5-2018

From Pieces To Paths: Combining Disparate Information in Computational Analysis of RNA-Seq.

Yifan Yang
Purdue University

FROM PIECES TO PATHS: COMBINING DISPARATE INFORMATION IN COMPUTATIONAL ANALYSIS OF RNA-SEQ

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Yifan Yang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Michael Gribskov, Chair

Department of Biological Sciences and Computer Science

Dr. Yuan Qi

Department of Computer Science and Statistics

Dr. Tony R. Hazbun

Department of Medicinal Chemistry and Molecular Pharmacology

Dr. Michael Y. Zhu

Department of Statistics

Approved by:

Dr. Stephen F. Konieczny

Head of Department of Biological Sciences

Approved by:

Dr. Jason R. Cannon

Head of Purdue University Life Sciences

For a better world.

ACKNOWLEDGMENTS

I want to show my greatest gratitude to both of my advisors, Professor Michael Gribskov and Professor Yuan (Alan) Qi, who supported me both academically and financially throughout my Ph.D. study.

I treasured the opportunities of working with two professors and receiving interdisciplinary training in both the Biological Sciences department and Computer science department. Dr. Gribskov, not only gave me invaluable guidance on how to conduct scientific research by choosing topics with great impact, and thinking creatively and critically, but more importantly, being a life mentor, selflessly sharing his life and career experience to me. Dr. Qi opened up an entirely new world – machine learning – for me, inspiring me with quantitative thinking for solving biological problems. Both professors have tremendous influence on my growth from a student to a young researcher.

I also want to express my sincere appreciation to two other professors in my committee, Professor Tony Hazbun and Professor Michael Yu Zhu. These two professors, who are experts in pharmaceutics and statistics respectively, gave me great advice on my research, oral presentations, and scientific writings in my preliminary examination and all the committee meetings over the years.

I would also like to thank all my research fellows and collaborators. To name a few, they are Shandian Zhe, Hao Peng, and Syed Abbas Z Naqvi. Among my best memories are the discussions that we had, each contributing from the perspective of our own domain, in which we learned from each other and worked towards common goals, and in the face of many deadlines.

I want to thank all my friends who I met at Purdue. I'm grateful that life has brought our paths crossing to each other's for sharing the happiness and getting through the difficulties of these years.

Lastly, I will say thanks to Ci Zhang, my parents and grandparents. Thanks for their long-lasting support and patience. Their endless love is the origin of the light in my heart, repelling all the darkness on the road.

PREFACE

In 2011, I was luckily admitted to the Purdue University Interdisciplinary Life Science program for my Ph.D. study. At that time, due to the rapid development of high throughput technologies in biology, researchers were beginning to apply statistical and machine learning algorithms to large-scale genomic data in order to model biological systems as interactive networks rather than by investigating each single gene. This systematic strategy greatly changed people's view of biological processes and gained exciting progress in many areas. I realized that the knowledge of computation and machine learning would play an essential role in biological studies in the near future. So, after four rotations during the first year, and being inspired by the post genomic era, I decided to devote my Ph.D. study to computational biology, and dedicate myself to the basic science of human health.

Yifan Yang 2017 Oct.

West Lafayette, IN, USA

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Yang, Yifan Ph.D., Purdue University, May 2018. From Pieces To Paths: Combining Disparate Information in Computational Analysis of RNA-Seq. Major Professor: Michael Gribskov.

As high-throughput sequencing technology has advanced in recent decades, large-scale genomic data with high-resolution have been generated for solving various problems in many fields. One of the state-of-the-art sequencing techniques is RNA sequencing, which has been widely used to study the transcriptomes of biological systems through millions of reads. The ultimate goal of RNA sequencing bioinformatics algorithms is to maximally utilize the information stored in a large amount of pieced-together reads to unveil the whole landscape of biological function at the transcriptome level.

Many bioinformatics methods and pipelines have been developed for better achieving this goal. However, one central question of RNA sequencing is the prediction uncertainty due to the short read length and the low sampling rate of underexpressed transcripts. Both conditions raise ambiguities in read mapping, transcript assembly, transcript quantification, and even the downstream analysis.

This dissertation focuses on approaches to reducing the above uncertainty by incorporating additional information, of disparate kinds, into bioinformatics models and modeling assessments. I addressed three critical issues in RNA sequencing data analysis. (1) we evaluated the performance of current *de novo* assembly methods and their evaluation methods using the transcript information from a third generation sequencing platform, which provides a longer sequence length but with a higher error rate than next-generation sequencing; (2) we built a Bayesian graphical model for improving transcript quantification and differentially expressed isoform identification

by utilizing the shared information from biological replicates; (3) we built a joint pathway and gene selection model by incorporating pathway structures from an expert database. We conclude that the incorporation of appropriate information from extra resources enables a more reliable assessment and a higher prediction performance in RNA sequencing data analysis.

1. INTRODUCTION

1.1 A brief history of sequencing technology

Each small step of a human being is built on the steps of pioneers in history. To learn about the history of sequencing technology or even of molecular biology will help us understand where we are in this long river of time, and why we carry out studies depicted in this dissertation.

For thousands of years, people persistently pursued the answer of one central question about life – what is life and how are the traits of lives inheritated on and on? Until 1953, when James D. Watson and Francis Crick first deduced the structure of DNA molecules (Watson and Crick, 1953), and when Francis Crick proposed the Central Dogma of biology (Crick, 1958), which first summarized how sequence information was transferred from DNA to RNA and from RNA to protein; the era of molecular biology had began. Researchers started to wonder whether these magic DNA and RNA molecules could be sequenced, and the sequences stored as a "Bible of Life" so that our descendants would be able to decipher this huge and treasured book, and answer the central question about life.

DNA sequencing is actually a fairly new field in our history (Metzker, 2008; Kulski, 2016). From 1977, about 25 years after the discovery of the DNA double helix structure, when Sanger and Maxam/Gilbert first invented the technology of DNA sequencing, until today, when sequencing techology has been widely industrialized and applied in many fields, it has only been about 40 years. I would like to divide these 40 years into three big stages, marked by different historical events as milestones. Of course, with the development of sequencing technology, the need for both computer hardware and software (e.g. databases and bioinformatics algorithms), has greatly increased.

The first stage of sequencing technology is from 1977 to 1990. During this period, people, for the first time, invented DNA sequencing techology with the ability to sequence short genomes up to thousands of base pairs; in the same period, databases for storing sequences and corresponding algorithms for searching sequences were initialized. The story should be backdated to 1965, when the first nucleic acid molecule – yeast alanine-tRNA – was sequenced (Holley et al., 1965). Believe it or not, it took researchers about 7 years to prepare a 1 gram tRNA sample from yeast, and the sequencing was purely based on chromatographic and spectrophotometric procedures, which was extremely time-consuming and labrious – only several bases could be sequenced per year ! In 1977, two sequencing methods – Maxam and Gilbert's chemical degradation method (Maxam and Gilbert, 1977) and Sanger's chain-termination sequencing method (Sanger et al., 1977) – were published, and competed with each other for years. Eventually, Sanger's method prevailed over the Maxam/Gilbert's method due to its greater simplicity and robustness, and dominated the sequencing field for the next 20 years. In 1986, Sanger's sequencing method was first incorporated into an automated instrument, which was marketed by Applied Biosystems Instruments (ABI), and sequencing speed reached 1,000 base pairs per day. Another epoch-making technology, which was invented by Mullis in 1983 (Saiki et al., 1985), is the polymerase chain reaction (PCR) technology, facilitating the development of sequencing technology by amplifying DNA molecules to a high concentration. In the area of databases and bioinformatics algorithms, GenBank was founded in 1982, and the BLAST algorithm (Altschul et al., 1990), the most widely used approach to identifying similar sequences in sequence databases, was developed in 1990.

The second stage is from 1990 to 2005. During this period, steady advances were made in sequencing technology, and the number of sequences submitted to the database explosively increased; Many sequencing centers around the world were established; meanwhile, Sanger's chain-termination sequencing method was still dominant in the field, although significant advances were made in increasing throughput by moving from gel to capillary electrophoretic methods. In these 15 years, researchers

started sequencing genomes of many species, ranging from viruses, protists, fungi, and plants, to animals (Primorose and Twyman, 2006), e.g. *Haemophilus influenzae*, the first genome of a bacterium, was sequenced in 1995; the genome of *Saccharomyces cerevisiae*, the first eukaryotic genome to be sequenced, was completed in 1996; *Caenorhabditis elegans*, the first genome of a multicellular organism, completed in 1998; *Arabidopsis thaliana*, the first plant genome, completed in 2000; *Homo sapiens*, the first mammalian genome, completed in 2001; *Oryza sativa* (rice), the first crop genome, completed in 2002; *Mus musculus* (mouse), a widely used mammalian model organism, completed in 2002/3; and *Pan troglodytes* (chimpanzee), the closest relative to humans, completed in 2005. One event that I cannot overemphasize is the Human Genome Project (HGP), which started in 1990 and ended in 2003. This 13-year project, for the first time, sequenced the genome of humankind, and mapped the human genome to many other genomes (Chial, 2008). As an international collaboration of eighteen countries, this project not only raised sequencing technology to a global scope, but also built a foundation for much of the human health research of today. Due to the global impact of the HGP, many centers and institutes for genome sequencing and genome study were established at that time, such as The Institute for Genome Research (TIGR), now known as the J. Craig Venter Institute, in the USA, the Sanger Center in the United Kingdom, and RIKEN in Japan, etc. By the end of 1998, sequencing speed had reached 500,000 to 1 million bases per day using the ABI Prism 3700 multiple capillary sequencer (Metzker, 2008). For data storage at that time, the genomic sequences were kept in relational database management systems. Researchers could access sequences from websites (and ftp). The data transfer was no longer accomplished by mailing magnetic tapes, but done using the internet. For bioinformatics algorithms, because most of the genomes were sequenced by the shotgun strategy, genome assembly algorithms were developed and widely applied (Kulski, 2016; W. Myers Jr, 2016). Meanwhile, the first technologies for measuring gene expression (i.e., RNA abundance), the high density oligonucleotide arrays (mi-

croarrays), appeared in around 1996. Microarray technology actually dominated the field of gene expression analysis for about a decade (Southern, 2001).

The third stage is from 2005 to 2017. During this period, a number of new sequencing methods were widely commercialized and replaced Sanger's method; RNA sequencing (RNA-seq) has become the primary approach for studying transcriptomes of biological organisms. The new sequencing methods introduced beginning in 2005 are the so-called "second generation sequencing" or "next generation sequencing" (NGS). The most commonly used platforms for second generation sequencing were Roche 454 pyrosequencing (discontinued in 2013), Illumina, SOLiD, DNA nanoball sequencing, and Ion torrent (Kulski, 2016; Heather and Chain, 2016; Liu et al., 2014). Instead of cloning individual DNA fragments via foreign host cells as in Sanger's method, the new methods have much easier and quicker library preparation procedures using adapters, barcodes, primers, new PCR methods, and novel sequencing mechanisms; high throughput sequencing can be achieved by amplifying DNA clusters on a solid substrate with readout by charge-coupled device (CCD) cameras, producing the sequences of hundreds of thousands (Roche 454) to hundreds of millions of fragments simultaneously. Though each platform has its own pros and cons, the cost of second generation sequencing, (e.g. Illumina HiSeq), is much lower – about 300 thousand times cheaper than Sanger's method – per million bases (Liu et al., 2014; Muir et al., 2016). The reduced cost of sequencing immediately led to a second wave of increasing amounts of data. To adapt to the drastically increased data scale, remote data analysis by cloud computing has become available, and is now starting to be widely used (Dai et al., 2012; O'Driscoll et al., 2013). Databases have also become more diverse and specialized than before (Zou et al., 2015). Moreover, new bioinformatics algorithms, such as read alignment algorithms, read assembly algorithms, data compression algorithms, and data mining algorithms for extracting useful information from the omics data have been developed (Berger et al., 2013).

Very recently, other leading-edge sequencing technologies, such as PacBio, Helicos, Nanopore, and electron microscopy sequencing, have emerged. These sequencing

methods are called "third generation sequencing" to differentiate them from the "second generation sequencing" (Liu et al., 2014; Heather and Chain, 2016). These new methods try to sequence DNA molecules at a single molecule level, without complex fragmentation, ligation, or amplification steps. In theory, such approaches should be less technically biased and able to produce longer read lengths than second generation sequencing. However, the problems with these approaches, such as high sequencing error rate and low read coverage, etc., are still present, and need to be solved in the future.

Looking back, over a short period of 40 years, sequencing technology has gone through from zero to one, and one to more. The overall trend is that the sequencing has become faster, cheaper, and more precise at determining and measuring the expression of real transcriptomes of organisms. As a result, the amount and scale of sequencing data will continue to increase; and bioinformatics algorithms for analyzing big genomic data are urgently needed.

My Ph.D. study focuses on developing and evaluating bioinformatics algorithms for RNA-seq data analysis. In the next section, I will briefly introduce RNA-seq and its applications.

1.2 Significance of RNA Sequencing and its applications

The NGS technology has a variety of applications depending on the goal of the research. Some common applications include whole genome sequencing (WGS), whole exome sequencing (WES), targeted sequencing of specific genes, chromatin immunoprecipitation sequencing (ChIP-seq), and RNA-seq. Among the above sequencing methods, RNA-seq is a state-of-the-art technique that takes advantage of the high-throughput of NGS for studying dynamic and tissue-specific transcriptomes. Another method, which was widely used in the 1990s to study transcriptomics, is the microarray technique, based on oligonucleotide hybridization. However, since around 2005, as the cost of per base of RNA-seq and the sequencing quality have continuously

improved, RNA-seq has taken over from microarray approaches, becoming the first choice for quantitatively assessing gene expressions in biological systems. Extensive reviews have been published on how RNA-seq has revolutionized our view of biological processes and pushed forward the biomedical field (Han et al., 2015). Most recently, due to the decreasing cost of RNA-seq and its powerful ability for providing fast and accurate quantification of RNA levels, the RNA-seq technique has been standardized and translated to the medicine and healthcare fields in the real world.

1.2.1 RNA-seq in scientific explorations

RNA-seq has completely changed our view of the landscape of the human transcriptome by discovering new transcripts, identifying novel mutations, quantifying transcripts at the isoform level, and enabling comprehensive differential expression and functional analysis. About 10 years ago, people thought that only 3% of human genome was transcribed as messenger RNA (mRNA), based on mapping of the expressed sequence tags (ESTs) back to the genome. Until very recently, when substantial RNA-seq data firmly established the reality of pervasive transcription – more than 85% of the human genome is transcribed, although only 3% is eventually translated to proteins (Hangauer et al., 2013). These non-coding transcripts have been identified as belonging to previously unknown classes of RNAs, such as long non-coding RNA (lncRNA), microRNA (miRNA), small interfering RNA (siRNA), and enhancer RNA (eRNA) (Iyer et al., 2015; Guo et al., 2015; Kim et al., 2010). These new discoveries have dramatically changed our understanding of the human genome and enriched our knowledge of gene regulation.

In addition to the "junk" regions of human genome, RNA-seq has also led to new discoveries in coding sequence regions. Using RNA-seq, many genes have been found to have alternative isoforms (an average of 3.4 alternative isoforms per gene according to GENCODE human annotation version 27[1]) (Li et al., 2014), although

[1]https://www.gencodegenes.org/stats/archive.html

most genes have only one dominant isoform in most conditions (Trapnell et al., 2010; Gonzàlez-Porta et al., 2013). Multiple isoforms can result from either the switch of transcriptional start sites (TSS) on the genome, or from the complex alternative splicing process during the transcript maturation. Many studies have found that isoforms of the same gene can have the opposite functions by regulating the same complex or pathway (Li et al., 2014; Tone et al., 2001). Aberrant splicing events, and fusion genes with abnormal exon-intron structures, or copy number variations, also have been found to be related to diseases, especially in cancers (Fackenthal and Godley, 2008; Eswaran et al., 2013). These discoveries have led to new hypotheses for human transcriptome and transcriptional regulation.

The single nucleotide resolution of RNA-seq enables highly sensitive and accurate quantification of transcripts. By comparing the transcriptomes of different tissues, or at different times, many interesting mechanisms have been unraveled at the transcriptome level.

1.2.2 RNA-seq in translational medicine

As the affordability and reliability of RNA-seq have improved, many research groups and companies have started to translate RNA-seq technology into healthcare. One promising direction is personalized medicine, in which a therapeutic plan is made based on the genomic information of each patient (Rabbani et al., 2016). Due to genomic variations in the human population, the conventional strategy of "same disease – same drug" has been gradually replaced by a concept of personalized medicine, or a "same disease – different drugs" strategy. One successful case, which has already entered the clinical trial stage, is the design of personalized cancer vaccines for treating melanoma[2] (Ott et al., 2017; Sahin et al., 2017). RNA-Seq and exome sequencing were used to identify tumor-specific mutations in each patient, and the mutant proteins that are most likely to trigger immune system responses to invading cancer cells, were

[2]doi:10.1038/nature.2017.22249

selected to generate a mixture of vaccines. In two independent studies (Ott et al., 2017; Sahin et al., 2017), four out of six and eight out of thirteen patients were tumor free one year after receiving the vaccine treatment.

Another direction that RNA-seq has been translated to is precision medicine, in which clinical diagnosis, prognosis and medical screening are made by precisely sequencing and measuring the biomarkers of individual patients (Byron et al., 2016). Based on the genomic information, healthcare providers can make better plans for disease treatment and lifestyle adjustment. For example, FoundationOne Heme, which has been approved by the US Food and Drug Administration (FDA), employs RNA-seq technology for detecting oncogenic fusions in hematologic malignancies and sarcomas. FoundationOne Heme provides useful information to physicians for diagnosing hematological cancers. An emerging area is the measurement of the levels of extracellular RNA (exRNA) by RNA-seq for diagnosing a disease or monitoring the process of a disease (Byron et al., 2016). It is appealing because exRNAs exist in biofluids, which can be non-invasively acquired from patients. The US National Center for Advancing Translational Sciences (NCATS) initiated an exRNA communication consortium in 2015 for developing diagnostic tools.

Aside from the above translational studies and applications using RNA-seq, many companies have been launched that seek to translate the sequencing technology into healthcare[3]. Some companies focus on providing direct healthcare services, including cancer gene profiling, pharmacogenomic toxicity analysis, and therapy counseling. For example, Veritas genetics provides whole genome sequencing, and links the disease risk analysis using smartphone apps; IBM Watson for genomics, announced in January 2017 that it will be integrated into Illumina's TruSight Tumor 170 tool for speeding up drug recommendations for cancer patients; Rosetta genomics employs a NGS-platform for providing genome-wide oncogenomic tumor profiling and pharmacogenomic toxicity analysis specifically for lung cancer patients. Some companies emphasize integrating machine learning methods into data analysis. For example,

[3]http://medicalfuturist.com/top-companies-genomics/

Verge genomics, which uses machine learning and genomics data for providing new treatment options specifically for neurodegenerative diseases; Verily Life Science, a healthcare company subsidiary of Google, aims to use mature google learning algorithms for solving healthcare issues.

1.3 General workflow of RNA-seq data analysis and related bioinformatics algorithms

Currently, the most typical RNA-seq platform provides sequencing of paired-end reads with the length of about 100 – 250 base pair (bp), and the numbers of reads ranging from 5 to 60 million per sample, depending on the goal of the sequencing. However, because of the broad applications of RNA-seq, as I introduced in Section 1.2, there is not a universal workflow for all RNA-seq data analyses. Considering the majority of studies, which aim to systematically interpret biological functions using RNA-seq data, I outline a general workflow of RNA-seq data analysis in Figure 1.1, which also highlights the focus of this dissertation.

Figure 1.1 (A) shows the major steps of RNA-seq data analysis including the construction of a transcriptome (assembled transcripts) from RNA-seq reads, the quantification of transcripts, and the interpretation of biological functions (e.g. using networks) based on the experimental design. Following such a workflow, the massive information stored as millions of short-read sequences will be transferred into organized biological networks, which can be visualized and analyzed by domain experts.

After sequencing, RNA-seq raw reads will first be pre-processed by removing low-quality reads (e.g. the reads with a total base quality in a window lower than a threshold) and artifacts (e.g. adaptors, contaminant DNAs and PCR duplicates) (Martin and Wang, 2011). Then the preprocessed reads will be used for constructing transcripts depending on the availability of a reference genome/transcriptome. Figure 1.1 (B) shows a detailed workflow of RNA-seq data analysis. If the reference genome or transcriptome are highly reliable, we can directly use the annotations for quanti-

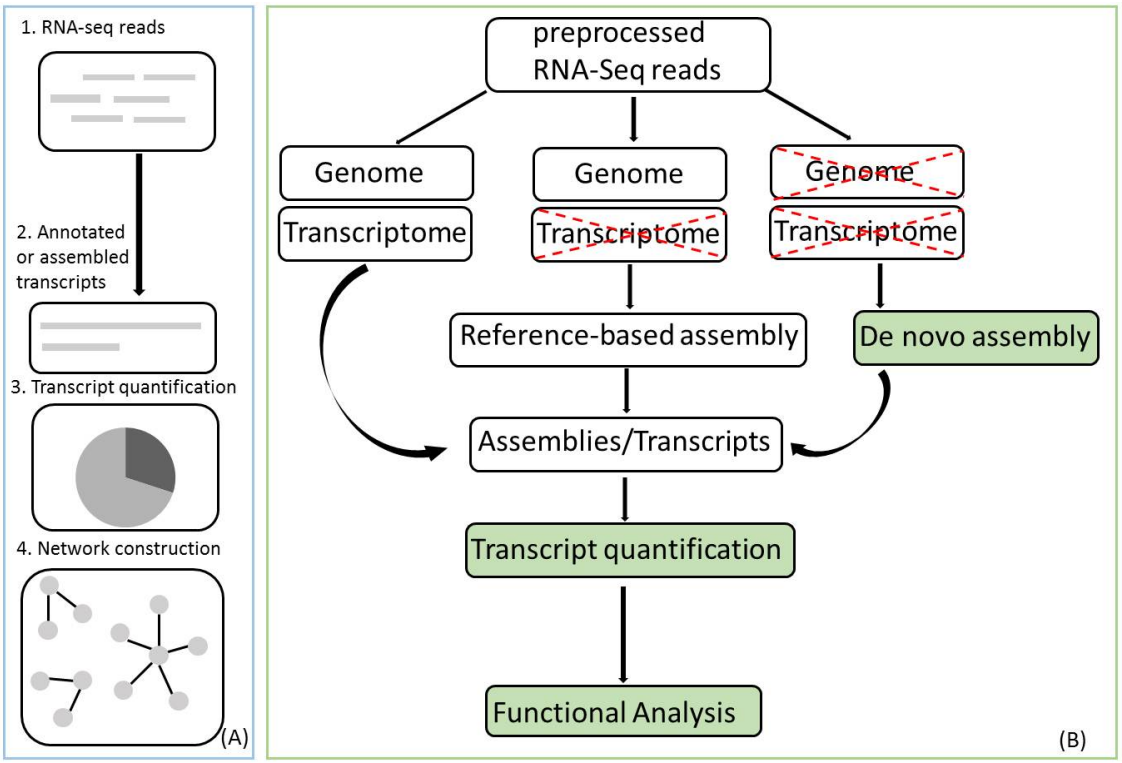


Fig. 1.1.: A general workflow of RNA-seq data analysis. (A) is a cartoon flowchart that shows how the information flows from RNA-seq short reads to transcripts, and then to gene expression and functional networks. (B) shows a detailed workflow based on the availability of reference genome and transcriptome (red crosses mark information that is unavailable). My three works (chapter 2, 3, and 4) focus on the parts with green color.

fying transcripts; if only the reference genome is available, and the transcriptome is unknown or unreliable, we can rely on a reference-based assembly method to obtain assemblies as transcripts; if neither the reference genome nor the transcriptome is available, *de novo* assembly becomes the only choice for identifying transcripts from RNA-seq reads. Then, based on the known or assembled transcripts (reference-based assemblies or *de novo* assemblies), we can quantify the transcripts by mapping pre-processed RNA-seq short reads back to the transcripts/assemblies and estimating the count of reads that are potentially coming from each transcript/assembly. Based on the expression levels of transcripts, functional analysis is performed for illustrating important genes/gene clusters/pathways associated with a phenotype or a disease.

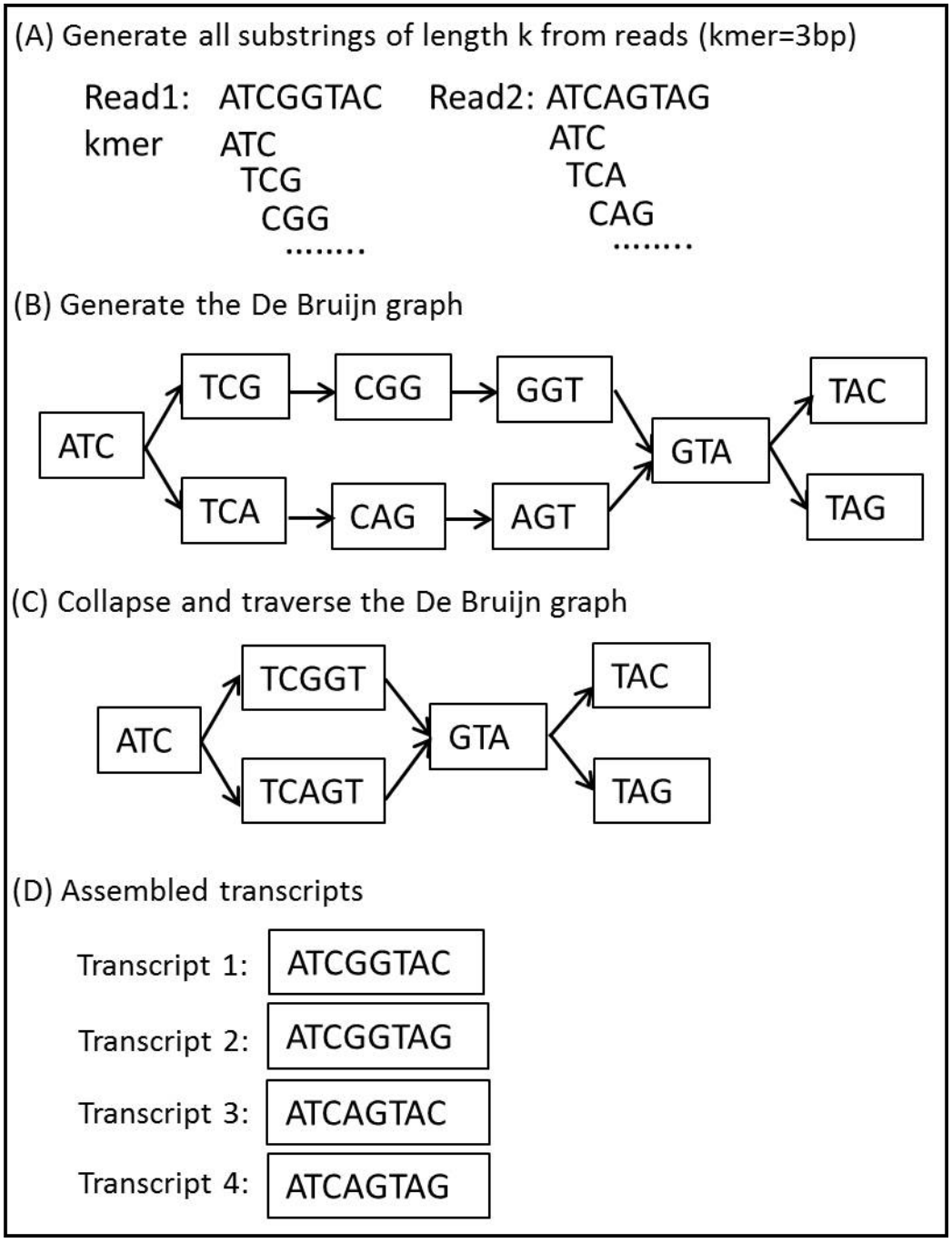


Fig. 1.2.: Overview of the RNA-seq *de novo* transcriptome assembly strategy. (A) shows all substrings of length k (kmers) are generated from every single read. (B) shows the generation of a De Bruijn graph using kmers. Each node represents a unique kmer and each arrow represents the overlap of k-1 bases between two kmers. (C) shows that the non-branching chains in a De Bruijn graph can be collapsed into a single node. (D) shows all the possible transcripts assembled by traversing the paths in the De Bruijn graph.

Because Chapter 2 focuses on RNA-seq *de novo* assembly, I will briefly introduce some basic concepts and nomenclatures used in RNA-seq *de novo* assembly. Figure 1.2 shows a typical strategy of RNA-seq *de novo* assembly with four steps. In the first step (A), all the RNA-seq reads will be chopped into substrings of length k (kmers) by shifting one base at each time; in the second step (B), De Bruijn graphs will be constructed using all or the most frequent kmers; Each node represents a unique kmer and each arrow represents the overlap of k-1 bases between two kmers; in the third step (C), De Bruijn graphs will be simplified by collapsing non-branching chains of nodes and trimming off the branches with low weights (e.g. less frequent and low-quality kmers); in the last step (D), transcripts will be generated by traversing paths in De Bruijn graphs. Some algorithms call these resulting transcripts as contigs, and connect contigs which are supported by read evidence for generating transcripts.

Transcriptome assembly is potentially more complicated than genome assembly. In genome assembly, the sequencing depth is presumably the same at each base if not considering the technical bias and sample decay, because there are no copy variations of DNA. In contrast, transcriptome assembly methods have to consider the variations of expression levels of transcripts in a sample, so we cannot assume the sequencing depths are even at all bases; the PCR amplification step may even enlarge these variations; also, low-expressed transcripts often have too few reads, which are hard to be assembled. Therefore, transcriptome assembly and its evaluation methods are still challenging questions in RNA-seq data analysis.

Lastly, for each part of the flowchart in Figure 1.1, I summarize the properties of state-of-the-art bioinformatics algorithms and the methods related to this dissertation in Table 1.1, including read preprocessing methods, read alignment methods, *de novo* assembly methods, *de novo* assembly evaluation methods, transcript quantification methods, differential analysis methods, and functional analysis methods.

Table 1.1.: Summary of commonly used bioinformatics programs and the methods related to this dissertation for RNA-seq data analysis.

Steps	**Methods**	**Properties and comments**
Read pre-processing	FastQC	• A sequencing base quality evaluator • Provides visualization of read statistics and qualities
	Trimmomatic	• Removes adapters • Trims low qualilty bases for RNA-seq reads
Read alignment (mapping)	BWA	• Maps low-divergent sequences against a large reference genome based on Burrows-Wheeler transform • BWA-backtrack is designed for Illumina sequence reads up to 100bp • BWA-SW and BWA-MEM mapped reads from 70bp to 1Mbp
	Bowtie	• A fast RNA-seq short read (< 50 bp) aligner • Aligns reads to a reference genome indexed by Burrows-Wheeler transform • Only non-gapped and end-to-end alignment
	Bowtie2	• A fast read aligner supporting longer read length ($< 1,000$ bp) • Both gapped and local alignment
	Tophat	• An RNA-seq read aligner built on Bowtie • Detects splice junctions

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	STAR	• An RNA-seq read aligner using maximum mappable prefix search • Handles both RNA-seq paired-end short reads and long single reads generated by the third generation sequencing technologies • Detects both splice junctions and chimeric transcripts
	GMAP	• Initially designed for cDNA alignment to reference genome (splice-aware) • Applicable to single RNA-seq short reads (< 75 bp) with specific parameter settings as suggested in manual[4] • Applicable to PacBio long reads with optimized parameters as recommended in tutorial[5]
De novo assembly	SOAPdenovo-Trans	• An RNA-seq short read assembler based on SOAPdenovo
	Trans-ABySS	• An RNA-seq short read assembler based on ABySS • TransABySS-merge can merge multiple *de novo* assemblies with different kmers • No gene-isoform relation preserved

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	IDBA-Tran	• Employs a progressive probabilistic approach to iteratively remove erroneous kmers in de Bruijn graph construction, instead of using a global threshold • Designed for better assembly of low-expression transcripts • No gene-isoform relationship preserved
	Trinity	• Both genome-guided and *de novo* transcriptome assembly • Three steps: searching paths in kmer graphs to generate linear contigs; clustering the contigs and constructing individual De Bruijn graph for each cluster; tracing the paths and reporting transcripts • Kmer size only varies from 25 bp to 32 bp
	Oases	• An RNA-seq short read assembler based on the Velvet assembler • Merges assemblies made with multiple kmers
De novo assembly evaluation	rnaQUAST	• A metric-based RNA-seq assembly evaluator • Only provides reference-based evaluation • Allows to *de novo* assemble transcripts using several thrid-party tools, such as BUSCO and GeneMarkS-T

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	DETONATE	• A model-based RNA-seq assembly evaluator • Module REF-EVAL is used for reference-based evaluation • Module RSEM-EVAL is used for *de novo* assembly evaluation without reference
	TransRate	• A model-based *de novo* asssembly evaluator • Provides contig scores for all contigs and an assembly score for the whole set of assemblies • Removes low quality assemblies by optimizing an empirical function
Transcript quantification	Cufflinks	• Provides reference based RNA-seq assembly for detecting novel transcripts • Estimates expression levels of genes/transcripts based on a given reference or a self-assemblied transcriptome • Cuffdiff used for differential expression analysis of genes/transcripts
	MISO	• No assembly function • Exon-centric model estimates the expression levels of exons • Isoform-centric model estimates the expression levels of spliced isoforms • Bayes factor (BF) evaluates the significance of differentially expressed (DE) isoforms

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	RSEM	• Estimates expressions of genes/transcripts with or without a reference genome • Employs EBSeq to evaluate DE genes/transcripts • Provides visualization tools • An RNA-seq read simulator if given the expression levels of trancripts
	BitSeq	• A 2-step model for estimating gene/transcript expression levels first in each replicate, and then the mean expressions in each condition • The probability of log-ratio of expression in condition 2 over expression in condition 1 is used to evaluate transcript DE levels
	DEIsoM	• A one step integrated model for estimating gene/transcript expression levels in a whole condition which is comprised of multiple biological replicates • No loss of any sources of variations from either ambiguous mapping or biological replication • Kullback-Leibler (KL) divergence is used to evaluate transcript DE levels between two conditions
Differential analysis	DESeq2	• Identifies DE genes based on RNA-seq count data • Models the count data as a negative binonial distribution and a shrinkage estimator for distribution variance

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	EdgeR	• Examines differential expression of replicated count data using an overdispersed Poisson model accounting for both biological and technical variability • Empirical Bayes methods are used to moderate the degree of overdispersion across transcripts • Applicable to other data (e.g. proteome peptide count data)
	EBSeq	• Identifies not only DE genes but also DE isoforms by considering the uncertainty from ambiguous read mapping using an empirical Bayesian method • Evaluates DE between two or more conditions
Pathway analysis	GSEA	• Determines whether an a pre-defined set of genes shows statistically significant, concordant differences between two biological states • The pre-defined set of genes can be from the Molecular Signature Database (MSigDb) or from users' gene set files.
	SeqGSEA	• Improved from GSEA to adapt to RNA-seq data with fewer biological replicates • Incorporates the absolute gene statistic in one-tailed GSEA to lower the false positive rate in the GSEA gene permutation method • Uses negative binomial distribution to model read count data

Continued on next page

Table 1.1 – continued from previous page

Steps	Methods	Properties and comments
	NaNOS	• Jointly selects both genes and pathways associated with a phenotype • Incorporates pathway structures encoded in the database • Efficient inference algorithm

In this dissertation, I will focus on three parts of the flowchart (Figure 1.1), shaded in green, including the assessment of *de novo* assembly methods, the modeling of transcript quantification and DE isoform identification, and the association of genes and pathways with phenotypes using high throughput genomic data. I will discuss the challenges and potential solutions in the next section.

1.4 Three critial issues in RNA-Seq analysis (Outline of the dissertation)

Despite many successful applications of RNA-seq in both scientific explorations and translational medicine, as I introduced in Section 1.2, multiple challenges in RNA-seq data analysis are still present. The central goal of RNA-seq data analysis is to maximally use the information stored in millions of RNA-seq reads for reconstructing the transcriptome and understanding the biological functions associated with the specific spatiotemporal phenotype.

