PREDICTING SPLICING REGULATION WITH LEARNING METHODS

by
Guangyu Yang

A dissertation submitted to The Johns Hopkins University in conformity
with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
July, 2021

Abstract

Alternative splicing is an important post-transcriptional process that serves to increase the diversity of proteins in different tissues and developmental stages, and its dysregulation is often associated with diseases. Large-scale RNA-seq experiments and bioinformatic approaches already found evidence of splice site selections and interaction among cis-regulatory elements and trans-acting factors. However, in most cases, the mechanisms behind are still incompletely understood and remain to be determined. Therefore, there is a great need to accurately map and quantify gene splice variants, identify differences in splicing between conditions and computationally reveal the splicing regulation. In this dissertation, we investigate those challenges and propose novel computational methods to mitigate them. I will highlight my Ph.D. works on alternative splicing and present machine learning and statistical methods to extract gene and alternative splicing features from large collections of RNA-seq data, determining statistically significant differences in expression and splicing measurements between conditions, and predicting the splicing regulations of cis-regulatory sequence elements and trans-acting factors.

Thesis Committee

Dr. Liliana Florea (Primary Advisor)
Associate Professor
Department of Genetic Medicine
Department of Computer Science
Johns Hopkins School of Medicine

Dr. Benjamin Langmead
Associate Professor
Department of Computer Science
Johns Hopkins Whiting School of Engineering

Dr. Sarven H. Sabunciyan
Assistant Professor
Department of Pediatrics
Johns Hopkins School of Medicine

Dr. Leslie Cope
Associate Professor
Department of Oncology
Johns Hopkins School of Medicine

To the God who gives me the strength, courage and wisdom to do my researches and write down the words.

Acknowledgements

First and foremost I am extremely grateful to my advisor, Liliana Florea, for her invaluable advice, continuous support, and patience during my PhD study. Her immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

I would like to thank my wife Yujing Xie, this would have not been possible without her unwavering support and belief in me. I would like to express my sincere gratitude to Ben Langmead, Sarven Sabunciyan, and Leslie Cope, for their willingness to serve on my thesis committee and for their insightful comments and suggestions.

I would like to express my gratitude to the rest of my family and close friends. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

More broadly, I would like to thank all the professors, collaborators and fellow students I met over the years during my Ph.D. study. It is their kind help and support that have made my study and life in Johns Hopkins a wonderful time.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we focus on the pre-messenger RNA (pre-mRNA) splicing process, and design computational methods to predict and reveal the splicing regulatory code.

To start, I will introduce the biological background, namely what is pre-mRNA splicing, why alternative splicing is important, what is the machinery that regulates splicing events, and where the gene data come from.

1.1 Biological background

Three major kinds of molecules are important for most of the living organisms, namely deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and protein. DNA is a long double-stranded molecule that contains the genetic code for the development, functioning, growth and reproduction of all known organisms. A set of special sequences in the DNA, called genes, carry the genetic instructions needed for making the proteins. Protein, on the other hand, is an essential functional part of organisms. Proteins are made of amino acids, function as a cell's "building blocks", and participate in virtually every process within cells. In between, RNA acts as a carrier that translates genetic information that is encoded in the DNA into protein.

According to the central dogma, DNA contains the information needed to make all of the proteins (Figure 1-1). RNA is "transcribed" from DNA, acting as a messenger that carries the information encoded in DNA to the ribosomes. The ribosomes serve as factories in the cell where the information is "translated" from a code into the functional product - proteins. This process by which the DNA instructions are converted into the functional product is called gene expression.

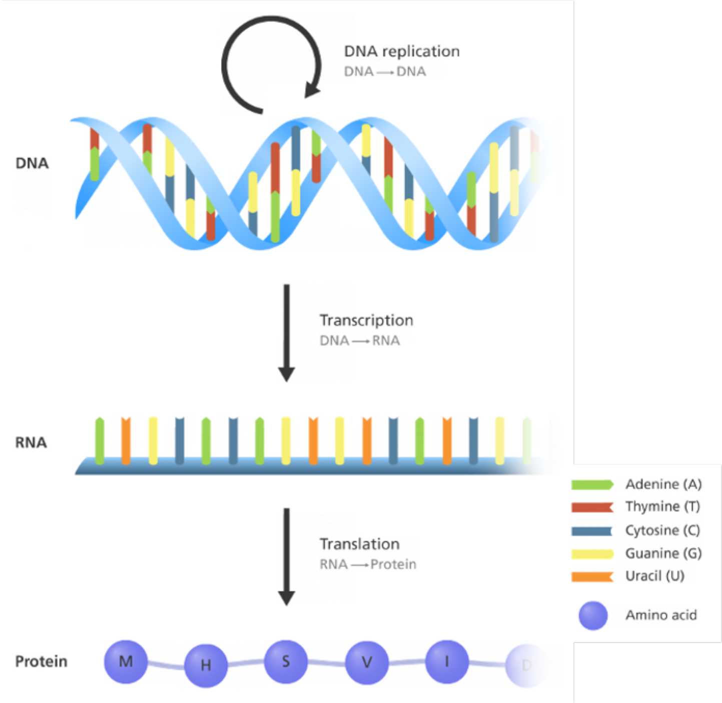


Figure 1-1. The central dogma.

Gene expression involves two key stages, transcription and translation. In transcription, the information in the DNA of every cell is converted into small, portable RNA messages. During translation, these messages travel from where the DNA is in the cell nucleus to the cytoplasm where ribosomes direct protein synthesis.

1.1.1 RNA transcription and RNA splicing

In humans and other eukaryotes, transcription of eukaryotic organisms contains two main stages. In the first stage, an RNA sequence is formed by reverse complementing the original DNA sequence. When an enzyme called RNA polymerase binds to a DNA strand of a gene in a region called the promoter, the RNA polymerase reads the DNA strand and builds the pre-mRNA molecule, using complementary base pairs. This process continues until the RNA polymerase crosses a stop (termination) sequence in the gene. This newly generated RNA is called pre-messenger RNA (pre-mRNA) and it is not quite ready to go. Pre-mRNA has to go through several processing steps to become a mature messenger RNA (mRNA) that can be translated into a protein.

These include the addition of a 5' cap and a 3' poly-A tail molecules to the two ends of the transcript to prevent degradation by the enzymes, and an excision process of removing non-coding sequences (called introns) and joining together the coding sequences (called exons) to form the mRNA.

The excision process is called the RNA splicing. The RNA splicing reactions are carried out by the spliceosome — a dynamic RNA–protein complex with a highly regulated functional cycle found in all eukaryotes [1]. There are two kinds of splicing events. Constitutive splicing events are recognized efficiently by the spliceosome and are spliced the same way in each pre-mRNA from a given gene. In contrast, in alternative splicing events, recognition and joining of a 5' and 3' splice site pair are in competition with at least one other 5' or 3' splice site.

1.1.2 Splicing, spliceosome, and RNA binding proteins

Splicing is the RNA regulatory process that we are mainly interested in, therefore it is necessary to describe this process in more detail. A splicing event can be described as the excision of the non-coding sequences (introns) from a pre-mRNA and joining of the coding sequences (exons). The special sequences at the intron/exon junctions are called the splice sites. Typically, the splice sites include a GU dinucleotide at the intron 5' end (5' splice site), a terminal AG at the 3' end (3' splice site) and a branch point (with an A) close to the 3' splice site [2] (Figure 1-2).

The splicing process involves a series of biochemical reactions that are catalyzed by the spliceosome, a large molecular complex composed of four small nuclear ribonucleoproteins (U1, U2, U4/U6 and U5 snRNPs) and approximately 50–100 non-snRNP splicing factors [3]. The spliceosome assembles onto each intron, with the binding of the U1 and U2 snRNP to the splice sites and the assistance of the splicing factors (e.g. SF1, U2AF65, U2AF35) to help locate the binding sites (Figure 1-3).

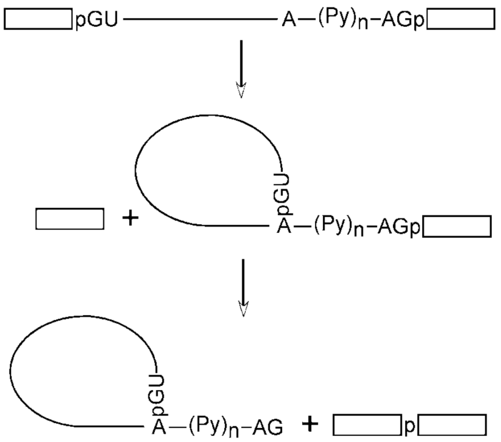


Figure 1-2. Illustration of pre-mRNA splicing (image from [4]).

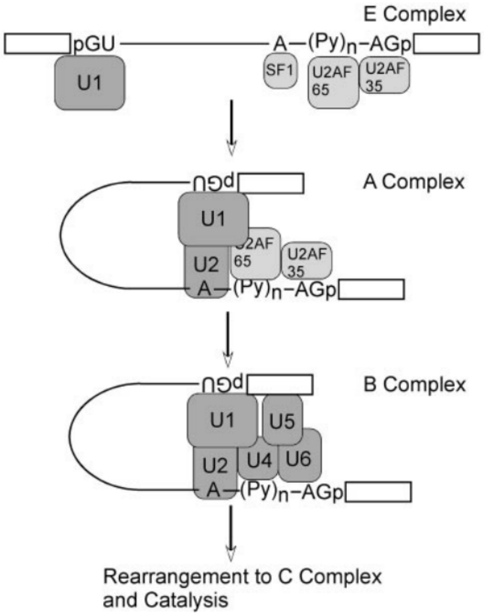


Figure 1-3. Illustration of spliceosome assembly (image from [4]).

1.1.3 Alternative splicing

Because of the alternative splicing events, the pre-messenger RNA is not always the same. A pre-messenger RNA segment may be spliced out or included in different ways (Figure 1-4), giving rise to several different possible mRNA products, potentially coding for different protein isoforms [4]. These protein isoforms differ in their peptide sequence and hence have different biochemical properties and biological functions.

Alternative splicing is a major contributor to both protein diversity and genetic regulation in higher eukaryotes. It is important in many cellular and developmental processes, including sex determination, apoptosis, axon guidance, cell excitation and contraction, and many others [5–7]. Estimates are that more than 90% of human genes undergo alternative splicing [8]. Moreover, many genes have multiple splicing patterns, and some have thousands [9, 10]. Splicing is tightly regulated in different tissues and developmental stages, and the mis-regulation of splicing is often involved in human diseases [11–13], with an estimate that as many as 50% of disease causing mutations affect splicing [14–16].

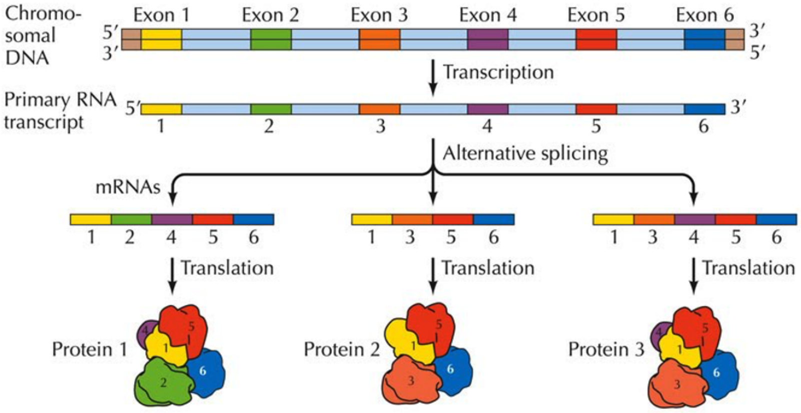


Figure 1-4. Illustration of alternative splicing.

