q., Tominaga, K., Blencowe, B. J., Pereira-Smith, O. M., and Misteli, T. (2010). Regulation of alternative splicing by histone modifications. *Science*, 327(5968):996–1000.

Mardon, H. J., Sebastio, G., and Baralle, F. E. (1987). A role for exon sequences in alternative splicing of the human fibronection gene. *Nucleic acids research*, 15(19):7725–7733.

Matera, A. G. and Wang, Z. (2014). A day in the life of the spliceosome. *Nat Rev Mol Cell Biol*, 15(2):108–121.

Meng, X.-L. and Van Dyk, D. (1997). The EM Algorithman Old Folk-song Sung to a Fast New Tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–567.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Morel, C. F., Thomas, M. A., Cao, H., ONeil, C. H., Pickering, J. G., Foulkes, W. D., and Hegele, R. A. (2006). A lmna splicing mutation in two sisters with severe dunnigan-type familial partial lipodystrophy type 2. *The Journal of Clinical Endocrinology & Metabolism*, 91(7):2689–2695.

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

Muchir, A., Bonne, G., van der Kooi, A. J., van Meegen, M., Baas, F., Bolhuis, P. A., de Visser, M., and Schwartz, K. (2000). Identification of mutations in the gene encoding lamins A/C in autosomal dominant limb girdle muscular dystrophy with atrioventricular conduction disturbances (LGMD1B). *Human molecular genetics*, 9(9):1453–1459.

Murphy, K. P. (2012). *Machine Learning: a Probabilistic Perspective*. MIT press.

Nellore, A., Jaffe, A. E., Fortin, J.-P., Alquicira-Hernández, J., Collado-Torres, L., Wang, S., Phillips III, R. A., Karbhari, N., Hansen, K. D., Langmead, B., et al. (2016). Human splicing diversity and the extent of unannotated splice junctions across human RNA-seq samples on the Sequence Read Archive. *Genome Biology*, 17(1):266.

Nicolae, M., Mangul, S., Măndoiu, I. I., and Zelikovsky, A. (2011). Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for molecular biology*, 6(1):9.

Nilsen, T. W. and Graveley, B. R. (2010). Expansion of the eukaryotic proteome by alternative splicing. *Nature*, 463(7280):457–463.

Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., Gillespie, S. M., Wakimoto, H., Cahill, D. P., Nahed, B. V., Curry, W. T., Martuza, R. L., et al. (2014). Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401.

Patro, R., Mount, S. M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature biotechnology*, 32(5):462–464.

Pollard, K. S., Hubisz, M. J., Rosenbloom, K. R., and Siepel, A. (2010). Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Research*, 20(1):110–121.

Qiu, X., Hill, A., Packer, J., Lin, D., Ma, Y.-A., and Trapnell, C. (2017). Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14(3):309–315.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, USA.

Roberts, A. and Pachter, L. (2013). Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature methods*, 10(1):71–73.

Roberts, A., Trapnell, C., Donaghey, J., Rinn, J. L., and Pachter, L. (2011). Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome biology*, 12(3):R22.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367.

Rogé, X. and Zhang, X. (2013). RNAseqViewer: visualization tool for RNA-Seq data. *Bioinformatics*, 30(6):891–892.

Schulz, D., Schwalb, B., Kiesel, A., Baejen, C., Torkler, P., Gagneur, J., Soeding, J., and Cramer, P. (2013). Transcriptome surveillance by selective termination of noncoding RNA synthesis. *Cell*, 155(5):1075–1087.

Scialdone, A., Tanaka, Y., Jawaid, W., Moignard, V., Wilson, N. K., Macaulay, I. C., Marioni, J. C., and Göttgens, B. (2016). Resolving early mesoderm diversification through single-cell expression profiling. *Nature*, 535(7611):284–293.

Scotti, M. M. and Swanson, M. S. (2015). RNA mis-splicing in disease. *Nature Reviews Genetics*.

Sefer, E., Kleyman, M., and Bar-Joseph, Z. (2016). Tradeoffs between Dense and Replicate Sampling Strategies for High-Throughput Time Series Experiments. *Cell systems*, 3(1):35–42.

Shalek, A. K., Satija, R., Shuga, J., Trombetta, J. J., Gennert, D., Lu, D., Chen, P., Gertner, R. S., Gaublomme, J. T., Yosef, N., et al. (2014). Single cell RNA Seq reveals dynamic paracrine control of cellular variation. *Nature*, 510(7505):363.

Shen, S., Park, J. W., Lu, Z.-x., Lin, L., Henry, M. D., Wu, Y. N., Zhou, Q., and Xing, Y. (2014). rMATS: robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proceedings of the National Academy of Sciences*, 111(51):E5593–E5601.

Song, Y., Botvinnik, O. B., Lovci, M. T., Kakaradov, B., Liu, P., Xu, J. L., and Yeo, G. W. (2017). Single-cell alternative splicing analysis with expedition reveals splicing dynamics during neuron differentiation. *Molecular Cell*, 67(1):148–161.