Many bioinformatics algorithms and pipelines have been developed to accomplish this goal. However, one critical challenge is the prediction uncertainty due to the short read length and the low sampling rate of weakly expressed transcripts. Both conditions lead to ambiguities in read mapping, transcript assembly, transcript quantification, and even downstream analyses, when dealing with RNA-seq data. A central idea behind reducing the uncertainty is to incorporate additional informa-

[4]Align Pacbio long reads using GMAP: `https://github.com/PacificBiosciences/cDNA_primer/wiki/Aligner-tutorial:-GMAP,-STAR,-BLAT,-and-BLASR`

[5]Align RNA-seq short reads using GMAP: `https://github.com/juliangehring/GMAP-GSNAP`

tion into bioinformatic models for consolidating the results. This extra information could be from other sequencing platforms, from technical or biological replicates, from databases containing expert knowledge, etc.

This dissertation will cover three critical issues in RNA-seq modeling and model assessments, and in each case, the problem is solved by incorporating additional information. Chapter 2 discusses how to assess *de novo* assembly methods and *de novo* assembly evaluation methods using a third generation sequencing technology; Chapter 3 discusses how to improve the transcript quantification and DE isoform identification by capturing the shared information from biological replicates; Chapter 4 discusses a joint pathway and gene selection model that incorporates pathway structures from an expert database.

1.5 References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410.

Berger, B., Peng, J., and Singh, M. (2013). Computational solutions for omics data. *Nature Reviews Genetics*, 14(5):333 – 346.

Byron, S. A., Van Keuren-Jensen, K. R., Engelthaler, D. M., Carpten, J. D., and Craig, D. W. (2016). Translating RNA sequencing into clinical diagnostics: opportunities and challenges. *Nature Reviews Genetics*, 17:257 EP –.

Chial, H. (2008). DNA sequencing technologies key to the human genome project. *Nature Education*, 1(1):219.

Crick, F. (1958). On protein synthesis. In Sanders, F., editor, *Symposia of the Society for Experimental Biology, Number XII: The Biological Replication of Macromolecules.*, pages 138 –163. Cambridge University Press.

Dai, L., Gao, X., Guo, Y., Xiao, J., and Zhang, Z. (2012). Bioinformatics clouds for big data manipulation. *Biology Direct*, 7:43 – 43.

Eswaran, J., Horvath, A., Godbole, S., Reddy, S. D., Mudvari, P., Ohshiro, K., Cyanam, D., Nair, S., Fuqua, S. A. W., Polyak, K., Florea, L. D., and Kumar, R. (2013). RNA sequencing of cancer reveals novel splicing alterations. *Scientific Reports*, 3:1689.

Fackenthal, J. D. and Godley, L. A. (2008). Aberrant RNA splicing and its functional consequences in cancer cells. *Disease Models & Mechanisms*, 1(1):37–42.

Gonzàlez-Porta, M., Frankish, A., Rung, J., Harrow, J., and Brazma, A. (2013). Transcriptome analysis of human tissues and cell lines reveals one dominant transcript per gene. *Genome Biology*, 14(7):R70.

Guo, C., Li, L., Wang, X., and Liang, C. (2015). Alterations in SiRNA and MiRNA expression profiles detected by deep sequencing of transgenic rice with SiRNA-mediated viral resistance. *PLOS ONE*, 10(1):e0116175.

Han, Y., Gao, S., Muegge, K., Zhang, W., and Zhou, B. (2015). Advanced applications of RNA sequencing and challenges. *Bioinformatics and Biology Insights*, 9(Suppl 1):29–46.

Hangauer, M. J., Vaughn, I. W., and McManus, M. T. (2013). Pervasive transcription of the human genome produces thousands of previously unidentified long intergenic noncoding RNAs. *PLOS Genetics*, 9(6):1–13.

Heather, J. M. and Chain, B. (2016). The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1 – 8.

Holley, R. W., Apgar, J., Everett, G. A., Madison, J. T., Marquisee, M., Merrill, S. H., Penswick, J. R., and Zamir, A. (1965). Structure of a ribonucleic acid. *Science*, 147(3664):1462–1465.

Iyer, M. K., Niknafs, Y. S., Malik, R., Singhal, U., Sahu, A., Hosono, Y., Barrette, T. R., Prensner, J. R., Evans, J. R., Zhao, S., Poliakov, A., Cao, X., Dhanasekaran, S. M., Wu, Y.-M., Robinson, D. R., Beer, D. G., Feng, F. Y., Iyer, H. K., and Chinnaiyan, A. M. (2015). The landscape of long noncoding RNAs in the human transcriptome. *Nature Genetics*, 47:199–208.

Kim, T.-K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., Harmin, D. A., Laptewicz, M., Barbara-Haley, K., Kuersten, S., Markenscoff-Papadimitriou, E., Kuhl, D., Bito, H., Worley, P. F., Kreiman, G., and Greenberg, M. E. (2010). Widespread transcription at neuronal activity-regulated enhancers. *Nature*, 465(7295):182–187.

Kulski, J. K. (2016). Next-generation sequencing - an overview of the history, tools, and "omic" applications. In Kulski, J. K., editor, *Next Generation Sequencing - Advances, Applications and Challenges*, chapter 01. InTech, Rijeka.

Li, H.-D., Menon, R., Omenn, G. S., and Guan, Y. (2014). The emerging era of genomic data integration for analyzing splice isoform function. *Trends in genetics : TIG*, 30(8):340 – 347.

Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., Lin, D., Lu, L., and Law, M. (2014). Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012.

Martin, J. A. and Wang, Z. (2011). Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682.

Maxam, A. M. and Gilbert, W. (1977). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(2):560–564.

Metzker, M. L. (2008). Sequencing technologies – the next generation. *Nature Review Genetics*, 9(11).

Muir, P., Li, S., Lou, S., Wang, D., Spakowicz, D. J., Salichos, L., Zhang, J., Weinstock, G. M., Isaacs, F., Rozowsky, J., and Gerstein, M. (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology*, 17(1):53.

O'Driscoll, A., Daugelaite, J., and Sleator, R. D. (2013). Big data, hadoop and cloud computing in genomics. *Journal of Biomedical Informatics*, 46(5):774 – 781.

Ott, P. A., Hu, Z., Keskin, D. B., Shukla, S. A., Sun, J., Bozym, D. J., Zhang, W., Luoma, A., Giobbie-Hurder, A., Peter, L., Chen, C., Olive, O., Carter, T. A., Li, S., Lieb, D. J., Eisenhaure, T., Gjini, E., Stevens, J., Lane, W. J., Javeri, I., Nellaiappan, K., Salazar, A. M., Daley, H., Seaman, M., Buchbinder, E. I., Yoon, C. H., Harden, M., Lennon, N., Gabriel, S., Rodig, S. J., Barouch, D. H., Aster, J. C., Getz, G., Wucherpfennig, K., Neuberg, D., Ritz, J., Lander, E. S., Fritsch, E. F., Hacohen, N., and Wu, C. J. (2017). An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, 547(7662):217–221.

Primorose, S. B. and Twyman, R. M. (2006). *Principles of gene manipulation and genomics*. Blackwell Publishing.

Rabbani, B., Nakaoka, H., Akhondzadeh, S., Tekin, M., and Mahdieh, N. (2016). Next generation sequencing: implications in personalized medicine and pharmacogenomics. *Molecular Biosystems.*, 12:1818–1830.

Sahin, U., Derhovanessian, E., Miller, M., Kloke, B.-P., Simon, P., Löwer, M., Bukur, V., Tadmor, A. D., Luxemburger, U., Schrörs, B., Omokoko, T., Vormehr, M., Albrecht, C., Paruzynski, A., Kuhn, A. N., Buck, J., Heesch, S., Schreeb, K. H., Müller, F., Ortseifer, I., Vogler, I., Godehardt, E., Attig, S., Rae, R., Breitkreuz, A., Tolliver, C., Suchan, M., Martic, G., Hohberger, A., Sorn, P., Diekmann, J., Ciesla, J., Waksmann, O., Brück, A.-K., Witt, M., Zillgen, M., Rothermel, A., Kasemann, B., Langer, D., Bolte, S., Diken, M., Kreiter, S., Nemecek, R., Gebhardt, C., Grabbe, S., Höller, C., Utikal, J., Huber, C., Loquai, C., and Türeci, Ö. (2017). RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature*, 547(7662):222–226.

Saiki, R., Scharf, S., Faloona, F., Mullis, K., Horn, G., Erlich, H., and Arnheim, N. (1985). Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science*, 230:1350–1354.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12):5463–5467.

Southern, E. (2001). *DNA Microarrays: History and Overview*, volume 170.

Tone, M., Tone, Y., Fairchild, P. J., Wykes, M., and Waldmann, H. (2001). Regulation of CD40 function by its isoforms generated through alternative splicing. *Proceedings of the National Academy of Sciences*, 98(4):1751–1756.

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*, 28:511–515.

W. Myers Jr, E. (2016). A history of DNA sequence assembly. 58.

Watson, J. and Crick, F. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738.

Zou, D., Ma, L., Yu, J., and Zhang, Z. (2015). Biological databases for human research. *Genomics, Proteomics & Bioinformatics*, 13(1):55–63.

2. CHAPTER 2. ASSESSMENTS ON RNA-SEQ *DE NOVO* ASSEMBLY BY PACBIO LONG READ SEQUENCING

2.1 Abstract

RNA-Seq *de novo* assembly is an important method to generate transcriptomes for non-model organisms before any downstream analysis. Given many great *de novo* assembly methods developed by now, one critical issue is that there is no consensus on the evaluation of *de novo* assembly methods yet. Therefore, to set up a benchmark for evaluating the quality of *de novo* assemblies is very critical. Addressing this challenge will help us deepen the insights on the properties of different *de novo* assemblers and their evaluation methods, and provide hints on choosing the best assembly sets as transcriptomes of non-model organisms for the further functional analysis.

In this article, we generate a "real time" transcriptome using PacBio long reads as a benchmark for evaluating five *de novo* assemblers and two model-based *de novo* assembly evaluation methods. By comparing the *de novo* assemblies generated by RNA-Seq short reads with the "real time" transcriptome from the same biological sample, we find that Trinity is best at the completeness by generating more assemblies than the alternative assemblers, but less continuous and having more misassemblies; Oases is best at the continuity and specificity, but less complete; The performance of SOAPdenovo-Trans, Trans-ABySS and IDBA-Tran are in between of five assemblers. For evaluation methods, DETONATE leverages multiple aspects of the assembly set and ranks the assembly set with an average performance as the best, meanwhile the contig score can serve as a good metric to select assemblies with high completeness, specificity, continuity but not sensitive to misassemblies; TransRate contig score is useful to remove misassemblies, and TransRate can optimize the assembly set by

filtering out the assemblies with low contig scores, yet often the assemblies in the optimal set is too few to be used as a transcriptome.

2.2 Introduction

With the rapid development of sequencing technology, transcriptome assembly by RNA-Seq short reads has become increasingly important in many fields, such as plant science (Brereton et al., 2016; Ranjan et al., 2014), animal science (Moreton et al., 2014) and disease related studies (Mittal and McDonald, 2017; Mamrot et al., 2017; Wang et al., 2015). Current transcriptome assembly methods mainly fall into three categories: reference-based assembly, *de novo* assembly and a hybrid assembly that merges the above two (Martin and Wang, 2011). For non-model organisms with no available reference genome or transcriptome, *de novo* assembly becomes the only choice to determine the transcriptome before any downstream analysis. Many *de novo* assembly methods have been developed, however, there is no consensus on how to evaluate these methods. Therefore, establishing a reliable benchmark for understanding the property of each *de novo* assembly tool has become a critical issue (Moreton et al., 2015).

Recently, powerful tools, such as Trinity (Grabherr et al., 2011), Oases (Schulz et al., 2012), SOAPdenovo-Trans (Xie et al., 2014), Trans-ABySS (Robertson et al., 2010), and IDBA-Tran (Peng et al., 2013), have been developed for *de novo* assembly of transcriptomes from RNA-Seq short reads. From the data perspective, when evaluating *de novo* assembly methods, researchers can either simulate RNA-Seq short reads base on a known reference genome or transcriptome (O'Neil and Emrich, 2013), or use real RNA-Seq datasets and evaluate the performance of assemblers by comparing the assemblies with the reference transcriptome or the transcriptome of a related species (Honaas et al., 2016; Wang and Gribskov, 2017). In the first case, even though it is convenient to control the properties of simulated data, such as the expression levels of transcripts, the sequencing error rate , the sequencing depth and etc, the

simulated data cannot completely represent the real data. In the latter case, the evaluation heavily relies on the quality of the reference transcriptome. Nevertheless, the expressed transcripts may even vary among biological replicates or different tissues (Melé et al., 2015). The presence of assemblies that are missed in the reference transcriptome does not necessarily mean that those assemblies are misassemblies. The novel transcript could be representing mutated or fusion transcripts that hasn't been annotated in the reference. Similarly, the absence of assemblies compared with the reference transcriptome does not necessarily indicate incompleteness of an assembly. It could be that the transcripts that are not expressed in a particular sample. Even though the reference transcriptome is well annotated for a species, e.g., *Homo sapiens*, the reference transcriptomes still vary between different instititional sites (Ensembl and RefSeq) and versions still exist, which complicate the issue from another perspective (Section 2.4.1).

Two types of methods are used for assessing *de novo* assemblies: metrics-based methods and model-based methods. However, without a reference transcriptome, metrics-based methods can only provide an empirical description rather than an assessment of the quality of the assemblies, such as the total number and the length information of assemblies. With a reference transcriptome, the metrics-based methods have the ability to comprehensively evaluate the accuracy, completeness, continuity, and misassembly rate of the assemblies. However, this analysis is based on the assumption that the reference transcriptome is complete and reliable (Martin and Wang, 2011; Bushmanova et al., 2016). Model-based methods, such as DETONATE (Li et al., 2014a) and TransRate (Smith-Unna et al., 2016), focus on how well the assemblies can be explained by the read evidence. However, each model-based method has its own definition of the "optimal" assembly, which is inconsistent among different models. Furthermore, model-based methods themselves are hard to evaluate if we do not have a reliable reference transcriptome in hand.

In this study, we utilize the PacBio long read sequencing technology to generate a "real time" transcriptome as a benchmark for assessing (1) the properties of five com-

monly used *de novo* assembly methods and (2) the effectiveness of two model-based evaluation methods. By comparing the assemblies from the short reads to the "real time" transcriptome from PacBio long reads of the same biological sample, we eliminate the biological uncertainties to a large extent. We conclude that Trinity is best at completeness, but assembled transcripts are less continuous and have more misassemblies than the alternative methods; Oases is best at continuity and specificity (we followed the nomenclature used in rnaQUAST (Bushmanova et al., 2016); the specificity refers to the percentage of the assemblies that can be well mapped back to the annotated transcripts), but less complete; The performance of SOAPdenovo-Trans, Trans-ABySS and IDBA-Tran are in between. For the model-based evaluation methods, DETONATE ranks the method with all aspects having the average performance as the best, while TransRate doesn't penalize any downsides but only encourages the good aspects of the assemblies; The contig scores of DETONATE can help select the assemblies with high completeness, specificity and continuity but not a low misassembly rate, while the contig scores of TransRate are helpful in removing misassemblies.

2.3 Methods

2.3.1 RNA-Seq datasets

The datasets we used were from the Sequencing Quality Control (SEQC)/MAQC-III Consortium, which sequenced a human brain sample by multiple platforms, including MiSeq short read sequencing and PacBio long read sequencing (Li et al., 2014b). MiSeq generated 7.85 million paired-end reads with the length equal to 250 bp. PacBio generated 0.68 million Reads of Insert (RoIs) with an average length equal to $1,640$ bp.

2.3.2 Quality control for short read *de novo* assemblies

We first trimmed the adapters and filtered out the low quality reads from the MiSeq dataset using Trimmomatic (version 0.32). Adaptors and low quality reads with average quality below 16 over a 5 base window were removed. And only trimmed reads with length over 30 bases were used for *de novo* assembly. FastQC (version 0.11.2) [1]was then used to visualize the read quality before and after cleaning, shown in Supplementary Figure 2.5.

To determine the best kmer for *de novo* assembly, we used Kmergenie (version 1.6982) (Chikhi and Medvedev, 2014). Kmergenie examines multiple kmers and counts the frequency of kmers under each k. Then Kmergenie estimates the best k value, which potentially could recover the most possible contigs. Our dataset has the best k = 31 bp, shown in Supplementary Figure 2.6.

Cleaned reads were used for *de novo* assembling by five different assemblers, including Trinity (version 2.2.0), Oases (version 0.2.08), SOAPdenovo-Trans (version 1.03), Trans-ABySS (version 1.5.1), and IDBA-Tran (version 1.1.2). All the methods were tested under the default parameters.

2.3.3 Quality control for the "real time" transcriptome generated by PacBio long reads

To obtain the real time transcriptome, we ran PacBio long reads through RS_IsoSeq (v2.3.0) pipeline (Gordon et al., 2015) using default parameters. After clustering, we filtered out the non-human genes by aligning both the full length high-quality and full length low-quality consensus sequences to the hg19 human reference genome using STAR (Dobin et al., 2013) and GMAP (Wu and Watanabe, 2005) as recommended by RS_IsoSeq. The detailed steps and the number of sequences generated in each step are shown in Supplementary Figure 2.7. Then we collapsed the aligned consensus

[1]FastQC is available at: `https://www.bioinformatics.babraham.ac.uk/projects/fastqc`

sequences by pbtranscript-tofu [2]with the minimum alignment identity equal to 0.85 and the minimum coverage to 0.90, as shown in Supplementary Figure 2.2.

2.4 Results

2.4.1 The real time transcriptome can be served as a reliable benchmark for assessing *de novo* assemblies

Analysis of PacBio long reads with RS_IsoSeq pipeline, produced 9,636 genes (33,307 transcripts). All 33,307 transcripts were corrected versus the hg19 human genome. 244 full length, low-quality transcripts can be aligned to the hg19 human genome by neither STAR nor GMAP. We pooled these 33,307 alignable sequences and 244 unalignable sequences together, rendering 33,551 transcripts and 9,880 genes in total as the "real time" transcriptome, shown in Supplementary Figure 2.7.

First, to show the relationship between the "real time" transcriptome generated from PacBio long reads and the well annotated human transcriptomes, we drew a Venn diagram between the reference transcriptomes from Ensembl and RefSeq and the "real time" transcriptome using vennBLAST (Zahavi et al., 2015). Ensembl reference transcriptome has 191,891 transcripts; RefSeq has 63,874 transcripts; the "real time" transcriptome has 33,551 transcripts. In Figure 2.1, Ensembl has the most transcripts, which almost cover RefSeq and the real time transcriptome. The real time transcriptome is about half the size of RefSeq and largely overlaps with RefSeq. Apparently, the three transcriptomes do not completely overlap each other, which indicates that the evaluations on the *de novo* assemblies would be very different if we chose different reference transcriptomes. Though the "real time" transcriptome is not the most complete set of human transcripts, it derives from the same biological sample as the short reads, which eliminates the uncertainty of sample variance.

[2]pbtranscript-tofu is available at: `https://github.com/PacificBiosciences/cDNA_primer/wiki/tofu-Tutorial-(optional).-Removing-redundant-transcripts`

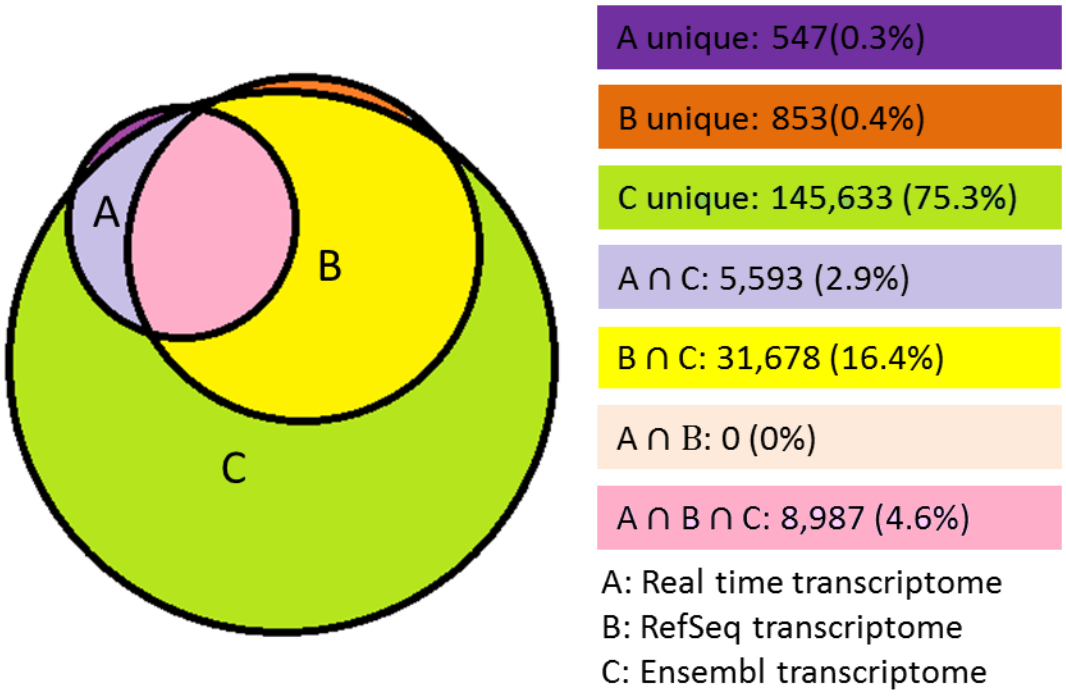


Fig. 2.1.: The Venn diagram of three different reference transcriptomes. A is the "real time" transcriptome. B is the RefSeq transcriptome. C is the Ensembl transcriptome. Note that the reason that the total number of transcripts in RefSeq and the real time transcriptome is smaller than the numbers mentioned in the text is because there are multiple transcripts in RefSeq and the real time transcriptome aligning to the same transcript in Ensembl.

Therefore, the real time transcriptome should be more optimal as a benchmark for assessing short read assemblies than the other two references.

Second, we checked whether the abundance of PacBio long reads was corresponds to that of the MiSeq short reads. If yes, it will provide another evidence that the "real time" transcriptome generated from PacBio long reads can serve as a reliable reference for assessing short read assemblies. A scatter plot of the ranks of the abundances estimated by PacBio long reads, and MiSeq short reads is shown in Figure 2.2. Each data point represents a gene from the "real time" transcriptome. Most highly expressed genes in PacBio also have high expressions as estimated by short reads, and lie in the right up corner. The low expression genes in PacBio have different expression patterns, ranging from low to high as estimated by short reads,

and lie along the bottom. This pattern is due to the different throughputs of two sequencing technologies. PacBio has a lower sequencing thoughput than the short read platform. Many transcripts have only one copy detected in PacBio, but these transcripts may have many short reads sampled in MiSeq. This relationship between the abundances of PacBio long reads and MiSeq short reads suggests that a majority of transcripts from the "real time" transcriptome should be recovered by the short read assembly.

In summary, the generation of the real time transcriptome agrees with both the well annotated reference transcriptome and the real time sampling. Therefore, the real time transcriptome can be a better benchmark for assessing short read *de novo* assembling in terms of both sufficency and specificity.

2.4.2 Assessments on short read *de novo* assembly methods

Assemblies were performed by each method and for each method the number and the length of predicted transcripts are compared. In Table 2.1, Trinity generates the most assemblies, while Oases generates the fewest assemblies, but with longest average length, median length and N50. The numbers of assemblies in SOAP-denovoTrans, and IDBA-Tran are between those of Trinity and Oases. The distribution of the assembly length in Figure 2.3 shows that the assemblers can be categorized into three groups. Trinity tends to give more assemblies in the range of $200-400$ bp than alternative methods; Oases tends to give the fewest assemblies in the range of $200-400$ bp but the curve gradually goes up, having the largest N50 $= 1,090$ bp. Trans-ABySS, SOAPdenovoTrans, and IDBA-Tran share very similar distributions yet IDBA-Tran reports a slightly higher number of assemblies in the range of $300-400$ bp than Trans-ABySS and SOAPdenovoTrans. This finding is consistent with the result in (Wang and Gribskov, 2017), which tested the above assemblers using two authentic RNA-Seq datasets from *Arabidopsis thaliana*. Also, by comparing the assemblies with three reference transcriptomes in Figure 2.3, including RefSeq, Ensembl, and the "real

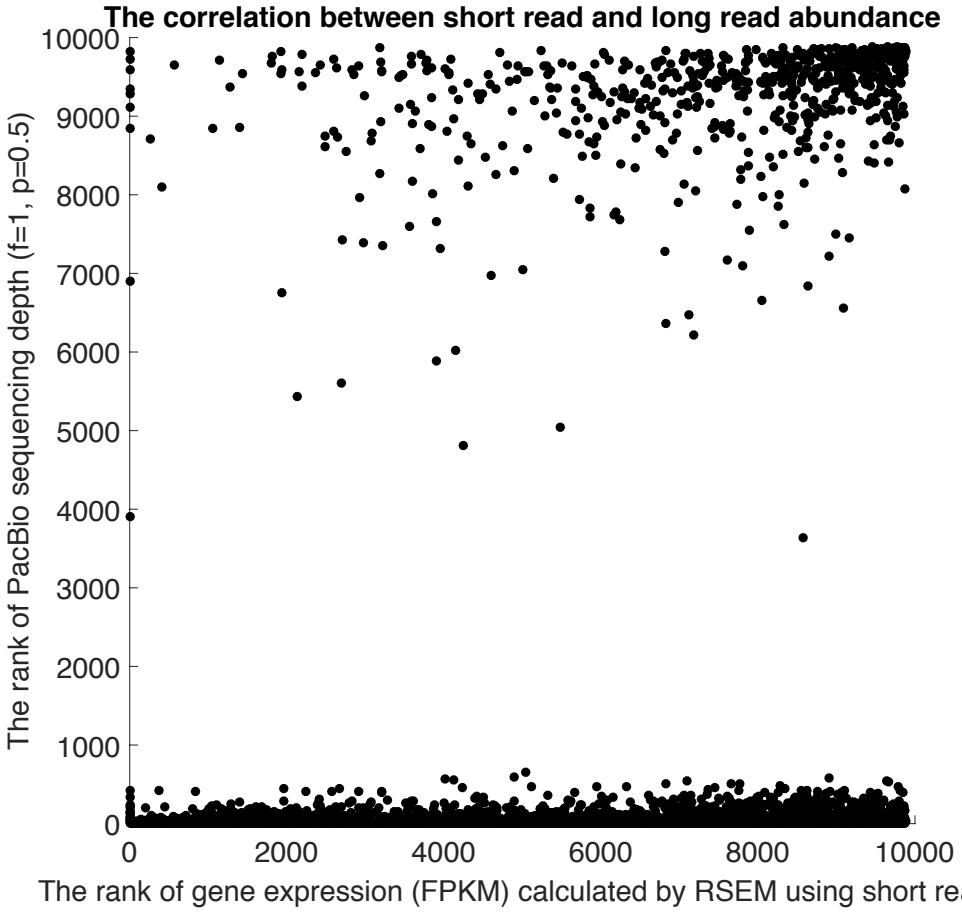


Fig. 2.2.: Correlation between the abundance ranks of PacBio long reads and MiSeq short reads. The gene with the lowest abundance is ranked as the first. X-axis shows the ranks of gene expression in Fragments Per Kilobase of transcript per Million mapped reads (FPKM) estimated by RSEM (Li and Dewey, 2011) using MiSeq short reads. Y-axis shows the ranks of gene counts from PacBio long reads. If a gene is supported by $1f5p$ in PacBio long read sequencing, it means this gene is supported by one full length read and five partial length reads in PacBio. The expression of this gene would be given as (1+ 0.5x5) = 3.5.

time" transcriptome, it is clear that all assemblers provide redundant assemblies; and the redundant assemblies are mostly in the short length range.

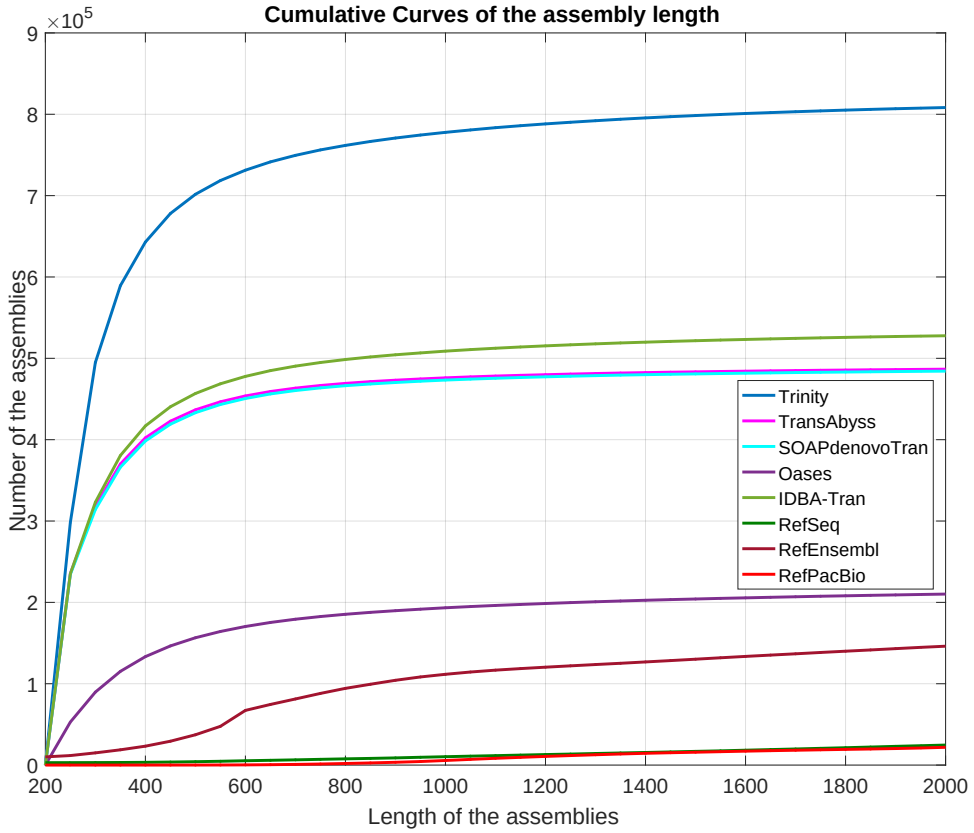


Fig. 2.3.: Cumulative curves of the assembly length from five *de novo* assembly methods. There reference transcriptomes are also plotted as quality controls.

The more short reads that can be aligned back to the assembly, the higher probability that the assembler has generated the correct assembly, if the gene expression level is not taken into account at this stage. Table 2.2 shows that Trinity, SOAPdenovo-Trans, and Trans-ABySS have $75\% - 78\%$ short reads that can be mapped back, while Oases and IDBA-Tran only have $56\% - 59\%$. If we only count the number of concordant reads (see the second column in Table 2.2), the trend is the same as when we count the total number of aligned reads. This suggests that Trinity, SOAPdenovo-Trans and Trans-ABySS assemblies potentially contain more

Table 2.1.: Metrics for the length and the total number of assemblies from five methods.

	min_len	**max_len**	**Total bases**	**Number of assemblies**	**avg_len**	**median_len**	**N50**
Trinity	200	22,834	332,493,521	**821,870**	405	273	388
IDBA-Tran	200	23,838	222,727,826	538,261	414	266	412
SOAPdenovo-Trans	200	27,428	181,427,977	490,473	370	255	350
Trans-ABySS	200	27,277	322,119,676	492,286	363	254	341
Oases	200	27,468	149,749,751	**224,847**	**666**	**343**	**1,090**

information from the short reads than those of Oases and IDBA-Tran. However, if we measure the number of reads mapped back per kilobase of assembly, SOAPdenovo-Trans, Trans-ABySS, and Oases have about $60 - 64$ short reads mapped back per kilobase of assembly, while IDBA-Tran has 39 and Trinity has only 10. This is because, even though Trinity covers the highest number of short reads, it reports many more predicted assemblies than the other methods.

Table 2.2.: The number of short reads that can be mapped back to the assemblies.

	Total number of read support	Number of concordant reads	Number of reads per 1K assmblies
Trinity	**11,897,117(78.45%)**	**3,318,984(21.89%)**	9.98
SOAPdenovo-Trans	11,355,630(74.88%)	2,807,074(18.51%)	62.59
Trans-ABySS	11,372,597(74.99%)	2,809,522(18.53%)	63.64
Oases	9,021,448(59.49%)	2,089,026(13.78%)	60.24
IDBA-Tran	8,595,231(56.68%)	1,992,794(13.14%)	38.59

The total number of read support include both the concordantly and disconcordantly mapped reads. Concordant read support means both of the paired end reads can be mapped into an assembly in the right orientation. The disconcordant reads mean either the paired end reads cannot map to the same assembly or map to an assembly in a reversed manner. The number of aligned reads per kilobases of assemblies = the total number of read support / the total number of assemblies.

We evaluated the qualities of assemblies by aligning them back to the “real time" transcriptome using rnaQUAST (Bushmanova et al., 2016) (version 1.4.0). We considered all main statistics reported by rnaQUAST to evaluate the quality of assemblies, including alignability, accuracy, completeness/sensitivity, specificity, continuity, and misassembly. The overall performance of SOAPdenovo-Trans, Trans-ABySS and IDBA-Tran (Table 2.3,) are similar; SOAPdenovo has the highest accuracy and the lowest number of misassemblies in five methods. Oases and Trinity perform very differently, yet each has its own advantages. Oases has the longest average alignment length, the best continuity, specificity, and mean isoform coverage, but Oases assemblies are less complete at both the gene and isoform level. Trinity has the highest completeness at both the gene and isoform level, and a slightly lower specificity than Oases, but a relatively poor continuity and the highest rate of misassemblies. Note

that the specificities are very low in all five methods, which indicates a redundancy of assemblies reported.

2.4.3 Assessments on model-based *de novo* assembly evaluation methods

There are two state-of-the-art methods having been assessed here, DETONATE (version 1.9) and TransRate (version 1.0.1). The RSEM-EVAL module of DETONATE is used for evaluating *de novo* assemblies without a reference transcriptome. The RSEM-EVAL score is the sum of three components; the likelihood estimates how well the assemblies are explained by the mapped short reads; the assembly prior assumes the assembly length follows a negative binomial distribution and the transcripts are independent from each other (the number of isoforms or homogenous genes will influence this component); the BIC penalty penalizes the prediction of too many bases and assemblies. Table 2.4 shows that the likelihood makes the largest contribution to the RSEM-EVAL score. Consistent with Table 2.2 and 2.3, Trinity has the highest likelihood, but the lowest assembly prior and BIC penalty, which lowers its overall RSEM-EVAL score. On the contrary, SOAPdenovo-Trans and Trans-ABySS do not score highly any component, which is consistent with Table 2.3, but achieve the best overall RSEM-EVAL score, because no single component dominates the final evaluation. IDBA-Tran and Oases have low RSEM-EVAL scores mainly due to the low likelihoods, though Oases has the best assembly prior and BIC penalty, which is also consistent with Table 2.2 and 2.3.

TransRate shows the opposite pattern compared with DETONATE. The TransRate assembly score is the geometric mean of the contig scores multiplied by the proportion of short reads that positively support the assemblies. Each contig score is the product of four components: the nucleotide score measuring the alignment distance between the assembly and the short reads, the coverage score measuring the fraction of the assembly length covered by reads, the order score measuring the orientation of the paired-end read mapping, and the segment score measuring the per-nucleotide

read coverage. In Table 2.5, we find that Oases has the highest TransRate score. However, after optimizing an empirical target function – $\mathrm{T} = \sqrt{\left(\prod_{c=1}^{n} S(C)\right)^{\frac{1}{n}} R_{valid}}$, where $S(C)$ is the contig score, n is the number of selected contigs, and R_{valid} is the proportion of reads that can be mapped to the selected contigs – as recommended by TransRate, the TransRate scores of all methods greatly increase, and Trinity has the best optimal score.

2.4.4 Contig scores can serve as a good metric for removing low quality assemblies

In Section 2.4.2, we found that the number of predicted assemblies produced by *de novo* approaches was about 15 times of the number of transcripts in the"real time" transcriptome, on average; In Section 2.4.3, we found that a large portion of assemblies have low DETONATE and TransRate contig scores. Together, this indicates a redundancy of assemblies. Therefore, the question is whether DETONATE and TransRate contig scores can serve as good metrics for removing low-quality assemblies.