1.1.4 Next generation sequencing

To understand splicing regulation and other gene processes, the sequencing reads from the next generation sequencing (NGS) platform are typically used. NGS is a technology for determining the sequence of DNA or RNA molecules to study genetic variation associated with diseases or other biological phenomena. With its ultra-high throughput, scalability, and accuracy, NGS has revolutionized genomics analyses and enabled new applications in genomic and clinical research, reproductive health, and many other areas.

Reads are pieces of sequencing fragments of DNA or RNA generated by NGS platforms. Bioinformatics analysis tools typically map the individual reads to the human reference genome to detect related genes (or features), alternatively spliced transcripts, allelic gene variants and single nucleotide polymorphisms.

Detection of splicing variations and mapping of the complex transcriptome have been greatly facilitated by the development of the NGS high-throughput RNA sequencing (RNA-seq) technology, which enables quantitative profiling of the transcriptome in a wide variety of cell types and conditions. Briefly, millions of RNA sequencing reads are generated from the gene features, thus providing an effective means to determine the set of genes and splice isoforms. For the purposes of evaluating differences in splicing patterns and expression levels of genes between conditions, RNA-seq read counts are summarized at the genomic level of interest, such as introns, exons or full isoforms. In particular, introns reflect both the structure and abundance of genes and their isoforms, and are the gene features to most accurately detect from sequencing data, and therefore can be used more reliably to detect alternative splicing variation. However, accurately quantifying intron abundance, determining the full extent of splicing variation, and comparing gene splicing profiles among biological states remains challenging.

1.2 Outline

My work entails developing statistical and machine learning models and tools for characterizing the cellular transcriptome and its variations along RNA processing pathways, in particular as it relates to primary RNA splicing, using large scale genomics data. In particular, we developed several methods under the umbrella of the JULiP project (JULiP, JULiP2 and MntJULiP) to determine and characterize alternative splicing variation of genes as represented by splice junctions (introns), and to compare them between experimental conditions in large-scale RNA-seq experiments. Second, we developed a deep learning model of sequences and alignments for the bioinformatics problem of predicting alternative splicing from the local genomic sequence context.

In this thesis, I will describe the primary projects that I worked on during my Ph.D. study; the remaining chapters are organized as follows.

1.2.1 JULiP and JULiP2

In Chapter 2, I will discuss the JULiP and JULiP2 methods for selecting, quantifying and comparing intron sets from large collections of RNA-seq data. JULiP and JULiP2 are based on the Generalized Linear Model (GLM), and leverage the latent gene information across all of the analysis samples. JULiP predicts an accurate set of introns from a large RNA-seq sample databases and JULiP2 uses the predicted introns to determine differential splicing of genes.

1.2.2 MntJULiP and Jutils

In Chapter 3, I will present MntJULiP and Jutils. MntJULiP is a differential splicing analysis tool that implements novel Dirichlet-multinomial and zero-inflated negative binomial models within a Bayesian framework to detect both changes in splicing ratios and in absolute splicing levels of introns with high accuracy, and can find classes of splicing variation overlooked by reference tools. Additionally, a mixture model allows multiple conditions to be compared simultaneously. MntJULiP is highly scalable, and can process hundreds of GTEx samples in <1 hour to reveal splicing constituents of phenotypic differentiation.

To visualize the results of MntJULiP and other popular differential splicing programs, we developed the toolkit Jutils. Jutils extracts the program predictions into a unified format file for visualization. Jutils can visualize alternative splicing events in several ways, including heatmaps, Venn diagrams and sashimi plots.

1.2.3 A Deep Learning (DL) splicing model

In Chapter 4, I will introduce a deep learning (DL) based model of alternative splicing, trained to predict splicing ratios of given splice junctions (introns). The previously described MntJULiP program receives as input the RNA-seq reads directly, from which it extracts the read counts and predicts the splicing ratio. In contrast, the

DL model receives as input the motif information of known RNA binding proteins (RBPs) as position weight matrices (PWMs) along with the RNA sequences around the alternative splice junctions. The network then learns the latent information on *trans*-factors (RBPs), *cis*-elements (motifs), and how they work together to regulate the splicing events, which it uses to predict the intron splicing ratios.

Chapter 2

JULiP and JULiP2

A typical RNA-seq data analysis aims to analyze the transcriptome in a given RNA-seq sample, condition, or experiment. In more specific terms, it aims to determine the expressed genes and transcripts, their expression levels and, in a multi-condition experiment, differences in expression levels and splicing patterns that could lead to reproducible markers. The RNA-seq data analysis component starts with the millions of short reads generated from a given RNA sample. Reads are mapped to a reference genome and to the genomic feature of interest (e.g., gene, transcript, exon or intron), and the normalized and/or estimated read counts are used to measure the abundance of the feature in the analyzed samples. A critical problem in the analysis work is determining a reliable set of features that can be used as the basis for measuring expression and splicing levels. Accurate detection of such features is hampered by sequencing and alignment artifacts. Further, once a reference 'database' of features has been established, determining biologically significant differences in expression and splicing levels of features, using rigorous statistical methods that take into account information from multiple biological replicates, is critical for identifying a set of informative and reproducible markers.

Our *first* major research interest in this area, therefore, involves developing machine learning and statistical methods to extract gene and alternative splicing features (herein, introns) from large collections of RNA-seq data and refine them into high-confidence reference sets. Our *second* major interest is in determining statistically significant differences in expression and splicing measurements between conditions.

In the following section, we present background information and previous work. We then describe our tools JULiP, for accurate intron selection from multiple RNA-seq samples, and JULiP2, which extends JULiP's statistical model and incorporates differential splicing prediction.

2.1 Background

2.1.1 Accurate prediction of alternatively spliced features

By far the most efficient means for detecting splice variation large scale is by computational prediction. When a reference genome is available, transcript assembly programs (*e.g.*, Cufflinks [17], CLASS2 [18], StringTie [19], FlipFlop [20] and others reviewed in [21]) can be used to assemble RNA-seq reads aligned to the genome into gene and transcript models. These methods create a graph representation of a gene and its possible alternative splicing combinations, from which a subset of transcripts is selected using linear or quadratic programs, dynamic programming, and network flow optimization [21]. Local splice variation can then be extracted from these annotations. The assembly process, however, is difficult and fraught with errors [18, 22]. As an alternative, introns represent the building blocks of full-length isoforms as well as of local alternative splicing events, such as exon skipping, mutually exclusive exons, and alternative exon and gene ends, and have been used to detect and characterize alternative splicing variation in practice [23, 24]. Therefore, accurate and comprehensive identification of the set of expressed introns in a given set of RNA-seq samples is critical for all gene and splice variation analyses downstream.

Current assembly-based methods depend critically on the quality of the reference genome, precision of the read mapping software, and depth of coverage with RNA-seq reads. False positives can result from spurious RNA fragments during library preparation, incorrect alignment, and intronic and intergenic 'noise' from unspliced RNA. False negatives can arise from low expression genes, which typically have only a handful of reads and are partially reconstructed. Therefore, analyzing each sample individually limits both the accuracy and the potential to identify splice variants, in particular rare or low expression events.

Batch sequencing of large numbers of RNA-seq samples from multiple replicates, tissues or populations is becoming increasingly common [25–27]. Current approaches that process one data set at a time are not capable of seamlessly and efficiently analyzing such massive collections. Designing tools that can simultaneously analyze multiple samples, however, is challenging due to the large number of artifacts compounded over the full collection of data sets and also to the sheer volume of data. So far only a handful of algorithms have been proposed to assemble reads across multiple sample: CLIIQ [28], an early prototype algorithm that uses an integer linear programming (ILP) approach with variables the full set of isoforms; MiTie [29], which builds a splicing graph representing the gene and maximizes a likelihood function using mixed integer programming with a regularization penalty; and ISP [30], which solves an LP or ILP problem iteratively on a weighted connectivity graph derived from the input samples. While marking significant conceptual advances, they scale poorly (MiTie) or otherwise have limited performance in detecting splicing variation (ISP). Therefore, a highly efficient and accurate feature selection algorithm is needed. Our JULiP algorithm selects a highly accurate subset of introns from a large collection of RNA-seq data directly from alignments, using latent gene information across the samples incorporated into generalized linear models.

2.1.2 Detection of differentially spliced features

Gene alternative splicing plays an important role in development, tissue specialization and disease, and differences in splicing patterns can reveal important factors for phenotypic differentiation. Aberrant alternative splicing has been associated with a wide spectrum of diseases, including cancers [31]. The importance of alternative splicing emphasizes the need to accurately map and quantify splice variations, and to detect differences in splicing patterns between cellular conditions.

There are three classes of methods for differential splicing detection, based on the splicing feature of interest (Figure 2-1). The *first* class is that of isoform based methods, such as the assembly-based Cufflinks/Cuffdiff pipeline [17, 32]. The Cufflinks/Cuffdiff pipeline constructs a set of isoforms, quantifies the expression that best explains the observed reads, and determines differential splicing in two ways. First, Cuffdiff measures differences in isoform expression between conditions, to determine instances of isoform-level regulation. Second, it determines differences in the relative usage of isoforms within a gene, using the Jensen–Shannon divergence to measure and compare the similarity between two probability distributions.

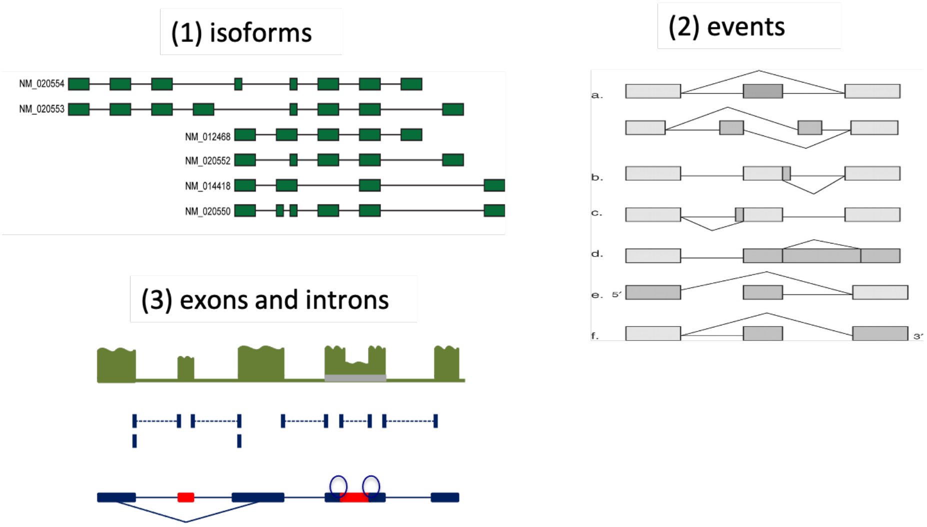


Figure 2-1. Methods may detect differential splicing at the isoform, alternative splicing event, and exon/intron level.