Stegle, O., Denby, K. J., Cooke, E. J., Wild, D. L., Ghahramani, Z., and Borgwardt, K. M. (2010). A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *Journal of Computational Biology*, 17(3):355–367.

Sturgill, D., Malone, J. H., Sun, X., Smith, H. E., Rabinow, L., Samson, M.-L., and Oliver, B. (2013). Design of RNA splicing analysis null models for post hoc filtering of Drosophila head RNA-Seq data with the splicing analysis kit (Spanki). *BMC Bioinformatics*, 14(1):320.

Thorvaldsdóttir, H., Robinson, J. T., and Mesirov, J. P. (2013). Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, 14(2):178–192.

Tilgner, H., Knowles, D. G., Johnson, R., Davis, C. A., Chakrabortty, S., Djebali, S., Curado, J., Snyder, M., Gingeras, T. R., and Guigó, R. (2012). Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. *Genome research*, 22(9):1616–1625.

Topa, H. and Honkela, A. (2016). Analysis of differential splicing suggests different modes of short-term splicing regulation. *Bioinformatics*, 32(12):i147–i155.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386.

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

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., Van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515.

Traunmüller, L., Gomez, A. M., Nguyen, T.-M., and Scheiffele, P. (2016). Control of neuronal synapse specification by a highly dedicated alternative splicing program. *Science*, 352(6288):982–986.

Tuomela, S., Salo, V., Tripathi, S. K., Chen, Z., Laurila, K., Gupta, B., Äijö, T., Oikari, L., Stockinger, B., Lähdesmäki, H., et al. (2012). Identification of early gene expression changes during human Th17 cell differentiation. *Blood*, 119(23):e151–e160.

Vallejos, C. A., Marioni, J. C., and Richardson, S. (2015). Basics: Bayesian analysis of single-cell sequencing data. *PLoS computational biology*, 11(6):e1004333.

Veloso, A., Kirkconnell, K. S., Magnuson, B., Biewen, B., Paulsen, M. T., Wilson, T. E., and Ljungman, M. (2014). Rate of elongation by RNA polymerase II is associated with specific gene features and epigenetic modifications. *Genome Research*, 24(6):896–905.

Venables, J. P., Klinck, R., Koh, C., Gervais-Bird, J., Bramard, A., Inkel, L., Durand, M., Couture, S., Froehlich, U., Lapointe, E., et al. (2009). Cancer-associated regulation of alternative splicing. *Nature structural & molecular biology*, 16(6):670–676.

Verdinelli, I. and Wasserman, L. (1995). Computing Bayes factors using a generalization of the Savage-Dickey density ratio. *Journal of the American Statistical Association*, 90(430):614–618.

Wahl, M. C., Will, C. L., and Lührmann, R. (2009). The spliceosome: design principles of a dynamic RNP machine. *Cell*, 136(4):701–718.

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

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63.

Welch, J. D., Hu, Y., and Prins, J. F. (2016). Robust detection of alternative splicing in a population of single cells. *Nucleic Acids Research*, 44(8):e73–e73.

White, E. S., Baralle, F. E., and Muro, A. F. (2008). New insights into form and function of fibronectin splice variants. *The Journal of pathology*, 216(1):1–14.

Windhager, L., Bonfert, T., Burger, K., Ruzsics, Z., Krebs, S., Kaufmann, S., Malterer, G., L'Hernault, A., Schilhabel, M., Schreiber, S., et al. (2012). Ultrashort and progressive 4sU-tagging reveals key characteristics of RNA processing at nucleotide resolution. *Genome research*, 22(10):2031–2042.

Wu, A. R., Neff, N. F., Kalisky, T., Dalerba, P., Treutlein, B., Rothenberg, M. E., Mburu, F. M., Mantalas, G. L., Sim, S., Clarke, M. F., et al. (2014). Quantitative assessment of single-cell RNA-sequencing methods. *Nature Methods*, 11(1):41–46.

Xiong, H. Y., Alipanahi, B., Lee, L. J., Bretschneider, H., Merico, D., Yuen, R. K., Hua, Y., Gueroussov, S., Najafabadi, H. S., Hughes, T. R., et al. (2015). The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218):1254806.

Zhang, R., Lahens, N. F., Ballance, H. I., Hughes, M. E., and Hogenesch, J. B. (2014). A circadian gene expression atlas in mammals: Implications for biology and medicine. *Proceedings of the National Academy of Sciences*, 111(45):16219–16224.

Zheng, W., Chung, L. M., and Zhao, H. (2011). Bias detection and correction in RNA-Sequencing data. *BMC bioinformatics*, 12(1):290.

Ziegenhain, C., Vieth, B., Parekh, S., Reinius, B., Guillaumet-Adkins, A., Smets, M., Leonhardt, H., Heyn, H., Hellmann, I., and Enard, W. (2017). Comparative analysis of single-cell rna sequencing methods. *Molecular cell*, 65(4):631–643.