We selected the top $40,000$ assemblies based on the DETONATE score, the TransRate score, and the FPKM of each assembly. An ideal selection would be an assembly set with no change in completeness, but with increased specificity and continuity, and decreased misassembly rate, compared with the full set of assemblies. Figure 2.4 shows the comparison between the full set and the selected set of assemblies in the completeness, specificity, continuity and the misassembly rate. For completeness, the database coverage rates are decreased in all selected sets compared to the full set, but DETONATE selections show higher completeness than TransRate and FPKM. For specificity, DETONATE selections show a generally higher mean fraction of matched assemblies than the full set of assemblies in all methods, but not the other two metrics. For continuity, DETONATE selections also show a generally higher mean fraction of isoform length assembled than the full set of assemblies; FPKM selections also have a higher continuity than the full set in all methods, except for Trinity. For the misas-

sembly rate, both TransRate and FPKM selections can greatly decrease the number of misassemblies but not DETONATE, which might because TransRate takes the order score into account.

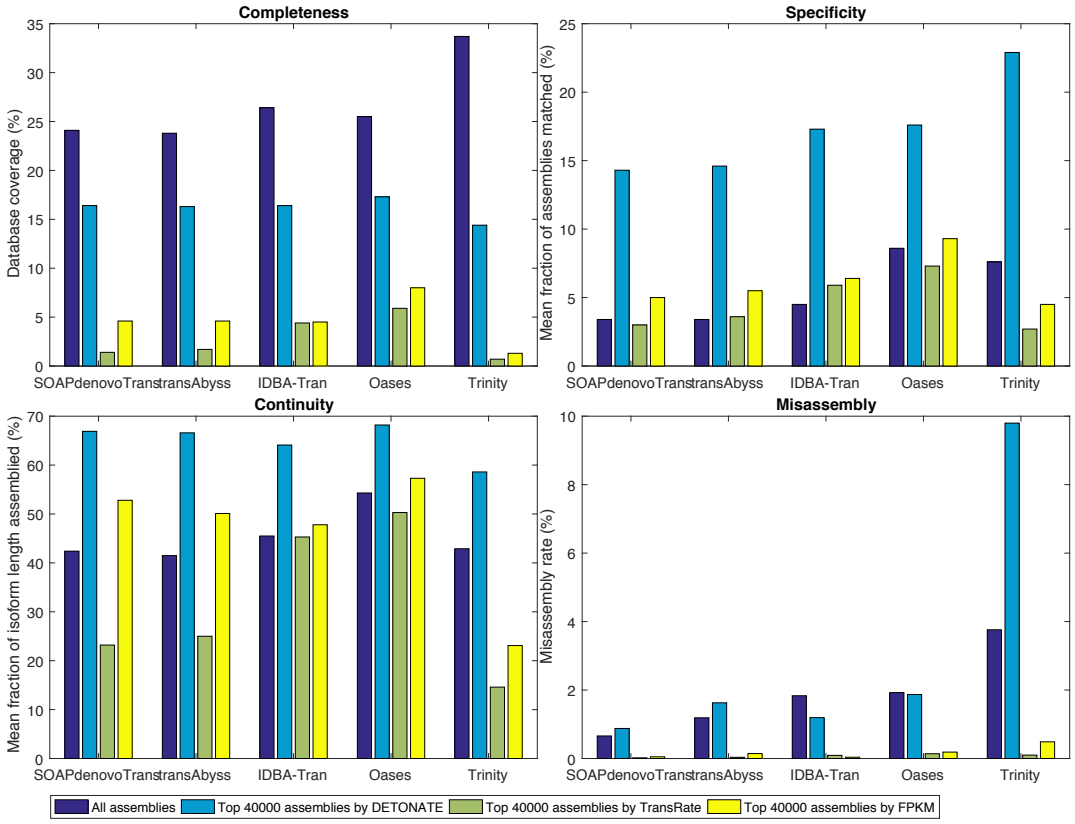


Fig. 2.4.: Efficiency to remove low quality assemblies by three different metrics, including DETONATE contig score, TransRate contig scores and the FPKMs of contigs. Four major aspects have been evaluated by comparing the top $40,000$ selected assemblies with the"real time" transcriptome. We evaluate the assembly quality in terms of the completeness, specificity, continuity, and the misassembly rate, as that in Table 2.3. The misassembly rate is calculated as the number of misassemblies divided by the number of assemblies.

Table 2.3.: Evaluations of assemblies from five *de novo* assembly methods by comparing with the "real time" transcriptome. We followed the nomenclatures used in rnaQUAST.

	SOAPdenovo-Trans	Trans-ABySS	IDBA-Tran	Oases	Trinity
Alignment					
Number of alignments (hit=50bp)	489,823(99.87%)	491,598(99.86%)	537,458 (99.85%)	224,564(99.87%)	820,611(99.85%)
Avg aligned length (bp)	367.8	360.1	408.8	649.8	368.4
Accuracy					
Avg mismatches (bp) per 1K alignment	1.4	1.6	1.7	1.5	3.0
Completeness/Sensitivity					
Gene level					
Number of > 50% covered genes	6,390(66.31%)	6,385(66.26%)	6,539(67.86%)	6,193(64.27%)	6,895(71.55%)
Number of > 95% covered genes	2,805(29.11%)	2,851(29.59%)	2,796(29.02%)	2,845(29.52%)	3,085(32.02%)
Isoform level					
Number of > 50% covered isoforms	7,020(21.08%)	6,952(20.87%)	7,964(23.91%)	7,726(23.20%)	10,396(31.21%)
Number of > 95% covered isoforms	2,865(8.60%)	2,885(8.66%)	2,974(8.93%)	3,104(9.32%)	3,423(10.28%)
Database coverage	24.1%	23.8%	26.4%	25.5%	33.7%
Mean isoform coverage	48.8%	47.9%	52.7%	59.9%	53.3%
Specificity					
Number of > 50% matched assemblies	16,175(3.30%)	16,309(3.31%)	24,250(4.51%)	19,694(8.76%)	62,758(7.64%)
Number of > 95% matched assemblies	10,391(2.12%)	10991(2.23%)	13003(2.42%)	9505(4.23%)	35316(4.30%)
Mean fraction of assemblies matched	3.4%	3.4%	4.5%	8.6%	7.6%
Unannotated assemblies	459,792(93.74%)	458,408 (93.12%)	492,515(91.50%)	192,489(85.61%)	708,017 (86.14%)
Continuity					
Gene Level					
Number of > 50% assembled genes	5,228(54.25%)	5,243(54.41%)	5,266(54.65%)	5,370(55.73%)	4,967(51.55%)
Number of > 95% assembled genes	2,270(23.56%)	2,243(23.28%)	2,123(22.03%)	2,381(24.71%)	1,800(18.68%)
Isoform Level					
Number of > 50% assembled isoforms	5,604(16.83%)	5,561(16.70%)	6,287(18.88%)	6,663(20.00%)	7,160(21.50%)
Number of > 95% assembled isoforms	2,313(6.94%)	2,268(6.81%)	2,265(6.80%)	2,605(7.82%)	2,002(6.01%)
Mean isoform continuity	42.4%	41.5%	45.5%	54.3%	42.9%
Misassemblies	3,233(0.66%)	5,859(1.19%)	9,874(1.83%)	4,332(1.93%)	30,915(3.76%)

- Completeness/sensitivity is calculated by aligning the assemblies to the genes/isoforms in the database, showing how completely the assemblies can cover the database.
- Specificity is calculated by aligning the isoforms/genes in the database to the assemblies, showing how specific or redundant the assemblies are in the database.
- Continuity is always calculated using the longest assemblies that can continuously mapped to the genes/isoforms in the database, showing whether the assemblies are integral.
- Misassemblies are confirmed by both GMAP and BLASTN, meaning partial alignments from the one assembly can be equally well mapped to different locations in the database.
- Genes/isoforms mean the transcripts from the real time transcriptome. Assemblies means the *de novo* assemblies generated by each assembler.
- Gene/isoform coverage is a percentage calculated as the number of bases on the gene/isoform covered by the assemblies divided by the length of this gene/isoform.
- x% covered genes/isoforms means the number of genes/isoforms that have at least x% gene/isoform coverage.
- Database coverage means the total number of bases covered by assemblies divided by the total length of all isoforms in the database.
- The matched fraction of each assembly is calculated as the number of matched bases on the assembly divided by the length of this assembly.
- x% matched assemblies means the total number of assemblies that have at least x% matched fraction.
- Unannotated assemblies mean the total number of assemblies that do not cover any isoform from the database.
- Gene/isoform continuity is also a percentage calculated as the number of bases on the gene/isoform covered by the longest continuous assembly divided by the length of this gene/isoform.
- x% assembled genes/isoforms means the number of genes/isoforms that have an at least x% gene/isoform continuity.

  Dark green: the best performance; Light green: good performance, slightly lower than the best, but better than the rest methods; Red: the lowest performance. For those having no color marked, their performance are comparable to each other, but obviously better than the red and worse than the green.

Table 2.4.: DETONATE RSEM-EVAL scores for five *de novo* assembly methods.

	SOAPdenovo-Trans	**Trans-ABySS**	**IDBA-Tran**	**Oases**	**Trinity**
Likelihood	-3,001,246,561	-3,016,249,561	-3,238,791,356	-3,225,800,940	**-2,841,562,473**
Assembly prior	-251,577,741	-247,841,006	-308,900,873	**-207,358,450**	-461,191,586
BIC penalty	-3,884,882	-3,899,242	-4,263,395	**-1,780,946**	-6,509,768
RSEM-EVAL score *	**-3,256,709,184**	-3,267,989,809	-3,551,955,624	-3,434,940,336	-3,309,263,827

* The RSEM-EVAL score is the sum of three components, including the likelihood, the assembly prior and the BIC penalty for each assembly set.

Table 2.5.: TransRate assembly and the assembly scores after optimization for five *de novo* assembly methods. The potential bridges show the number of potential links between contigs that are supported by the reads.

	SOAPdenovo-Trans	**Trans-ABySS**	**IDBA-Tran**	**Oases**	**Trinity**
TransRate score	0.00173	0.00166	0.00087	**0.00271**	0.00221
Optimal score	0.02787	0.02757	0.00877	0.01748	**0.03122**
Potential Bridges	2,546	1,129	8,014	5,379	**42,296**

2.5 Discussion

In this study, we propose a reliable benchmark – a real time transcriptome, produced by PacBio long read sequencing – for assessing the *de novo* assembly and evaluation methods. As opposed to other *de novo* assembly assessment strategies, which either simulate RNA-Seq data or utilize well annotated reference transcriptome as a ground truth for real data, our study takes the advantage of sequencing the same biological sample using both the short read and long read technologies to eliminate the biological uncertainty. The real time transcriptome relies on both the well annotated reference transcriptome and the real time sampling, thus, the real time transcriptome can serve as a better reference for assessing *de novo* assemblies than the alternative simulation or a reference transcriptome.

By comparing the *de novo* assemblies from five commonly used methods to the real time transcriptome, we find that the properties of the tested assemblers vary significantly. For instance, Trinity has the highest read mapping rate (shown in Table 2.2), and the best completeness, but generates too many short assemblies in the range between $200 - 400$ bases (shown in Figure 2.3). This makes Trinity assemblies less continuous, and potentially increasing the number of assemblies that can be linked by short reads (shown in Table 2.5). Trinity also has the highest misassembly rate of the five methods (shown in Table 2.3). An improvement to Trinity would be to decrease the number of misassemblies while increasing the continuity. Oases generally generates the longest and the fewest assemblies in all five methods (shown in Table 2.1), which gives it the best continuity and specificity (shown in Table 2.3). However, Oases has a low read mapping rate (shown in Table 2.2), which makes it less complete than the other methods. An improvement to Oases would be to increase the completeness of the assemblies. The performance of SOAPdenovo-Trans, Tran-AByss, and IDBA-Tran are very similar, but SOAPdenovo-Trans has the lowest number of mismatches and misassemblies of the five methods (shown in Table 2.3).

Because of the overall redundancy of *de novo* assemblies in all the methods, DETONATE and TransRate can serve as good metrics to evaluate and remove low-quality assemblies, but with different patterns. The DETONATE assembly score mainly considers the read mapping rate, the independence of transcripts, the total number of assemblies, and the number of assembled bases when evaluates the assemblies. DETONATE ranks the method with no extreme disadvantages in the above aspects as the best (shown in Table 2.4). The TransRate assembly score is an empirical function that takes many different aspects into account, mainly including the mapping accuracy, the mapping orientation, the mapping depth, the mapping coverage, and the fraction of mapped reads. By taking the product of the first four terms as the contig score, TransRate actually treats the first four aspects equally, then weights the contig score by the fraction of mapped reads. TransRate only encourages the advantages but doesn't penalize the disadvantages of the assembly, as the way DETONATE does. The optimization of the TransRate assembly score is a good way to select the best quality assemblies, but the number of selected assemblies is often low, and cannot be controlled by users.

Both DETONATE and TransRate provide contig scores as an evaluation for each assembly. The contig scores can be used as metrics for removing redundant low quality assemblies. When the top $40,000$ assemblies ranked by DETONATE, TransRate and FPKM are examined, we find that the DETONATE contig score can effectively remove the redundant assemblies while keeping a high completeness and continuity rate, but not be able to remove misassemblies. The TransRate contig score is very sensitive in removing misassemblies but not helpful in the completeness, specificity and continuity.

There is weakness in this study. For instance, only one dataset has been tested here, because it is not very easy to obtain the datasets which have been sequenced by both short read and long read technologies. It would be better to include further benchmark datasets to eliminate any bias from the sequencing platforms or organisms. Also, we evaluated the assemblies from several major perspectives, including

length, the total number of assemblies, the read mapping rate, completeness, specificity, continuity and misassembly, by comparing the assemblies with the real time transcriptome. There may be additional perspectives that the model-based evaluation methods take into account, but are not included in our metrics.

2.6 Supplementary materials

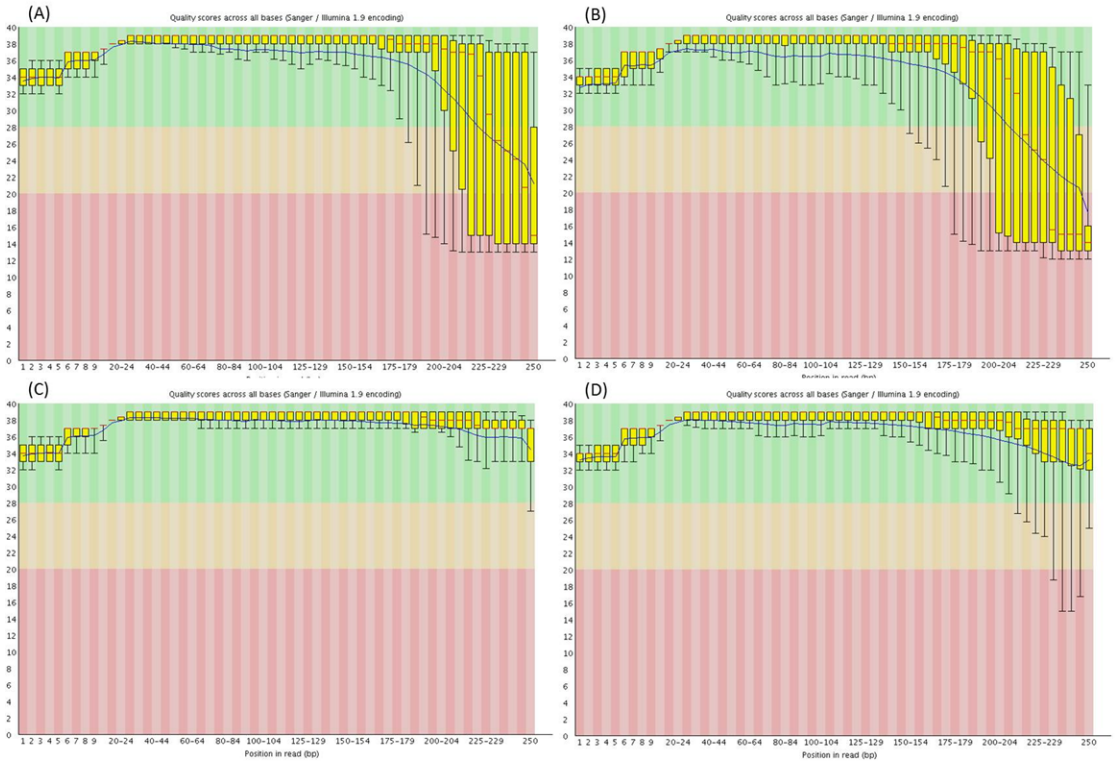


Fig. 2.5.: MiSeq read quality visualization by FastQC before and after trimming. (A) and (B) are the positional qualities of forward and backward reads in the raw dataset. (C) and (D) are the positional read qualities of forward and backward reads after trimming.

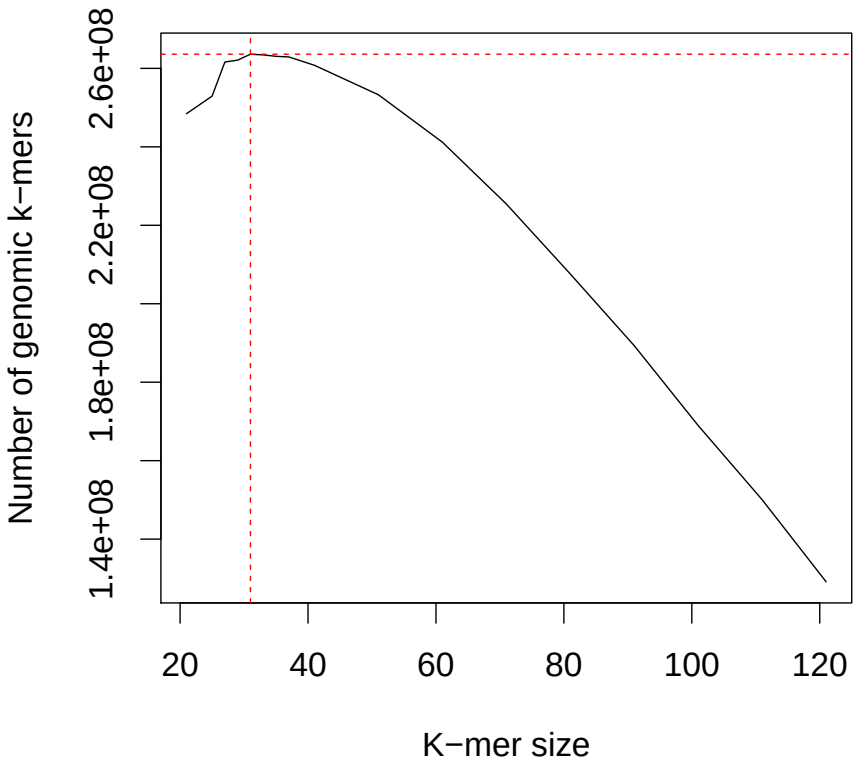


Fig. 2.6.: Kmergenie shows kmer = 31bp is the best choice for short read assembly.

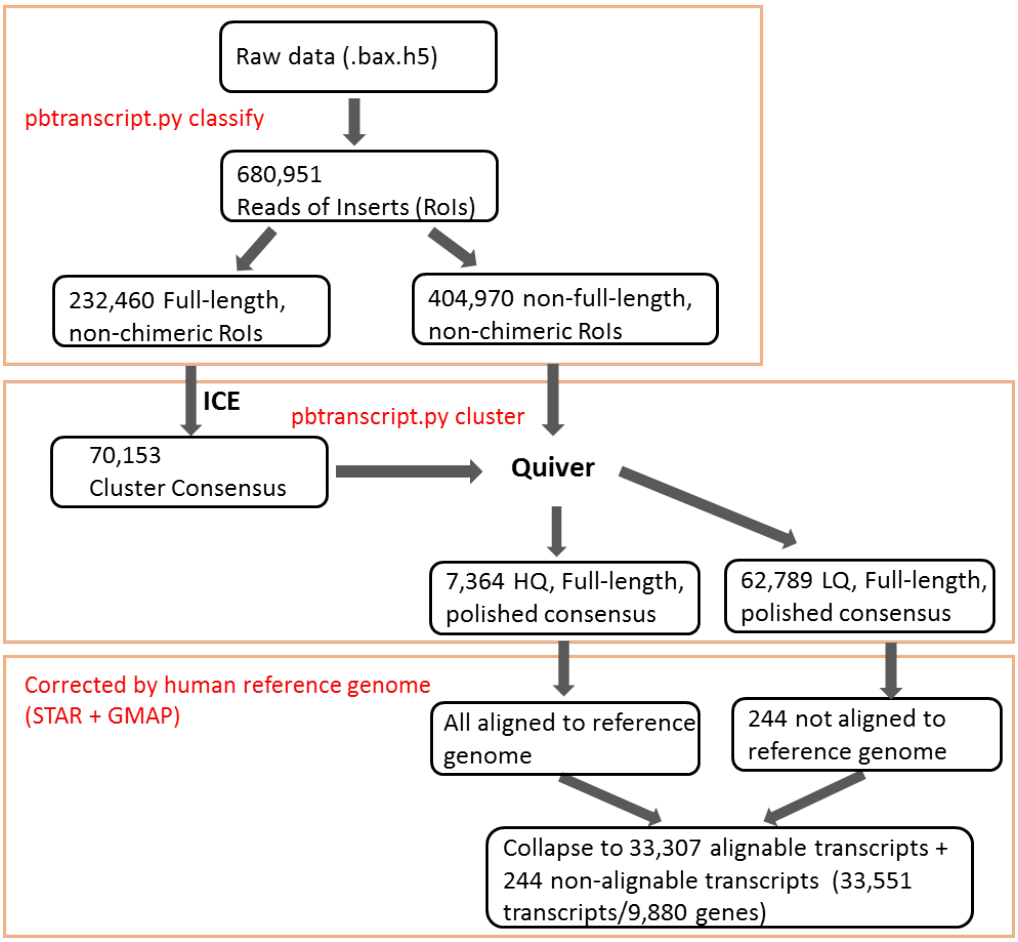


Fig. 2.7.: The flowchart of processing PacBio long reads into real time transcriptome. We begin from the .bax.h5 raw data. The first step is classification, namely to classify Reads of Inserts (RoIs) into full-length and non-full-length RoIs based on the adaptors, meanwhile removing the chimeric RoIs. The second step is clustering, namely to cluster RoIs into consensus, while each consensus can be viewed as a transcript. The third step is collapsing and correction, namely to align the consensus sequences back to the reference genome and get the real time transcriptome.

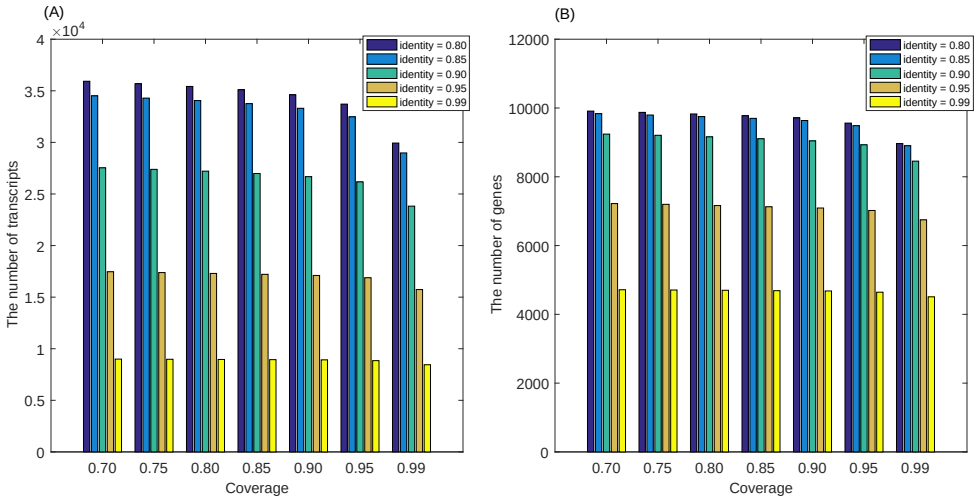


Fig. 2.8.: Selections of the thresholds for coverage and identity when align the consensus sequences back to the human reference genome. Because the coverage and identity are the only parameters the user has to set when running the collapse step in pbtranscript-tofu.py, and these parameters will eventually influence the numbers of transcripts and genes in the real time transcriptome, we carefully select the values of these two parameters. (A) and (B) show the number of transcripts and genes in the real time transcriptome when we set different coverages and identities, respectively. Note that when the coverage is from 0.90 to 0.95, more transcripts/genes will be dropped out than the previous columns, which indicates many consensus sequences having a coverage between 0.90 and 0.95. To keep as much information from PacBio long reads as possible, we consider coverage $= 0.9$ is a long enough to represent a transcript/gene. Similarly for identity, when identity is from 0.85 to 0.90, more transcripts/genes will be dropped out than the previous columns. Taking the facts that the error rate of PacBio sequencing was about 10%-15% in 2013 and the average length of consensus sequence ($1,640$ bp) was long enough to align the consensus sequence to the right position on the genome into account, we choose identity $= 0.85$ in our dataset.

2.7 References

Brereton, N. J. B., Gonzalez, E., Marleau, J., Nissim, W. G., Labrecque, M., Joly, S., and Pitre, F. E. (2016). Comparative transcriptomic approaches exploring contamination stress tolerance in Salix sp. reveal the importance for a metaorganismal *de novo* assembly approach for nonmodel plants. *Plant Physiology*, 171(1):3–24.

Bushmanova, E., Antipov, D., Lapidus, A., Suvorov, V., and Prjibelski, A. D. (2016). rnaQUAST: a quality assessment tool for *de novo* transcriptome assemblies. *Bioinformatics*, 32(14):2210–2212.

Chikhi, R. and Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1):31–37.

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.

Gordon, S. P., Tseng, E., Salamov, A., Zhang, J., Meng, X., Zhao, Z., Kang, D., Underwood, J., Grigoriev, I. V., Figueroa, M., Schilling, J. S., Chen, F., and Wang, Z. (2015). Widespread polycistronic transcripts in fungi revealed by single-molecule mRNA sequencing. *PLOS ONE*, 10(7):1–15.

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B. W., Nusbaum, C., Lindblad-Toh, K., Friedman, N., and Regev, A. (2011). Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nature Biotechnology*, 29(7):644–652. 10.1038/nbt.1883.

Honaas, L. A., Wafula, E. K., Wickett, N. J., Der, J. P., Zhang, Y., Edger, P. P., Altman, N. S., Pires, J. C., Leebens-Mack, J. H., and dePamphilis, C. W. (2016). Selecting superior *de novo* transcriptome assemblies: Lessons learned by leveraging the best plant genome. *PLOS ONE*, 11(1):1–42.

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

Li, B., Fillmore, N., Bai, Y., Collins, M., Thomson, J., Stewart, R., and Dewey, C. (2014a). Evaluation of *de novo* transcriptome assemblies from RNA-seq data. *Genome Biology*, 15(12):553.

Li, S., Tighe, S. W., Nicolet, C. M., Grove, D., Levy, S., Farmerie, W., Viale, A., Wright, C., Schweitzer, P. A., Gao, Y., Kim, D., Boland, J., Hicks, B., Kim, R., Chhangawala, S., Jafari, N., Raghavachari, N., Gandara, J., Garcia-Reyero, N., Hendrickson, C., Roberson, D., Rosenfeld, J. A., Smith, T., Underwood, J. G., Wang, M., Zumbo, P., Baldwin, D. A., Grills, G. S., and Mason, C. E. (2014b). Multi-platform assessment of transcriptome profiling using RNA-seq in the ABRF next-generation sequencing study. *Nature Biotechnology*, 32(9):915–925.

Mamrot, J., Legaie, R., Ellery, S. J., Wilson, T., Seemann, T., Powell, D. R., Gardner, D. K., Walker, D. W., Temple-Smith, P., Papenfuss, A. T., and Dickinson, H. (2017). *De novo* transcriptome assembly for the spiny mouse (Acomys cahirinus). *Scientific Reports*, 7(1):8996.

Martin, J. A. and Wang, Z. (2011). Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682.

Melé, M., Ferreira, P. G., Reverter, F., DeLuca, D. S., Monlong, J., Sammeth, M., Young, T. R., Goldmann, J. M., Pervouchine, D. D., Sullivan, T. J., Johnson, R., Segrè, A. V., Djebali, S., Niarchou, A., Consortium, T. G., Wright, F. A., Lappalainen, T., Calvo, M., Getz, G., Dermitzakis, E. T., Ardlie, K. G., and Guigó, R. (2015). The human transcriptome across tissues and individuals. *Science*, 348(6235):660–665.

Mittal, V. K. and McDonald, J. F. (2017). *De novo* assembly and characterization of breast cancer transcriptomes identifies large numbers of novel fusion-gene transcripts of potential functional significance. *BMC Medical Genomics*, 10(1):53.

Moreton, J., Dunham, S. P., and Emes, R. D. (2014). A consensus approach to vertebrate *de novo* transcriptome assembly from RNA-seq data: assembly of the duck (Anas platyrhynchos) transcriptome. *Frontiers in Genetics*, 5:190.

Moreton, J., Izquierdo, A., and Emes, R. D. (2015). Assembly, assessment, and availability of *de novo* generated eukaryotic transcriptomes. *Frontiers in Genetics*, 6:361.

O'Neil, S. T. and Emrich, S. J. (2013). Assessing *de novo* transcriptome assembly metrics for consistency and utility. *BMC Genomics*, 14(1):1–12.

Peng, Y., Leung, H. C. M., Yiu, S.-M., Lv, M.-J., Zhu, X.-G., and Chin, F. Y. L. (2013). IDBA-tran: a more robust *de novo* de Bruijn graph assembler for transcriptomes with uneven expression levels. *Bioinformatics*, 29(13):i326.

Ranjan, A., Ichihashi, Y., Farhi, M., Zumstein, K., Townsley, B., David-Schwartz, R., and Sinha, N. R. (2014). *De Novo* assembly and characterization of the transcriptome of the parasitic weed dodder identifies genes associated with plant parasitism. *Plant Physiology*, 166(3):1186–1199.

Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., Mungall, K., Lee, S., Okada, H. M., Qian, J. Q., Griffith, M., Raymond, A., Thiessen, N., Cezard, T., Butterfield, Y. S., Newsome, R., Chan, S. K., She, R., Varhol, R., Kamoh, B., Prabhu, A.-L., Tam, A., Zhao, Y., Moore, R. A., Hirst, M., Marra, M. A., Jones, S. J. M., Hoodless, P. A., and Birol, I. (2010). *De novo* assembly and analysis of RNA-seq data. *Nature Methods*, 7:909–912.

Schulz, M. H., Zerbino, D. R., Vingron, M., and Birney, E. (2012). Oases: Robust *de novo* RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*, 28(8):1086–1092.

Smith-Unna, R. D., Boursnell, C., Patro, R., Hibberd, J. M., and Kelly, S. (2016). Transrate: reference free quality assessment of de-novo transcriptome assemblies. *Genome Research*, 26(8):1134–1144.

Wang, L., Wang, Z., Chen, J., Liu, C., Zhu, W., Wang, L., and Meng, L. (2015). *De Novo* transcriptome assembly and development of novel microsatellite markers for the traditional chinese medicinal herb, Veratrilla baillonii Franch (Gentianaceae). *Evolutionary Bioinformatics*, 11(Suppl 1):39–45.

Wang, S. and Gribskov, M. (2017). Comprehensive evaluation of *de novo* transcriptome assembly programs and their effects on differential gene expression analysis. *Bioinformatics*, 33(3):327–333.

Wu, T. D. and Watanabe, C. K. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, 21(9):1859–1875.

Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., Huang, W., He, G., Gu, S., Li, S., Zhou, X., Lam, T.-W., Li, Y., Xu, X., Wong, G. K.-S., and Wang, J. (2014). SOAPdenovo-Trans: *de novo* transcriptome assembly with short RNA-seq reads. *Bioinformatics*, 30:1660–1666.

Zahavi, T., Stelzer, G., Strauss, L., Salmon, A. Y., and Salmon-Divon, M. (2015). VennBLAST-Whole transcriptome comparison and visualization tool. *Genomics*, 105(3):131 – 136.

3. CHAPTER 3. DEISOM: A HIERARCHICAL BAYESIAN MODEL FOR IDENTIFYING DIFFERENTIALLY EXPRESSED ISOFORMS USING BIOLOGICAL REPLICATES

DEIsoM: A hierarchical Bayesian model for identifying differentially expressed isoforms using biological replicates

Hao Peng [1,*,†], Yifan Yang [1,3,†], Shandian Zhe [1], Jian Wang [4], Michael Gribskov [1,3] and Yuan(Alan) Qi [1,2]

[1]Department of Computer Science, [2]Department of Statistics, [3]Department of Biological Sciences, Purdue University, West Lafayette, IN 47907, USA, and [4]Eli Lilly and Company, Indianapolis, IN 46285, USA

[*]To whom correspondence should be addressed.
[†]These two authors contributed equally to this paper.

Received on 16 December 2016;
Revised on 05 May 2017;
Accepted on 02 June 2017.

Associate Editor: Alfonso Valencia

3.1 Abstract

Motivation: High-throughput mRNA sequencing (RNA-Seq) is a powerful tool for quantifying gene expression. Identification of transcript isoforms that are differentially expressed in different conditions, such as in patients and healthy subjects, can provide insights into the molecular basis of diseases. Current transcript quantification approaches, however, do not take advantage of the shared information in the biological replicates, potentially decreasing sensitivity and accuracy.

Results: We present a novel hierarchical Bayesian model called DEIsoM (Differentially Expressed Isoform detection from Multiple biological replicates) for identifying DE (Differentially Expressed) isoforms from multiple biological replicates representing two conditions, *e.g.*, multiple samples from healthy and diseased subjects. DEIsoM first estimates isoform expression within each condition by (1) capturing common patterns from sample replicates while allowing individual differences, and (2) modeling the uncertainty introduced by ambiguous read mapping in each replicate. Specifically, we introduce a Dirichlet prior distribution to capture the common expression pattern of replicates from the same condition, and treat the isoform expression of individual replicates as samples from this distribution. Ambiguous read mapping is modeled as a multinomial distribution, and ambiguous reads are assigned to the most probable isoform in each replicate. Additionally, DEIsoM couples an efficient variational inference and a post-analysis method to improve the accuracy and speed of identification of DE isoforms over alternative methods. Application of DEIsoM to an HCC (Hepatocellular Carcinoma) dataset identifies biologically relevant DE isoforms. The relevance of these genes/isoforms to HCC are supported by PCA (Principal Component Analysis), read coverage visualization, and the biological literature.

Availability: The software is available at : https://github.com/hao-peng/DEIsoM

Contact: pengh@purdue.edu

3.2 Introduction

RNA-seq is a powerful tool for investigating the transcriptomes of various organisms. There are many complex issues in RNA-seq and transcriptome analysis ranging from RNA-seq read correction (Le et al., 2013), transcriptome assembly (Martin and Wang, 2011) to alternative splicing and gene fusion detection (Ozsolak and Milos, 2011). However, one of the most fundamental issues is to quantify and identify isoforms differentially expressed in two conditions, while each containing multiple replicates. Most DE isoform quantification methods treat each replicate independently, ignoring the fact that, because the underlying biological mechanism is the same in a given condition, the replicates tend to share similar expression patterns. DEIsoM improves DE isoform identification and quantification by catching the information shared between replicate samples; rather than separately estimating the isoform expression for each replicate, it captures the common expression pattern of the whole condition in one single model.