The *second* class of methods focuses on specific types of alternative splicing events which have been categorized into several common types, including exon skipping, mutually exclusive exons, intron retention and alternative 3'/5' splice sites. Alternative splicing events can be detected from RNA-seq reads mapped to exons or exon junctions, either starting from a reference set of gene annotations or by building a gene schematic representation such as a splice graph, annotated with the number of observed reads that unambiguously support the presence or absence of each splicing event. Comparing the read counts between mutually exclusive paths (isoforms) gives an estimate of the relative contribution of each isoform, which can then be compared between conditions.

As an example, the software rMATS [33] retrieves candidate alternative splicing events from an input annotation, extending the set with introns from the input alignments. It then uses the counts of reads mapped to the two mutually exclusive isoforms, for instance exon inclusion or exon skipping in the case of an exon skipping event, in a bayesian framework to estimate a value called the percent splicing inclusion levels (ψ), and determine statistically significant differential events.

Another tool, MAJIQ [34], quantifies RNA splicing in units of local splicing variations (LSV). LSVs are defined in the splice graph where several edges share either the start or the end endpoint of a same exon. MAJIQ models LSVs as structural network motifs and estimates an a posterior marginal distribution over the fractions on LSVs, defined by the percent selected index (PSI), and further determines changes in PSI between two conditions (dPSI).

Further, DiffSplice [35] defines so called Alternative Spliced Modules (ASM) in splice graphs. An ASM is a region in a splice graph where isoforms differ from each other, hence each ASM has at least two alternative paths. ASM seeks to minimize the ambiguity in isoform resolution by only considering regions that are not shared by all isoforms. DiffSplice tests for differential splicing of each ASM instead of whole transcripts. The relative abundances of alternative paths are estimated using the maximum likelihood method. As with Cuffdiff, the difference of the relative abundance composition is measured using the Jensen-Shannon divergence metric (JSD) at the level of ASMs.

The *third* type of methods do not directly quantify the expression levels of transcripts or AS events, rather they use differential exon/intron usage as a surrogate to infer differential isoform usage. These methods divide a gene into typically disjoint counting bins, and the number of observed reads overlapping each bin is counted. Bins can be full or truncated exonic regions, junction regions, or both. To infer differential exon/intron (bin) usage between conditions, these methods often make use of (generalized) linear models with the assumption of Poisson or negative binomial distributions on read mappings. The models contain an interaction term between the bin identifiers and the condition of interest to search for non-proportionality of the bin counts within a gene between the conditions.

One such method, DEXSeq [36], collects reads by bins of exons or subexons, and uses a generalized linear model to detect the differential usage of counting units across conditions. Similar to DEXSeq, JunctionSeq [37] constructs bins for read counts on individual exons and/or splice junctions from the counts of the whole gene. In contrast, LeafCutter [38] clusters introns that share a donor or an acceptor splice site. Introns are represented by splice junction read counts and jointly modeled by a Dirichlet-multinomial generalized linear model.

In the following sections, we present our tools JULiP and JULiP2, two new statistical models to select introns and to determine differential alternative splicing events at the intron-level from two-group RNA-seq data with replicates. As a brief introduction, JULiP and JULiP2 work in multiple steps (Figure 2-2)

- receive as input RNA-seq read alignments (the BAM file) and construct the set of candidate introns and their splice junction read counts;

- use the gene regions extracted from a reference genome annotation to group the introns and their read counts;

- for each gene region intron group, estimate intron abundance using the regularized program and select an optimal set of introns;

- in addition, for the differential analysis task in JULiP2, estimate the differential usage of introns by a generalized linear model (GLM) coupled with a Wald test.

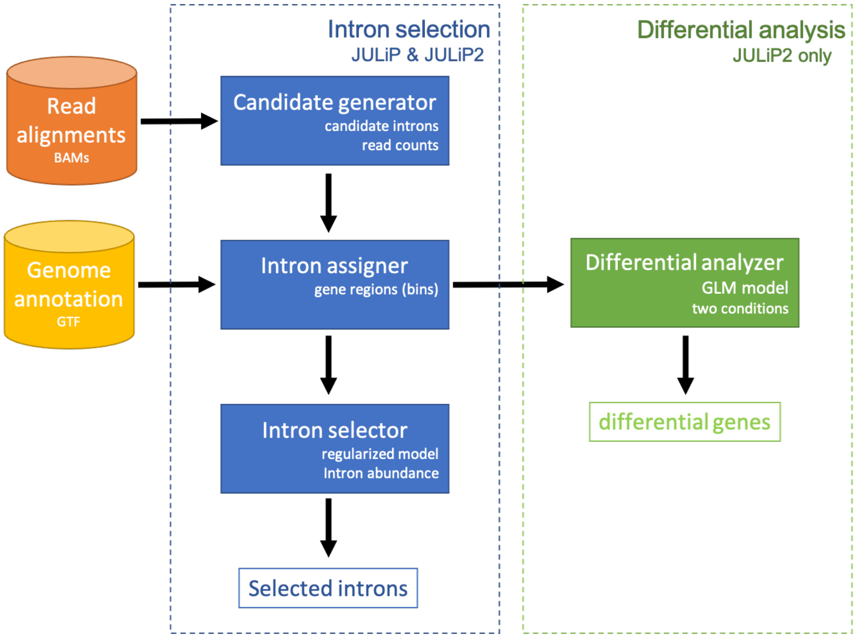


Figure 2-2. The architecture of JULiP and JULiP2

2.2 JULiP: An efficient model for accurate intron selection from multiple RNA-seq samples

Determining a high confidence and complete set of features is critical for the accuracy of downstream differential and functional analyses. Our tool JULiP (JUnction prediction using an L_1-regularized Program) implements a L_1-regularized model to identify and select a highly accurate set of introns from large scale RNA-seq data sets [39]. Unlike traditional approaches that extract introns from each sample and then merge them per sample sets, our model selects introns directly by criss-crossing information across all input samples. Specifically, for each gene region, JULiP solves an L_1-regularized program iteratively on the aggregate set of introns extracted from the multi-sample RNA-seq data set. When evaluated on simulated and real data, JULiP detected introns with both very high precision (>98%) and high sensitivity (>89%). In particular, it detected at least 30% more introns in each sample compared to traditional assembly-based approaches, and 10% more than the cumulative intron set of all samples, at higher or comparable precision [39].

We next introduce the mathematical model and provide details of the algorithm and implementation. We then compare JULiP with existing approaches on both simulated and real RNA-seq data sets, and discuss scalability and practical implications.

2.2.1 Methods and implementation

2.2.1.1 The core optimization formulation

We assume that reads from RNA-seq samples are generated independently and follow a Poisson distribution. Given a gene region, we denote V the candidate set of introns, and S the sample set. We define a set of observations $X = \{x_v^s \mid v \in V, s \in S\}$, where x_v^s is a random variable representing the number of reads aligned to intron v in sample s. Each variable x_v^s follows a Poisson distribution with mean λ_v^s, which is the expected count of intron v in sample s. More specifically:

$$\lambda_v^s = N^s \beta_v, \tag{2.1}$$

where N^s is the total number of mapped reads in sample s and β_v is a coefficient that describes the abundance ratio of intron v in transcripts of the gene, which needs to be estimated. We assume samples are derived from the same cell type or condition, and therefore intron utilization is similar across all samples. Thereby, the likelihood function is:

$$\begin{aligned} P(X = x_v^s, v \in V, s \in S \mid \beta) &= \prod_{s \in S} \prod_{v \in V} P(X = x_v^s \mid \beta) \\ &= \prod_{s \in S} \prod_{v \in V} \frac{e^{-\lambda_v^s} (\lambda_v^s)^{x_v^s}}{x_v^s!} \end{aligned} \tag{2.2}$$

Taking the logarithm of the above equation, we have:

$$F(\beta) = -\sum_{s \in S} \sum_{v \in V} (\lambda_v^s - x_v^s ln \lambda_v^s + ln(x_v^s!)) \tag{2.3}$$

Since x_v^s does not depend on β, maximizing $F(\beta)$ with respect to β is equivalent to minimizing:

$$L(\beta) = \sum_{s \in S} \sum_{v \in V} (\lambda_v^s - x_v^s ln \lambda_v^s) \tag{2.4}$$

The total number of candidate introns collected from all samples is large, but only a limited number of introns are expected to be real. Therefore, the solution must be sparse, and L_1 (Lasso) regularization can be used to encode sparseness. Together with $L(\beta)$, we propose:

$$J(\beta; t) = \sum_{s \in S} \sum_{v \in V} (\lambda_v^s - x_v^s ln \lambda_v^s) + t \|\beta\|_1 \tag{2.5}$$

where $t > 0$ is the regularization parameter, and $\beta = [\beta_1, \beta_2, ..., \beta_{|V|}]$.

Since the read count for each intron cannot be negative, $\beta_v \geq 0$ for $v \in V$. Hence,

$$t \|\beta\|_1 = t \sum_{v \in V} \beta_v \tag{2.6}$$

To summarize, the optimization problem can be expressed as:

$$\begin{aligned} &arg\,min_{\beta}\ J(\beta;t) \\ &s.t.\ t > 0, \\ &\beta_v \geq 0, \\ &\lambda_v^s = N^s \beta_v \end{aligned} \tag{2.7}$$

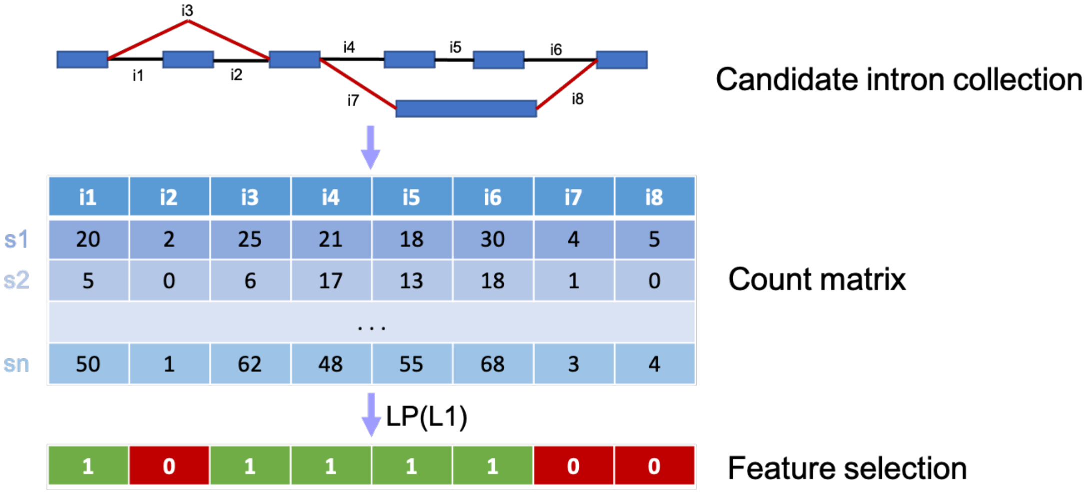
2.2.1.2 Implementation

JULiP works in three steps (Figure 2-3):

(1) construct the set of candidate introns and their read counts;

(2) assign reads from multiple samples into gene regions (as bins); and

(3) estimate intron abundance using the regularized program and select an optimal set of introns.



	i1	i2	i3	i4	i5	i6	i7	i8
s1	20	2	25	21	18	30	4	5
s2	5	0	6	17	13	18	1	0
	. . .							
sn	50	1	62	48	55	68	3	4

1	0	1	1	1	1	0	0

Figure 2-3. Overview of JULiP implementation.