Although many computational tools have been developed for quantifying and identifying DE isoforms using RNA-seq data, nearly all approaches estimate the isoform abundance in each replicate separately, and do not attempt to actively capture the aforementioned shared information. For instance, MISO (Mixture of ISOforms) (Katz et al., 2010) infers the isoform fractions for each replicate and evaluates the DE of every pair of replicates using the Bayes Factor, not considering replicates as a group. Additionally, MISO is slow due to its use of MCMC sampling, which is computationally challenging to adapt to the rapid growth in the amount of RNA-seq data (Kakaradov et al., 2012). DRIMSeq (a Dirichlet-Multinomial framework) (Nowicka and Robinson, 2016) infers the isoform fractions for each replicate in a Dirichlet-Multinomial model with a fixed hyperparameter and evaluates DE between two conditions by likelihood ratio test. Cufflinks (Trapnell et al., 2012) quantifies the isoform abundance in individual replicates by maximum a posteriori (MAP) and detects DE isoforms by the hypothesis test based on Jensen-Shannon divergence. RSEM

(RNA-Seq by Expectation Maximization) (Li and Dewey, 2011) estimates isoform abundance for each replicate using an Expectation Maximization (EM) algorithm. EBSeq (Empirical Bayesian Seq) (Leng et al., 2013) then takes the expected counts from all replicates to fit a joint model and estimates the probability of DE for each isoform between multiple conditions. However, the variance of the expected counts stemming from ambiguous read mapping is simply lost in this process, compromising the DE isoform detection. BitSeq (Bayesian inference of transcripts from Sequencing data) (Glaus et al., 2012) (Hensman et al., 2015) estimates the per condition mean isoform abundance from multiple replicates. However, BitSeq accomplishes this estimation in two stages rather than in an integrated model, which could potentially lose information when the "pseudo-data" from each fitted model in stage 1 is fed to the conjugate normal-gamma model in stage 2. Some other models do take the strategy of utilizing the shared information from multiple biological replicates, such as rMATS (Shen et al., 2014), and MAJIQ (Vaquero-Garcia et al., 2016). However, they are both exon-centric, quantifying and identifying alternative splicing at the exon level not the isoform level.

Here, we present DEIsoM, a hierarchical Bayesian model for quantifying and identifying DE isoforms between two conditions. Other than estimating the isoform abundance in each replicate separately, DEIsoM actively captures the shared information of perconditioned replicates in one principle framework. Specifically, DEIsoM uses a Dirichlet prior distribution to capture the shared information among replicates in each condition, and implements a fast VB (Variational Bayesian) method to gain computational efficiency instead of MCMC sampling when computing the posterior distributions of isoform fractions. Figure 3.1(A) shows a typical design for an RNA-Seq experiment with three replicates in each condition. Because we assume that the replicates in one condition share the same underlying biological mechanism, their expression patterns tend to be the same within a certain sample variance. We capture this common pattern through a Dirichlet prior with a tracable and effeciently updated hyperparameter. Additionally, we evaluate the DE isoforms by computing

the KL (Kullback–Leibler) divergence between the posterior distributions of the two conditions, which is intrinsically fast in our model. Figure 3.1(B) gives a qualitative idea of how KL divergence is used to evaluate DE; the DE level is represented as the non-overlapping areas between the two posterior distributions.

Simulations in Section 3.4 demonstrate the superior performance of DEIsoM over alternative methods for quantifying and predicting DE isoforms, as well as the improved computational speed of VB method compared to MCMC sampling. Furthermore, on a real HCC dataset (Section 3.5), DEIsoM identifies HCC relevant DE isoforms which are supported by PCA, read coverage visualization, and the biological literature.

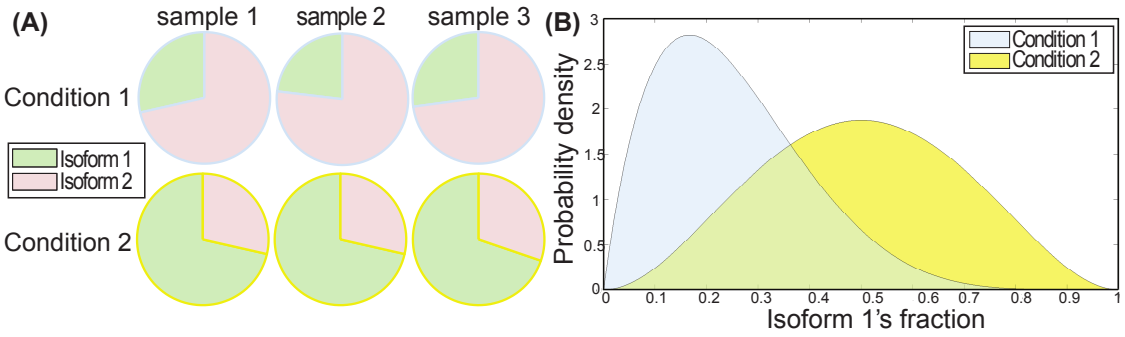


Fig. 3.1.: DEIsoM estimation concept. (A) shows a typical RNA-Seq experimental setting targetted by DEIsoM. There are two conditions, each of which comprises three replicates shown as pie charts representing the expression fractions of two isoforms of a particular gene. We assume that the replicates in one condition are more likely to share a similar expression pattern, which will be modelled by the Dirichlet prior distribution. (B) shows the posterior distribution of fractional isoform expression for each condition. The DE level of the isoform between two conditions can be represented by the non-overlapping regions (purely blue and yellow) under the two curves. In other words, the smaller the overlapping region is, i.e., the more distinct the two posteriors, the more differentially expressed the isoforms of this gene. We measure this distinction by KL divergence, which is a widely recognized method to capture the difference between two probability distributions.

3.3 Methods

DEIsoM consists of three parts: the hierarchical graphical model for isoform quantification (Section 3.3.1), the VB algorithm for model estimation (Section 3.3.2) and the identification of DE isoforms between two conditions (Section 3.3.3).

3.3.1 Model

Suppose we have collected RNA-seq data from M replicates in each condition. For the m^{th} replicate, there are in total $N^{(m)}$ paired-end reads that can be aligned to a given gene with K isoforms. Here, we utilize the previous annotated or assembled isoforms, so K is known for each gene. We use a K-dimensional binary vector, $\mathbf{R}_n^{(m)}$, to represent the read alignment to isoforms. If the n^{th} read from the m^{th} replicate maps to the k^{th} isoform, the k^{th} element of $\mathbf{R}_n^{(m)}$, $R_{n,k}^{(m)}$, is set to be 1, and 0 otherwise. The unsequenced fragment length between the n^{th} paired-end reads is denoted as $\boldsymbol{\lambda}_n^{(m)} = [\lambda_{n,1}^{(m)}, \ldots, \lambda_{n,K}^{(m)}]$.

First, we model how a read is generated from an isoform. We use a binary random variable $Z_{n,k}^{(m)}$ to represent whether the n^{th} read of the m^{th} replicate is actually generated from the k^{th} isoform. We call $Z_{n,k}^{(m)}$ the latent read origin. Although a read can map to multiple isoforms, it can only be sequenced from one isoform. Therefore, $\mathbf{Z}_n^{(m)}$ is a K-dimensional vector with exactly one element equal to 1 and all the others equal to 0, where $\sum_{k=1}^{K} Z_{n,k}^{(m)} = 1$. We assume that for the m^{th} replicate, $\mathbf{Z}_n^{(m)}$ follows a multinomial distribution $p(\mathbf{Z}_n^{(m)}|\boldsymbol{\psi}^{(m)})$, where $\boldsymbol{\psi}^{(m)}$ is a K-dimensional vector representing the fractions of isoforms in the m^{th} replicate for a given gene. Thus, $\boldsymbol{\psi}_k^{(m)} \in [0, 1]$ for all k and $\sum_{k=1}^{K} \boldsymbol{\psi}_k^{(m)} = 1$. The fractions of isoforms $\{\boldsymbol{\psi}^{(m)}\}_{m=1..M}$ can vary among replicates, but we assume that the replicates all follow the same Dirichlet prior distribution $p(\boldsymbol{\psi}|\boldsymbol{\alpha})$ in each condition. Different from MISO, which uses one fixed prior $p(\boldsymbol{\psi})$ for each replicate, DEIsoM shares the same prior among replicates. The underlying reason is that the distributions of isoforms from different replicates of the same condition are not independent, but share some common patterns. DEIsoM

summarizes the shared information in the hyperparameter $\boldsymbol{\alpha}$. In Section 3.3.2, we will further explain how the hyperparameter $\boldsymbol{\alpha}$ is updated using the information from all replicates.

We assume that the observed read alignments $R_{n,k}^{(m)}$ and the unsequenced fragment length $\lambda_{n,k}^{(m)}$ are conditionally independent given the corresponding latent read origin $\mathbf{Z}^{(m)}$ and some fixed parameters Θ:

$$p(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)} | Z_{n,k}^{(m)}, \Theta) = p(R_{n,k}^{(m)} | Z_{n,k}^{(m)}, \Theta) p(\lambda_{n,k}^{(m)} | \Theta)$$

where Θ includes l_k, L, μ and σ^2. l_k is the length of the k^{th} isoform; L is the sequenced read length; μ and σ^2 are the mean and variance of $\lambda_n^{(m)}$ respectively. The first part, $p(R_{n,k}^{(m)} | Z_{n,k}^{(m)}, \Theta)$, represents the probability that a read can be aligned to a specific region of the k^{th} isoform conditioned on whether it is generated from this isoform. If the n^{th} read is generated from the k^{th} isoform, this read is assumed to be uniformly generated from one of all the possible positions in this isoform. Otherwise, $p(R_{n,k}^{(m)} | Z_{n,k}^{(m)}, \Theta)$ is 0. The number of all possible positions is $\tilde{l}_{n,k}^{(m)} = l_k - (2L + \lambda_{n,k}^{(m)}) + 1$, Then the conditional distribution is:

$$p(R_{n,k}^{(m)} = 1 | Z_{n,k}^{(m)}, \Theta) = \begin{cases} 1/\tilde{l}_{n,k}^{(m)} & \text{if } Z_{n,k}^{(m)} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The second part, $p(\lambda_{n,k}^{(m)} | \Theta)$, is the probability of observing a paired-end read with unsequenced length $\lambda_n^{(m)}$, which follows a normal distribution with mean μ and variance σ^2. Both μ and σ^2 can be given or estimated from the aligned RNA-seq data. As a result, we have the following generative process for each of M replicates (Figure 3.2):

1. $\boldsymbol{\psi}^{(m)} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
2. For each of $N^{(m)}$ reads:
   (a) $\mathbf{Z}_n^{(m)} \sim \text{Multinomial}(1, \boldsymbol{\psi}^{(m)})$

(b) $R_{n,k}^{(m)} \sim p(R_{n,k}^{(m)}|Z_{n,k}^{(m)}, \Theta)$

(c) $\lambda_{n,k}^{(m)} \sim \text{Normal}(\mu, \sigma^2)$

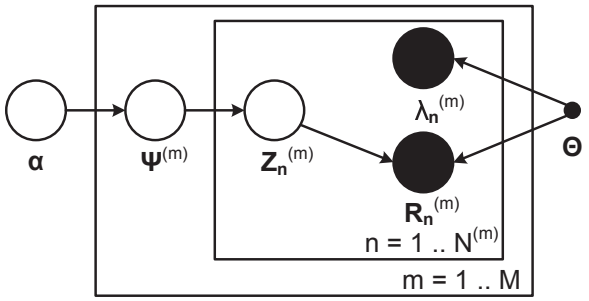


Fig. 3.2.: The graphical model representation of DEIsoM.

3.3.2 Estimation

To compute the posterior distribution of isoform fractions and read assignments,

$$p(\boldsymbol{\psi}, \mathbf{Z}|\mathbf{R}, \boldsymbol{\alpha}, \Theta) = \frac{p(\boldsymbol{\psi}, \mathbf{Z}, \mathbf{R}|\boldsymbol{\alpha}, \Theta)}{p(\mathbf{R}|\boldsymbol{\alpha}, \Theta)}$$

we need to compute the denominator:

$$p(\mathbf{R}|\boldsymbol{\alpha}, \Theta) = \prod_m \int p(\boldsymbol{\psi}^{(m)}|\boldsymbol{\alpha}) \prod_n \sum_k \left[p(Z_{n,k}^{(m)} = 1|\psi_k^{(m)}) \times p(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)}|Z_{n,k}^{(m)} = 1, \Theta) \right] d\boldsymbol{\psi}^{(m)}$$

which is computationally intractable, so we have to use approximate inference techniques, such as Markov Chain Monte Carlo (MCMC) sampling method or Variational Bayesian method. Classical MCMC methods may take a long time to converge due to the high correlation between the latent variables (Section 3.4.2). The Variational Bayesian method (Jordan et al., 1999) tends to be faster and better scalable to large data for many graphical models. The VB algorithm approximates the intractable posterior p by a proposed distribution q, where q belongs to a family of distributions controlled by the variational parameters. We can optimize the variational parameters

to minimize the Kullback-Leibler divergence between q and the posterior p, KL$(q||p)$. This is equivalent to maximizing a variational evidence lower bound. In such a way, the inference problem is cast to an optimization problem, which can be efficiently solved by gradient-based optimization algorithms.

For our model, we propose a family of variational distributions, which has the form:

$$q(\boldsymbol{\psi}, \mathbf{Z}) = \prod_m q(\boldsymbol{\psi}^{(m)}; \boldsymbol{\beta}^{(m)}) \prod_n q(\mathbf{Z}_n^{(m)}; \mathbf{r}_n^{(m)}),$$

where $q(\boldsymbol{\psi}^{(m)}; \boldsymbol{\beta}^{(m)})$ is a Dirichlet distribution parameterized by $\boldsymbol{\beta}^{(m)}$ and $q(\mathbf{Z}_n^{(m)}; \mathbf{r}_n^{(m)})$ is a multinomial distribution parameterized by $\mathbf{r}_n^{(m)}$.

We use the following iterative variational EM algorithm updates to find the optimal parameters for our model:

1. (E-step) For each replicate, estimate the variational parameters $\mathbf{r}_n^{(m)}$, $\boldsymbol{\beta}^{(m)}$;
2. (M-step) Maximize the variational evidence lower bound with respect to the hyperparameter $\boldsymbol{\alpha}$.

In E-step, we estimate the posterior distribution using a very commonly used algorithm, coordinate ascent variational inference (CAVI) (Bishop, 2006). We iteratively update:

$$r_{n,k}^{(m)} = \frac{\rho_{n,k}^{(m)}}{\sum_{l=1}^{K} \rho_{n,l}^{(m)}} \quad \text{and} \quad \beta_k^{(m)} = \alpha_k + \sum_{n=1}^{N^{(m)}} r_{n,k}^{(m)} \tag{3.1}$$

where

$$\begin{aligned} \boldsymbol{\rho}_{n,k}^{(m)} =& p\left(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)} | Z_{n,k}^{(m)} = 1, \Theta\right) \times \\ & \exp\left[\digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K} \beta_l^{(m)})\right] \end{aligned} \tag{3.2}$$

and $\digamma$ denotes the digamma function which is the derivative of the log-gamma function.

In M-step, we can use the Newton-Raphson method to update the hyperparameter $\boldsymbol{\alpha}$. This method is widely used for parameter estimation of models with Dirichlet priors (Ronning, 1989; Minka, 2000; Blei et al., 2003). Here, we initialize the hyperparameter $\boldsymbol{\alpha} = \mathbf{1}$. The Newton-Raphson method finds the stationary point of an objective function using the iterative updates:

$$\boldsymbol{\alpha}_{\text{new}} = \boldsymbol{\alpha}_{\text{old}} - \boldsymbol{H}(\boldsymbol{\alpha}_{\text{old}})^{-1}\boldsymbol{g}(\boldsymbol{\alpha}_{\text{old}}) \tag{3.3}$$

where $\boldsymbol{g}$ and $\boldsymbol{H}$ denote the gradient and the Hessian matrix of the objective function respectively. However, some new α_k may become non-positive during the iterative updates, which is invalid for Dirichlet distributions. Therefore, instead of working on $\boldsymbol{\alpha}$ directly, we update $\log(\boldsymbol{\alpha})$ first and then take the exponential of it. Let $\boldsymbol{\gamma} = \log(\boldsymbol{\alpha})$. The gradient and the Hessian of the variational lower bound with respect to $\boldsymbol{\gamma}$ can be computed as:

$$\begin{aligned} g_k(\boldsymbol{\gamma}) = & M \quad \digamma\Big(\sum_{l=1}^{K}\alpha_l\Big) - \digamma(\alpha_k)\Big)\Big(\alpha_k + \\ & \sum_{m=1}^{M}\Big(\alpha_k \quad \digamma(\beta_k^{(m)}) - \digamma\Big(\sum_{l=1}^{K}\beta_l^{(m)}\Big)\Big)\Big(\end{aligned} \tag{3.4}$$

$$H_{i,j}(\boldsymbol{\gamma}) = M \quad \digamma'\Big(\sum_{l=1}^{K}\alpha_l\Big)\alpha_i\alpha_j\Big)\Big(+ \sigma(i,j)\Delta_i(\alpha) \tag{3.5}$$

where we define $\sigma(i,j) = 1$ if $i = j$, otherwise $\sigma(i,j) = 0$, $\digamma'$ is the trigamma function, and

$$\begin{aligned} \Delta_i(\boldsymbol{\alpha}) = M \quad & \digamma\Big(\sum_{l=1}^{K}\alpha_l\Big) - \digamma'(\alpha_i)\alpha_i - \digamma(\alpha_i)\Big)\Big(\alpha_i \\ & + \alpha_i \sum_{m=1}^{M} \quad \digamma(\beta_i^{(m)}) - \digamma\Big(\sum_{l=1}^{K}\beta_l^{(m)}\Big)\Big)\Big(\end{aligned} \tag{3.6}$$

A drawback of taking the logarithm is that we can no longer use the special structure of Hessian to compute $\boldsymbol{H}^{-1}\boldsymbol{g}$ efficiently as in Blei et al. (2003). Since Hessian computation can be expensive for large K, we update $\boldsymbol{\gamma}$ with L-BFGS method using the gradient only. Updates for $\boldsymbol{\alpha}$ will terminate when the maximum number of iterations is reached or the change in evidence lower bound is smaller than our threshold.

3.3.3 Identification

The DE level of an isoform can be represented as the difference between the posterior distributions of isoform fractions under two conditions. As used in the Variational Bayesian method, KL divergence measures the difference between any two distributions. Therefore, we compute the KL divergence between the posterior distributions of isoform fractions under the two conditions to evaluate the DE level of the isoforms. A higher KL divergence implies that the isoforms of this gene are more differentially expressed under the two conditions. Specifically, we train the model and estimate the posterior distribution $p(\boldsymbol{\psi}|\mathbf{R}, \boldsymbol{\alpha}, \Theta)$ with data from healthy and diseased conditions respectively. As described in Section 3.3.2, although the exact posterior distribution cannot be computed, we use the approximate posterior distributions from two conditions, $q(\boldsymbol{\psi}; \boldsymbol{\beta})$ and $q'(\boldsymbol{\psi}'; \boldsymbol{\beta}')$, to compute the KL divergence. Because $q(\boldsymbol{\psi}^m)$ or $q'(\boldsymbol{\psi}^m)$ are independent Dirichlet distributions, the KL divergence, D_{KL} can be computed analytically as:

$$D_{KL}(q||q') = \sum_{m=1}^{M} \Bigg\{ \Bigg(\log \frac{\sum_{k=1}^{K} \boldsymbol{\beta}_k^{(m)}}{\sum_{k=1}^{K} \boldsymbol{\beta}_k'^{(m)}} + \sum_{k=1}^{K} \log \frac{\Gamma(\boldsymbol{\beta}_k'^{(m)})}{\Gamma(\boldsymbol{\beta}_k^{(m)})} + \sum_{k=1}^{K} [\boldsymbol{\beta}_k^{(m)} - \boldsymbol{\beta}_k'^{(m)}][F(\boldsymbol{\beta}_k^{(m)}) - F(\sum_{l=1}^{K} \boldsymbol{\beta}_l^{(m)})] \Bigg\} \Bigg(\tag{3.7}$$

To remove the asymmetry of D_{KL} between two conditions, we further compute the Jensen-Shannon divergence $D_{JS} = \frac{1}{2}[D_{KL}(q||q') + D_{KL}(q'||q)]$.

3.4 Simulations

In this section, we present four simulation studies to test that (1) whether DEIsoM benefits from the shared information from the multiple biological replicates compared with alternative methods; (2) whether the VB inference speeds up the computation without loss of accuracy; (3) whether DEIsoM is robust to different simulation settings; (4) whether the quantification of DEIsoM outperforms alternative methods under a more realistic setting.

3.4.1 Comparison of five methods on synthetic data

To test whether the shared information contributes to DE isoform detection, we generate synthetic data and compare DEIsoM with four commonly used programs: Cufflinks (v2.2.1), MISO (v0.5.3), RSEM (v1.2.30), and BitSeqVB (v0.7.5). The synthetic data are generated as follows. We first randomly select 200 genes (1395 isoforms) from the annotation of chromosome 1 in the hg19 human reference genome, in which 100 genes are labeled as containing DE isoforms and the rest are non-DE. To make the synthetic data more realistic, we sample the expression levels of genes from a log-normal distribution (Gierliński et al., 2015). Isoform fractions are generated from a symmetric Dirichlet distribution with $\boldsymbol{\alpha} = \mathbf{1}$, which means the chance of sampling any fraction of isoforms is equally probable. For instance, if there are three isoforms, the probability of sampling the isoform fraction as (0.1, 0.2, 0.7) is the same as (0.2, 0.3, 0.5). For DE isoforms, we draw two different samples for two conditions respectively; for non-DE, we draw only one sample shared by both conditions. To model the variation among replicates, we add Gaussian noise with a standard deviation equal to 10% of the expression level of each replicate. According to Standards, Guidelines and Best Practices for RNA-Seq V1.0[1], the number of paired-end RNA-Seq reads used in current studies is around 30 million per replicate. And for each tissue, it is generally

[1]Standards, Guidelines and Best Practices for RNA-Seq V1.0 can be found at: `https://genome.ucsc.edu/ENCODE/protocols/dataStandards/ENCODE_RNAseq_Standards_V1.0.pdf`

expected more than $10,000$ genes are expressed (Consortium, 2015). Following the above empirical read numbers, we generate $600,000$ RNA-Seq reads for 200 genes using RNASeqReadSimulator [2] for each of five replicates in both conditions, using default settings.[3] To test the robustness of DEIsoM, we repeat the above simulation process 10 times. For RSEM, BitSeq, MISO, and DEIsoM, the simulated reads are mapped back to the reference transcriptom using Bowtie2 (Langmead and Salzberg L, 2012). For Cuffdiff, the reads are mapped back to the hg19 reference genome using Tophat (Trapnell et al., 2009). The machine used to run all experiments has two 8-Core Intel Xeon-E5 processors and 64GB memory.

First, we compare the quantification performance of DEIsoM with MISO, Cuffdiff, RSEM and BitSeqVB in terms of the correlations between the predicted isoform fractions and the ground truth on the synthetic data. Figure 3.3 (A) summarizes the means and the standard errors of the correlation coefficients in 10 replicates. They show that the correlation coefficients in DEIsoM is higher than the alternative methods.

Second, we compare the DE isoform identification performance of DEIsoM with MISO, Cuffdiff, RSEM-EBSeq, and BitSeqVB in terms of the AUC (Area Under Curve) of ROC (Receiver Operating Characteristic) curves on the synthetic data. The ROC curves are computed based on different ranking criteria for the four methods. DEIsoM uses the KL divergence; MISO uses both the average of Bayes factors of all pairs of subjects (MISO-BF) and the average of KL divergences of posteriors of isoform factions (MISO-KL); Cuffdiff uses a log-fold-change based p-value; RSEM-EBSeq uses the PPDE (Posterior Probability of Differential Expression); BitSeqVB uses the PPLR (Probability of Positive Log Ratio). And we choose the "isoform-centric" mode for MISO. Also, PPLR is more sensitive to the upregulated DE isoforms than the downregulated ones by definition. Figure 3.4 (A) shows the ROC curves

[2]RNASeqReadSimulator is available at: `http://alumni.cs.ucr.edu/~liw/rnaseqreadsimulator.html`

[3]Our simulation code is available at: `https://github.com/hao-peng/DEIsoM/tree/master/simulation`

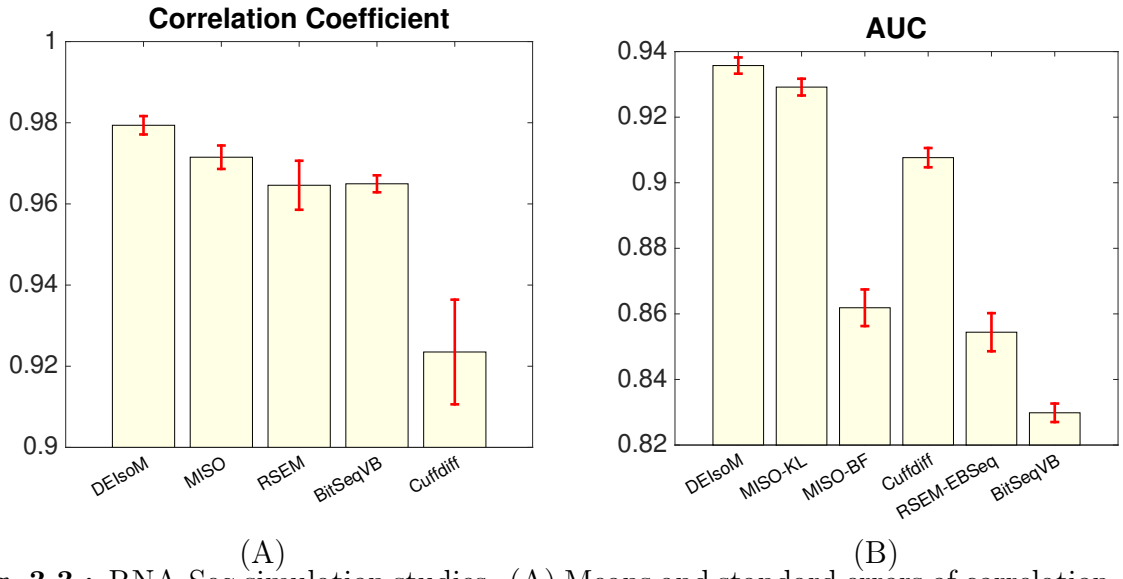
for one of the 10 repeated experiments. Figure 3.3 (B) summarizes the means and standard errors of the AUCs over 10 runs. They show that DEIsoM consistently outperforms MISO-BF, MISO-KL, Cuffdiff, and RSEM-EBSeq on the synthetic data under our settings.

Third, we compare the CPU time of DEIsoM, Cuffdiff, RSEM-EBSeq and BitSeqVB. The time we count is from the point we give the alignment files as input to the point that the programs generate the quantification results. We summarize it in Supplementary Table 3.2 for one run of the simulated data and the real data which will be discussed in Section 3.5. The numbers of hours used by the three algorithms are comparable, where Cuffdiff is always the fastest in all methods. However, DEIsoM has better DE isoform identification and quantification performance than the alternative methods, which is shown in both Section 3.4 and Section 3.5.

3.4.2 Comparison of VB and MCMC on synthetic data

To test whether the VB inference algorithm speeds up the computation over MCMC sampling without loss of accuracy, we compare the ROC curves and running time of the two implementations. We set the maximum iteration number as 1500 for both VB and MCMC. The burn-in time of MCMC is 150 iterations. Note that the MCMC sampling here is not completely the same as MISO. MISO combines the Metropolis-Hasting algorithm with a Gibbs sampler. We follow the same approach to estimate $\boldsymbol{\psi}$, but we iteratively sample $\boldsymbol{\alpha}$ from its posterior distribution given a non-informative prior which depends on all five replicates. Details of our MCMC sampling method are described in the Supplementary 3.8.1. The VB inference shows an advantage over MCMC in both the ROC curve and computing time within the limited number of iterations. Figure 3.4(B) shows the ROC curves for both implementations; VB inference achieves an AUC=0.9445 in 1.4 CPU hours, whereas the MCMC method has AUC=0.8844 in 56 CPU hours. Although MCMC theoretically can give samples from the exact target posterior distribution, it converges slowly on

this dataset, which may cause inaccurate predictions and long running time. However, VB usually converges before the limit is reached under the same number of maximum iterations. Therefore, the VB method achieves a faster speed and a higher accuracy than the MCMC sampling.

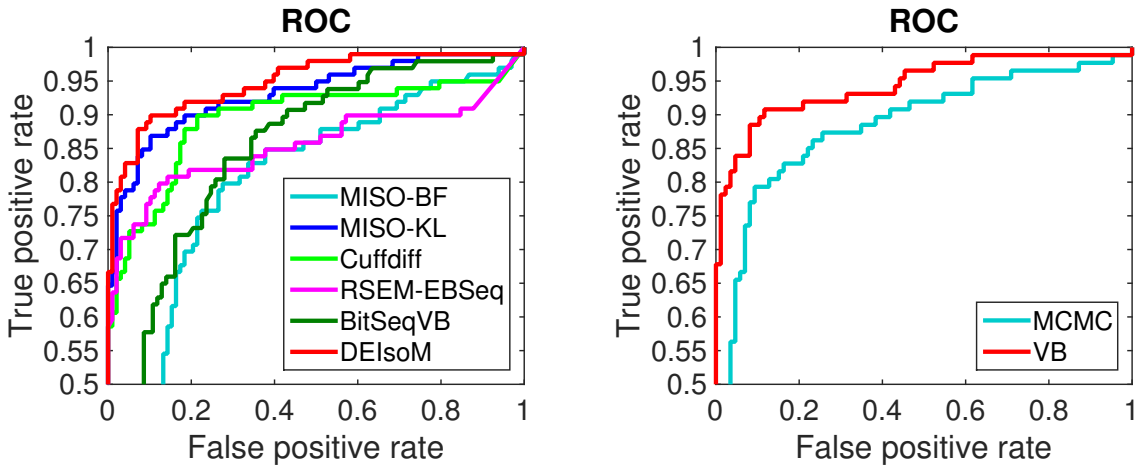


(A) (B)

Fig. 3.3.: RNA-Seq simulation studies. (A) Means and standard errors of correlation coefficients between the estimation and the ground truth in 10 replicates, using DEIsoM, MISO, RSEM, BitSeqVB and Cuffdiff. (B) Means and standard errors of AUCs of 10 repeated simulations for DEIsoM, MISO-KL, MISO-BF, Cuffdiff, RSEM-EBSeq and BitSeqVB.

3.4.3 Comparison of sensitivity of five methods

To demonstrate the robustness of DEIsoM, we vary the parameter of Dirichlet distribution $\boldsymbol{\alpha}$ used for generating isoform fractions. When we increase $\boldsymbol{\alpha}$, the variance of generated isoform fractions under two conditions becomes smaller, but the mean remains the same. As a result, the difficulty of distinguishing DE genes from non-DE genes increases. In this experiment, we set $\boldsymbol{\alpha} = \mathbf{1}$, $\mathbf{3}$ and $\mathbf{5}$ and keep the other settings unchanged to simulate the data. We test all above five methods on the simulated reads to see whether they are sensitive to the change of $\boldsymbol{\alpha}$. Table 3.1 shows that as $\boldsymbol{\alpha}$ increases, the AUCs of all methods decrease, since the task becomes harder.



(A) (B)

Fig. 3.4.: RNA-Seq simulation studies. (A) ROC curve comparison of MISO, Cuffdiff, RSEM-EBSeq, BitSeqVB and DEIsoM from one run of 10 repeated experiments. For MISO we use two evaluation methods, MISO-KL and MISO-BF. MISO-KL denotes the average of KL divergences of the posteriors of isoform fractions. MISO-BF denotes the average of Bayes factors. (B) ROC curve comparison of VB and MCMC implementations of DEIsoM on the same dataset.

However, DEIsoM consistently outperforms the alternative methods throughout all $\boldsymbol{\alpha}$ settings.

$\boldsymbol{\alpha}$	**1**	**3**	**5**
MISO-BF	0.849	0.727	0.673
MISO-KL	0.912	0.878	0.844
Cuffdiff	0.890	0.834	0.815
RSEM-EBSeq	0.873	0.798	0.762
BitSeqVB	0.807	0.771	0.704
DEIsoM	**0.931**	**0.915**	**0.887**

Table 3.1.: AUCs for MISO, Cuffdiff, RSEM-EBSeq, BitSeqVB, and DEIsoM on simulated data with different $\boldsymbol{\alpha}$.

3.4.4 Comparison of abundance estimation

To test the quantification performance of DEIsoM under a more realistic setting, we simulate RNA-Seq reads using real data. Two RNA-Seq datasets of human stom-

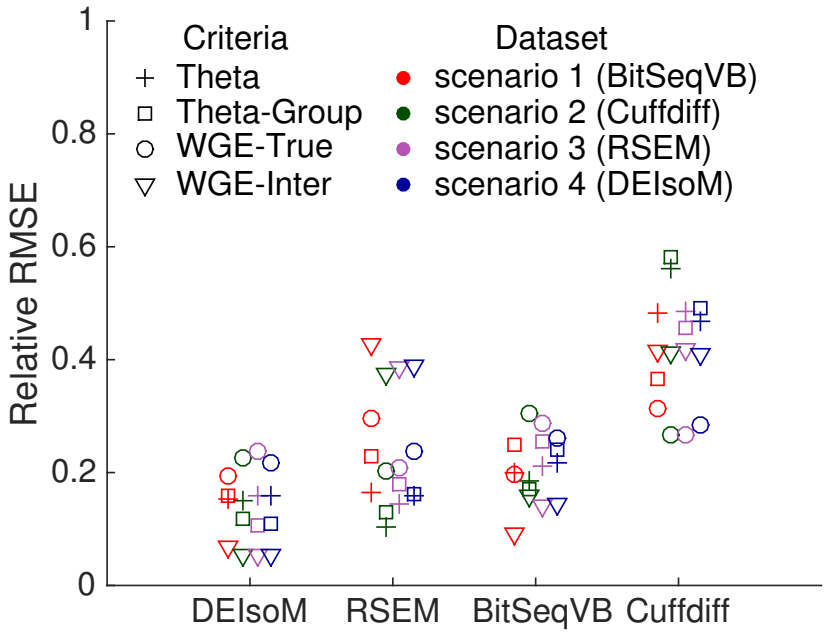


Fig. 3.5.: Relative root mean squared errors of DEIsoM, RSEM, BitSeqVB and Cuffdiff on four simulated datasets. Theta: estimated transcript fractional expression compared with the ground truth for all the replicates. Theta-Group: mean estimated transcript fractional expression of the whole group compared with the true group mean. WGE-True: within-gene relative estimates compared with the ground truth. WGE-Inter: inter-replicate consistency of within-gene relative estimates.

ach tissue were chosen from the ENCODE project[4]. Following the same percedure in Hensman et al. (2015), we estimate the abundance of 196, 317 transcripts using four models, RSEM, Cuffdiff, BitSeqVB and DEIsoM, as the ground truth for each scenario. By feeding the ground truth to Spanki (Sturgill et al., 2013), we generate about 10 millions paired-end reads for each of the five replicates under each scenario. Four different evaluation criteria are used, see Supplementary 3.8.2: Theta, Theta-Group, WGE-True and WGE-Inter. Theta measures the accuracy of transcript fraction estimation for all the replicates; Theta-Group measures the accuracy of transcript fraction estimation for the whole group; WGE-True measures the accuracy of within-gene relative fractional estimation; WGE-Inter measures the predictive con-

[4]The datasets from ENCODE project can be found at: `https://www.encodeproject.org/experiments/ENCSR853WOM/` and `https://www.encodeproject.org/experiments/ENCSR752UNJ/`

sistency among all replicates. Figure 3.5 summarizes the relative root mean square errors (RMSE) of DEIsoM, RSEM, BitSeqVB and Cuffdiff on four simulated datasets. They show that the DEIsoM RMSEs in both Theta-Group and WGE-Inter are lower than the other three methods, indicating that DEIsoM tends to give more consistent and accurate estimates for the whole condition. This result is consistent with the one in (Hensman et al., 2015). A similar result evaluated by the relative mean absolute errors (MAE) is shown in Supplementary Figure 3.9.

3.5 Real data experiments and results

In this section, we test whether DEIsoM successfully identifies DE isoforms in real data. We apply DEIsoM and alternative programs to a Hepatocellular Carcinoma (HCC) RNA-seq dataset, and evaluate the predicted DE isoforms by PCA, read coverage visualization, and comparison to the biological literature. Aberrant alternative splicing is known to be involved in HCC (Berasain et al., 2010), so DE isoforms should be present.

3.5.1 Data pre-processing

RNA-seq data was collected from nine pairs of HCC tumors and their matched adjacent normal tissues (Sung et al., 2012) (Kan et al., 2013). The mRNA of each sample was extracted, amplified and sequenced using the Illumina HiSeq 2000 platform. 150 base paired-end reads were generated and aligned to the hg19 human reference genome using RUM (RNA-Seq Unified Mapper) (Grant et al., 2011). The aligned reads are used as input to three methods, Cuffdiff, RSEM-EBSeq, and DEIsoM, for DE isoform detection. MISO is not included because it cannot perform a group-wise analysis.

3.5.2 PCA

Because there is no exact ground truth for the HCC real data, we evaluate the quantification ability of each method by PCA plots. We first choose 38 significantly DE genes that are verified by polymerase chain reaction (PCR) from the previous publications (Dong et al., 2009; Wang et al., 2015; Huang et al., 2017; Wang et al., 2017). For each gene, we sum up the Fragments Per Kilobase of transcript per Million mapped reads (FPKM) of all the child isoforms as the gene expression. If the gene/isoform expressions associated with the HCC are correctly estimated, these gene/isoforms can be used as features to distinguish between the normal and tumor samples in PCA plots. Figure 3.6 shows that DEIsoM and RSEM can linearly separate tumor samples from their matched normal samples; BitSeqVB has one tumor sample (9) very closed to the normal cluster; Cuffdiff misses three tumor samples (4,5,6) in the normal cluster.

3.5.3 Read coverage visualization

To understand the expression patterns of the DE isoforms selected by DEIsoM, we visualize the read coverage on the hg19 reference genome. Because it may be possible to align a read to multiple isoforms, it is hard to determine the exact expression level of each isoform from the read coverage visualization. But it is possible to tell the change in isoform expression in some cases. A previous study successfully identified the genes with DE isoforms by testing the difference in read coverage between two conditions (Stegle et al., 2010). Following the same logic, we assume that if the read coverage of a gene is similar in the two conditions, the isoforms of that gene will be predicted as non-DE. Otherwise, they are more likely to be DE.

First, we examine the read coverage of IGF2, a gene identified by DEIsoM as having DE isoforms. IGF2 is the 2^{nd} most DE gene identified by DEIsoM. Eight isoforms of IGF2 have been observed according to the human transcriptome annotation. Figure 3.7 (A, B) shows the read coverage of IGF2 in nine pairs of normal and tumor

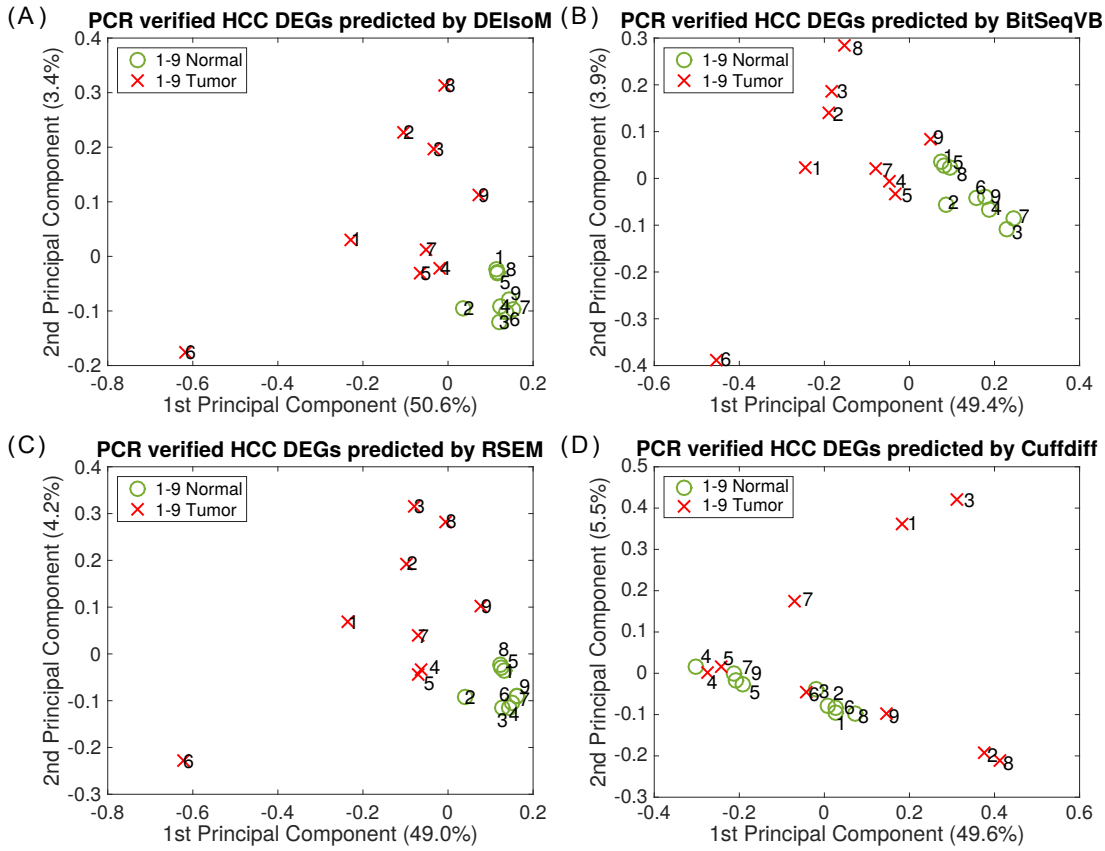


Fig. 3.6.: PCA plots for nine pairs of HCC samples and their matched normal samples. Each sample is represented by a vector with 38 gene expressions. All these 38 genes are PCR verified DE genes in HCC. A, B, C, D are PCA plots using the estimations from DEIsoM, BitSeqVB, RSEM and Cuffdiff, respectively. Circle: normal sample. Cross: tumor sample. Percentage: the proportion of variance of the corresponding principle component.

samples. Note that the reads aligned to the last two exons (in the box) can only contribute to isoform 4 (ENST00000300632). Figure 3.7(B) shows that the absolute numbers of reads aligned to the last two exons in all tumor samples are much lower than that in normal samples. Figure 3.7(C) is the same as Figure 3.7(B) but with an automatically scaled y-axis. (C) shows that in eight of nine tumor samples (1T, 2T, 4T – 9T), the fractions of reads aligned to the last two exons are much lower in the HCC samples than that in the normal samples. This indicates that IGF2 isoform 4 is down-regulated in HCC tumors. However, in the Cuffdiff results, this isoform has

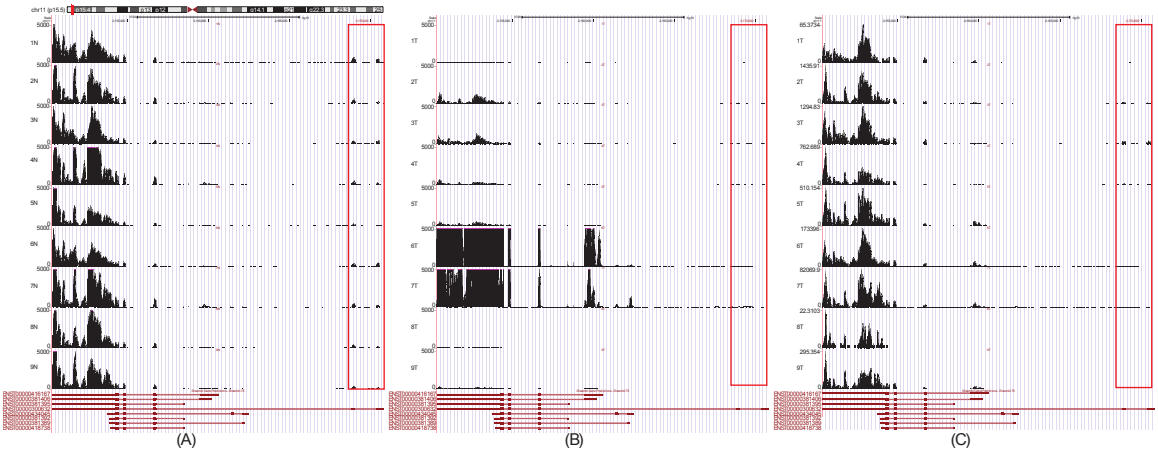


Fig. 3.7.: Read coverage of IGF2 – a top selection by DEIsoM. The data was normalized across replicates by scaling the total number of reads to that of 1N (replicate 1 under normal condition). (A) Read coverage patterns of nine normal samples with y-axis scaled to 5000. (B) Read coverage patterns of nine matched tumor samples with y-axis scaled to 5000. (C) is the same as (B) but uses an automatically scaled y-axis. This illustrates that 1-5 and 8-9 tumor samples have very low read abundance in the last two exons, and the low signals are not due to the imposition of a fixed large y-axis scale. The exon positions of eight isoforms are listed under each panel.

a p-value of 0.039 with rank 95; in RSEM-EBSeq, the PPDE equal to 1 out of 1147 DE isoforms all with PPDE = 1. But if we further rank the RSEM-EBSeq result by transcript real fold change (condition 1 over condition 2) as recommended, this isoform ranks 671 out of 1147 DE isoforms.

Second, we show the read coverage of IGF2BP1, a gene identified by Cuffdiff as having DE isoforms. Isoform 1 (ENST00000290341) of IGF2BP1 is the 6^{th} most DE gene. Supplementary Figure 3.10 shows the read coverage of IGF2BP1 in normal and tumor samples. Note that the reads aligned to the last exon only contribute to isoform 1 (the box indicates the last exon). However, only four of nine tumor samples show moderate differential expression of isoform 1 (lower than 500), and the expression level is near zero in all normal samples and five of the tumor samples (1T – 4T, 8T). Cuffdiff evaluates DE level using the log-fold-change between the conditions. This "fold" will be extremely large when the expression of one condition is near zero and the other is slightly higher. However, due to the low count numbers in both conditions, the confidence of calling this gene as having DE isoforms is low. Often, an empirical value is set to avoid low signals (NOTEST or LOWDATA). On the contrary, DEIsoM ranks IGF2BP1 as 244. Because both large sample variance and low read coverage lead to relatively "flat" posterior distributions in both normal and tumor conditions, which are close to the prior distribution. Thus, the KL divergence between two posterior distributions is small and the isoforms are not identified as DE.

Lastly, we visualize the five least differentially expressed isoforms identified by DEIsoM, showing that the low ranked isoforms have very similar read coverage patterns in both normal and tumor samples. Supplementary Figure 3.11 shows COX16 has a similar read coverage pattern among all samples in both normal and tumor conditions. This is because a low KL divergence requires a high similarity between two posterior distributions of isoform fraction.

3.5.4 Biological relevance of predicted DE isoforms

To further understand the functions of DE isoforms selected by DEIsoM, we examine whether they are supported by HCC relevant literature. PubMed searches were performed using the keywords "gene name + hepatocellular carcinoma". Since most current experimental work focuses on the expression levels of genes rather than isoforms, we associate the DE isoforms identified by DEIsoM, Cuffidiff and RSEM-EBSeq with their gene names. Also, we assume that if the expression of a gene changes, it is very likely caused by a change of its isoforms. DE isoforms/genes are then categorized into four groups (3, 2, 1, 0) according to their relevance to HCC. "Category 3" refers to a gene whose function in HCC has been well studied and can be used as a potential biomarker for prognosis or diagnosis. "Category 2" indicates that differential expression of a gene has been detected *in vivo*, but not used as a biomarker. "Category 1" indicates a gene whose function has only been studied *in vitro* but not in patient biopsies. "Category 0" indicates a gene for which we found no HCC relevant literature.

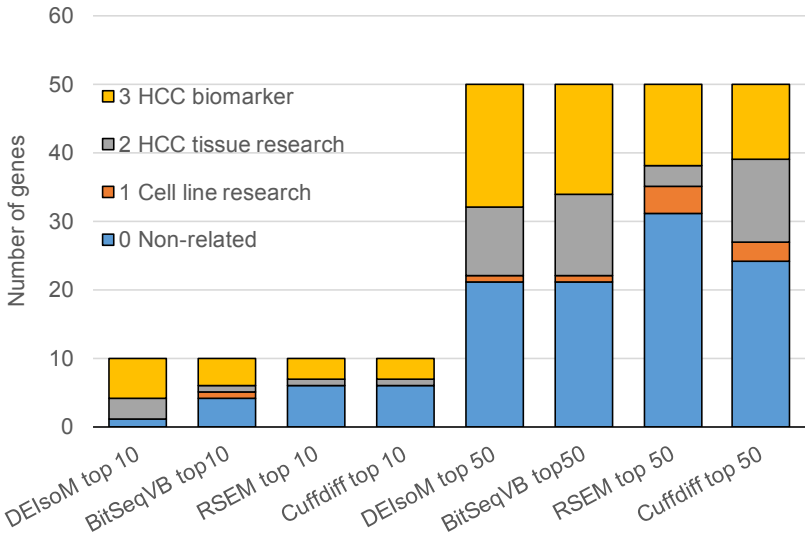


Fig. 3.8.: HCC relevance of DE isoforms identified by DEIsoM, BitSeqVB, Cuffdiff and RSEM-EBSeq. Relevance is defined as Category 3: HCC biomarkers, Category 2: DE genes verified in HCC tissues, Category 1: DE genes verified in HCC cell lines, and Category 0: HCC non-related genes. We analyze both the top 10 and top 50 selections for all four methods.

First, we compare the number of genes that are HCC biomarkers (Category 3) in the predictions by DEIsoM, BitSeqVB, RSEM-EBSeq and Cuffdiff (the first four columns in Figure 3.8). In the top 10 lists, more genes are identified as HCC biomarkers by DEIsoM than BitSeqVB, RSEM-EBSeq or Cuffdiff. Specifically, 6/10 genes identified by DEIsoM (ASS1, TTR, IGF2, AHSG, GPC3, CRP) vs. 4/10 genes identified by BitSeqVB (GPC3, AFP, IGF2BP3, UBE2C), 3/10 genes identified by Cuffdiff (SKP2, C-FOS, SOCS2) and 3/10 genes identified by RSEM-EBSeq (PEG10, TERT, ACAN) and belong to Category 3.

Second, we have examined the six specific HCC biomarkers status (ASS1,TTR, IGF2, AHSG, GPC3, CRP) in the top 10 list of DEIsoM. Specifically, ASS1 is detected to be down-regulated in HCC liver samples, which can be used to predict metastatic relapse with a high sensitivity and specificity (Tan et al., 2014); TTR is down-regulated in HCC patient serum (Qiu et al., 2008); Seon-Hee Yim *et al.* (Yim and Chung, 2010) state that both IGF2 and GPC3 are effective biomarkers for HCC – particularly, circulating IGF2 mRNA is positive in 34% of HCC patients and 100% correlated with the extrahepatic metastasis; GPC3 has been reported to interact with the Wnt signaling pathway to stimulate cell growth in HCC; GPC3 has also been used combined with PEG10, MDK, SERPINI1, and QP-C as a classifier that successfully distinguishes noncancerous hepatic tissues from HCCs (Yim and Chung, 2010); AHSG combined with two other HCC-associated antigens – KRT23 and FTL – can be used to diagnose HCC with sensitivity up to 98.2% in joint tests and specificity up to 90.0% in serial tests. (Wang et al., 2009); CRP, an inflammatory cytokine, is highly expressed in HCC and its expression is correlated with tumor size, Child-Pugh function and survival time (Jang et al., 2012).

Generally, DEIsoM ranks genes/isoforms highly associated with HCC on the top. In the top 10 list (the first four columns in Figure 3.8), 60% of genes identified by DEIsoM as having DE isoforms are experimentally proven HCC biomarkers (Category 3), and 90% are HCC biomarkers plus DE genes verified *in vivo* (Category 3 + 2). On the contrary, BitSeqVB, RSEM-EBSeq and Cuffdiff show a lower performance

than DEIsoM – 30% to 40% of genes having DE isoforms that are experimentally proved HCC biomarkers (Category 3), and 40% to 50% are HCC biomarkers plus DE genes verified *in vivo* (Category 3 + 2).

Even if we expand this search to top 50 lists (the fifth column in Figure 3.8 and Supplementary Table 3.5), DEIsoM still identifies 18 genes (36%) as HCC biomarkers, and 10 genes (20%) as DE genes verified *in vivo*. However, BitSeqVB, RSEM-EBSeq and Cuffdiff identify fewer literature proven DE genes than DEIsoM in the top 50 list (the last three columns in Figure 3.8 and Supplementary Table **??**). BitSeqVB identifies 16 genes (32%) as HCC biomarkers, 12 genes (24%) as DE genes *in vivo*; RSEM-EBSeq identifies 12 genes (24%) as HCC biomarkers and 3 genes (6%) as DE genes verified *in vivo*; Cuffdiff identifies 11 genes (22%) as HCC biomarkers, 12 genes (24%) as DE genes *in vivo*. Therefore, DEIsoM has a clear superior ability to select DE genes that are supported by the published literature.

Moreover, the isoforms of four genes (FGFR2, survivin, ADAMTS13 and CD44) identified as DE by DEIsoM have been found to be up or down-regulated in HCC. This provides additional support for DE genes identified by DEIsoM. In the case of FGFR2 (ranked 62 of 11950 genes), the FGFR2-IIIb isoform is down-regulated and has been related to HCC aggressive growth, while the FGFR2-IIIc isoform is expressed at the same level in normal and HCC tissues (Amann et al., 2010). All three isoforms of survivin (ranked 120 of 11950 genes), survivin normal, survivin 2B and survivin Delta Ex3 have been detected in well, moderately and poorly differentiated HCC but none of these are found in normal tissues (Takashima et al., 2005). RT-PCR results are available for ADAMTS13 (ranked 201 of 11950 genes) showing differences in the expression of three known isoforms (WT and 1, 2) between normal liver tissue and hepatoma cell lines (Shomron et al., 2010). For CD44 (ranked 607 of 11950 genes), CD44-v6 is up-regulated in HCC, while CD44 standard form remains stable (Zhang et al., 2010).

To more clearly understand the performance of different methods, we also examine the overlapping DE genes in the top 200 lists from the compared methods.

Supplementary Table 3.3 shows the overlapping DE genes by feeding the FPKM of all isoforms from each method to EBSeq. This tests the quantification similarity between any two methods. According to the number of overlapping DE genes, the quantification performance of RSEM and BitSeqVB are the most similar, followed by RSEM and DEIsoM. Supplementary Table 3.4 shows the overlapping DE genes using the DE evaluation methods of their own. This tests the performance of both the quantification and DE identification. After changing the DE evaluation method, the number of overlapping DE genes between RSEM and BitSeq decreases from 96 to 62, while this number between RSEM and DEIsoM decreases from 74 to 14, which suggests that KL divergence performs differently from PPDE or PPLR. PPDE and PPLR are only sensitive to the absolute abundance change of an isoform, while KL divergence is sensitive to the overall isoform fractional pattern change within a gene, not limited to the absolute abundance change. This is useful in searching isoform switching events in many cases.

3.6 Discussion

In contrast to the models that treat each biological replicate separately, DEIsoM incorporates all biollogical replicates in one seamless framework. By capturing the shared information across multiple biological replicates, DEIsoM achieves a higher prediction accuracy and inter-replicate consistency than the alternative methods in the simulation studies (Section 3.4.1, 3.4.3, 3.4.4). This shared information comes from the intrinsic fact that all the replicates in one condition share the same underlying biological mechanism. As described in model construction (Section 3.3.1), we use a Dirichlet prior to represent a base fraction–which is characterized by the hyperparameter $\boldsymbol{\alpha}$ and learned from data—and then sample the instance-specific fraction for each replicate. The fractions for different replicates are not necessarily the same — because we allow some within-condition variance — however, those fractions retain underlying coherence since they are sampled from the same Dirichlet prior (or the

base fraction). In addition, as the conjugate prior for the multinomial distribution, the Dirichlet prior enables close-form, efficient updates in our VB inference, which greatly benefits the computation. Furthermore, faster computing speed is gained using the VB algorithm, instead of the MCMC sampling used in MISO, during the inference step. The VB method converts a sampling problem to an optimization problem and speeds up the estimation (Section 3.4.2). DEIsoM is also promising in real applications. On the HCC dataset, by PCA plotting, we find that the normal and tumor samples can be linearly separated by the estimated expression levels of PCR verified DE genes, suggesting an accurate quantification of DE isoforms in DEIsoM. Using read coverage visualization, we find that the DEIsoM KL divergence is capable of identifying isoforms whose read coverage patterns change, and does not give false positive results for isoforms with low read abundance in both conditions. This property is desirable in practice, since a low number of reads causes a large uncertainty in estimation. In DEIsoM, the posterior distributions of both conditions are close to the uniformly distributed prior if the read number is low, which reduces the KL divergence between the two conditions. However, neither Cuffdiff nor RSEM-EBSeq will automatically prune such isoforms (Section 3.6, 3.7). Moreover, a great number of isoforms predicted to be DE by DEIsoM are supported by the biological literature, providing encouraging results for real applications.

However, there are still some improvements that could be incorporated into DEIsoM. First, DEIsoM builds on the approach of MISO, which considers the quantification of isoforms gene by gene. In order to handle the reads multi-mapped to different gene loci, we have also added a variant version of DEIsoM that simultaneously considers all transcript isoforms, rather than performing a gene by gene analysis. This enhancement will allow the inclusion of multiply mapped reads into the analysis. However, the KL divergence is not applicable to this version, since KL divergence measures the isoform pattern change within a gene. Second, the KL divergence as a DE evaluation method is not based on a hypothesis test, but rather on the difference of the posterior distributions of fractional isoform expression between two conditions,

so it only provides a rank instead of p-values to infer "significantly" DE genes. However, KL divergence is sensitive to the overall isoform pattern change within a gene, and more differentiable for ranking isoforms/genes than p-values, which tend to give the same rank to many genes. DEIsoM allows the estimated isoform levels to be reported as FPKM, thereby allowing p-values to be calculated by many existing differential expression analysis methods. Lastly, DEIsoM assumes a known reference genome/transcriptome and the uniform read distribution. The misannotation or the non-uniformity of the read data may compromise the estimation accuracy in DEIsoM. We are considering including the novel isoform construction and the modeling of non-uniformly distributed read data into our future versions.

3.7 Conclusion

We propose a hierarchical Bayesian model, DEIsoM, for detecting DE isoforms using multiple biological replicates from two conditions. DEIsoM captures the information shared across replicates, and provides fast and accurate prediction compared to alternative methods in simulations. On the HCC real dataset, the estimated expression levels of PCR verified DE genes can be used as features to separate the tumor samples from their matched normal samples in PCA plots; read coverage visualization confirms that DEIsoM KL divergence is capable of identifying DE isoforms. DEIsoM is relatively resistant, compared to alternative methods, to identifying isoforms with low read abundance in both conditions as DE. Biological literature review suggests that the DE isoforms selected by DEIsoM have high relevance to HCC.

3.8 Supplementary materials

3.8.1 Model

Derivations for the variational Bayesian inference

First we compute a variational lower bound for the log model evidence:

$$\log p(\mathbf{R}|\boldsymbol{\alpha},\Theta) = \log \prod_m \Bigg(\int p(\boldsymbol{\psi^{(m)}}|\boldsymbol{\alpha}) \prod_n \sum_{\boldsymbol{Z}_n^{(m)}} \Big[p(\boldsymbol{Z}_n^{(m)}|\psi_k^{(m)}) p(\boldsymbol{R}_n^{(m)}, \lambda_n^{(m)}|\boldsymbol{Z}_n^{(m)},\Theta) \Big] d\boldsymbol{\psi}^{(m)} \tag{3.8}$$

$$\geq \sum_{m=1}^{M} \Big\{ \mathrm{E}_q[\log p(\boldsymbol{\psi^{(m)}}|\boldsymbol{\alpha})] - \mathrm{E}_q[\log q(\boldsymbol{\psi^{(m)}})] + \sum_{n=1}^{N^{(m)}} \Big[\mathrm{E}_q[\log p(\boldsymbol{Z}_n^{(m)}|\psi_k^{(m)})] + \mathrm{E}_q[\log p(\boldsymbol{R}_n^{(m)}, \boldsymbol{\lambda}_n^{(m)}|\boldsymbol{Z}_n^{(m)},\Theta)] - \mathrm{E}_q[\log q(\boldsymbol{Z}_n^{(m)})] \Big] \Bigg(\tag{3.9}$$

$$\begin{aligned} =& \sum_{m=1}^{M} \Big\{ \log\Gamma(\sum_{k=1}^{K}\alpha_k) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k - 1)(\digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K}\beta_l^{(m)})) \\ & - \log\Gamma(\sum_{k=1}^{K}\beta_k^{(m)}) + \sum_{k=1}^{K}\Gamma(\beta_k^{(m)}) - \sum_{k=1}^{K}(\beta_k^{(m)} - 1)(\digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K}\beta_l^{(m)})) \\ & + \sum_{n=1}^{N^{(m)}} \Big[\sum_{k=1}^{K} r_{n,k}^{(m)} (\digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K}\beta_l^{(m)})) \\ & + \sum_{k=1}^{K} r_{n,k}^{(m)} \log p(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)}|Z_{n,k}^{(m)} = 1, \Theta) \\ & - \sum_{k=1}^{K} r_{n,k}^{(m)} \log r_{n,k}^{(m)} \Big] \Big\} \end{aligned} \tag{3.10}$$

$$= \mathcal{L}. \tag{3.11}$$

The gradient with respect to $\beta_k^{(m)}$ is

$$\frac{\partial \mathcal{L}}{\partial \beta_k^{(m)}} = (\alpha_k - \beta_k^{(m)} + \sum_{n=1}^{N^{(m)}} r_{n,k}^{(m)}) \digamma'(\beta_k^{(m)}) - \sum_{l=1}^{K} (\alpha_l - \beta_l^{(m)} + \sum_{n=1}^{N^{(m)}} r_{n,l}^{(m)}) \digamma'(\sum_{l=1}^{K} \beta_k^{(m)}). \tag{3.12}$$

Setting the gradient to zero for $k = 1, \ldots, K$, we have the optimum

$$\beta_k^{(m)} = \alpha_k + \sum_{n=1}^{N} r_{n,k}^{(m)}. \tag{3.13}$$

The gradient with respect to $r_{n,k}^{(m)}$ is

$$\frac{\partial \mathcal{L}}{\partial r_{n,k}^{(m)}} = \digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K} \beta_l^{(m)}) + \log p(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)} | Z_{n,k}^{(m)} = 1, \Theta) - \log r_{n,k}^{(m)} - 1 \tag{3.14}$$

Setting the gradient to zero, we have the optimum

$$r_{n,k}^{(m)} \propto \rho_{n,k}^{(m)} = p(R_{n,k}^{(m)}, \lambda_{n,k}^{(m)} | Z_{n,k}^{(m)} = 1, \Theta) \exp \left[\digamma(\beta_k^{(m)}) - \digamma(\sum_{l=1}^{K} \beta_l^{(m)}) \right] \tag{3.15}$$

Since $\sum_{k=1}^{K} r_{n,k}^{(m)} = 1$, we have

$$r_{n,k}^{(m)} = \frac{\rho_{n,k}^{(m)}}{\sum_{l=1}^{K} \rho_{n,l}^{(m)}}. \tag{3.16}$$

The gradient with respect to α_k is

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = M \left(\digamma(\sum_{l=1}^{K} \alpha_l) - \digamma(\alpha_k) \right) + \sum_{m=1}^{M} \left(\digamma(\boldsymbol{\beta}_k^{(m)}) - \digamma(\sum_{l=1}^{K} \boldsymbol{\beta}_l^{(m)}) \right) \tag{3.17}$$

The $(i,j)^{\text{th}}$ element of the Hessian matrix is

$$\frac{\partial \mathcal{L}}{\partial \alpha_i \partial \alpha_j} = M \quad \digamma'(\sum_{l=1}^{K} \alpha_l) - \sigma(i,j)\digamma'(\alpha_j) \Bigg), \tag{3.18}$$

while $\sigma(i,j) = 1$ if and only if $i = j$, otherwise $\sigma(i,j) = 0$. To ensure $\boldsymbol{\alpha}$ is always positive during optimization, we let $\boldsymbol{\gamma} = \log(\boldsymbol{\alpha})$ and optimize $\boldsymbol{\gamma}$ instead. Taking the gradient with respect to γ_k, we have

$$\frac{\partial \mathcal{L}}{\partial \gamma_k} = M \quad \digamma(\sum_{l=1}^{K} \alpha_l) - \digamma(\alpha_k) \Bigg) \alpha_k + \sum_{m=1}^{M} \alpha_k \quad \digamma(\boldsymbol{\beta}_k^{(m)}) - \digamma(\sum_{l=1}^{K} \boldsymbol{\beta}_l^{(m)}) \Bigg) \Bigg(\tag{3.19}$$

The $(i,j)^{\text{th}}$ element of the Hessian matrix is

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \gamma_i \partial \gamma_j} = & M \quad \digamma'(\sum_{l=1}^{K} \alpha_l)\alpha_i\alpha_j \Bigg) \\ & + \sigma(i,j)\alpha_i \Bigg[\Bigg(M \quad \digamma(\sum_{l=1}^{K} \alpha_l) - \digamma'(\alpha_i)\alpha_i - \digamma(\alpha_i) \Bigg) + \sum_{m=1}^{M} \quad \digamma(\beta_i^{(m)}) - \digamma(\sum_{l=1}^{K} \boldsymbol{\beta}_l^{(m)}) \Bigg) \Bigg] \Bigg(\end{aligned} \tag{3.20}$$

Model inference with MCMC sampling

We use the Markov chain Monte Carlo (MCMC) inference method to compare with our proposed Variational Bayesian (VB) inference method. Similarly to MISO, we use a combination of Metropolis-Hastings (MH) algorithm and a Gibbs sampling algorithm. Within a replicate, the sampling steps are exactly the same as MISO. To sample the target posterior distribution, $p(\boldsymbol{\psi}^{(m)}|\boldsymbol{R}, \boldsymbol{\lambda})$, we use a softmax-normal distribution as the purposed the distribution for the MH algorithm. To sample the target posterior distribution, $p(\boldsymbol{Z}|\boldsymbol{R}, \boldsymbol{\lambda})$, we use the usual Gibbs steps for Dirichlet-Multinomial models. Different from MISO, we assume a non-informative prior for $\boldsymbol{\alpha}$, such that $p(\boldsymbol{\alpha})$ is a constant. We use an additional MH sampling step to sample $\boldsymbol{\alpha}$ from its posterior distributions given the samples of $\{\boldsymbol{\psi}^{(m)}\}_M$, where we use a log-normal distribution as the proposal distribution. The sampling scheme follows:

1. Initialize $\boldsymbol{\alpha}_0$, and for $m = 1..M$ initialize $\boldsymbol{\mu}_0^{(m)}$, $\boldsymbol{\psi}_0^{(m)}$ and $\boldsymbol{Z}_0^{(m)}$

2. For $t = 1..$[max number of iterations]:

   (a) For $m = 1..M$:

      i. Propose $\boldsymbol{\mu}_{t+1}^{(m)}$ and $\boldsymbol{\psi}_{t+1}^{(m)}$ as:

$$\begin{aligned}\boldsymbol{\mu}_{t+1}^{(m)} &\sim \mathcal{N}(\boldsymbol{\mu}_t^{(m)}, \Sigma) \\ \boldsymbol{\psi}_{t+1}^{(m)} &\sim \sigma(\boldsymbol{\mu}_{t+1}^{(m)})\end{aligned}$$

      where $\sigma(\cdot)$ is the softmax function $\sigma(\boldsymbol{v}) = \frac{e^{\boldsymbol{v}}}{\sum_{k=1}^{K} e^{v_k}}$.

      ii. Accept $\boldsymbol{\mu}_{t+1}^{(m)}$ and $\boldsymbol{\psi}_{t+1}^{(m)}$ with probability:

$$P_{\text{accept}} = \min \quad \frac{p(\boldsymbol{\psi}_{t+1}^{(m)}, \boldsymbol{Z}_t^{(m)}, \boldsymbol{\alpha}_t) q(\boldsymbol{\psi}_t^{(m)} | \boldsymbol{\psi}_{t+1}^{(m)})}{p(\boldsymbol{\psi}_t^{(m)}, \boldsymbol{Z}_t^{(m)}, \boldsymbol{\alpha}_t) q(\boldsymbol{\psi}_{t+1}^{(m)} | \boldsymbol{\psi}_t^{(m)})}, 1 \Bigg) \Bigg($$

      where $q(\cdot)$ is the proposed softmax-normal distribution.

      iii. For $n = 1..N^{(m)}$:

         A. Compute the conditional posterior of assigning a read for every isoform $1 \leq k \leq K$:

$$\theta_i = p(Z_{n,k}^{(m)} = 1 | \boldsymbol{\psi}_{t+1}^{(m)}, R_n^{(m)}, \lambda_n^{(m)})$$

         B. Sample an assignment for this read:

$$Z_{n,t+1}^{(m)} \sim \text{Multinomial}(1, [\theta_1, ...\theta_K])$$

   (b) Propose $\boldsymbol{\alpha}_{t+1}$ as:

$$\boldsymbol{\alpha}_{t+1} \sim \ln \mathcal{N}(\ln(\boldsymbol{\alpha}_t), \tilde{\Sigma})$$

   where $\ln \mathcal{N}(\cdot)$ denotes the log-normal distribution.

(c) Accept $\boldsymbol{\alpha}_{t+1}$ with probability:

$$\tilde{P}_{\text{accept}} = \min\left(\frac{p(\boldsymbol{\psi}_{t+1}, \boldsymbol{\alpha}_{t+1}) q(\boldsymbol{\alpha}_t | \boldsymbol{\alpha}_{t+1})}{p(\boldsymbol{\psi}_{t+1}, \boldsymbol{\alpha}_t) q(\boldsymbol{\alpha}_{t+1} | \boldsymbol{\alpha}_t)}, 1\right)\Bigg($$

where $\boldsymbol{\psi}_{t+1}$ includes $\boldsymbol{\psi}_{t+1}^{(1)}..\boldsymbol{\psi}_{t+1}^{(M)}$ and $q(\cdot)$ is the proposed log-normal distribution.

3.8.2 Simulated data

Evaluation measures

The following evaluation measures with root mean square errors are used:

Theta: $\sqrt{\frac{1}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\left(\psi_k^{(m)} - \hat{\psi}_k^{(m)}\right)^2}$, where $\psi_k^{(m)}$ and $\hat{\psi}_k^{(m)}$ denote the true and estimated fraction for transcript k of replicate m, $k = 1, \ldots, K$ and $m = 1, \ldots, M$.

Theta-Group: $\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\bar{\psi}_k - \frac{1}{M}\sum_{m=1}^{M}\hat{\psi}_k^{(m)}\right)^2}$, where $\hat{\psi}_k^{(m)}$ denotes the estimated fraction for transcript k of replicate m, $k = 1, \ldots, K$ and $m = 1, \ldots, M$, and $\bar{\psi}_k$ denotes the true group mean fraction for transcript k before generating the fraction for each replicate using negative binomial process.

WGE-True: $\sqrt{\frac{1}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\left(q_k^{(m)} - \hat{q}_k^{(m)}\right)^2}$, where $q_k^{(m)} = \frac{\psi_k^{(m)}}{\sum_{j\in T_k}\psi_j^{(m)}}$ and $\hat{q}_k^{(m)} = \frac{\hat{\psi}_k^{(m)}}{\sum_{j\in T_k}\hat{\psi}_j^{(m)}}$ denote the within gene true and estimated fraction for transcript k of replicate m, $k = 1, \ldots, K$ and $m = 1, \ldots, M$. Also T_k denotes the set of transcripts in the same parent gene as transcript k: $T_k = \{j : \text{transcript } j \text{ is in the same gene as transcript } k\}$.

WGE-Inter: $\sqrt{\frac{2}{M(M-1)}\sum_{1\le i<j\le M}\frac{1}{K}\sum_{k=1}^{K}\left(q_k^{(i)} - \hat{q}_k^{(j)}\right)^2}$.