In *step 1*, JULiP aligns all reads to the genome, separately for each sample, using the spliced alignment program TopHat2 [40]. A set of candidate introns and their read counts in each sample are then inferred from the spliced alignments. To reduce ambiguity due to reads mapping to multiple locations, we use the counts of uniquely mapped reads.

In *step 2*, overlapping read alignments in each sample are merged to form contiguous exonic regions, which are then connected via the introns from step 1 into larger gene regions. Regions from individual samples are further merged across all samples and used as 'bins' for clustering introns. This process may create long regions possibly containing several genes. Large 'bins' harboring hundreds of introns can significantly affect performance as well as the accuracy of the program, for instance when genes with varying expression levels are being clustered together. Therefore, before intron selection JULiP splits a region if the sequence of intron counts changes abruptly over a fixed window.

Lastly, *step 3* selects a subset of candidate introns believed to be expressed in the samples and estimates their abundance levels, using the L_1-regularized program from the previous section. The algorithm iteratively estimates the abundance ratio β of introns based on their read counts. The process updates β and reduces the regularization parameter t simultaneously, and is iterated until convergence.

To further speed up the program for application to very large RNA-seq collections, we also implemented our model using the Hadoop distributed framework. With this implementation, JULiP can solve individual programming problems from hundreds and potentially thousands of samples simultaneously across tens or hundreds of computers.

2.2.2 Evaluation

An accurate set of introns is critical for building complete transcript models and for identifying alternative splicing variation. We evaluated JULiP and several transcript assembly methods including Cufflinks (v2.2.1) [32], CLASS2 (v2.1.2) [18], StringTie (v1.2.2) [19] and FlipFlop (v1.9.6) [20], as well as the multi-sample assembler ISP (v0.3) [30] on both simulated and real RNA-seq data. (MiTie [29] was prohibitively slow, requiring more than a week to process a single gene region, and was unable to handle long gene regions, and therefore was omitted.)

For the single-sample programs, we assemble transcripts for individual samples, then extract and aggregate introns from all samples. We ran each program with the default settings and, where applicable, in 'sensitive' mode (denoted '_F001'), where we adjusted the minimum isoform fraction (parameter '-F 0.01') to report more splice isoforms. For ISP, we extract introns directly from the isoforms predicted from the multiple samples.

In the following sections we assess the accuracy of intron selection as a general indicator for the programs' ability to reconstruct as complete as possible a set of transcripts and to detect splicing variation.

2.2.2.1 Performance on simulated data

We simulated 25 RNA-seq samples, each with roughly 85.9 million 100 bp paired-end reads, starting from the expression profile of GENCODE v.22 [41] genes and transcripts in five hippocampus samples (data not shown) and using the program Polyester [42]. Polyester incorporates typical biases from library preparation and sequencing, generating reads from the reference genome and randomly introducing errors. Reads were mapped to the entire human genome (GRCh38) using TopHat2 (v2.1.0). For illustration purposes, we restricted our analysis to reads mapping to chromosome 12. For evaluation, we consider GENCODE junctions contained in the reads for each sample as the gold reference, and declare a match between a predicted feature and a reference intron if their genomic coordinates match exactly. Hence, a predicted intron is a true positive (TP) if it exactly matches an intron in the reference, a false positive (FP) if it has no counterpart in the reference, whereas a reference intron is deemed a false negative (FN) if it was not reported by the program. We use the standard sensitivity $Sn = TP/(TP + FN)$ and precision $Pr = TP/(TP + FP)$ measures as well as the combined $F - value = 2 * Sn * Pr/(Sn + Pr)$ [43] to measure accuracy.

We first assessed the potential of methods to uncover introns from single samples when using local information only versus information from multiple samples. When each sample is considered individually, programs find between 7,795-10,113 (per sample average) of the introns in a sample, for 78%-95% sensitivity, while precision is very high for all programs, at >96% (Figure 2-4 and Table 2-I, column 2). JULiP's sensitivity is 16-30% higher than those of the single-sample assemblers, and a remarkable 13% higher than the sensitivity of the multi-assembler ISP.

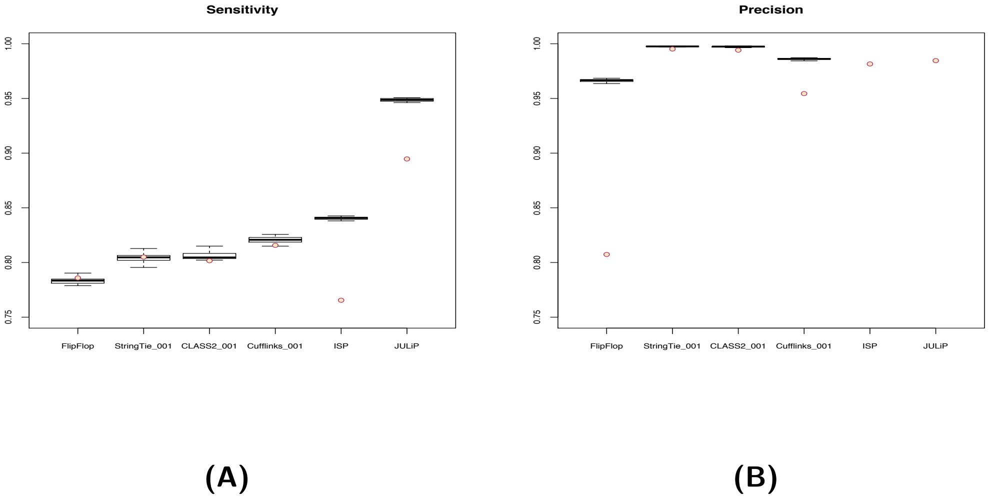


Figure 2-4. Sensitivity (A) and precision (B) of programs for each sample. Per sample precision cannot be calculated for ISP and JULiP, as they have access to additional information and predict true introns in the simulated model that may not have been sampled in an individual data set. Boxplots indicate the minimum, maximum and median values, and the .25 and .75 quantiles for each program over the 25 samples. Circles indicate the values for the pooled set of samples.

Set	TP (sample avg.)	TP	FP	Sn	Pr	F-val
Gold reference	10,659	12,996	NA	NA	NA	NA
Cufflinks	7,795	9,832	248	0.757	0.975	0.852
CLASS2	8,193	10,194	51	0.784	0.995	0.877
StringTie	8,245	10,093	29	0.777	0.997	0.873
Cufflinks_F001	8,749	10,601	506	0.816	0.954	0.880
CLASS2_F001	8,593	10,418	61	0.802	0.994	0.888
StringTie_F0.01	8,575	10,462	50	0.805	0.995	0.890
FlipFlop	8,351	10,214	2,437	0.786	0.807	0.797
ISP	8,961	9,949	187	0.766	0.982	0.860
JULiP	10,113	11,628	182	0.895	0.985	0.938

Table 2-I. Performance of methods on the simulated data. TP = true positives, per sample average (column 2) and pooled across all samples (column 3); FP = false positives; Sn = TP/(TP + FN), Pr = TP/(TP + FP) and F-val = 2 * Sn * Pr/(Sn + Pr).

Even when junctions are pooled across all samples, JULiP significantly outperforms all other methods (Table 2-I, columns 3-7). It detects 89.5% of the introns encoded in the data, compared to 81.6% or lower for the rest of the methods, at higher or comparable precision (98.5%). Therefore, JULiP takes advantage of the latent information in multiple samples to improve the sensitivity and precision of predictions simultaneously and in a significant way.

Last but not least, JULiP found a total of 11,628 of the 12,996 introns generated by the simulator, which is >30% more introns than found on average by any of the other programs in a single sample. This difference represents introns missed by the conventional assemblers but also new introns, *i.e.* which were included in the simulation set but are not present in that sample, thus illustrating the power of a multi-sample approach.

	Genes detected only by JULiP	Genes detected only by the counterpart
JULiP vs CLASS2_F001	136	2
JULiP vs StringTie_F001	193	4
JULiP vs Cufflinks_F001	218	1
JULiP vs FlipFlop	252	4
JULiP vs ISP	129	2

Table 2-II. Performance in gene detection.

To more specifically assess the contribution to gene and transcript reconstruction, we mapped the introns found from the pooled data back to the reference annotations to determine the set of genes they represent. In pairwise comparisons, JULiP found between 129-252 genes that were not discovered by the other method, and only missed 1-4 genes in each comparison (Table 2-II). Therefore, JULiP was significantly more sensitive and more robust in detecting the genes expressed in the set of samples, utilizing the hidden information from multiple samples to identify genes with weak signal that could not be detected by other methods.

2.2.2.2 Performance on real data

To observe the behavior of programs in a more realistic setting, we applied all methods to 50 randomly chosen lymphoblastoid RNA-seq samples sequenced as part of the GEUVADIS population variation project [25]. Samples contained between 23-57 million 75 bp paired-end Illumina reads. As before, we mapped all reads to the reference genome with TopHat2, and extracted alignments on chromosome 12 for analysis.

Unlike with simulated data, the true set of introns in the samples is not known. As a notable consequence, it is impossible to distinguish between novel junctions not yet recorded in the annotations and artifacts, or false positives. Nevertheless, we compile a comprehensive reference set of introns from transcript annotations in the GENCODE v.22, RefSeq and KnownGenes repositories, the latter two obtained from the Genome

Browser at the University of California Santa Cruz (http://genome.ucsc.edu).

Set	TP	FP	Sn	Pr	F-val	Sn'	Pr'	F'-val
Reference	20,171	NA	NA	NA	NA	NA	NA	NA
Cufflinks	9,507	2,424	0.471	0.797	0.592	0.575	0.971	0.722
CLASS2	9,917	3,351	0.492	0.747	0.593	0.633	0.962	0.763
StringTie	9,206	1,948	0.456	0.825	0.588	0.550	0.994	0.708
Cufflinks_F001	10,883	8,739	0.540	0.555	0.547	0.842	0.866	0.854
CLASS2_F001	10,521	5,286	0.522	0.666	0.585	0.737	0.940	0.826
StringTie_F001	9,688	3,089	0.480	0.758	0.588	0.629	0.992	0.770
FlipFlop	9,960	13,656	0.494	0.422	0.455	0.684	0.584	0.630
ISP	9,030	1,723	0.448	0.840	0.584	0.507	0.952	0.662
JULiP	11,846	5,049	0.587	0.701	0.639	0.825	0.985	0.898

Table 2-III. Performance of programs on the GEUVADIS data set. Except for ISP and JULiP, introns were pooled across all samples. Sn (Pr) = sensitivity (precision) on the combined GENCODE, KnownGenes and RefSeq reference database; Sn' (Pr') = potential sensitivity (precision) with additional EST, mRNA and multi-sample support.

Programs report between 9,000–12,000 junctions across all samples, with sensitivity values ranging between 45.6% and 58.7%, where JULiP is the most sensitive of the programs while ISP and StringTie are the least sensitive (Table 2-III). Precision varies more widely across methods, with StringTie and ISP seemingly being the most precise. Even so, JULiP has the best overall accuracy as measured by the F-value, 4% more than the runner ups, CLASS2 and Cufflinks. We hypothesize, however, that most of the additional introns predicted by JULiP are in fact true but unannotated splice junctions.