Supplementary figures and tables

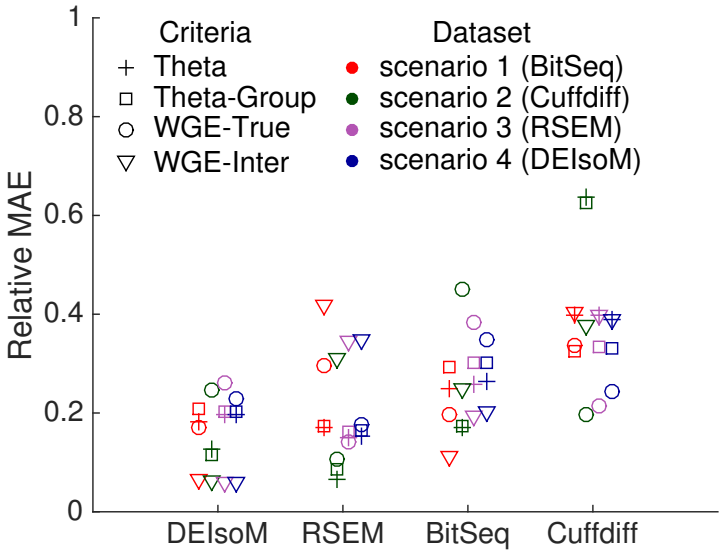


Fig. 3.9.: Relative mean absolute errors (MAEs) of DEIsoM, RSEM, BitSeq and Cuffdiff on four simulated datasets. Theta: estimated relative transcript expression compared with the ground truth for replicates. Theta-Group: mean estimated relative transcript expression of replicates compared with the true group means. WGE-True: within gene estimates compared with the ground truth. WGE-Inter: inter-replicate consistency of within gene estimates.

Table 3.2.: A comparison between the total CPU times for methods evaluated on synthetic data and real data. We include the user time and system time in computing the total CPU time.

	DEIsoM	Cuffdiff	RSEM-EBSeq	BitSeq
Simulated data	1.4h	1.1h	1.5h	3.2h
Real HCC data	137h	56.9h	353h	95.8h

3.8.3 Real data

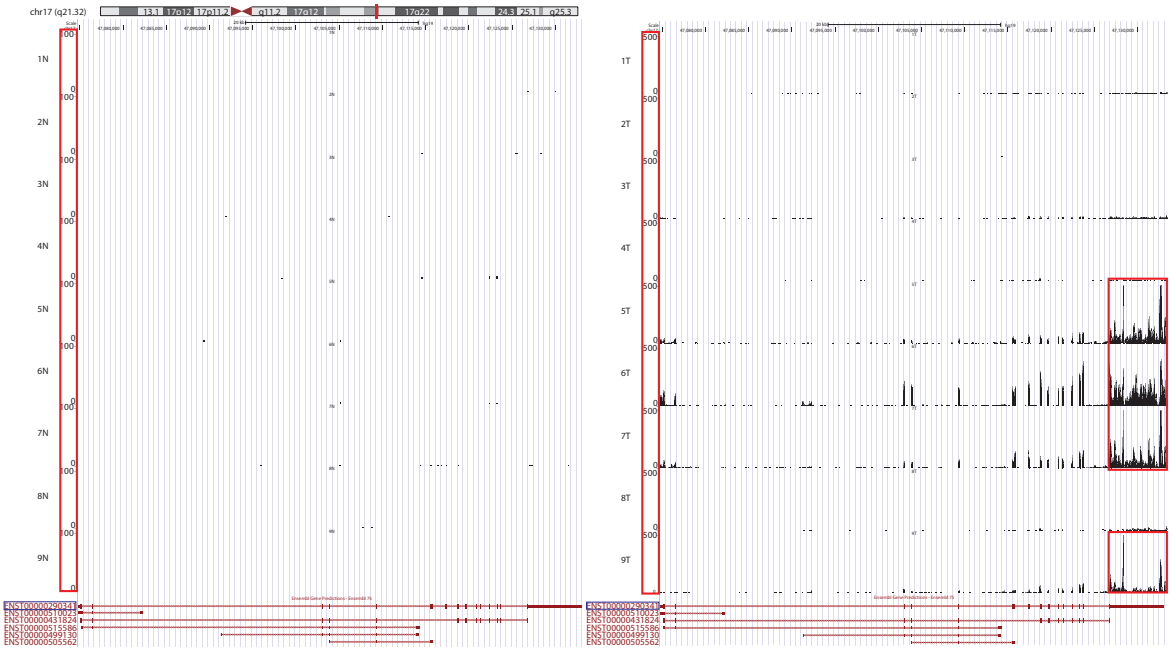


Fig. 3.10.: The read coverage visualization of the top DE isoform selected by Cuffdiff. The isoform 1 (blue box) of IGF2BP1 has been identified as the 6^{th} most DE isoform by Cuffdiff. The left panels show the read coverage patterns of nine normal samples, whereas the right panels show the read coverage of nine matched tumor samples.

Table 3.3.: The number of overlapped genes in the top selected DE genes between methods. All methods use the same evaluation method to rank the gene. The FPKM have been computed by DEIsoM, Cuffdiff, RSEM and BitSeqVB first. Then EBSeq is used to rank the DE genes.

# of selected genes	10	20	50	100	200
BitSeq and RSEM-EBSeq	1	1	14	39	96
RSEM-EBSeq and DEIsoM	0	2	12	33	74
RSEM-EBSeq and Cuffdiff	1	4	11	34	73
BitSeq and DEIsoM	1	2	10	34	71
BitSeq and Cuffdiff	1	3	9	24	62
Cuffdiff and DEIsoM	2	3	6	23	51

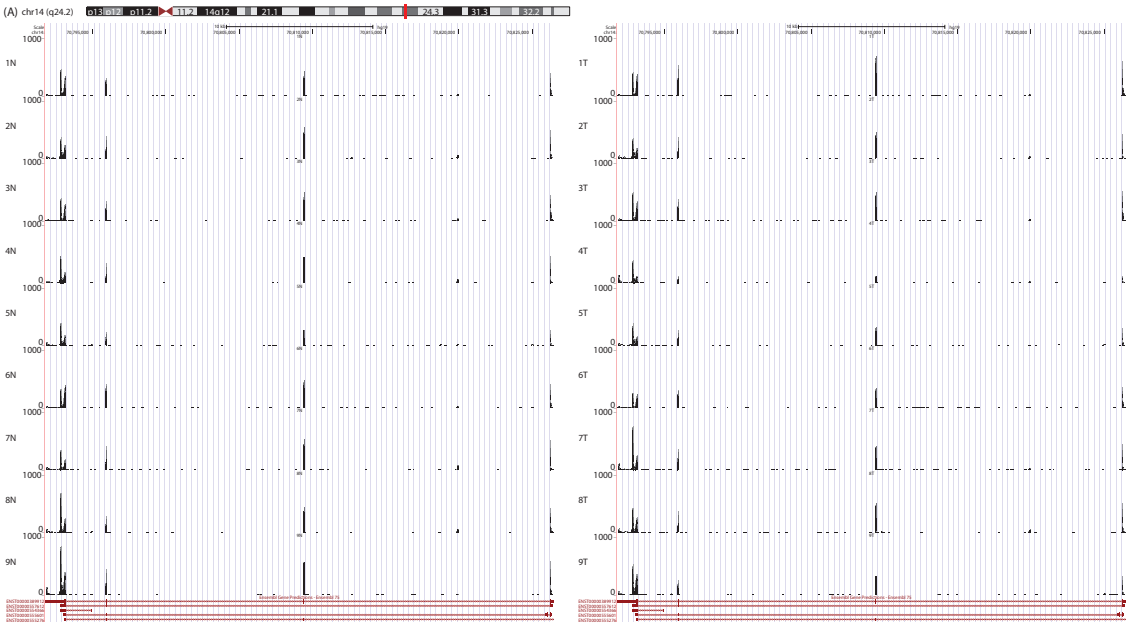


Fig. 3.11.: The read coverage visualization of the bottom selection by DEIsoM. The isoforms of COX16 are selected as non-DE. The left panels show the read coverage patterns of nine normal samples, whereas the right panels show nine matched tumor samples. It illustrates that the read coverage patterns are very similar between two conditions.

Table 3.4.: The number of overlapped genes in the top selected DE genes between methods. Different methods use different DE evaluations to rank the gene. Cuffdiff uses log-fold-change based p-values; RSEM uses PPDE and further real fold change estimated by EBSeq; BitSeqVB uses PPLR; DEIsoM uses KL divergence.

# of selected genes	10	20	50	100	200
BitSeq and RSEM-EBSeq	2	5	11	26	62
RSEM-EBSeq and Cuffdiff	0	0	1	3	23
RSEM-EBSeq and DEIsoM	0	0	1	4	14
BitSeq and Cuffdiff	0	0	0	2	13
BitSeq and DEIsoM	1	2	2	6	11
Cuffdiff and DEIsoM	0	0	0	1	3

Table 3.5.: Biological relevance of the top 50 DE genes selected by DEIsoM on HCC data and the corresponding references (Date: by 2016-11-12).

Rank	Gene ID	Symbol	Level	Reference
1	ENSG00000130707	ASS1	3	https://www.ncbi.nlm.nih.gov/pubmed/24946162
2	ENSG00000118271	TTR	3	https://www.ncbi.nlm.nih.gov/pubmed/17828420
3	ENSG00000167244	IGF2	3	https://www.ncbi.nlm.nih.gov/pubmed/21769080
4	ENSG00000145192	AHSG	3	https://www.ncbi.nlm.nih.gov/pubmed/19304375
5	ENSG00000011465	DCN	2	https://www.ncbi.nlm.nih.gov/pubmed/12521301
6	ENSG00000243649	CFB	2	https://www.ncbi.nlm.nih.gov/pubmed/24195504
7	ENSG00000188257	PLA2G2A	0	-
8	ENSG00000250722	SEPP1	2	https://www.ncbi.nlm.nih.gov/pubmed/19304375
9	ENSG00000147257	GPC3	3	https://www.ncbi.nlm.nih.gov/pubmed/22706665
10	ENSG00000132693	CRP	3	https://www.ncbi.nlm.nih.gov/pubmed/1337988
11	ENSG00000019582	CD74	0	-
12	ENSG00000138115	CYP2C8	0	-
13	ENSG00000166710	B2M	2	https://www.ncbi.nlm.nih.gov/pubmed/10879242
14	ENSG00000055957	ITIH1	0	-
15	ENSG00000081051	AFP	3	https://www.ncbi.nlm.nih.gov/pubmed/22620007
16	ENSG00000151655	ITIH2	0	-
17	ENSG00000159403	C1R	0	-
18	ENSG00000100197	CYP2D6	2	https://www.ncbi.nlm.nih.gov/pubmed/16048566
19	ENSG00000244255		0	-
20	ENSG00000169439	SDC2	0	-
21	ENSG00000167711	SERPINF2	2	https://www.ncbi.nlm.nih.gov/pubmed/16980951
22	ENSG00000185813	PCYT2	0	-
23	ENSG00000166741	NNMT	3	https://www.ncbi.nlm.nih.gov/pubmed/19216803
24	ENSG00000160862	AZGP1	3	https://www.ncbi.nlm.nih.gov/pubmed/22625427
25	ENSG00000167996	FTH1	0	https://www.ncbi.nlm.nih.gov/pubmed/12029631
26	ENSG00000122786	CALD1	0	-
27	ENSG00000142541	RPL13A	2	https://www.ncbi.nlm.nih.gov/pubmed/16820872
28	ENSG00000117601	SERPINC1	3	https://www.ncbi.nlm.nih.gov/pubmed/16820872
29	ENSG00000109971	HSP70	3	https://www.ncbi.nlm.nih.gov/pubmed/14673798
30	ENSG00000185624	P4HB	0	-
31	ENSG00000142192	APP	1	https://www.ncbi.nlm.nih.gov/pubmed/9243801
32	ENSG00000160868	CYP3A4	3	https://www.ncbi.nlm.nih.gov/pubmed/23891548
33	ENSG00000204628	GNB2L1	2	https://www.ncbi.nlm.nih.gov/pubmed/16820872
34	ENSG00000008394	MGST1	0	-
35	ENSG00000197111	PCBP2	3	https://www.ncbi.nlm.nih.gov/pubmed/27748915
36	ENSG00000148672	GLUD1	0	-
37	ENSG00000136011	STAB2	2	https://www.ncbi.nlm.nih.gov/pubmed/23870052
38	ENSG00000116171	SCP2	0	-
39	ENSG00000110492	MDK	3	https://www.ncbi.nlm.nih.gov/pubmed/17317821
40	ENSG00000213494	CCL14	0	-
41	ENSG00000166278	C2	0	-
42	ENSG00000114867	EIF4G1	0	-
43	ENSG00000142748	FCN3	2	https://www.ncbi.nlm.nih.gov/pubmed/17006932
44	ENSG00000003436	TFPI	0	-
45	ENSG00000198363	ASPH	3	https://www.ncbi.nlm.nih.gov/pubmed/22245894
46	ENSG00000116882	HAO2	3	https://www.ncbi.nlm.nih.gov/pubmed/26658681
47	ENSG00000197746	PSAP	0	-
48	ENSG00000198848	CES1	3	https://www.ncbi.nlm.nih.gov/pubmed/19658107
49	ENSG00000138674	SEC31A	0	-
50	ENSG00000127831	VIL1	3	https://www.ncbi.nlm.nih.gov/pubmed/22530999

Table 3.6.: Biological relevance of the genes of the top 50 DE isoforms selected by RSEM-EBSeq on HCC data and the corresponding references (Date: by 2016-11-12).(*: Clone-based (Vega))

Rank	Gene ID	Symbol	Level	Reference
1	ENSG00000242265	PEG10	3	https://www.ncbi.nlm.nih.gov/pubmed/24369324
2	ENSG00000187243	MAGED4B	0	-
3	ENSG00000130829	DUSP9	0	-
4	ENSG00000164362	TERT	3	https://www.ncbi.nlm.nih.gov/pubmed/26099527
5	ENSG00000206557	TRIM71	0	-
6	ENSG00000225546	LVCAT5	0	-
7	ENSG00000157766	ACAN	3	https://www.ncbi.nlm.nih.gov/pubmed/22912547
8	ENSG00000225210	AL589743.1*	0	-
9	ENSG00000159217	IGF2BP1	2	https://www.ncbi.nlm.nih.gov/pubmed/24395596
10	ENSG00000238107	RP11-495P10.5*	0	-
11	ENSG00000139219	COL2A1	1	https://www.ncbi.nlm.nih.gov/pubmed/21731504
12	ENSG00000185686	PRAME	0	-
13	ENSG00000231196	RP11-495P10.8*	0	-
14	ENSG00000223572	CKMT1A	0	-
15	ENSG00000096088	PGC	0	-
16	ENSG00000126752	SSX1	1	https://www.ncbi.nlm.nih.gov/pubmed/24798046
17	ENSG00000254233	LVCAT8	0	-
18	ENSG00000253293	HOXA10	1	https://www.ncbi.nlm.nih.gov/pubmed/25120782
19	ENSG00000214814	FER1L6	0	-
20	ENSG00000136231	IGF2BP3	3	https://www.ncbi.nlm.nih.gov/pubmed/18802962
21	ENSG00000233539	LOC730338	0	-
22	ENSG00000110347	MMP12	3	https://www.ncbi.nlm.nih.gov/pubmed/21683576
23	ENSG00000228651	RP11-556E13.1*	0	-
24	ENSG00000181617	FDCSP	0	-
25	ENSG00000081051	AFP	3	https://www.ncbi.nlm.nih.gov/pubmed/22620007
26	ENSG00000043355	ZIC2	1	https://www.ncbi.nlm.nih.gov/pubmed/26426078
27	ENSG00000106031	HOXA13	3	https://www.ncbi.nlm.nih.gov/pubmed/25341685
28	ENSG00000107984	DKK1	3	https://www.ncbi.nlm.nih.gov/pubmed/27458854
29	ENSG00000133063	CHIT1	0	-
30	ENSG00000147485	PXDNL	0	-
31	ENSG00000179083	FAM133A	0	-
32	ENSG00000264424	MYH4	0	-
33	ENSG00000172016	REG3A	2	https://www.ncbi.nlm.nih.gov/pubmed/16314847
34	ENSG00000226674	TEX41	0	-
35	ENSG00000074211	PPP2R2C	0	-
36	ENSG00000086548	CEACAM6	0	-
37	ENSG00000154277	UCHL1	2	https://www.ncbi.nlm.nih.gov/pubmed/18666234
38	ENSG00000178999	AURKB	3	https://www.ncbi.nlm.nih.gov/pubmed/20799978
39	ENSG00000168955	TM4SF20	0	-
40	ENSG00000183837	PNMA3	0	-
41	ENSG00000163993	S100P	3	https://www.ncbi.nlm.nih.gov/pubmed/23785431
42	ENSG00000168243	GNG4	0	-
43	ENSG00000112818	MEP1A	3	https://www.ncbi.nlm.nih.gov/pubmed/26660154
44	ENSG00000236849	LINC01474	0	-
45	ENSG00000171243	SOSTDC1	0	-
46	ENSG00000198074	AKR1B10	3	https://www.ncbi.nlm.nih.gov/pubmed/27672277
47	ENSG00000204832	ST8SIA6-AS1	0	-
48	ENSG00000251049	RP11-685F15.1*	0	-
49	ENSG00000123496	IL13RA2	0	-
50	ENSG00000229183	PGA4	0	-

Table 3.7.: Biological relevance of the gene of the top 50 DE isoforms selected by Cuffdiff on HCC data and the corresponding references (Date: by 2016-11-12).(*: Clone-based (Vega))

Rank	Gene ID	Symbol	Level	Reference
1	ENSG00000145604	SKP2	3	https://www.ncbi.nlm.nih.gov/pubmed/27779207
2	ENSG00000170345	C-FOS	3	https://www.ncbi.nlm.nih.gov/pubmed/22582734
3	ENSG00000171848	R2	0	-
4	ENSG00000232001	AC108868.6*	0	-
5	ENSG00000116761	CTH	0	-
6	ENSG00000197408	CYP2B6	2	https://www.ncbi.nlm.nih.gov/pubmed/25024626
7	ENSG00000130635	COL5A1	0	-
8	ENSG00000120833	SOCS2	3	https://www.ncbi.nlm.nih.gov/pubmed/23475171
9	ENSG00000238106		0	-
10	ENSG00000236786	TSPY15P	0	-
11	ENSG00000171408	PDE7B	0	-
12	ENSG00000121691	CAT	3	https://www.ncbi.nlm.nih.gov/pubmed/21985599
13	ENSG00000174992	ZG16	2	https://www.ncbi.nlm.nih.gov/pubmed/17307141
14	ENSG00000183748	LOC101928757	0	-
15	ENSG00000216649	GAGE12E	0	-
16	ENSG00000167780	ACAT2	2	https://www.ncbi.nlm.nih.gov/pubmed/24163426
17	ENSG00000090889	KIF4A	2	https://www.ncbi.nlm.nih.gov/pubmed/25998931
18	ENSG00000263585	RP11-498C9.13*	0	-
19	ENSG00000205362	MT1A	3	https://www.ncbi.nlm.nih.gov/pubmed/16703398
20	ENSG00000219607	PPP1R3G	0	-
21	ENSG00000249842	CTD-2331D11.4*	0	-
22	ENSG00000226164	FGFR3P6	0	-
23	ENSG00000260886	TAT-AS1	0	-
24	ENSG00000128266	GNAZ	2	https://www.ncbi.nlm.nih.gov/pubmed/16264227
25	ENSG00000162769	FLVCR1	2	https://www.ncbi.nlm.nih.gov/pubmed/27387388
26	ENSG00000184374	COLEC10	0	-
27	ENSG00000232164	LOC729348	0	-
28	ENSG00000072080	SPP2	0	-
29	ENSG00000092621	PHGDH	1	https://www.ncbi.nlm.nih.gov/pubmed/25872475
30	ENSG00000116017	ARID3A	2	https://www.ncbi.nlm.nih.gov/pubmed/27458175
31	ENSG00000156510	HKDC1	3	https://www.ncbi.nlm.nih.gov/pubmed/27155152
32	ENSG00000162409	PRKAA2	2	https://www.ncbi.nlm.nih.gov/pubmed/27216817
33	ENSG00000136040	PLXNC1	0	-
34	ENSG00000157131	C8A	3	https://www.ncbi.nlm.nih.gov/pubmed/26414287
35	ENSG00000143842	SOX13	2	https://www.ncbi.nlm.nih.gov/pubmed/24160375
36	ENSG00000230328	RP11-35N6.6*	0	-
37	ENSG00000224902	GAGE12H	0	-
38	ENSG00000099860	GADD45B	3	https://www.ncbi.nlm.nih.gov/pubmed/12759252
39	ENSG00000172073	TEX37	0	-
40	ENSG00000104549	SQLE	2	https://www.ncbi.nlm.nih.gov/pubmed/25787749
41	ENSG00000169174	PCSK9	3	https://www.ncbi.nlm.nih.gov/pubmed/26674961
42	ENSG00000130222	GADD45G	2	https://www.ncbi.nlm.nih.gov/pubmed/23897841
43	ENSG00000108448	TRIM16L	0	-
44	ENSG00000126231	PROZ	2	https://www.ncbi.nlm.nih.gov/pubmed/22689435
45	ENSG00000006074	CCL18	1	https://www.ncbi.nlm.nih.gov/pubmed/26449829
46	ENSG00000143369	ECM1	3	https://www.ncbi.nlm.nih.gov/pubmed/27460906
47	ENSG00000236362	GAGE12F	0	-
48	ENSG00000126752	SSX1	1	https://www.ncbi.nlm.nih.gov/pubmed/24798046
49	ENSG00000235494	RP11-498P14.4*	0	-
50	ENSG00000130427	EPO	3	https://www.ncbi.nlm.nih.gov/pubmed/26097591

Table 3.8.: Biological relevance of the gene of the top 50 DE isoforms selected by BitSeqVB-PPLR on HCC data and the corresponding references (Date: by 2016-11-12).(*: Clone-based (Vega))

Rank	Gene ID	Symbol	Level	Reference
1	ENSG00000159217	IGF2BP1	2	https://www.ncbi.nlm.nih.gov/pubmed/24395596
2	ENSG00000260518	BMS1P8	0	-
3	ENSG00000147257	GPC3	3	http://www.ncbi.nlm.nih.gov/pubmed/22706665
4	ENSG00000158402	CDC25C	0	-
5	ENSG00000043355	ZIC2	1	https://www.ncbi.nlm.nih.gov/pubmed/26426078
6	ENSG00000081051	AFP	3	http://www.ncbi.nlm.nih.gov/pubmed/22620007
7	ENSG00000136231	IGF2BP3	3	https://www.ncbi.nlm.nih.gov/pubmed/18802962
8	ENSG00000206557	TRIM71	0	-
9	ENSG00000099953	MMP11	0	-
10	ENSG00000175063	UBE2C	3	https://www.ncbi.nlm.nih.gov/pubmed/17354233
11	ENSG00000198074	AKR1B10	3	https://www.ncbi.nlm.nih.gov/pubmed/27672277
12	ENSG00000143228	NUF2	2	https://www.ncbi.nlm.nih.gov/pubmed/25374179
13	ENSG00000024526	DEPDC1	3	https://www.ncbi.nlm.nih.gov/pubmed/25605201
14	ENSG00000164362	TERT	3	https://www.ncbi.nlm.nih.gov/pubmed/26099527
15	ENSG00000175329	ISX	2	https://www.ncbi.nlm.nih.gov/pubmed/23221382
16	ENSG00000112742	TTK	3	https://www.ncbi.nlm.nih.gov/pubmed/24859455
17	ENSG00000034063	UHRF1	3	https://www.ncbi.nlm.nih.gov/pubmed/28060737
18	ENSG00000074410	CA12	0	-
19	ENSG00000101057	MYBL2	3	https://www.ncbi.nlm.nih.gov/pubmed/18624722
20	ENSG00000130829	DUSP9	0	-
21	ENSG00000109805	NCAPG	2	https://www.ncbi.nlm.nih.gov/pubmed/28238542
22	ENSG00000085831	TTC39A	0	-
23	ENSG00000089685	BIRC5	2	https://www.ncbi.nlm.nih.gov/pubmed/28238542
24	ENSG00000125780	TGM3	0	-
25	ENSG00000175793	SFN	2	https://www.ncbi.nlm.nih.gov/pubmed/24859455
26	ENSG00000123485	HJURP	0	-
27	ENSG00000117650	NEK2	3	https://www.ncbi.nlm.nih.gov/pubmed/28101574
28	ENSG00000113296	THBS4	3	https://www.ncbi.nlm.nih.gov/pubmed/28177895
29	ENSG00000154545	MAGED	0	-
30	ENSG00000011426	ANLN	3	https://www.ncbi.nlm.nih.gov/pubmed/23717429
31	ENSG00000163808	KIF15	2	https://www.ncbi.nlm.nih.gov/pubmed/24859455
32	ENSG00000111206	FOXM1	2	https://www.ncbi.nlm.nih.gov/pubmed/26289845
33	ENSG00000198758	EPS8L3	0	-
34	ENSG00000142945	KIF2C	0	-
35	ENSG00000187243	MAGED4B	0	-
36	ENSG00000165480	SKA3	0	-
37	ENSG00000174371	EXO1	0	-
38	ENSG00000185686	PRAME	0	-
39	ENSG00000156970	BUB1B	2	https://www.ncbi.nlm.nih.gov/pubmed/25753876
40	ENSG00000228651	RP11-556E13.1*	0	-
41	ENSG00000111665	CDCA3	2	https://www.ncbi.nlm.nih.gov/pubmed/25236463
42	ENSG00000169213	RAB3B	0	-
43	ENSG00000135451	TROAP	0	-
44	ENSG00000066279	ASPM	3	https://www.ncbi.nlm.nih.gov/pubmed/18676753
45	ENSG00000198203	SULT1C2	0	-
46	ENSG00000176092	CRYBG2	0	-
47	ENSG00000166851	PLK1	3	https://www.ncbi.nlm.nih.gov/pubmed/19725153
48	ENSG00000169679	BUB1	2	https://www.ncbi.nlm.nih.gov/pubmed/28238542
49	ENSG00000129173	E2F8	2	https://www.ncbi.nlm.nih.gov/pubmed/20068156
50	ENSG00000072571	HMMR	3	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4906898/

Acknowledgments

We would like to thank Eli Lilly and company for sharing the HCC data and providing the very helpful discussions.

Funding

This work was supported by NSF CAREER Award IIS-1054903, and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

3.9 References

Amann, T., Bataille, F., Spruss, T., Dettmer, K., Wild, P., Liedtke, C., Mühlbauer, M., Kiefer, P., Oefner, P. J., Trautwein, C., Bosserhoff, A. K., and Hellerbrand, C. (2010). Reduced expression of fibroblast growth factor receptor 2IIIb in hepatocellular carcinoma induces a more aggressive growth. *American Journal of Pathology*, 176(3):1433 – 1442.

Berasain, C., Goñi, S., Castillo, J., Latasa, M. U., Prieto, J., and Avila, M. A. (2010). Impairment of pre-mRNA splicing in liver disease: mechanisms and consequences. *World Journal of Gastroenterology*, 16(25):3091–3102.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Consortium, G. (2015). The genotype-tissue expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235):648–660.

Dong, H., Ge, X., Shen, Y., Chen, L., Kong, Y., Zhang, H., Man, X., Tang, L., Yuan, H., Wang, H., Zhao, G., and Jin, W. (2009). Gene expression profile analysis of human hepatocellular carcinoma using sage and longsage. *BMC Medical Genomics*, 2(1):5.

Gierliński, M., Cole, C., Schofield, P., Schurch, N. J., Sherstnev, A., Singh, V., Wrobel, N., Gharbi, K., Simpson, G., Owen-Hughes, T., Blaxter, M., and Barton, G. J. (2015). Statistical models for RNA-seq data derived from a two-condition 48-replicate experiment. *Bioinformatics*.

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

Grant, G. R., Farkas, M. H., Pizarro, A. D., Lahens, N. F., Schug, J., Brunk, B. P., Stoeckert, C. J., Hogenesch, J. B., and Pierce, E. A. (2011). Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27(18):2518–2528.

Hensman, J., Papastamoulis, P., Glaus, P., Honkela, A., and Rattray, M. (2015). Fast and accurate approximate inference of transcript expression from RNA-seq data. *Bioinformatics*, 31(24):3881.

Huang, Y., Pan, J., Chen, D., Zheng, J., Qiu, F., Li, F., Wu, Y., Wu, W., Huang, X., and Qian, J. (2017). Identification and functional analysis of differentially expressed genes in poorly differentiated hepatocellular carcinoma using RNA-seq. *Oncotarget*.

Jang, J. W., Oh, B. S., Kwon, J. H., You, C. R., Chung, K. W., Kay, C. S., and Jung, H. S. (2012). Serum interleukin-6 and C-reactive protein as a prognostic indicator in hepatocellular carcinoma. *Cytokine*, 60(3):686–693.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kakaradov, B., Xiong, H. Y., Lee, L. J., Jojic, N., and Frey, B. J. (2012). Challenges in estimating percent inclusion of alternatively spliced junctions from RNA-seq data. *BMC Bioinformatics*, 13(Suppl 6):S11.

Kan, Z., Zheng, H., Liu, X., Li, S., Barber, T. D., Gong, Z., Gao, H., Hao, K., Willard, M. D., Xu, J., Hauptschein, R., Rejto, P. A., Fernandez, J., Wang, G., Zhang, Q., Wang, B., Chen, R., Wang, J., Lee, N. P., Zhou, W., Lin, Z., Peng, Z., Yi, K., Chen, S., Li, L., Fan, X., Yang, J., Ye, R., Ju, J., Wang, K., Estrella, H., Deng, S., Wei, P., Qiu, M., Wulur, I. H., Liu, J., Ehsani, M. E., Zhang, C., Loboda, A., Sung, W. K., Aggarwal, A., Poon, R. T., Fan, S. T., Wang, J., Hardwick, J., Reinhard, C., Dai, H., Li, Y., Luk, J. M., and Mao, M. (2013). Whole-genome sequencing identifies recurrent mutations in hepatocellular carcinoma. *Genome Research*, 23(9):1422–1433.

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.

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

Le, H. S., Schulz, M. H., McCauley, B. M., Hinman, V. F., and Bar-Joseph, Z. (2013). Probabilistic error correction for RNA sequencing. *Nucleic Acids Research*, 41:e109.

Leng, N., Dawson, J. A., Thomson, J. A., Ruotti, V., Rissman, A. I., Smits, B. M. G., Haag, J. D., Gould, M. N., Stewart, R. M., and Kendziorski, C. (2013). EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, 29(8):1035.

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

Martin, J. A. and Wang, Z. (2011). Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682.

Minka, T. P. (2000). Estimating a Dirichlet distribution. Technical report, M.I.T.

Nowicka, M. and Robinson, M. (2016). DRIMSeq: a Dirichlet-multinomial framework for multivariate count outcomes in genomics [version 2; referees: 2 approved]. *F1000Research*, 5(1356).

Ozsolak, F. and Milos, P. M. (2011). RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, 12:87–98.

Qiu, J. G., Fan, J., Liu, Y. K., Zhou, J., Dai, Z., Huang, C., and Tang, Z. Y. (2008). Screening and detection of portal vein tumor thrombi-associated serum low molecular weight protein biomarkers in human hepatocellular carcinoma. *Journal of Cancer Research and Clinical Oncology*, 134(3):299–305.

Ronning, G. (1989). Maximum likelihood estimation of Dirichlet distributions. *Journal of Statistical Computation and Simulation*, 34(4):215–221.

Shen, S., Park, J. W., Lu, Z.-x., Lin, L., Henry D., M., 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 of the United States of America*, 111(51):E5593.

Shomron, N., Hamasaki-Katagiri, N., Hunt, R., Hershko, K., E. Pommier, S. G., Blaisdell, A., Dobkin, A., Marple, A., Roma, I., Newell, J., Allen, C., Friedman, S., and Kimchi-Sarfaty, C. (2010). A splice variant of ADAMTS13 is expressed in human hepatic stellate cells and cancerous tissues. *Thrombosis and Haemostasis*, 104(3):531–535.

Stegle, O., Drewe, P., Bohnert, R., Borgwardt, K., and Rätsch, G. (2010). Statistical tests for detecting differential RNA-transcript expression from read counts. Nature Precedings.

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.

Sung, W. K., Zheng, H., Li, S., Chen, R., Liu, X., Li, Y., Lee, N. P., Lee, W. H., Ariyaratne, P. N., Tennakoon, C., Mulawadi, F. H., Wong, K. F., Liu, A. M., Poon, R. T., Fan, S. T., Chan, K. L., Gong, Z., Hu, Y., Lin, Z., Wang, G., Zhang, Q., Barber, T. D., Chou, W. C., Aggarwal, A., Hao, K., Zhou, W., Zhang, C., Hardwick, J., Buser, C., Xu, J., Kan, Z., Dai, H., Mao, M., Reinhard, C., Wang, J., and Luk, J. M. (2012). Genome-wide survey of recurrent HBV integration in hepatocellular carcinoma. *Nature Genetics*, 44:765–769.

Takashima, H., Nakajima, T., Moriguchi, M., Sekoguchi, S., Nishikawa, T., Watanabe, T., Katagishi, T., Kimura, H., Minami, M., Itoh, Y., Kagawa, K., and Okanoue, T. (2005). In vivo expression patterns of survivin and its splicing variants in chronic liver disease and hepatocellular carcinoma. *Liver International*, 25(1):77–84.

Tan, G. S., Lim, K. H., Tan, H. T., Khoo, M. L., Tan, S. H., Toh, H. C., and Ching Ming Chung, M. (2014). Novel proteomic biomarker panel for prediction of aggressive metastatic hepatocellular carcinoma relapse in surgically resectable patients. *Journal of Proteome Research*, 13(11):4833–4846. PMID: 24946162.

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

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., Pimentel, H., Salzberg, S. L., Rinn, J. L., and Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, 3(7):562–578.

Vaquero-Garcia, J., Barrera, A., Gazzara R, M., González-Vallinas, J., Lahens F, N., Hogenesch B, J., Lynch W, K., and Barash, Y. (2016). A new view of transcriptome complexity and regulation through the lens of local splicing variations. *eLife*, (5):e11752.

Wang, F., Wang, R., Li, Q., Qu, X., Hao, Y., Yang, J., Zhao, H., Wang, Q., Li, G., Zhang, F., Zhang, H., Zhou, X., Peng, X., Bian, Y., and Xiao, W. (2015). Meta-analysis of gene expression profiles indicates genes in spliceosome pathway are up-regulated in hepatocellular carcinoma (HCC). *Medical Oncology*, 32(96).

Wang, F., Wang, R., Li, Q., Qu, X., Hao, Y., Yang, J., Zhao, H., Wang, Q., Li, G., Zhang, F., Zhang, H., Zhou, X., Peng, X., Bian, Y., and Xiao, W. (2017). A transcriptome profile in hepatocellular carcinomas based on integrated analysis of microarray studies. *Diagnostic Pathology*, 12(1):4.

Wang, K., Xu, X., Nie, Y., Dai, L., Wang, P., and Zhang, J. (2009). Identification of tumor-associated antigens by using SEREX in hepatocellular carcinoma. *Cancer Letter*, 281(2):144–150.

Yim, S. H. and Chung, Y. J. (2010). An overview of biomarkers and molecular signatures in HCC. *Cancers*, 2(2):809–823.

Zhang, T., Huang, X. H., Dong, L., Hu, D., Ge, C., Zhan, Y. Q., Xu, W. X., Yu, M., Li, W., Wang, X., Tang, L., Li, C. Y., and Yang, X. M. (2010). PCBP-1 regulates alternative splicing of the CD44 gene and inhibits invasion in human hepatoma cell line HepG2 cells. *Molecular Cancer*, 9(72).

4. CHAPTER 4. JOINT NETWORK AND NODE SELECTION FOR PATHWAY-BASED GENOMIC DATA ANALYSIS

This is a side project of my dissertation. The main tasks I did in this project were searching and processing three cancer datasets, extracting and integrating KEGG pathway structures as matrices, analyzing the biological results, and writing the section of "application to expression data". Meanwhile, I learned the basic ideas of building a graphical Bayesian model and its inference method from two other members in this team.

Joint network and node selection for pathway-based genomic data analysis

Shandian Zhe [1], Syed A.Z. Naqvi [1], Yifan Yang [3] and Yuan Qi [1,2]

[1]Department of Computer Science,Purdue University, West Lafayette, IN,47907,USA.
[2]Department of Statistics, Purdue University, West Lafayette,IN,47907,USA.
[3]Department of Biology, Purdue University, West Lafayette,IN,47907,USA.

*To whom correspondence should be addressed.

Received on 9 Feb 2013;
Revised on 05 Jun 2013;
Accepted on 06 June 2013.