Set	FP	ESTs, mRNAs	≥2 samples	≥5 samples	Explained
Cufflinks	2,424	295	1,493	941	0.738
CLASS2	3,351	344	2,160	1,258	0.747
StringTie	1,948	278	1,326	916	0.823
Cufflinks_F001	8,739	665	4,770	2,360	0.622
CLASS2_F001	5,286	487	3,361	1,876	0.728
StringTie_F001	3,089	412	2,168	1,440	0.835
FlipFlop	13,656	495	2,845	1,673	0.245
ISP	1,723	146	913	516	0.615
JULiP	5,049	546	3,698	2,003	0.841

Table 2-IV. False positive introns explained by other data sources.

To test this assumption, we searched the predicted but unexplained introns for all programs against the collection of introns extracted from spliced alignments of ESTs and full-length mRNA sequences, obtained from the UCSC Table Browser. Also, introns that occur in two or more of the samples are more likely to represent true but rare or cell type specific introns, not yet included in the databases. These two categories accounted for more than 84% of the unannotated introns predicted by JULiP, and smaller fractions for the other programs (Table 2-IV). When these additional introns are considered, the fraction of predicted introns for each program (Table 2-III, Sn') grows to 50.7–82.5%, and similarly for precision (Pr'), 58.4–99.2%. JULiP has slightly lower sensitivity than Cufflinks_F001, which is the most sensitive, however its precision is higher by a significant 12%, namely 98.5% compared to 86.6%. Overall, JULiP once again has the best F-value, 89.8%, and has the best tradeoff between sensitivity and precision, and therefore is the method best suited to extract splice information from the large collection of RNA-seq data.

2.2.2.3 Scalability with large collections of data

To test JULiP's scalability with large collections of RNA-seq data, we simulated 100 samples using the protocol earlier. We tested the parallel version of JULiP on the Johns Hopkins University MARCC computing cluster, using 20, 40, 60, 80 and 100 simulated samples, with varying levels of resources. Tests were performed on 4 Linux cluster nodes with 24 cores each, to capture the potential of JULiP under a resource-rich scenario, and on a single 24-core node, to assess performance under a typical bioinformatics computing environment. For comparison, we also ran the non-parallel, single-threaded version of JULiP on all data sets. All nodes had 2.5 GHz CPUs and 128 GB of memory.

No. samples	20	40	60	80	100
4 x 24 CPUs	1m1.5s	1m8.3s	1m22.0s	1m41.0s	1m54.3s
	(0m38.1s)	(0m41.5s)	(0m51.8s)	(1m7.3s)	(1m17.2s)
1 x 24 CPUs	1m14.0s	2m15.1s	3m11.9s	4m12.5s	5m15.9s
	(0m50.3s)	(1m37.8s)	(2m21.9s)	(3m7.9s)	(3m50.4s)
1 x 1 CPU	(1 thread)	6m14.7s	12m20.1s	18m21.8s	24m11.9s 29m15.8s
	(1m39.2s)	(3m19.3s)	(4m32.7s)	(5m58.9s)	(7m38.9s)
Max memory	597 Mb	605 Mb	608 Mb	607 Mb	616 Mb

Table 2-V. Run time and memory requirements for JULiP with 20, 40, ... , 100 simulated data sets with varying resource levels (Numbers in parentheses indicate run times for the LP solver only).

JULiP took less than 2 minutes to complete each run in the resource-rich environment, and less than 6 minutes on the 24 CPU single server (Table 2-V). Most importantly, JULiP took less than 30 minutes to complete the largest run sequentially, of which only 7m39s solving the LP problems and the rest preparing the read count and region information. Therefore, JULiP can be used on a variety of computing platforms and with varying amounts of resources, from a single desktop to large-scale computing clusters. In contrast, Cufflinks, CLASS2 and StringTie took roughly 20, 12 and 10 minutes *per sample* on average, all with 4 threads, FlipFlop required more than 7 hours *per sample* single-threaded, and ISP took 1h 10min with 48 threads for the 25 sample data set. While these numbers are not directly comparable, they showcase the capabilities and potential of JULiP to perform calculations on very large collections of data.

2.3 JULiP2: A robust model for intron selection and detection of differential alternative splicing from multiple RNA-seq samples

Once an accurate set of features is generated, the next step in an RNA-seq analysis involves comparing and identifying differences in usage between two conditions. To address this problem, we developed the JULiP2 method that incorporates differential splicing detection. Additionally, the Poisson read count model in JULiP may not be suitable to capture the over-dispersion in some biological data; therefore, in JULiP2 we formulated a new intron selection method, based on the negative binomial distribution to model read counts.

JULiP2 detects introns from the read alignments and uses the read counts of splice junctions to estimate the relative expression of introns under the experimental conditions. JULiP2 adopts a generalized linear model (GLM) to model read counts and test for differential usage of individual introns for each gene. More specifically, to detect differential isoform usage represented as differential intron abundance in different conditions, JULiP2 tests for differentials in read counts supporting splice junctions (introns). Testing for differential usage of introns, compared to whole isoforms or even local alternative splicing events, has a number of benefits. Introns as splicing events are discrete and identifiable, capture a variety of splicing patterns, and make it easy to incorporate novel splicing variants. This allows us to indirectly query for differential regulation in unknown isoforms, improving performance on unknown genes or sparsely annotated genomes.

Following the general approach of its predecessor JULiP, JULiP2 extracts the introns in a gene region from read alignments, calculates the read counts for these introns, and fits the values to a generalized linear model (GLM). In the model, the parameters mean μ and variance σ are estimated, and then tested by a likelihood ratio test to detect differences between conditions, using the p-value of the test to report statistically significant results.

JULiP2 has the following key features that distinguish it from other similar RNA-seq analysis methods:

- JULiP2 can be run without a reference annotation, accurately detecting novel splice junctions as well as infering gene regions, which benefits analyses where the available transcript annotation is flawed or incomplete.
- JULiP2 can detect differentials in novel introns without the need for an additional isoform assembly step or the assistance of a reference annotation.
- JULiP2 is efficient, lightweight and faster than existing approaches while providing estimates of equal accuracy and substantially reducing parametric complexity.

2.3.1 Methods and implementation

2.3.1.1 Core statistical methodology

We assume the RNA-seq reads are generated independently for each samples. Given a gene region, we define a set of observation $X = \{x_v^s | v \in V, s \in S\}$, where V is a set of candidate introns, S is the sample set, and x_v^s is a random variable representing the splice junction reads counts of intron v in sample s. Unlike the Poisson regression model in JULiP with the assumption that variable x_v^s has equal mean and variance. Here, we consider an over-dispersion model that variance can differ from the mean, because of our observations of the real RNA-seq data, which almost reject the restriction that the variance equals to the mean.

In detail, we assume x_v^s follows a negative binomial distribution (NB) with mean μ_v^s and variance σ_v^s,

$$x_v^s \sim NB(\mu_v^s, \sigma_v^s)$$

It is natural to model the variance σ as a function of the mean, $\sigma = \mu + \alpha\mu^2$, where α is the dispersion parameter.

The joint distribution of $P(X|\mu, \alpha)$ can be represent as,

$$\begin{aligned} P(X|\mu, \alpha) &= \prod_{s \in S} \prod_{v \in V} P(X = x_v^s | \mu_v^s, \alpha_v) \\ &= \prod_{s \in S} \prod_{v \in V} \frac{\Gamma(x_v^s + \alpha_v^{-1})}{\Gamma(x_v^s + 1)\Gamma(\alpha_v^{-1})} (\frac{\alpha_v^{-1}}{\alpha_v^{-1} + \mu_v^s})^{\alpha_v^{-1}} (\frac{\mu_v^s}{\alpha_v^{-1} + \mu_v^s})^{x_v^s} \end{aligned}$$

Where $\Gamma()$ is the gamma function. Taking the logarithm with respected to all samples and splice junctions in given gene region,

$$\begin{aligned} F(\mu, \alpha) &= log(P(X|\mu, \alpha)) \\ &= \sum_{s \in S} \sum_{v \in V} log(P(X = x_v^s | \mu_v^s, \alpha_v)) \\ &= \sum_{s \in S} \sum_{v \in V} (log(\Gamma(x_v^s + \alpha_v^{-1})) - log(\Gamma(x_v^s + 1)) - log(\Gamma(\alpha^{-1})) \\ &- (x_v^s + \alpha_v^{-1}) log(1 + \alpha_v \mu_v^s) + x_v^s log(\alpha_v \mu_v^s)) \end{aligned}$$

Ignoring x_v^s from the above equation, we have the loss function,

$$L(\mu, \alpha) = \sum_{s \in S} \sum_{v \in V} (log(\Gamma(\alpha_v^{-1})) - log(\Gamma(x_v^s + \alpha_v^{-1})) + (x_v^s + \alpha_v^{-1}) log(1 + \alpha_v \mu_v^s) - x_v^s log(\alpha_v \mu_v^s))$$

The splice junction representing the true introns should be consistent and significant between replicates. We use a single parameter β_v to represent μ_v^s ($\mu_v^s = \beta_v$) for splice junction (intron) v across multiple samples and the above equation can be simplify as,

$$\begin{aligned} &L(\mu, \alpha) = L(\beta, \alpha) \\ &= \sum_{s \in S} \sum_{v \in V} (log(\Gamma(\alpha_v^{-1})) - log(\Gamma(x_v^s + \alpha_v^{-1})) + (x_v^s + \alpha_v^{-1}) log(1 + \alpha_v \beta_v) - x_v^s log(\alpha_v \beta_v)) \end{aligned}$$

2.3.1.2 Model for intron detection

For the intron detection task, the total number of candidate splice junctions collected from all samples is large, but only a few splice junctions are expected to be real.

Therefore, the solution must be sparse, and we can use L_1(Lasso) regularization to encode sparseness. Together with $L(\beta, \alpha)$, we propose:

$$J(\beta, \alpha; t) = L(\beta, \alpha) + t_\beta ||\beta||_1 + t_\alpha ||\alpha||_1$$

where $t = (t_\beta, t_\alpha)$, $t > 0$ is the regularization parameter. Since $\beta \geq 0$ and $\alpha \geq 0$, for $v \in V$

$$\begin{aligned} t_\beta ||\beta||_1 &= t_\beta \sum_{v \in V} \beta_v \\ t_\alpha ||\alpha||_1 &= t_\alpha \sum_{v \in V} \alpha_v \end{aligned}$$

Finally, for intron detection task, the optimization problem can be expressed as:

$$\begin{aligned} &argmin_{\beta,\alpha} J(\beta, \alpha; t) \\ s.t.\ \ & t > 0 \\ & \beta \geq 0 \\ & \alpha > 0 \end{aligned}$$

2.3.1.3 Model for differential splicing

To further estimate differential usage of introns and gene expression in different condition groups, we redesign the mean μ_v^s to include more information. Specifically, μ_v^s is predicted via a log-linear model, $log\ \mu_v^s = \beta_1 + \beta_2 + \beta_3 + \cdots$, where the parameter β_i is used to model the potential factors of interest. The factors we considered for our model included (1) the fraction of the reads mapped to splice junction, β_v; (2) the sample the reads came from, β_s; (3) the treatment condition, β_{cv}; (4) the control condition, β_{tv}; (5) the baseline expression strength of the gene, β_g; (6) the fraction of the reads mapped to the gene that overlap with the splice junction, β_{fg}; and (7) the logarithm of the fold change in the overall expression of the gene, β_{fcg}. We evaluated multiple combinations of parameters and parameter values, to select a subset that had a critical impact on the results, which led to:

$$log\ \mu_v^s = \beta_v + \beta_s + \beta_{cv} + \beta_{tv}$$

Where, μ_v^s is decompose into four factors, β_v is the logarithm of the expected fraction of the reads mapped to splice junction v. β_s is the logarithm of the expected fraction of the reads came from sample s, β_{tv} is the effect of the test condition has on the fraction of reads falling into splice junction v. β_{cv} is the effect of the control condition has on the fraction of reads falling into splice junction v. hence, $L(\mu, \alpha)$ change to,

$$\begin{aligned}
L(\mu, \alpha) = L(\beta, \alpha) = & \sum_{s \in S} \sum_{v \in V} (log(\Gamma(\alpha_v^{-1})) - log(\Gamma(x_v^s + \alpha_v^{-1})) \\
& + (x_v^s + \alpha_v^{-1}) log(1 + \alpha_v exp(\beta_v + \beta_s + \beta_{cv} + \beta_{tv})) \\
& - x_v^s log(\alpha_v exp(\beta_v + \beta_s + \beta_{cv} + \beta_{tv}))) \\
s.t.\ \ \beta_v, \beta_s, \beta_{cv}, \beta_{tv} & \geq 0 \\
\alpha_v & > 0
\end{aligned}$$

2.3.1.4 Implementation and parameter selection

Like JULiP, JULiP2 works in four phases:

(1) construct the set of candidate introns and their read counts;

(2) detect mapping regions and merge into gene regions;

(3) for the intron detection task, estimate intron abundance using the GLM and select an optimal set of introns;

(4) for the differential analysis task, estimate the differential usage of introns by GLM coupled with a Wald test.

In *step 1*, a list of RNA-seq BAM files and optionally a transcriptome database are accepted as the JULiP2 input. A set of candidate introns and their read counts in each sample are inferred from the spliced alignments (in BAM file). To reduce ambiguity due to reads mapping to multiple locations, we use the count of primary mapping reads.

In *step 2*, if a transcriptome database is provided, genes and exons info will be extracted, otherwise, genic and exonic regions will be inferred by JULiP2. For the inference, overlapping read alignments in each sample are merged to form contiguous exonic regions. Exonic regions from individual samples are then merged across all samples and used as 'bins' for clustering introns. Exonic regions are further connected via the candidate introns from step 1 into larger gene regions. This process may create long regions that contain several genes. Large 'bins' harboring hundreds of introns can significantly affect performance as well as the accuracy of the program, for instance when genes with varying expression levels are being clustered together. Therefore, an optional approach can be adopted to split regions. The idea is to cut region if intron counts change abruptly over a fixed window.

In *step 3*, the model selects a subset of candidate introns believed to be expressed in the samples and estimates their abundance levels, using the L_1-regularized linear program described in the previous section. The algorithm iteratively estimates the abundance ratio β of introns based on their read counts. The process updates β and reduces the regularization parameters t accordingly, and is iterated until convergence.

In *step 4*, the GLM algorithms described in the previous section are used to reconstruct the read counts of introns under the null and the alternative hypotheses. The null hypothesis is that there is no difference in intron usage between the conditions, while the alternative hypothesis assumes some introns are differentially used between the two conditions. A likelihoood ratio test is applied to evaluate the results. A p-value is calculated on each intron within a gene and a Benjamini-Hochberg (BH) correction for multiple testing is used to reduce the false discovery rate and calculate a gene-wise q-value.

JULiP2 is designed take advantage of multiple CPU cores and scales well with the input RNA-seq alignment data. JULiP2 can quantify abundance from pre-computed alignments provided in SAM or BAM format. JULiP2 is written in Pyhton, is open-source and free licensed (GPL v3). It has been developed and tested on Linux and Macintosh OS X. The software and user manual are freely available at https://github.com/Guangyu-Yang/JULiP2 .

2.3.2 Evaluation

2.3.2.1 Performance of feature selection

We used the framework developed for evaluating JULiP, including measurements on both simulated and real data, to compare the performance of JULiP2 vis-à-vis its predecessor and several transcript assembly methods, including Cufflinks, CLASS2, StringTie, FlipFlop and the multi-sample assembler ISP. As before, introns were extracted from transcript predictions for the single-sample assemblers and were then combined across all samples, whereas for the multi-assembler ISP introns were extracted from the reported joint transcript set. For JULiP2, we used four types of models and parameter values for estimating the parameters of the negative binomial distributions (nb1-4).

Set	TP	FP	Sn	Pr	F-val
Gold reference	12,996	NA	NA	NA	NA
Cufflinks	9,832	248	0.757	0.975	0.852
CLASS2	10,194	51	0.784	0.995	0.877
StringTie	10,093	29	0.777	0.997	0.873
Cufflinks_F0.01	10,601	506	0.816	0.954	0.880
CLASS2_F0.01	10,418	61	0.802	0.994	0.888
StringTie_F0.01	10,462	50	0.805	0.995	0.890
FlipFlop	10,214	2,437	0.786	0.807	0.797
ISP	9,949	187	0.766	0.982	0.860
JULiP	11,628	182	0.895	0.985	0.938
JULiP2 (nb1)	12,089	294	0.930	0.976	0.953
JULiP2 (nb2)	11,899	235	0.916	0.980	0.947
JULiP2 (nb3)	11,671	212	0.898	0.982	0.938
JULiP2 (nb4)	11480	185	0.883	0.984	0.931

Table 2-VI. Performance of JULiP2 (versions nb1, nb2, nb3 and nb4) and other methods on 25 simulated RNA-seq data sets.

The first, accuracy test, on 25 simulated RNA-seq samples, showed JULiP2 to achieve better or comparable results with JULiP, as measured by the F-value as well as by the individual sensitivity and precision measures, and both JULiP programs in turn were significantly more sensitive than the other methods, namely >30% more than the most sensitive single-sample method, and 10% more introns in the cumulative set of samples (Table 2-VI). The second test, on real data, assessed the programs on 50 RNA-seq data sets from lymphoblastoid RNA-seq samples sequenced as part of the GEUVADIS population variation project. JULiP2 detected a number of known introns comparable to JULiP, but produced a larger number of additional introns, apparent false positives. Most of the additional introns, however, could be explained by new sources of information (dbEST, GenBank RNA-seq, recurrence), thus rendering JULiP2 as the tool with the highest capacity for finding intron features (Table 2-VII).

Set	TP	FP	Sn	Pr	F-val	Sn'	Pr'	F-val'
Reference (GENCODE)	20,171	NA	NA	NA	NA	NA	NA	NA
Cufflinks	9,507	2,424	0.471	0.797	0.592	0.575	0.971	0.722
CLASS2	9,917	3,351	0.492	0.747	0.593	0.633	0.962	0.763
StringTie	9,206	1,948	0.456	0.825	0.588	0.550	0.994	0.708
Cufflinks_F0.01	10,883	8,739	0.540	0.555	0.547	0.842	0.866	0.854
CLASS2_F0.01	10,521	5,286	0.522	0.666	0.585	0.737	0.940	0.826
StringTie_F0.01	9,688	3,089	0.480	0.758	0.588	0.629	0.992	0.770
FlipFlop	9,960	13,656	0.494	0.422	0.455	0.684	0.584	0.630
ISP	9,030	1,723	0.448	0.840	0.584	0.507	0.952	0.662
JULiP	11,846	5,049	0.587	0.701	0.639	0.825	0.985	0.898
JULiP2 (nb1)	11,954	10,526	0.593	0.532	0.561	0.995	0.893	0.941
JULiP2 (nb2)	11,631	7,336	0.577	0.613	0.594	0.909	0.967	0.937
JULiP2 (nb3)	11,271	5,433	0.559	0.675	0.611	0.821	0.992	0.898
JULiP2 (nb4)	10,967	4,390	0.544	0.714	0.617	0.759	0.995	0.861

Table 2-VII. Performance of JULiP2 (nb1, nb2, nb3 and nb4) and other methods on the GEUVADIS data set (50 lymphoblastoid RNA-seq samples).

2.3.2.2 Performance of differential splicing on simulated data

We evaluated the performance of JULiP2, comparing it with the programs JunctionSeq, rMATS, LeafCutter and MAJIQ for the differential analysis task, and with Cufflinks, Stringtie, and CLASS for the intron detection task. All the approaches were run with the default settings in the experiments.

We first evaluate the programs in a controlled setting where the true transcripts' expression and splicing levels are known. For this task, we generate 50 synthetic samples (7 million 101 bp paired-end reads/sample) by Polyester. The simulator was trained on GenBank SRR493366 reads, with gene expression levels (in FPKM) estimated with Ballgown [44] from Tophat2 alignments. For the simulation pipeline, we randomly select 2000 protein coding genes from among those containing at least two different expressed isoforms. We set the expression levels of the selected genes in the control samples to be the same as in the original sample. For the 'test' samples, we assign the selected protein coding genes into 4 groups, each with 500 genes, as follows. 1) The first group had no change in FPKM values. 2) For the second group, we simulated differential gene expressions (DE) by changing (doubling or halving) the

FPKM values of the genes, randomly. 3) In the third group, we simulated differential splicing (DS) by swapping the expression levels of the two most highly expressed isoforms for the gene. 4) In the fourth group, we applied the modification in 2) and 3) to create a mixture of differential expression and differential splicing events.

A true positive (TP) occurs when the prediction matches the gene settings, a false positive (FP) when the prediction does not appear in the gene settings, and lastly, a false negative (FN) occurs when a true differential event is not reported by the program. We calculate the standard sensitivity, precision and the F-value for each program. As seen in (Figure 2-5A), JULiP2 has the best performance in our testing, and sensitivity, precision and F-value are all very high. Of the programs tested, only JunctionSeq achieves a similar level of performance, but we note that JunctionSeq incorporates external information about the annotations and exon counts. Additionally, as a drawback, JunctionSeq has a large running time and cannot process large datasets.

2.3.2.3 Performance of differential splicing on real data

Further, to test the programs for their abilities in detecting differential splicing events in a real setting, we ran JULiP2, rMATS, MAJIQ and LeafCutter on an RNA-seq data set from hippocampus tissue of healthy and epileptic mice (PRJEB18790) [45]. We used 10 randomly selected control samples and 10 epilepsy samples. We aligned the reads with STAR [46] and applied each method to identify splicing differences between the two conditions (chr2 only, for simplicity). Figure 2-5B shows the correspondence between the sets of results, indicating that many of the predictions are program specific.