Associate Editor: Martin Bishop

4.1 Abstract

Motivation: By capturing various biochemical interactions, biological pathways provide insight into underlying biological processes. Given high-dimensional microarray or RNA-sequencing data, a critical challenge is how to integrate them with rich information from pathway databases to jointly select relevant pathways and genes for phenotype prediction or disease prognosis. Addressing this challenge can help us deepen biological understanding of phenotypes and diseases from a systems perspective.

Results: In this article, we propose a novel sparse Bayesian model for joint network and node selection. This model integrates information from networks (e.g. pathways) and nodes (e.g. genes) by a hybrid of conditional and generative components. For the conditional component, we propose a sparse prior based on graph Laplacian matrices, each of which encodes between network nodes. For the generative component, we use a spike and slab prior over network nodes. The integration of these two components, coupled with efficient variational inference, enables the selection of networks as well as correlated network nodes in the selected networks.

Simulation results demonstrate improved predictive performance and selection accuracy of our method over alternative methods. Based on three expression datasets for cancer study and the KEGG pathway database, we selected relevant genes and pathways, many of which are supported by biological literature. In addition to pathway analysis, our method is expected to have a wide range of applications in selecting relevant groups of correlated high-dimensional biomarkers.

Availability: The code can be downloaded at :

http://www.cs.purdue.edu/homes/szhe/software.html

Contact: alanqi@purdue.edu

4.2 Introduction

With the popularity of high-throughput biological data such as microarray and RNA-sequencing data, many variable selection methods – such as lasso (Tibshirani, 1996) and elastic net (Zou and Hastie, 2005) – have been proposed and applied to select relevant genes for disease diagnosis or prognosis. Nevertheless, these approaches ignore invaluable biological pathway information accumulated over decades of research; hence their selection results can be difficult to interpret biologically and their predictive performance can be limited by a small sample size of expression profiles. To overcome these limitations, a promising direction is to integrate expression profiles with rich biological knowledge in pathway databases. Because pathways organize genes into biologically functional groups and model their interactions that capture *correlation* between genes, this information integration can improve not only the predictive performance but also interpretability of the selection results. Thus, a critical need is to integrate pathway information with expression profiles for joint selection of pathways and genes associated with a phenotype or disease.

Despite their success in many applications, previous sparse learning methods are limited by several factors for the integration of pathway information with expression profiles. For example, group lasso (Yuan and Lin, 2007) can be used to utilize memberships of genes in pathways via a $l_{1/2}$ norm to select groups of genes, but they ignore pathway structural information. An excellent work by Li and Li (2008) overcomes this limitation by incorporating pathway structures in a Laplacian matrix of a global graph to guide the selection of relevant genes. In addition to graph Laplacians, binary Markov random field priors can be used to represent pathway information to influence gene selection (Wei and Li, 2007, 2008; Li and Zhang, 2010; Stingo and Vannucci, 2010). These network-regularized approaches do not explicitly select pathways. However, not all pathways are relevant and pathway selection can yield insight into underlying biological processes. A pioneering approach to joint pathway and gene selection by Stingo et al. (2011) uses binary Markov random field priors and couples

gene and pathway selection by hard constraints – for example, if a gene is selected, all the pathways it belongs to will be selected. However, this consistency constraint might be too rigid from a biological perspective: an active gene for cancer progression does not necessarily imply that *all* the pathways it belongs to are active. Given the Markov random field priors and the nonlinear constraints, posterior distributions are inferred by a Markov Chain Monte Carlo method (Stingo et al., 2011). But the convergence of MCMC for high dimensional problems is known to take a long time.

To overcome these limitations, we propose a new sparse Bayesian approach, called Network and NOde Selection (NaNOS), for joint pathway and gene selection. NaNOS is a sparse hybrid Bayesian model that integrates conditional and generative components in a principled Bayesian framework Lasserre et al. (2006). For the conditional component, we use a graph Laplacian matrix to encode information of each network (e.g. a pathway) and incorporate it into a sparse prior to select individual networks. For the generative component, we use a spike and slab prior to choose relevant nodes (e.g. genes) in selected networks. For this hybrid model, we do not impose the hard consistency constraints used by Stingo et al. (2011). Furthermore, the prior distribution of our model does not contain intractable partition functions. This enables us to give a full Bayesian treatment over model parameters and develop an efficient variational inference algorithm to obtain approximate posterior distributions for Bayesian estimation. As described in Section 4.4, our inference algorithm is designed to handle both continuous and discrete outcomes.

Simulation results in Section 4.5 demonstrate superior performance of our method over alternative methods for predicting continuous or binary responses, as well as comparable or improved performance for selecting relevant genes and pathways. Furthermore, on real expression data for large B cell lymphoma, pancreatic ductal adenocarcinoma, and colorectal cancer, our results yield meaningful biological interpretations supported by biological literature.

4.3 Model

In this section, we present the hybrid Bayesian model, NaNOS, for network and node selection. First, let us start from the classical variable selection problem. Suppose we have N independent and identically distributed samples $\mathcal{D} = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$, where $\mathbf{x}_i$ and t_i are the explanatory variables and the response of the i-th sample, respectively. The explanatory variables can be various biomarkers, such as gene expression levels or single-nucleotide polymorphisms. Following the tradition in variable selection, we normalize the values of each variable so that its mean and standard deviation are zero and one, respectively. The response can be certain phenotype or disease status. We aim to predict the response vector $\mathbf{t} = [t_1, \ldots, t_N]^\top$ based on the explanatory variables $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$ and to select a small number of variables relevant for the prediction. Because the number of variables (e.g. genes) is often much bigger than the number of samples, the prediction and selection tasks are statistically challenging.

To reduce the difficulty of variable selection, we can use valuable information from networks, each of which contains certain variables as nodes and represents their interactions. For example, biological pathways cluster genes into functional groups, revealing various gene interactions. Based M networks, we organize the explanatory variables $\mathbf{x}_i$ into M subvectors, each of which comprises the values of explanatory variables in its corresponding network. If a variable (i.e. a gene) appears in multiple networks (i.e. pathways), we duplicate its value in these networks. Note that networks here are exchangeable with graphs; we can use them to represent not only biological pathways but also linkage disequilibrium structures for genetic variation analysis.

Our model is a Bayesian hybrid of conditional and generative models based on a general framework proposed by (Lasserre et al., 2006). The conditional component selects individual networks via “discriminative" training; the generative component chooses relevant nodes in the selected networks; and the two models are glued to-

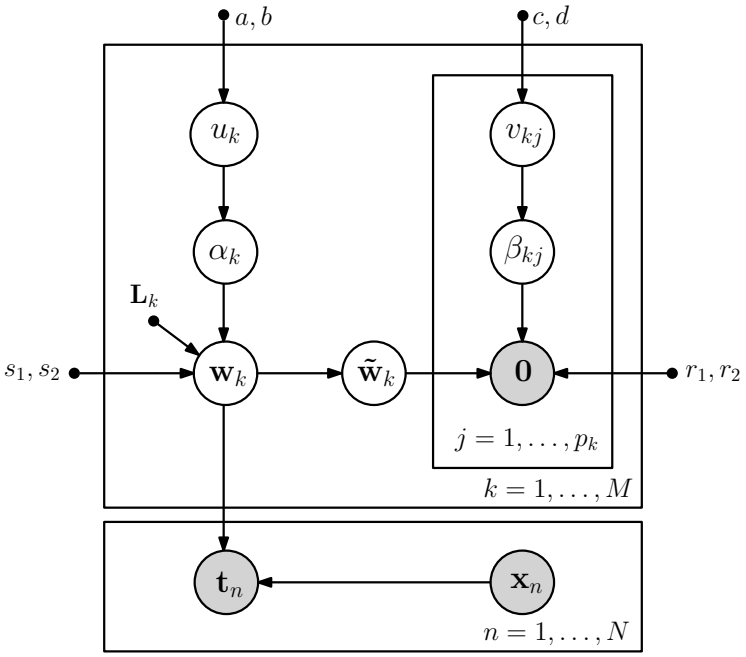


Fig. 4.1.: The graphical model representation of NaNOS.

gether through a joint prior distribution, so that the selected networks can guide node selection and, in return, the selected nodes can influence network selection.

Specifically, for the conditional model, we use a Gaussian data likelihood function for the continuous response

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \tau) = \prod_{i=1}^{N} \mathcal{N}(t_i|\mathbf{x}_i^{\mathrm{T}}\mathbf{w}, \tau^{-1}). \tag{4.1}$$

where $\mathbf{w}$ are regression weights, each of which represents the contribution of the corresponding node to the response, and τ is the precision parameter. For the unknown variance τ, we assign an uninformative diffuse Gamma prior, $\mathrm{Gam}(\tau|g, h)$ with $g = h = 10^{-6}$.

For the binary response, we use a logistic likelihood

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})^{t_i}[1 - \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})]^{1-t_i}, \tag{4.2}$$

where $t_i \in \{0, 1\}$, $\mathbf{w}$ are classifier weights, and $\sigma(\cdot)$ is the logistic function (i.e. $\sigma(y) = (1 + \exp(-y))^{-1}$). Based on the M networks, we partition $\mathbf{w}$ into M groups, so that $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]^{\top}$ where $\mathbf{w}_k$ are the weights for the explanatory variables in the k-th network.

To incorporate the topological information of a network, we use its normalized Laplacian matrix representation. Specifically, given an adjacent matrix $\mathbf{G}_k$ that represents the edges (i.e. interactions) between nodes in the k-th network, the normalized Laplacian matrix $\mathbf{L}_k$ is defined as

$$\mathbf{L}_k(i, j) = \begin{cases} 1 & i = j \text{ and } \deg(i) \neq 0 \\ \frac{1}{\sqrt{\deg(i)\deg(j)}} & i \neq j \text{ and } \mathbf{G}_k(i, j) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\deg(i) = \sum_j \mathbf{G}_k(i, j)$ is the degree of the i-th node in the k-th network.

Based on the graph Laplacian matrices, we design the following mixture prior over $\mathbf{w}_k$ to select relevant networks:

$$p(\mathbf{w}_k|\alpha_k)=\mathcal{N}(\mathbf{w}_k|\mathbf{0}, s_1\mathbf{L}_k^{-1})^{\alpha_k}\mathcal{N}(\mathbf{w}_k|\mathbf{0}, s_2\mathbf{I}_k)^{1-\alpha_k} \tag{4.3}$$

where α_k is a binary variable indicating whether the k-th network is selected, $s_1 > s_2$, $s_2 \approx 0$, and $\mathbf{I}_k$ is an identity matrix. We set the hyperparameters s_1, and s_2 based on cross-validation in our experiments. To make sure $\mathbf{L}_k$ is strictly positive-definite, we add a diagonal matrix $10^{-6}\mathbf{I}_k$ to $\mathbf{L}_k$. In (4.3), $\mathbf{L}_k$ captures the correlation information between nodes in the k-th network. Note that if we replace $\mathbf{L}_k$ by $\mathbf{I}_k$ in the slab component, the prior (4.3) becomes a simple generalization of the classical spike and slab prior (George and McCulloch, 1997) for group selection. When $\alpha_k = 1$, the k-th network is selected and the elements of $\mathbf{w}_k$ are encouraged to be similar to each other due to the Laplacian matrix $\mathbf{L}_k$; when $\alpha_k = 0$, because s_2 is close to zero, the corresponding Gaussian prior prunes $\mathbf{w}_k$. We use a Bernoulli prior distribution to reflect the uncertainty in α_k, $p(\alpha_k) = (u_k)^{\alpha_k}(1-u_k)^{1-\alpha_k}$ where $u_k \in [0,1]$ is the selection probability. Without any prior preference over selecting or pruning the k-th network, we assign a uniform prior over u_k: $p(u_k) = 1$ (i.e. $p(u_k) = \text{Beta}(u_k; a, b)$ where $a = b = 1$).

To identify relevant nodes, we introduce a latent vector $\tilde{\mathbf{w}}_k$ in the generative model for each network k, which is tightly linked to $\mathbf{w}_k$ as explained later. We use a spike and slab prior:

$$\begin{aligned} p(\tilde{\mathbf{w}}_k|\boldsymbol{\beta}_k) &= \prod_{j=1}^{p_k}\Big(\mathcal{N}(\tilde{w}_{kj}|0, r_1)^{\beta_{kj}}\mathcal{N}(\tilde{w}_{kj}|0, r_2)^{1-\beta_{kj}} \\ &= \prod_{j=1}^{p_k}\Big(\mathcal{N}(0|\tilde{w}_{kj}, r_1)^{\beta_{kj}}\mathcal{N}(0|\tilde{w}_{kj}, r_2)^{1-\beta_{kj}} \\ &= p(\mathbf{0}|\tilde{\mathbf{w}}_k, \boldsymbol{\beta}_k) \end{aligned} \tag{4.4}$$

where p_k is the number of nodes in the k-th network, $r_2 \approx 0$, and β_{kj} is a binary variable indicating whether to select the j-th node in the k-th network. We give β_{kj} a Bernoulli prior, $p(\beta_{kj}) = (v_{kj})^{\beta_{kj}}(1 - v_{kj})^{1-\beta_{kj}}$, and a uniform prior over v_{kj}: $p(v_{kj}) = 1$ (i.e., $p(v_{kj}) = \text{Beta}(v_{kj}|c, d)$ where $c = d = 1$). As shown above, the spike and slab prior $p(\tilde{\mathbf{w}}_k|\boldsymbol{\beta}_k)$ has the same form as $p(\mathbf{0}|\tilde{\mathbf{w}}_k, \boldsymbol{\beta}_k)$, which can be viewed as a generative model – in other words, the observation $\mathbf{0}$ is sampled from $\tilde{\mathbf{w}}_k$. This view enables us to combine the sparse conditional model for network selection with the sparse generative model for node selection via a principled hybrid Bayesian model.

Specifically, to link the conditional and generative models together, we introduce a prior on $\tilde{\mathbf{w}}_k$:

$$p(\tilde{\mathbf{w}}_k|\mathbf{w}_k) = \mathcal{N}(\tilde{\mathbf{w}}_k|\mathbf{w}_k, \lambda\mathbf{I}) \tag{4.5}$$

where the variance λ controls how similar $\tilde{\mathbf{w}}_k$ and $\mathbf{w}_k$ are in our joint model. For simplicity, we set $\lambda = 0$ so that $p(\tilde{\mathbf{w}}_k|\mathbf{w}_k) = \delta(\tilde{\mathbf{w}}_k - \mathbf{w}_k)$ where $\delta(f) = 1$ if $f = 0$ and $\delta(f) = 0$ otherwise. The graphical model representation of the joint model is given in Figure 4.1.

The network and node selections are consistent with each other in a probabilistic sense. If a network is pruned, all its node are removed. Because $\mathbf{w}_k = \tilde{\mathbf{w}}_k$ is enforced by the prior $\delta(\tilde{\mathbf{w}}_k - \mathbf{w}_k)$, when $\alpha_k = 0$, $\mathbf{w}_k = \mathbf{0}$ implies $\tilde{\mathbf{w}}_k = \mathbf{0}$. As a result, the spike component in (4.4) will be selected for all the nodes in the k-th network (i.e., $\beta_{kj} = 0$ for $j = 1, \ldots, p_k$) with a higher probability than the slab component. On the other hand, it is easy to see that, if one or multiple nodes in a network are selected, then this network will be selected too. Note that, if a node appears in multiple networks and is selected, our model will not force *all* the networks that contain this node to be chosen. The reason is that we duplicate the value of this node in the networks and treat their corresponding regression or classification weights as separate model parameters.

4.4 Algorithm

In this section, we present the variational Bayesian algorithm for model estimation. Specifically, we develop the variational updates to efficiently approximate the posterior distribution of weights $\mathbf{w}$, the network-selection indicators $\boldsymbol{\alpha}$, the node-selection indicators $\boldsymbol{\beta}$, the network- and node-selection probabilities $\mathbf{u}$ and $\mathbf{v}$, and the precision parameter τ for regression. Based on the posteriors of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we can decide which networks and nodes are selected.

For regression, based on the model specification in Section 4.3, the posterior distribution of our model is

$$\begin{aligned}
&p(\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{u}, \mathbf{v}, \tau | \mathbf{t}, \mathbf{X}) \\
=&\frac{1}{Z}\mathcal{N}(\mathbf{t}|\mathbf{X}\mathbf{w}, \tau^{-1}\mathbf{I})\text{Gamma}(\tau)\cdot \\
&\prod_k \Big(p(\mathbf{w}_k|\boldsymbol{\alpha}_k)p(\tilde{\mathbf{w}}_k|\mathbf{w}_k)p(\mathbf{0}|\tilde{\mathbf{w}}_k, \boldsymbol{\beta}_k)\text{Bern}(\alpha_k|u_k)\text{Beta}(u_k)\cdot \\
&\prod_j \Big(\text{Bern}(\beta_{kj}|v_{kj})\text{Beta}(v_{kj}) \qquad (4.6)
\end{aligned}$$

where $p(\mathbf{w}_k|\boldsymbol{\alpha}_k)$ and $p(\mathbf{0}|\tilde{\mathbf{w}}_k, \boldsymbol{\beta}_k)$ are defined in (4.3) and (4.4), $p(\tilde{\mathbf{w}}_k|\mathbf{w}_k) = \delta(\tilde{\mathbf{w}}_k - \mathbf{w}_k)$, and Z is the normalization constant. For classification, the posterior distribution is similar to (4.6), except that we replace the Gaussian likelihood (4.1) by the logistic function (4.2) and remove the precision parameter τ and its prior for regression in (4.6).

Classical Markov chain Monte Carlo methods can be applied to approximate the posterior distribution. However, given the high dimensionality of the parameters (*e.g.*, $\mathbf{w}$ and α), it would take a long time for a sampler to converge. In practice it is even difficult to judge the sampler's convergence. Thus, we resort to a computationally efficient variational approximation to (4.6).

Specifically, we approximate the exact posterior distribution in (4.6) by a factorized distribution: $Q(\boldsymbol{\theta}) = Q(\mathbf{w})Q(\boldsymbol{\alpha})Q(\boldsymbol{\beta})$
$Q(\mathbf{u})Q(\mathbf{v})Q(\tau)$, where $\boldsymbol{\theta}$ denotes all the latent variables. Note that, for classification,

we do not have $Q_\tau(\tau)$. Since we set $p(\tilde{\mathbf{w}}|\mathbf{w}) = \delta(\tilde{\mathbf{w}} - \mathbf{w})$, we do not need a separate distribution $Q(\tilde{\mathbf{w}})$. To solve $Q(\boldsymbol{\theta})$, we minimize the Kullback-Leibler (KL) divergence between the exact and approximate posterior distributions of $\boldsymbol{\theta}$:

$$\mathrm{KL}(Q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{t}, \mathbf{X})) = \int Q(\boldsymbol{\theta}) \ln \frac{Q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{t}, \mathbf{X})} \mathrm{d}\boldsymbol{\theta}. \tag{4.7}$$

Applying coordinate descent for the minimization of (4.7), we obtain efficient updates for the variational distributions as described in the following sections. The updates are iterative: we update one of the variational distributions at a time while having all the other variational distributions fixed, and iterate these updates until convergence. Since these updates monotonically decrease the value of the KL divergence (4.7), which is lower bounded by zero, they are guaranteed to converge in terms of the KL value (Bishop, 2006).

4.4.1 Regression

The variational distributions for regression have the following forms:

$$\begin{aligned}
Q(\mathbf{w}) &= \mathcal{N}(\mathbf{w}|\mathbf{m}, \boldsymbol{\Sigma}) && (4.8)\\
Q(\boldsymbol{\alpha}) &= \prod_k {\gamma_k}^{\alpha_k} (1-\gamma_k)^{1-\alpha_k} && (4.9)\\
Q(\boldsymbol{\beta}) &= \prod_k \prod_j (\eta_{kj})^{\beta_{kj}} (1-\eta_{kj})^{1-\beta_{kj}} && (4.10)\\
Q(\mathbf{u}) &\propto \prod_k (u_k)^{\tilde{a}_k - 1} (1-u_k)^{\tilde{b}_k - 1} && (4.11)\\
Q(\mathbf{v}) &\propto \prod_k \prod_j (v_{kj})^{\tilde{c}_{kj} - 1} (1-v_{kj})^{\tilde{d}_{kj} - 1} && (4.12)\\
Q(\tau) &= \Gamma(\tau|\tilde{g}, \tilde{h}) && (4.13)
\end{aligned}$$

Their parameters are iteratively updated as follows:

$$\boldsymbol{\Sigma} = (\mathbf{A} + \langle\tau\rangle\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} \qquad \mathbf{m} = \langle\tau\rangle\boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}}\mathbf{t} \tag{4.14}$$

$$\tilde{a}_k = \gamma_k + a \qquad \tilde{b}_k = 1 - \gamma_k + b \tag{4.15}$$

$$\tilde{c}_{kj} = \eta_{kj} + c \qquad \tilde{d}_{kj} = 1 - \eta_{kj} + d \tag{4.16}$$

$$\begin{aligned}\gamma_k = {} & 1/(1 + \exp(\langle\ln(1-u_k)\rangle - \langle\ln u_k\rangle + \frac{p_k}{2}\ln\frac{s_1}{s_2} \\ & -\frac{1}{2}\ln|\mathbf{L}_k| + \frac{1}{2}\mathrm{tr}(\langle\mathbf{w}_k\mathbf{w}_k{}^{\mathrm{T}}\rangle(\frac{1}{s_1}\mathbf{L}_k - \frac{1}{s_2}\mathbf{I}_k))\end{aligned} \tag{4.17}$$

$$\begin{aligned}\eta_{kj} = {} & 1/(1 + \exp(\langle\ln(1-v_{kj})\rangle - \langle\ln v_{kj}\rangle \\ & +\frac{1}{2}\ln\frac{r_1}{r_2} + \frac{1}{2}\langle(w_{kj})^2\rangle(\frac{1}{r_1} - \frac{1}{r_2})))\end{aligned} \tag{4.18}$$

$$\tilde{h} = h + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} - \mathbf{m}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\sum\nolimits_i\Big(\mathbf{x}_i^{\mathrm{T}}\langle\mathbf{w}\mathbf{w}^{\mathrm{T}}\rangle\mathbf{x}_i \tag{4.19}$$

$$\tilde{g} = g + \frac{N}{2} \tag{4.20}$$

where $\mathbf{A} = \frac{1}{s_1}\mathrm{diag}(\{\gamma_k\mathbf{L}_k\}_k) + \frac{1}{s_2}\mathrm{diag}(\{(1-\gamma_k)\mathbf{I}_k\}_k) + \frac{1}{r_1}\mathrm{diag}(\boldsymbol{\eta}) + \frac{1}{r_2}\mathrm{diag}(1-\boldsymbol{\eta})$ (note that $\mathrm{diag}(\{\gamma_k\mathbf{L}_k\}_k)$ is a block-diagonal matrix), $\langle\cdot\rangle$ means expectation over the corresponding variational distribution, and the required moments in the above equations are

$$\langle\mathbf{w}\mathbf{w}^{\mathrm{T}}\rangle = \boldsymbol{\Sigma} + \mathbf{m}\mathbf{m}^{\mathrm{T}} \qquad \langle\tau\rangle = \tilde{g}/\tilde{h}$$

$$\langle\ln u_k\rangle = \psi(\tilde{a}_k) - \psi(\tilde{e}_k) \qquad \langle\ln(1-u_k)\rangle = \psi(\tilde{b}_k) - \psi(\tilde{e}_k)$$

$$\langle\ln v_{kj}\rangle = \psi(\tilde{c}_{kj}) - \psi(\tilde{f}_{kj}) \qquad \langle\ln(1-v_{kj})\rangle = \psi(\tilde{d}_{kj}) - \psi(\tilde{f}_{kj})$$

where $\psi(x) = \frac{\mathrm{d}}{\mathrm{dx}}\ln\Gamma(x)$, $\tilde{e}_k = \tilde{a}_k + \tilde{b}_k$ and $\tilde{f}_{kj} = \tilde{c}_{kj} + \tilde{d}_{kj}$.

4.4.2 Classification

Compared to regression, the classification task is more challenging. Because of the logistic function (4.2), we cannot directly solve the variational distribution $Q(\mathbf{w})$.

Therefore, we use a lower bound proposed by (Jaakkola and Jordan, 2000) to replace the logistic function in the joint distribution:

$$\begin{aligned}&\sigma(y)^t\big(1-\sigma(y)\big)^{1-t}\\ &\geq \sigma(\xi)\exp\Big(\frac{(2t-1)y-\xi}{2}-f(\xi)\big((2t-1)^2y^2-\xi^2\big)\Big)\end{aligned} \tag{4.21}$$

where $f(\mathbf{x})=\frac{1}{4\xi}\tanh(\xi/2)$, and ξ is a variational parameter. Note that the equality is achieved when $\xi=(2t-1)y$. Since the logarithm of the lower bound (4.21) is quadratic in y, it essentially converts the logistic function into a Gaussian form so that the variational inference becomes tractable.

Combining the maximization of the lower bound (4.21) with the minimization of the KL divergence (4.7), we obtain the variational updates for classification. They are the same as those for the regression task, except for that $Q(\mathbf{w})=\mathcal{N}(\mathbf{w}|\mathbf{m},\mathbf{\Sigma})$, now we have

$$\mathbf{\Sigma}=\Big(\mathbf{A}+2\sum\nolimits_i f(\xi_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\Big)^{-1}\quad \mathbf{m}=\frac{1}{2}\mathbf{\Sigma}\mathbf{X}^{\mathrm{T}}(2\mathbf{t}-\mathbf{1}) \tag{4.22}$$

where $\mathbf{A}$ is the same as in the regression.

In addition, maximization of the lower bound of the logistic function gives the update for the variational parameter ξ_i:

$$\xi_i^2=\mathbf{x}_i^{\mathrm{T}}\langle\mathbf{w}\mathbf{w}^{\mathrm{T}}\rangle\mathbf{x}_i. \tag{4.23}$$

4.4.3 Computational cost

The computational cost of the proposed algorithm is dominated by (4.14) for regression and (4.22) for classification. For both cases, it takes $O(p^3)$ for matrix inversion to obtain $\mathbf{\Sigma}$ and $O(Np+p^2)$ to obtain $\mathbf{m}$ for each iteration. Thus, the total cost is $O(p^3+Np)$ and, for most applications where $p>N$, it simplifies to $O(p^3)$.

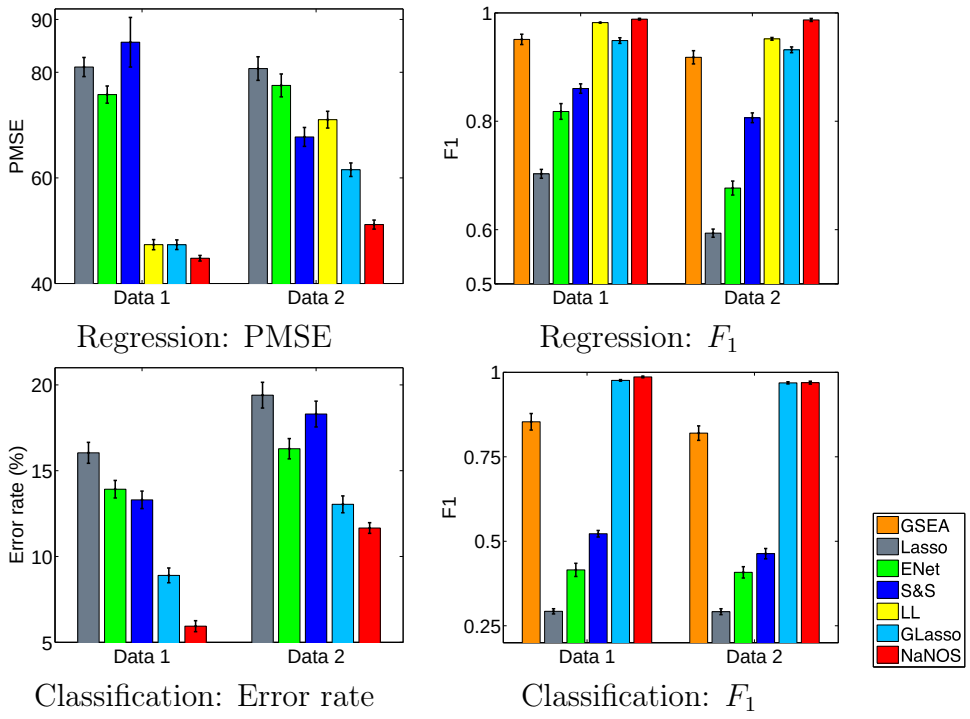
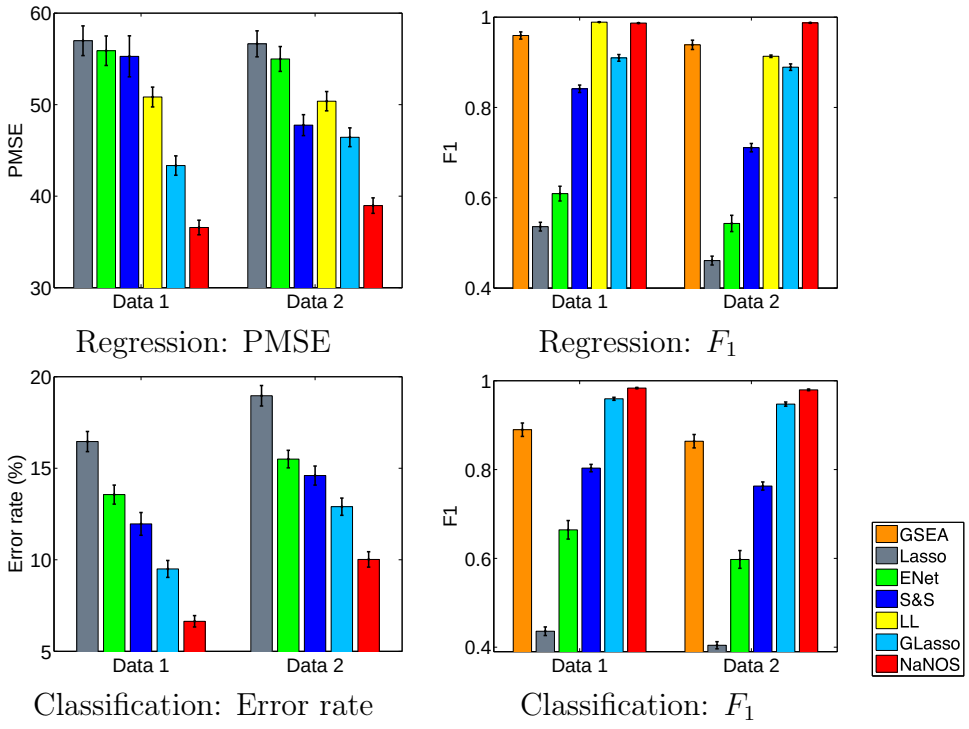


Fig. 4.2.: Prediction errors and F_1 scores for gene selection in Experiment 1. ENet, S&S, and GLasso stand for elastic net, the spike and slab model, and group lasso, respectively; and Data 1 and 2 indicate the first and second data generation models.



Regression: PMSE Regression: F_1

Classification: Error rate Classification: F_1

Fig. 4.3.: Prediction errors and F_1 scores for gene selection in Experiment 2.

4.5 Experiments

In this section, we apply NaNOS to synthetic and real gene expression data to select pathways (i.e., networks) and genes (i.e., nodes), and provide biological analysis of our results. We also compare NaNOS with alternative methods, including lasso Tibshirani (1996), elastic net Zou and Hastie (2005), group lasso Yuan and Lin (2007); Jacob et al. (2009), the network-constrained regularization approach (Li and Li (2008), henceforth "LL"), and the sparse Bayesian model with the classical spike and slab prior (George and McCulloch, 1997). For lasso and elastic net, we used the Glmnet software package [1]. For group lasso, we treat each pathway as a group. To handle genes appearing in multiple pathways (i.e., groups), we first duplicated their expression levels for each group – as suggested by (Jacob et al., 2009) – and then used the SLEP software package [2] for group lasso estimation. For the spike and slab model, we implemented variational inference similar to our updates in Section 4.4. Just as NaNOS, all these software packages use the Gaussian likelihood for regression and the logistic likelihood for classification. We used the default configuration of these software packages for the maximum number of iterations, initial values, and the threshold for convergence. To tune regularization weights in lasso, group lasso and the LL approach, we conducted thorough 10-fold cross validation (CV) on training data (i.e., not using the test data) using a large computer cluster. The CV grids on the free parameters are summarized here: for lasso, $\alpha = [0 : 0.01 : 1]$; for elastic net, $\alpha = [0 : 0.01 : 1]$ and $\beta = [0 : 0.01 : 1]$; for group lasso (both regression and logistic regression), $\alpha = [0 : 0.01 : 1]$; and for the LL approach, $\lambda_1 = [1 : 25 : 300]$ and $\lambda_2 = [1 : 25 : 300]$ (we also did a second-level CV after we pruned the range of λ_1 and λ_2 values based on the first-level CV). Finally, for NaNOS, the cross-validation grids are $s_1 = r_1 = [0.1, 1, 3]$ and $s_2 = r_2 = [10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$.

On the synthetic data for which we knew the true relevant pathways, we also compared NaNOS with the popular tool for gene set enrichment analysis (GSEA)

[1] `www-stat.stanford.edu/~tibs/glmnet-matlab/`
[2] `www.public.asu.edu/~jye02/Software/SLEP/`

(Mootha et al., 2003; Subramanian et al., 2005). We treated each pathway as a set, used GSEA's default configuration, and applied its suggested criterion FDR $< 25\%$ to discover enriched pathways. We then identified all the genes in these enriched pathways as target genes. Because GSEA cannot provide predictions on responses $\mathbf{t}$, we did not include it for comparison on the real data.

4.5.1 Simulation studies

We first compare all the methods on synthetic data in the following three experiments.

Experiment 1. We followed the first and second data generation models used by Li and Li (2008). Specifically, we simulated expression levels of 200 transcription factors (TFs), each controlling 10 genes in a simple tree-structured regulatory network, and assumed that 4 pathways – including *all* of their genes – have effect on the response $\mathbf{t}$. We sampled the expression levels of each TF from a standard normal distribution, $x_{TF} \sim \mathcal{N}(0, 1)$ and the expression level of each gene that this TF regulates from $\mathcal{N}(0.7x_{TF}, 0.51)$. This implies a correlation of 0.7 between the TF and its target genes.

For the first model with the continuous response, we designed a weight vector for each pathway, $\boldsymbol{\rho} = [1, \frac{1}{\sqrt{10}}, \ldots, \frac{1}{\sqrt{10}}]$, corresponding to the TF and 10 genes it regulates, and then sampled $\mathbf{t}$ as follows:

$$\begin{aligned} \mathbf{w} &= [5\boldsymbol{\rho}, -5\boldsymbol{\rho}, 3\boldsymbol{\rho}, -3\boldsymbol{\rho}, \mathbf{0}^\top]^\top \\ \mathbf{t} &= \mathbf{X}\mathbf{w} + \epsilon \end{aligned}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_e^2)$ and $\mathbf{0}$ is a vector of all zeros.

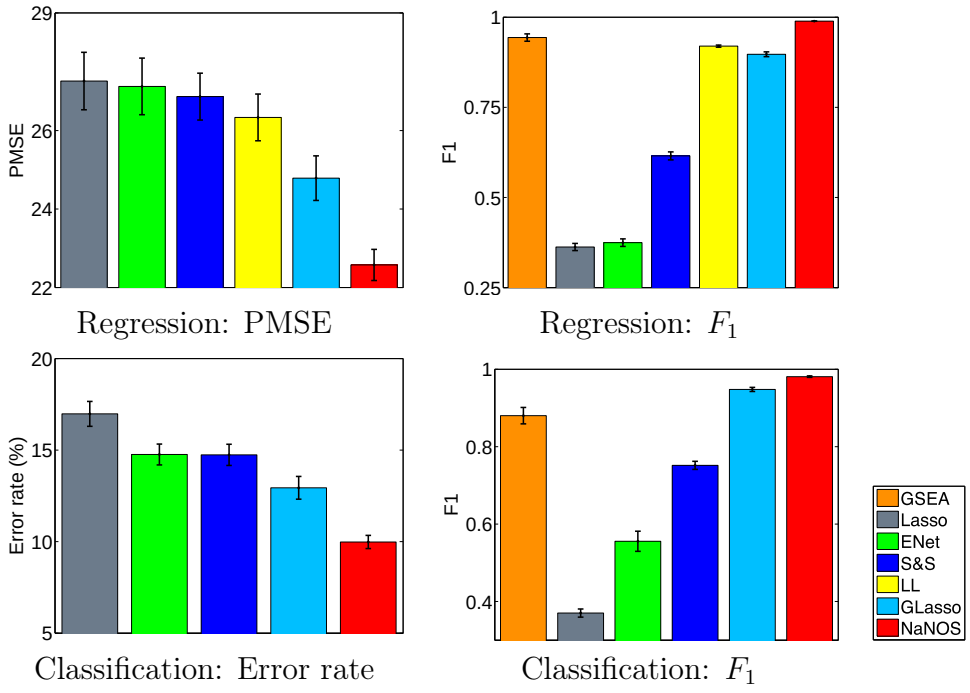


Fig. 4.4.: Prediction errors and F_1 scores for gene selection in Experiment 3.