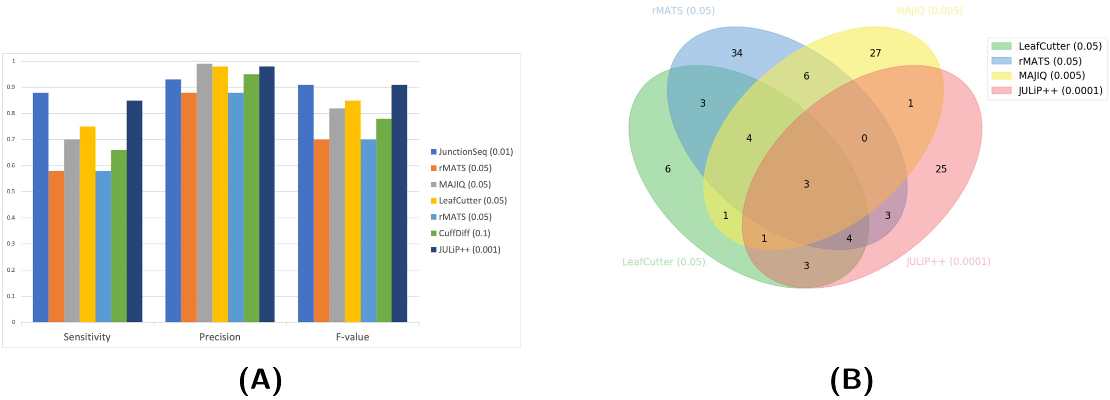


Figure 2-5. Evaluation of differential splicing prediction methods. (A) *Simulated data:* 2D column plots of the sensitivity, precision and F-values for the selected programs. The p-value thresholds are chosen based on the recommendation in the papers. (B) *Real data:* Correlations of differential splicing predictions by different methods.

2.3.2.4 Detection of non-canonical splice junctions from a collection of human liver RNA-seq data

To further assess the usefulness of our method, we applied JULiP2 to a collection of post-mortem human liver tissue samples from individuals with mental illness (schizophrenia, 21 samples) and unaffected controls (17 samples) (GenBank accession: PRJNA575230). We chose to focus on non-canonical splice junctions because they are difficult to sift from a large number of false positives in the initial set of alignments.

Filter	All	Non-canonical
Original	1,058,787	382,272
Alignment	830,791	154,276
JULiP2 ($\beta \geq 0.1$)	195,938	5,328
JULiP2 (DS, sufficient input)	183,632	3,457
Context (C=1)	182,136	3,445
Diff. spliced	11,258	720

Table 2-VIII. Non-canonical intron detection and selection from 21 RNA-seq liver samples from schizophrenia and 17 unaffected individuals.

Specifically, our approach was as follows (Table 2-VIII). We used STAR to generate genome-wide spliced alignments, from which we extracted candidate splice junctions (introns) with the tool ‘junc’ [18], modified and included in the JULiP2 package. This procedure generated 1,058,787 splice junctions across all samples, including 382,272 candidate non-canonical (NC) splice junctions. To reduce the number of false positive NC junctions and select an accurate subset for further investigation, we applied a three step filtering procedure. First, we employed the alignment-based filters implemented in the tool ‘junc’ to remove alignment artifacts, namely inconsistent mappings between reads within the same pair, reads with low alignment scores, and reads from non-concordant pairs. This step reduced the number of unique introns to 830,791, including 154,276 NC introns. Second, we applied the intron selection procedures implemented in JULiP2 to select a subset of high-confidence introns (183,632 introns, of which 3,457 non-canonical). Third, we selected only those introns that were identified as ‘reliable’ based on sufficient original read mapping support in at least C samples (default: 1). This resulted in 183,136 introns, including 3,445 NC introns. Of these remaining candidates, we further prioritized 720 putative NC junctions that JULiP2’s differential splicing module identified as differentially spliced between schizophrenic and unaffected individuals ($C = 1$). Figure 2-6 shows the alignments of one such example, intron chr6:25137823-25138073 with the genomic signal CT-GC (GC-AG on the gene’s strand) at the CMAHP gene.

2.4 Discussion

In this section, we described JULiP and JULiP2, two novel methods for intron selection from multiple RNA-seq samples and for differential splicing detection. The methods simultaneously model splice junction information across the samples into linear models, namely a regularized linear program for JULiP and a GLM for JULiP2, taking advantage of the latent information in the set of samples to extract a highly accurate set of introns.

When evaluated on simulated data, JULiP significantly outperformed current

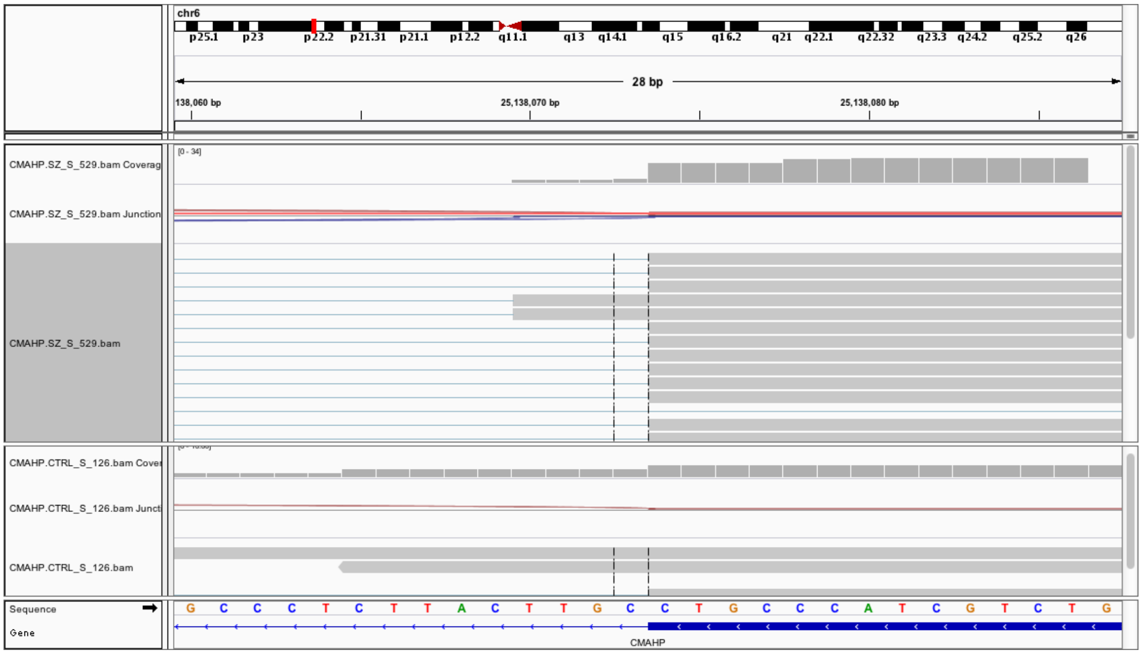


Figure 2-6. IGV plot showing the non-canonical splice signal CT-GC at the CMAHP gene.

assembly based approaches in the ability to select an accurate set of features (introns). In particular, JULiP could identify 16-30% more introns from a single sample compared to reference (single-sample) programs at precision comparable to the highest, and 9-18% more features when introns were pooled together across all samples. When applied to real data, JULiP had the highest potential to identify splice variation from multiple samples, demonstrating both high sensitivity and very high precision, >98%.

An implementation of JULiP using the Hadoop parallel framework scaled well with the number of nodes and samples, and ran in under 2 minutes for up to 100 samples. Therefore, JULiP provides a highly efficient model for intron selection from multiple, potentially hundreds and thousands of RNA-seq samples, and can be used as a standalone tool or can be efficiently integrated into a transcript assembly method for comprehensive annotation of splice variation.

While JULiP is restricted to the problem of intron selection, JULiP2 offers a more comprehensive and flexible approach, with the addition of a differential splicing model and a more realistic read count model based on the negative binomial distribution. The negative binomial distribution facilitates an intuitively interpretable separation of biological from technical variation, while the GLM allows for arbitrarily complex experiments.

Using both simulated and real data, we benchmarked JULiP2 against Cufflinks, Stringtie, and CLASS for the intron detection task, and against CuffDiff, JunctionSeq, and DEXSeq for the differential analysis task. The experiments demonstrated that our method has high sensitivity and precision, while better controlling the rate of false positives.

In summary, both JULiP and JULiP2 are novel and representative models for leveraging latent gene information from multiple related RNA-seq samples to significantly increase the accuracy of feature selection. Additionally, JULiP2 provides a robust and flexible framework for differential splicing analysis at the intron level. Our tools are annotation and assembly free, and therefore avoid the pitfalls of assembly while allowing for the detection of new splice variants. Lastly, they are lightweight, memory efficient and highly scalable, offering a powerful and practical solution for the analysis of large scale RNA-seq experiments.

Chapter 3

MntJULiP and Jutils

In Chapter 2, we discussed JULiP and JULiP2 and their applications to intron selection and differential splicing analysis. While our assessments indicate that JULiP and JULiP2 perform well, there are still limitations. For instance, because they rely on the gene context for estimating intron parameters and for testing, their performance suffers when used without a reference gene annotation to accurately determine the boundaries of the gene. To address this and other challenges, we developed MntJULiP, a comprehensive and scalable tool for differential splicing detection at the intron level, based on generalized linear models. Further, there is a scarcity of tools to present differential splicing events to biologists in an intuitive way. To fill this gap, we developed the visualization toolkit Jutils, to create representations of results produced by MntJULiP and other differential splicing tools as heatmaps, sashimi plots and Venn diagrams.

3.1 Background

Gene alternative splicing is a fundamental biological process that gives rise to a wide array of protein isoforms with modified properties in plant and animal systems. Most splicing variations are tissue specific, but splicing is also altered by external stimuli [15] and aberrant splicing has been associated with diseases [31]. Therefore, there is a great need to accurately map and quantify gene splice variants, as well as to identify differences in splicing between conditions.

As described in Chapter 2, current methods for differential splicing detection can detect and quantify alternative splicing from RNA sequencing (RNA-seq) data at the level of transcripts (isoforms), splicing events (exon skipping, mutually exclusive exons, alternative exon ends, intron retention), or primitive features (subexons, introns). Isoform-level quantification methods (Cuffdiff, Cuffdiff2, MISO [17, 32, 47]) require a reference annotation or a reconstructed set of transcripts, and their performance suffers from incompleteness and inaccuracies in the assemblies. Event level methods (DiffSplice, rMATS [23, 33]) are less affected by assembly errors, but represent only a subset of alternative splicing variations. For both of these methods, quantification is further complicated by the ambiguity in assigning reads that map to multiple locations in the genome and multiple transcripts of a gene. In contrast, more recent methods (LeafCutter, MAJIQ, JunctionSeq [34, 37, 38]) target introns, which can be more reliably identified from read alignments, capture a wider variety of splicing variations, and are less ambiguous to quantify, as intron-spanning reads associate with unique gene splice patterns.

Methods further differ in how they define splicing differences. Most methods quantify changes in the relative splicing levels of the target feature within a group of mutually exclusive local splicing patterns (LeafCutter, MAJIQ, rMATS, DiffSplice), or alternately identify features with splicing usage inconsistent with the rest of the gene (JunctionSeq, DEXseq [48]). Yet others look for changes in the overall (absolute) abundance levels, as a means to identify changes in isoform regulation leading to functional effects (Cuffdiff, Cuffdiff2, ALEXA-seq [17, 32, 49]).

Lastly, to increase accuracy, some methods rely on a pre-existing set of gene annotations to identify relevant splicing variations, limiting discovery of novel and potentially condition-specific features. The rich spectrum of methods for alternative splicing quantification and differential analysis offer a multifaceted yet inconsistent view of alternative splicing variation [50].