The second model is the same as the first one, except that the genes regulated by the same TF can have either positive or negative effect on the response $\mathbf{t}$. Specifically, we set

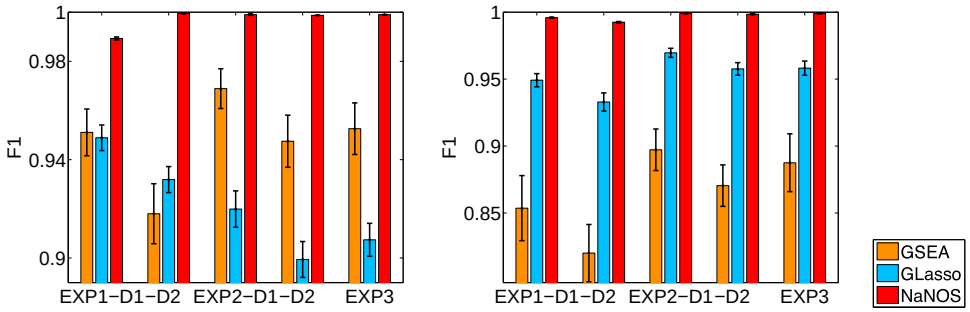
$$\boldsymbol{\rho} = [1, \frac{-1}{\sqrt{10}}, \frac{-1}{\sqrt{10}}, \frac{-1}{\sqrt{10}}, \underbrace{\frac{1}{\sqrt{10}}, \ldots, \frac{1}{\sqrt{10}}}_{7}].$$

For the first and second models, the noise variance was set to be $\sigma_e^2 = (\Sigma_j w_j^2)/4$ so that the signal-to-noise ratio was 12.85 and 7.54, respectively.

For the binary response, we followed the same procedure as for the continuous response to generate expression profiles $\mathbf{X}$ and the parameters $\mathbf{w}$. Then we sampled $\mathbf{t}$ from (4.2).

For each of the settings, we simulated 100 samples for training and 100 samples for test. We repeated the simulation 50 times. To evaluate the predictive performance, we calculated the prediction mean-squared error (PMSE) for regression and the error rate for classification. To examine the accuracy of gene and pathway selection, we

also computed sensitivity and specificity and summarized them in the F_1 score, $F_1 = 2\ (\text{sensitivity} \times \text{specificity})/(\text{sensitivity} + \text{specificity})$. The bigger the F_1 score, the higher the selection accuracy.



Regression: F_1 Classification: F_1

Fig. 4.5.: F_1 scores for pathway selection. "EXP" stands for "Experiment" and "D" stands for "Data model".

All the results are summarized in Figure 4.2, in which the error bars represent the standard errors. For all the settings, NaNOS gives smaller errors and higher F_1 scores for gene selection than the other methods, except that, for classification of the samples from the second data model, NaNOS and group lasso obtain the comparable F_1 scores. All the improvements are significant under the two-sample t-test ($p < 0.05$). We also show the accuracy of group lasso, GSEA and NaNOS for pathway selection in Figure 4.5. Again, NaNOS achieves significantly higher selection accuracy. Because the LL approach was developed for regression, we did not have its classification results. While the LL approach uses the topological information of all the pathways, they are merged together into a *global* network for regularization. In contrast, using a sparse prior over individual pathways, NaNOS can explicitly select pathways relevant to the response, guiding the gene selection. This may contribute to its improved performance.

Experiment 2. For the second experiment, we did not require all genes in relevant pathways to have effect on the response. Specifically, we simulated expression levels of 100 transcription factors (TFs), each regulating 21 genes in a simple regulatory

network. We sampled the expression levels of the TFs, the regulated genes, and their response in the same way as in Experiment 1, except that we set

$$\boldsymbol{\rho} = [1, \underbrace{\frac{1}{\sqrt{21}}, \ldots, \frac{1}{\sqrt{21}}}_{10}, \underbrace{0, \ldots, 0}_{11}]$$

for the first data generation model and

$$\boldsymbol{\rho} = [1, \frac{-1}{\sqrt{21}}, \frac{-1}{\sqrt{21}}, \frac{-1}{\sqrt{21}}, \underbrace{\frac{1}{\sqrt{21}}, \ldots, \frac{1}{\sqrt{21}}}_{7}, \underbrace{0, \ldots, 0}_{11}] \tag{4.24}$$

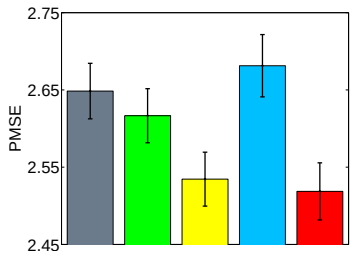
for the second data generation model. Note that the last eleven zero elements in $\boldsymbol{\rho}$ indicate that the corresponding genes have no effect on the response $\mathbf{t}$, even in the four relevant pathways.

The results for both the continuous and binary responses are summarized in Figures 4.3 and 4.5. For regression based on the first data model, NaNOS and LL obtain the comparable F_1 scores; for all the other cases, NaNOS significantly outperforms the alternative methods in terms of both prediction and selection accuracy ($p < 0.05$).

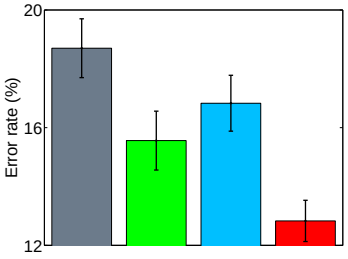
Experiment 3. Finally, we simulated the data as in Experiment 2, except that we replaced $\sqrt{21}$ in the denominators in (4.24) with 21, to obtain a weaker regulatory effect of the TF. Again, as shown in Figures 4.4 and 4.5, NaNOS outperforms the competing methods significantly.

4.5.2 Application to expression data

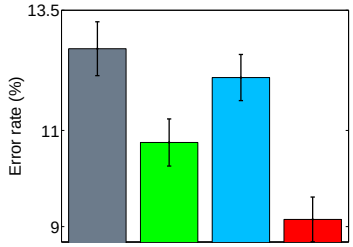
Now we demonstrate the proposed method by analyzing gene expression datasets for the cancer studies of diffuse large B cell lymphoma (DLBCL) (Rosenwald et al., 2002), colorectal cancer (CRC) (Ancona et al., 2006), and pancreatic ductal adenocarcinoma (PDAC) (Badea et al., 2008). We used the probeset-to-gene mapping provided in these studies. For the CRC and PDAC datasets in which multiple probes were mapped to the same genes, we took the average expression level of these probes. We used the pathway information from the KEGG pathway database



(a) Diffuse large B cell lymphoma



(b) Colorectal cancer



(c) Pancreatic ductal adenocarcinoma

Fig. 4.6.: Predictive performance on three gene expression studies of cancer.

(`www.genome.jp/kegg/pathway.html`) by mapping genes from the cancer studies into the database, particularly in the categories of Environmental Information Processing, Cellular Processes and Organismal Systems.

Diffuse large B cell lymphoma. We used gene expression profiles of 240 DLBCL patients from an uncensored study in the Lymphoma and Leukemia Molecular Profiling Project (Rosenwald et al., 2002). From 7399 probes, we found 752 genes and 46 pathways in the KEGG dataset. The median survival time of the patients is 2.8 years after diagnosis and chemotherapy. We used the logarithm of survival times of patients as the response variable in our analysis.

We randomly split the dataset into 120 training and 120 test samples 100 times and ran all the competing methods on each partition. The test performance is visualized in Figure 4.6.a. NaNOS significantly outperforms lasso, elastic net and group lasso. Although the results of the LL approach can contain connected sub-networks, these sub-networks do not necessarily correspond to (part of) a biological pathway. For instance, they may consist of components from multiple overlapped pathways. In contrast, NaNOS explicitly selects relevant pathways. Four pathways had the selec-

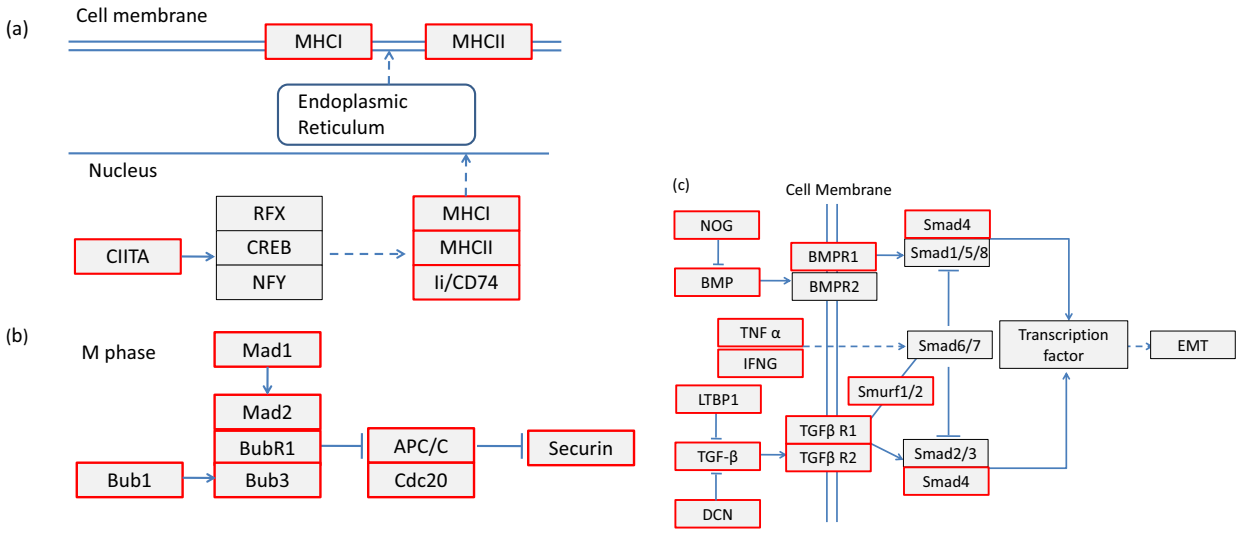


Fig. 4.7.: Examples of part of identified pathways. (a): the antigen processing and presentation pathway for DLBCL; (b): the cell cycle pathway for CRC;

(c): the TGF-β signaling pathway for PDAC. Red and black boxes indicate selected and not selected genes, respectively.

tion posterior probabilities larger than 0.95 and they were consistently chosen in all the 100 splits. Two of these pathways are discussed below.

First, NaNOS selected the antigen processing and presentation pathway. The part of this pathway containing selected genes is visualized in Figure 4.7.a. A selected regulator CIITA was shown to regulate two classes of antigens MHC I and II in DLBCL (Cycon et al., 2009). The loss of MHC II on lymphoma cells – including the selected HLA-DMB, -DQB1, -DMA, -DRA, -DRB1, -DPA1, -DPB1, and -DQA1 – was shown to be related to poor prognosis and reduced survival in DLBCL patients (Rosenwald et al., 2002). The selected MHC I (*e.g.*, HLA-A,-B,-C,-G) was reported to be absent from the cell surface, allowing the escape from immunosurveillance of lymphoma (Amiot et al., 1998). And the selected Ii/CD74 and HLA-DRB were proposed to be monoclonal antibody targets for DLBCL drug design (Dupire and Coiffier, 2010).

Second, NaNOS chose cell adhesion molecules (CAMs). Adhesive interactions between lymphocytes and the extracellular matrix are essential for lymphocytes' migration and homing. For example, the selected CD99 is known to be over-expressed in DLBCL and correlated with survival times (Lee et al., 2011), and LFA-1 (ITGB2/ITGAL) can bind to ICAM on the cell surface and further lead to the invasion of lymphoma cells into hepatocytes (Terol et al., 1999).

Colorectal cancer. We applied our model to a colorectal cancer dataset (Ancona et al., 2006). It contains gene expression profiles from 22 normal and 25 tumor tissues. We mapped 2455 genes from 22,283 probes into 67 KEGG pathways. The goal was to predict whether a tissue has the colorectal cancer or not and select relevant pathways and genes.

We randomly split the dataset into 23 training and 24 test samples 50 times and ran all the methods on each partition. The test performance is visualized in Figure 4.6.b. Again, based on a two-sample t-test, NaNOS outperforms the alternatives significantly ($p < 0.05$). Three out of the four pathways with the selection posterior

probabilities larger than 0.95 are discussed below. They were selected 20, 50 and 50 times in the 50 splits.

First, NaNOS selected the cell cycle pathway. This selection is consistent with the original result by Ancona et al. (2006). As shown in 4.7.b, NaNOS selected mitotic spindle assembly related genes. Specifically, Bub1 and Mad1 may regulate the checkpoint complex (MCC) containing Mad2, BubR1 and Bub3. The upregulated MCC may in turn inhibit ability of APC/C to ubiquitinate securin and further lead to mitotic event extension in CRC (Menssen et al., 2007). NaNOS also chose cyclin/CDK complexes, among which CycD/CDK4 overexpression is found in mouse colon tumor and CDK1, CDK2, CycE are increased in human CRC Wang et al. (1998); Vermeulen et al. (2003). NaNOS further identified MCM (minichromosome maintenance) complex – MCM2 and MCM5 – which are biomarkers for the CRC stage identification (Giaginis et al., 2009). Moreover, the selected TP53 and c-Myc are known to be closely related to CRC (Menssen et al., 2007).

Second, NaNOS chose the intestinal immune network for IgA production. A greatly increased level of IgA – as a result of long-term intestinal inflammation – can increase the chance of CRC (Rizzo et al., 2011) and serve as an effective biomarker for early diagnosis of CRC (Chalkias et al., 2011). Also, selected chemkines in this pathway, such as CXCR4 and CXCL12, may contribute to CRC progression (Sakai et al., 2012).

Third, NaNOS selected the cytokine-cytokine receptor interaction pathway as well as several well-known CRC-related molecules in this pathway. For instance, CXCL13 is a biomarker for stage II CRC prognosis (Agesen et al., 2012); CXCL10 dramatically increases with CRC progression (Toiyama et al., 2012); and IL10 secreted by CRC cells can accelerate tumor proliferation and be used for the prognosis of CRC progression (Toiyama et al., 2010).

Pancreatic ductal adenocarcinoma. This cancer dataset includes expression profiles from 39 PDAC and 39 normal subjects (Badea et al., 2008). By mapping 2781 genes from 54677 probes into KEGG pathways, we obtained 67 pathways. Our goal

was to predict whether a subject has the pancreatic cancer and select relevant pathways and genes. We randomly split the dataset into 39 training and 39 test samples 50 times and ran all the methods on each partition. The test performance is visualized in Figure 4.6.c. Based on a two-sample t-test, NaNOS significantly outperforms lasso, elastic net and group lasso.

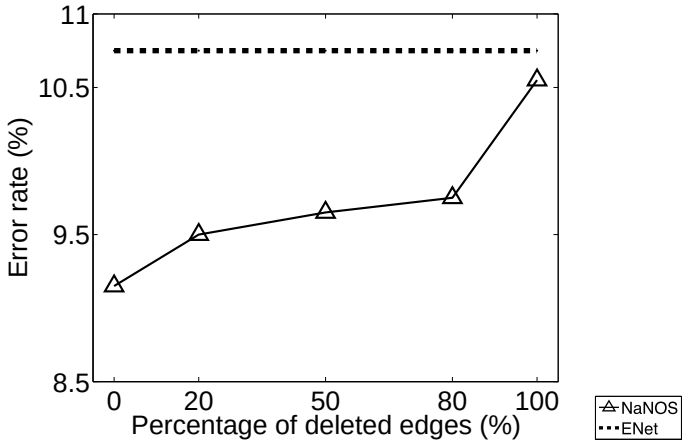


Fig. 4.8.: The predictive performance of NaNOS when the pathway structures are inaccurate. When more edges are randomly selected and removed from each pathway, the performance of NaNOS degrades smoothly, but still better than the competing methods.

To investigate the sensitivity of NaNOS to the structural noise in the pathway database, we randomly chose 20%, 50%, 80% and 100% edges in each pathway and removed them. We tested NaNOS for each case and reported the average test error rate in the new Figure 4.8. As expected, the error rate of NaNOS gradually increases with more edges being removed because less topological information in pathways is available. But NaNOS still consistently outperformed all the alternative methods such as elastic net, the second best method on this dataset. This experiment demonstrates i) that, by exploiting subtle correlation information embedded in the pathway topology, NaNOS can boost its modeling power and predictive performance, and ii) that NaNOS is robust to small perturbation in pathway topology.

We also examined the impact of the important prior distributions on pathway and gene selection probabilities u_k and v_{kj}. As described in Section 4.3, we used the uniform priors (i.e., the Beta(1,1) prior) over u_k and v_{kj}, indicating no prior preference over selecting a pathway or gene or not. The average test error based on the uninformative priors is 9.15 ± 0.5, as visualized in Figure 4.6.c. If we change the prior to a very informative one Beta(1,10) (mean 0.09 and standard deviation 0.083) that strongly prefers sparsity, then the average test error increases slightly to 10.0 ± 0.4. This *minor* increase in error may stem from the over-sparification caused by the sparsity prior that are over-confident (suggested by a small variance). Now if we use another informative prior Beta(10,1) (mean 0.91 and standard deviation 0.083) that strongly prefers dense – instead of sparse – estimation, then the average test error increases to 11.2 ± 0.5. This relatively larger error increase is exactly what we expected because now the *wrong* dense prior aims to select most pathways and genes. What is important is that, no matter which of these two informative priors we chose, NaNOS consistently outperformed lasso and group lasso in 4.6.c. Between these two extreme cases, if we use an uninformative or weak sparse prior (*e.g.*, Beta(0.5,0.5)), we find that similar prediction error rates were obtained for NaNOS as in 4.6.c. The above analysis indicates that NaNOS is robust to the prior choice.

In addition to using the even splitting strategy with the same number of training and test samples, we also tested the performance of all the algorithms in another setting with more training samples – specifically, 62 training and 16 test samples. We repeated the random partitioning 50 times. The average error rates for NaNOS, elastic net, lasso and group lasso are 8.00 ± 0.89, 9.90 ± 1.00, 12.0 ± 1.0 and 11.0 ± 0.14, respectively. Again, the two-sample t test indicates that NaNOS outperforms the alternative methods significantly ($p < 0.05$).

Three out of the five pathways with the selection posterior probabilities larger than 0.95 are discussed below. They were selected 35, 50 and 50 times in the 50 splits.

The first selected pathway was the TGF-β signaling pathway. It is essential in epithelial-mesenchymal transition (EMT) – a critical component for developmental and cancer processes – and related to PDAC (Krantz et al., 2012). The selected part of this pathway is visualized in Figure 4.7.c. It shows that IFNG, TNF-α, LTBP1, DCN, TGF-β, and its receptor TGF-β R1 were selected. The TGF-β ligand – via its receptor – propagates the signal through phosphorylation of Smads including the selected Smad 4, which in turn translocate into the nucleus and interact with Snail TFs to regulate EMT (Krantz et al., 2012). The selected BMP ligand (i.e., BMP2) is bound to BMP R1 and R2 receptors to activate Smad1, which is in a protein complex including Smad4. (Gordon et al., 2009) showed that in PANC-1 cell line this protein complex mediates EMT partially by increasing the activity of MMP-2.

The second identified pathway was extracellular matrix (ECM)-receptor interaction. It is associated with desmoplastic reaction, a hallmark in PDAC (Shields et al., 2012). In this pathway, NaNOS selected the integrin receptors – including ITGB1, ITGA2, ITGA3, ITGA5, ITGA6 – and the ECM proteins – collagens inlcuding COL1A1 and COL1A2 and laminins including LAMC2 and LAMB3. Important interactions among them were revealed in a previous study by Weinel et al. (1992).

The third chosen pathway was CAMs. CAMs are pivotal in pancreatic cancer invasion by mediating cell-cell signal transduction and cell-matrix communication (Keleg et al., 2003). In this pathway, the selected molecules include calcium-dependent cadherin family molecules (CDH2, CDH3) and neural-related molecules (MAG); both of them have shown to be related to PDAC (Kameda et al., 1999) (Keleg et al., 2003).

4.6 Discussion

As shown in the previous section, the new Bayesian approach, NaNOS, outperformed the alternative sparse learning methods on both simulation and real data by a large margin. Now we discuss three factors that may contribute to the improved performance of NaNOS.

First, the spike and slab prior (4.3) and its generalization (4.4) in NaNOS separate weight regularization from the selection of variables (pathways or genes). Both the (generalized) spike and slab prior and elastic net can be viewed as mixture models, in which one component encourages the selection of variables and the other helps remove irrelevant ones. However, unlike the elastic net where the weights over l_1 and l_2 penalty functions are fixed, the spike and slab prior has the selection indicators over these two components estimated from data. When a variable is selected, the model has a Gaussian prior over its value (i.e., weight) that is equivalent to a l_2 regularizer (as in ridge regression) and does not shrink the value of the selected variable as l_1 penalty would do. By contrast, lasso or elastic net, with a fixed mixture weight, has sparsity penalty over both pruned and selected variables, which can greatly shrink the values of selected variables and hurt predictive performance.

Second, NaNOS incorporates correlation structures encoded in pathways for variable selection. Specifically, it uses pathway structures into the extended spike and slab prior to explicitly model the detailed relationships between correlated genes. In contrast, Lasso and elastic net do not use this valuable correlation information in their models. By comparing prediction accuracies of NaNOS when 0% and 100% edges are removed from pathways (See Figure 4.8), we can see that the detailed correlation information captured by the pathway topology can greatly improve modeling quality.

Third, NaNOS has the capability of selecting both relevant pathways and genes due to its two-layer sparse structure. By contrast, with l_1/l_2 penalty, group lasso encourages the selection of all the genes in chosen pathways, leading to *dense* estimation. This may be undesirable in practice and deteriorate the predictive performance of group lasso. NaNOS enhances the *flexibility* of group lasso by conducting sparse estimation at both the pathway (or group) and gene levels. Meanwhile, our Bayesian estimation effectively avoids overfitting, a problem often plaguing flexible models.

NaNOS has been applied to joint pathway and gene selection in this paper. Inspired by the seminal works in (Frohlich et al., 2006; Chuang et al., 2007; Srivastava et al., 2008; Zycinski et al., 2013), we can use NaNOS in a variety of biomedi-

cal applications where there are abundant high-dimensional biomarkers of individual samples and other information sources – for example, the gene ontology (GO) and protein-protein interaction networks information – that capture correlation in the high-dimensional space. Here we discuss two approaches to apply NaNOS when we have only GO or other group information without network topology. The first approach is to compute some distance or similarity scores between genes based on the GO information (*e.g.*, following the approach by Srivastava et al. (2008)) and then estimate the network topology based on a network learning method, for example, graphical lasso (Friedman et al., 2008). With the estimated network topology, we can compute the graph Laplacian matrices and apply NaNOS to select genes and groups of genes. The second approach is to directly use the group membership information in NaNOS by replacing the graph Laplacian matrices with identity matrices. This approach becomes useful when we even do not have any information available to learn the network topology. As shown in Figure 8, even when all the edges were removed and we had only group information, NaNOS still outperformed the second best method, elastic net, in terms of prediction accuracy.

4.7 Funding

This work was supported by NSF IIS-0916443, NSF CAREER Award IIS-1054903, and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

4.8 References

Agesen, T. H., Sveen, A., Merok, M. A., Lind, G. E., Nesbakken, A., Skotheim, R. I., and Lothe, R. A. (2012). ColoGuideEx: a robust gene classifier specific for stage II colorectal cancer prognosis. *Gut*.

Amiot, L., Onno, M., Lamy, T., Dauriac, C., LE Prise, P.-Y., Fauchet, R., and Drenou, B. (1998). Loss of HLA molecules in B lymphomas is associated with an aggressive clinical course. *British Journal of Haematology*, 100(4):655–663.

Ancona, N., Maglietta, R., Piepoli, A., D'Addabbo, A., Cotugno, R., Savino, M., Liuni, S., Carella, M., Pesole, G., and Perri, F. (2006). On the statistical assessment

of classifiers using DNA microarray data. *BMC Bioinformatics*, 7(387).

Badea, L., Herlea, V., Dima, S., Dumitrascu, T., and Popescu, I. (2008). Combined gene expression analysis of whole-tissue and microdissected pancreatic ductal adenocarcinoma identifies genes specifically overexpressed in tumor epithelia. *Hepatogastroenterology*, 55(88):2016–2027.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Chalkias, A., Nikotian, G., Koutsovasilis, A., Bramis, J., Manouras, A., Mystrioti, D., and Katergiannakis, V. (2011). Patients with colorectal cancer are characterized by increased concentration of fecal hb-hp complex, myeloperoxidase, and secretory IgA. *American Journal of Clinical Oncology*, 34(6):561–566.

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, 3(140).

Cycon, K. A., Rimsza, L. M., and Murphy, S. P. (2009). Alterations in CIITA constitute a common mechanism accounting for downregulation of MHC class II expression in diffuse large B-cell lymphoma (DLBCL). *Experimental Hematology*, 37(2):184–194.

Dupire, S. and Coiffier, B. (2010). Targeted treatment and new agents in diffuse large B cell lymphoma. *International Journal of Hematology*, 92(1):12–24.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Frohlich, H., Speer, N., Spieth, C., and Zell, A. (2006). Kernel based functional gene grouping. In *International Joint Conference on Neural Networks*, pages 3580–3585.

George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica Sinica*, 7(2):339–373.

Giaginis, C., Georgiadou, M., Dimakopoulou, K., Tsourouflis, G., Gatzidou, E., Kouraklis, G., and Theocharis, S. (2009). Clinical significance of MCM-2 and MCM-5 expression in colon cancer: association with clinicopathological parameters and tumor proliferative capacity. *Digestive Diseases and Sciences*, 54(2):282–291.

Gordon, K. J., Kirkbride, K. C., How, T., and Blobe, G. C. (2009). Bone morphogenetic proteins induce pancreatic cancer cell invasiveness through a Smad1-dependent mechanism that involves matrix metalloproteinase-2. *Carcinogenesis*, 30(2):238–248.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation through varational methods. *Statistics and Computing*, 10(1):25–37.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, pages 433–440, New York.

Kameda, K., Shimada, H., Ishikawa, T., Takimoto, A., Momiyama, N., Hasegawa, S., Misuta, K., Nakano, A., Nagashima, Y., and Ichikawa, Y. (1999). Expression of highly polysialylated neural cell adhesion molecule in pancreatic cancer neural invasive lesion. *Cancer Letter*, 137(2):201–207.

Keleg, S., Büchler, P., Ludwig, R., Büchler, M. W., and Friess, H. (2003). Invasion and metastasis in pancreatic cancer. *Molecular Cancer*, 2(14).

Krantz, S. B., Shields, M. A., Dangi-Garimella, S., Munshi, H. G., and Bentrem, D. J. (2012). Contribution of epithelial-to-mesenchymal transition and cancer stem cells to pancreatic cancer progression. *Journal of Surgical Research*, 173(1):105–112.

Lasserre, J. A., Bishop, C. M., and Minka, T. P. (2006). Principled hybrids of generative and discriminative models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 87–94.

Lee, S., Park, S., Park, J., Hong, J., and Ko, J. (2011). Clinicopathologic characteristics of CD99-positive diffuse large B-cell lymphoma. *Acta Haematologica*, 125(3):167–174.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomics data. *Bioinformatics*, 24(9):1175–1182.

Li, F. and Zhang, N. (2010). Bayesian variable selection in structured high-dimensional covariate space with applications in genomics. *Journal of the American Statistical Association*, 105(491):1202–1214.

Menssen, A., Epanchintsev, A., Lodygin, D., Rezaei, N., Jung, P., Verdoodt, B., Diebold, J., and Hermeking, H. (2007). c-MYC delays prometaphase by direct transactivation of MAD2 and BubR1: identification of mechanisms underlying c-MYC-induced DNA damage and chromosomal instability. *Cell Cycle*, 6(3):339–352.

Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrale, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., Altshuler, D., and Groop, L. C. (2003). PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34:267–273.

Rizzo, A., Pallone, F., Monteleone, G., and Fantini, M. C. (2011). Intestinal inflammation and colorectal cancer: a double-edged sword? *World Journal of Gastroenterology*, 17(26):3092–3100.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., Hurt, E. M., Zhao, H., Averett, L., Yang, L., Wilson, W. H., Jaffe, E. S., Simon, R., Klausner, R. D., Powell, J., Duffey, P. L., Longo, D. L., Greiner, T. C., Weisenburger, D. D., Sanger, W. G., Dave, B. J., Lynch, J. C., Vose, J., Armitage, J. O., Montserrat, E., López-Guillermo, A., Grogan, T. M., Miller, T. P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T., and Staudt, L. M. (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *The New England Journal of Medicine*, 346(25):1937–1947.

Sakai, N., Yoshidome, H., Shida, T., Kimura, F., Shimizu, H., Ohtsuka, M., Takeuchi, D., Sakakibara, M., and Miyazaki, M. (2012). CXCR4/CXCL12 expression profile is associated with tumor microenvironment and clinical outcome of liver metastases of colorectal cancer. *Clinical & Experimental Metastasis*, 29(2):101–110.

Shields, M. A., Dangi-Garimella, S., Redig, A. J., and Munshi, H. G. (2012). Biochemical role of the collagen-rich tumour microenvironment in pancreatic cancer progression. *Biochemical Journal*, 441(2):541–552.

Srivastava, S., Zhang, L., Jin, R., and Chan, C. (2008). A novel method incorporating gene ontology information for unsupervised clustering and feature selection. *PLOS ONE*, 3(12).

Stingo, F. C., Chen, Y. A., et al. (2011). Incorporating biological information into linear models: A Bayesian approach to the selection of pathways and genes. *Annals of Applied Statistics*, 5(3):1978–2002.

Stingo, F. C. and Vannucci, M. (2010). Variable selection for discriminant analysis with Markov random field priors for the analysis of microarray data. *Bioinformatics*, 27(4):495–501.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550.

Terol, M., López-Guillermo, A., Bosch, F., Villamor, N., Cid Xutgla, M., Campo, E., and Montserrat, E. (1999). Expression of beta-integrin adhesion molecules in non-Hodgkin's lymphoma: correlation with clinical and evolutive features. *Journal of Clinical Oncology*, 17(6):1869–1875.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(B):267–288.

Toiyama, Y., Fujikawa, H., Kawamura, M., Matsushita, K., Saigusa, S., Tanaka, K., Inoue, Y., Uchida, K., Mohri, Y., and Kusunoki, M. (2012). Evaluation of CXCL10 as a novel serum marker for predicting liver metastasis and prognosis in colorectal cancer. *International Journal of Oncology*, 40(2):560–566.

Toiyama, Y., Miki, C., Inoue, Y., Minobe, S., Urano, H., and Kusunoki, M. (2010). Loss of tissue expression of interleukin-10 promotes the disease progression of colorectal carcinoma. *Surgery Today*, 40(1):46–53.

Vermeulen, K., Van Bockstaele, D. R., and Berneman, Z. N. (2003). The cell cycle: a review of regulation, deregulation and therapeutic targets in cancer. *Cell Proliferation*, 36(3):131–149.

Wang, Q.-J., Papanikolaou, A., Sabourin, C., and Rosenberg, D. W. (1998). Altered expression of cyclin D1 and cyclin-dependent kinase 4 in azoxymethane-induced mouse colon tumorigenesis. *Carcinogenesis*, 19(11):2001–2006.

Wei, Z. and Li, H. (2007). A Markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537–1544.

Wei, Z. and Li, H. (2008). A hidden spatial-temporal Markov random field model for network-based analysis of time course gene expression data. *Annals of Applied Statistics*, 2(1):408–429.

Weinel, R. J., Rosendahl, A., Neumann, K., Chaloupka, B., Erb, D., Rothmund, M., and Santoso, S. (1992). Expression and function of VLA-α_2, -α_3, -α_5 and -$alpha_6$-integrin receptors in pancreatic carcinoma. *International Journal of Cancer*, 52(5):827–833.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal Of The Royal Statistical Society, Series B*, 67(2):301–320.

Zycinski, G., Barla, A., Squillario, M., Sanavia, T., Camillo, B. D., and Verri, A. (2013). Knowledge Driven Variable Selection (KDVS) – a new approach to enrichment analysis of gene signatures obtained from high-throughput data. *Source Code for Biology and Medicine*, 8(2).

5. SUMMARY

5.1 Discussions

The ultimate goal of RNA-seq data analysis is to interpret biological mechanisms based on the massive information stored in RNA-seq reads. Despite many successful applications of RNA-seq, how to extract solid and consistent information from millions of pieced-together reads, and meanwhile to reduce the noise level is still a critical challenge in RNA-seq data analysis. One of the most efficient strategies is to utilize the additional information to facilitate this process. In this dissertation, I have presented three studies in RNA-seq data analysis by incorporating disparate information. The first study takes the advantage of long read information in PacBio sequencing for assessing the performance of *de novo* assembly by RNA-seq short reads. The second study relies on the common information shared in sample replicates for accurately quantifying the expression levels of transcripts, while keeping the sample variations in estimation. The last study utilizes the pathway structural information summarized by domain experts for selecting phenotype-associated genes and pathways from high-throughput genomic data. All the above studies demonstrate that, by incorporating disparate information, the performance of RNA-seq data analysis can be better assessed or improved in the steps from assembly to quantification, and to functional analysis.

5.2 Perspectives

Aside from RNA sequencing, which quantifies and identifies the transcriptome of a sample, many new sequencing techniques have been developed for measuring different omics of a sample. For instance, exome sequencing only quantifies the protein-coding

genes in a genome rather than quantifies the whole transcriptome as RNA-seq does; ChIP-seq combines the chromatin immunoprecipitation with DNA sequencing for identifying targeted protein binding sites on DNA; Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq) identifies DNA regions which are accessible to transposase, namely the regions having no proteins binding with; DNA-methylation sequencing measures the epigenomic status of a sample.

One interesting direction is to combine multiple omics data to enhance the prediction performance, and to better illustrate the mechanism of gene regulation. Even though each of the above sequencing techniques has focused on measuring different aspects of a sample, they should provide consistent information for mutually supporting each other when the signal is true. For example, if we find high levels of methylations in the promoter regions of a certain genes, the transcriptional rates of these genes should be low in RNA-seq data.

From the perspective of computer science, machine learning and deep learning algorithms can be applied to the multi-omics data. By reorganizing the multi-omics data into meaningful and efficient data structures, and by appropriately choosing the variables, both supervised learning and unsupervised learning algorithms, which have been successfully used in many other areas, such as image processing, natural language processing, etc., can be applied to sequence data, for illustrating new biological mechanisms. However, as the data has boomed, the curse of dimensionality has become increasingly severe. How to properly reduce the dimensionality by capturing the correlative information remains as a critical question.

The other interesting direction is to investigate the "individual –population" relations of genomes or transcriptomes. Here, the "individual" can refer to a human or a single cell. The 1000 genomes project sequenced the genomes of 1092 people from different races/populations from all over the world. Single cell RNA-seq has the ability to sequence the small amount of RNAs from each single cell. By clustering the samples using each individual omics information, the history of human evolution and the mechanism of disease progression can be revealed.

Again, during only four decades, the field of DNA sequencing has become more prosperous than ever, by quickly absorbing knowledge from diverse areas, including but not limited to chemistry, materials science, computer science, and engineering. In the near future, the sequencing technology can be envisioned to enter the real life of each person and better serve the world.

VITA

Yifan Yang, was born in Tianjin, China. She received her B.S. degree from Biological Sciences department in China Agricultural University in 2008. She started her Ph.D. study in both the Biological Sciences department and Computer Science department at Purdue from 2011. Her research direction is computational biology focusing on quantitative modeling and model assessments in RNA sequencing.