3.2 MntJULiP: Comprehensive and scalable quantification of splicing differences from RNA-seq data

We introduce MntJULiP, a statistical learning method based on a novel mixture Bayesian framework, for detecting differences in splicing between large collections of RNA-seq samples. MntJULiP represents splicing variation at intron level, thus capturing most splicing variations while avoiding the pitfalls of assembly. It infers intron annotations directly from the alignments, making it possible to discover new unannotated candidate markers. MntJULiP detects both differences in intron splicing levels, herein called differential splicing abundance (DSA), and differences in intron splicing ratios relative to the local gene output, termed differential splicing ratio (DSR) (Figure 3-1). Salient features of MntJULiP include: i) a novel statistical framework, including a zero-inflated negative binomial mixture model for individual introns, in the DSA model, and a Dirichlet multinomial mixture model for groups of alternatively spliced introns, in the DSR model; ii) it captures significantly more alternative splicing variation, and more types of variation, than existing tools; iii) superior performance compared to reference methods, including increased sensitivity in control experiments, and high reproducibility and reduced false positives in comparisons on real data; iv) a unique mixture model that allows comparison of multiple conditions simultaneously, to aptly capture global variation in complex and time-series experiments; and v) highly scalable, it could process hundreds of GTEx samples in less than half an hour.

MntJULiP differs from and improves upon the model implemented in JULiP and JULiP2 in multiple ways. The main difference is that, unlike JULiP and JULiP2 that jointly model the set of all introns at a gene, MntJULiP targets individual introns (DSA) or groups of introns that share the same splice site (DSR). In this way, MntJULiP can extract splicing events directly from the RNA-seq alignments, without the need for a reference gene annotation. Moreover, such setup can avoid grouping large numbers of introns from a gene region, which usually results in poor performance for some low expressed splice junctions or genes. Further, unlike JULiP/JULiP2 that select introns directly via the L1 regulated model, MntJULiP can be used for intron selection in an implicit way. Specifically, the MntJULiP DSA model estimates the mean expression level of an intron in given condition(s), which it then uses to filter out introns below a user specified threshold, likely caused by 'noise' or alignment errors.

3.2.1 Methods

MntJULiP consists of two components, a 'builder' and a 'quantifier'. The *builder* takes as input the aligned RNA-seq reads, extracts the splice junctions (introns) and their supporting read counts, and filters introns with low support (≤ 3 reads in each of the samples). A second, context-based filter for low support introns is embedded in the statistical model below. Individual introns are the subject of DSA analysis. For the DSR analysis, introns that share an endpoint are grouped into 'bunches'. If a reference gene annotation is provided, both individual introns and bunches are associated with an annotated gene if they share at least one intron coordinate. The *quantifier* subsequently evaluates candidate introns, building a learning model for each intron and performing two statistical tests: i) a test for a change in intron abundance between the two (or more) conditions (*DSA*), and ii) a test for a change in the splicing level of the intron relative to its 'bunch' (*DSR*). For DSA, MntJULiP uses a mixture zero-inflated negative binomial model to estimate individual introns' abundance levels from the raw read counts. For DSR, it estimate the relative splicing ratios with a mixture multinomial Dirichlet distribution. In both models, log-likelihood ratio tests are used to determine differential splicing events. P-values are then adjusted for

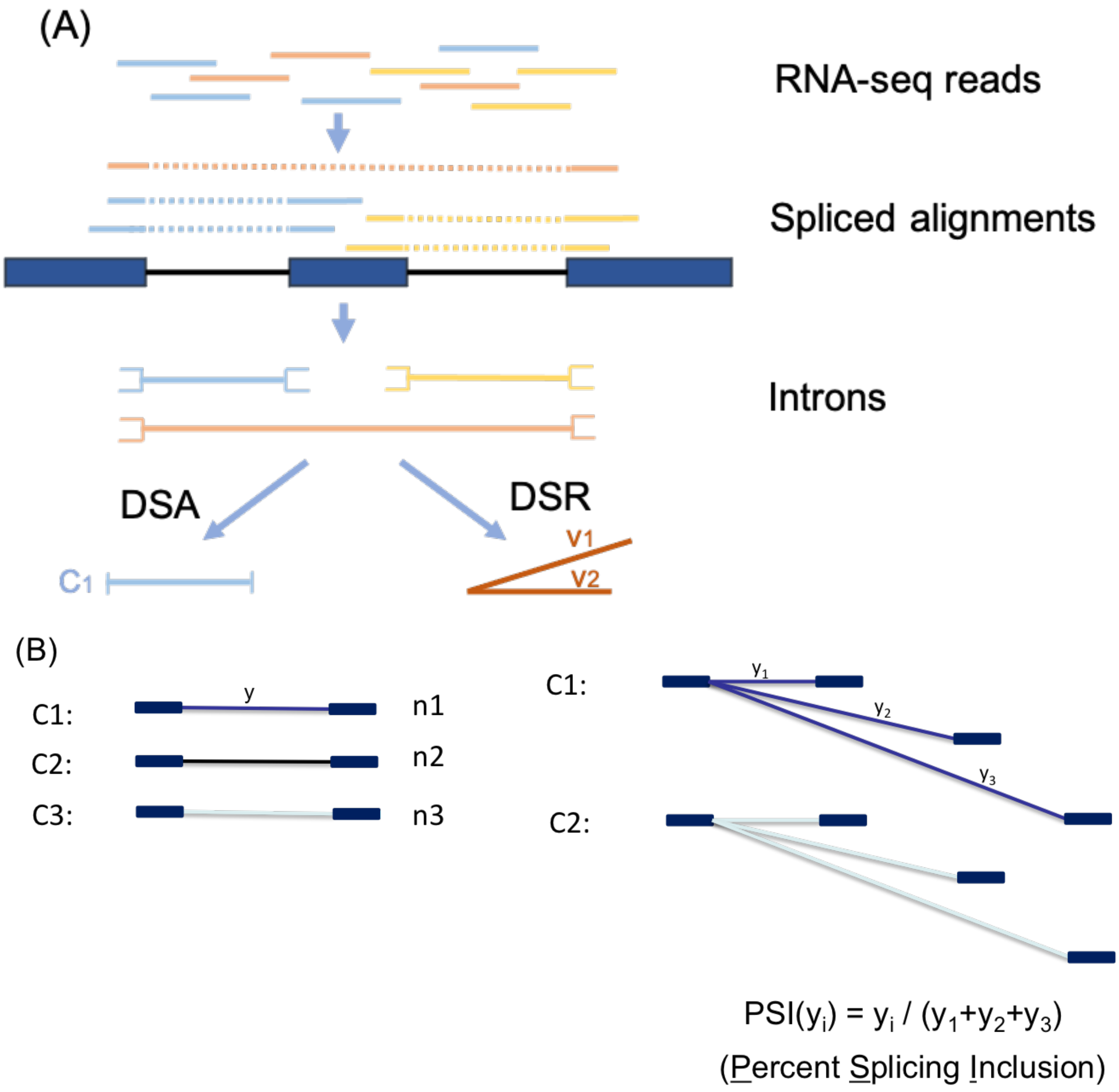


Figure 3-1. (A) RNA-seq reads are aligned to the genome and spliced alignments are used to detect the genomes and calculate their read counts. MntJULiP then tests individual introns for differential intron abundance (DSA), and groups of introns sharing a splice site ('bunches') for differential splicing ratio (DSR). (B) Left, DSA: Each intron is analyzed individually, and the expression (abundance) level is compared between conditions. Right, DSR: Introns that share a splice junction ('bunch') are collectively analyzed, and the PSI value for each intron is compared between conditions. Shown are: an individual exon in a three-condition experiment, in the DSA diagram, and a three-intron 'bunch' in a two-condition experiment, in the DSR diagram.

multiple testing using the Benjamini-Hochberg correction. The entire framework is described in detail below.

3.2.1.1 Bayesian read count model

We use a Bayesian statistical framework to estimate intron splicing levels for differential analyses. This framework also provides another way to distinguish true introns from sequencing and systematic errors, as a second context-based intron filter. To start, we assume that the read count y of intron v in a given sample follows a negative binomial distribution $NB(\mu,\, \theta)$. We next add a loose prior with an empirical $\hat{\mu}$ (the sample mean) modeled by a normal distribution: $\mu \sim N(\hat{\mu},\, \sqrt{\hat{\mu}/10})$ to model subtle variability between conditions and among the individual samples, and a restriction to the dispersion parameter ϕ as an inverse Half-Cauchy distribution: $\phi^{-1} \sim \text{Half Cauchy}(0, 5)$. Lastly, to account for low expression genes and transcripts, and for low read count introns from library preparation and sequencing artifacts, we introduce a zero inflated modifier on the negative binomial Bayesian model [51]:

$$y \sim \begin{cases} 0, & \text{with probability } \pi \\ NB(y), & \text{with probability } 1 - \pi \end{cases}$$

where $NB(y)$ is the probability density function of the negative binomial model described above.

Let $p(y)$ denote the probability density function for this model. For n samples and intron read count y_j in sample j, we define the log likelihood:

$$L(\theta) = \log p(y_1, y_2, \ldots, y_n) = \sum_{j=1}^{n} \log p(y_j)$$

We maximize the log likelihood function using the Limited memory Broyden Fletcher Goldfarb Shanno (LM-BFGS) optimization method [52] and obtain point estimates for parameter θ over the samples.

3.2.1.2 The differential splicing abundance (DSA) model

The previous section established a general Bayesian model to estimate intron abundance. Next we describe the framework for modeling individual intron abundance and for DSA testing in a multi-condition experiment. Assume that samples are drawn from m (typically 2) conditions. Given an intron v and a sample generated from condition i, its intron read count y follows a negative binomial distribution $NB_i(y)$ with the condition specific parameters μ_i, θ_i , ϕ_i and π_i, as defined earlier.

Let $p_i(y)$ be the probability density function for the complete model for condition $i \in \{1, \ldots, m\}$. We define a mixture probability model for y:

$$\bar{p}(y) = \prod_i p_i(y)^{z_i}$$

where z_i is the indicator variable for that sample, equal to 1 iff the sample belongs to condition i and 0 otherwise.

To formulate the problem, given n samples, m conditions and y_j the intron read count in sample $j \in \{1, \ldots, n\}$, we define the log likelihood:

$$L(\theta) = \log \bar{p}(y_1, y_2, \ldots, y_n) = \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \log p_i(y_j)$$

with $z_{ij} \in \{0, 1\}$ the indicator variable for sample j and condition i.

Having these two Bayesian models, we establish a hypothesis test for differential intron abundance given the data: the null hypothesis is that samples are generated from the same condition, and the alternative hypothesis is that the samples belong to different conditions, and apply a likelihood-ratio test:

$$LR = -2[L(\theta_0) - L(\theta_1)],$$

where $L(\theta_0), L(\theta_1)$ are the log likelihoods of the null and alternative hypothesis models, respectively, with parameters θ_0 and θ_1 .

Lastly, since the parameter μ_j of the alternative hypothesis model is the expected read count (mean) of the intron in condition j, we can establish an additional intron filter by setting a threshold for μ_j (e.g., $\mu_j \geq 1$), to separate a ‘true’ intron from noise.

3.2.1.3 The differential splicing ratio (DSR) model

We next formulate a framework to test for differences in splicing ratios of introns within a ‘bunch’, or groups of introns sharing an endpoint. For simplicity, we start by assuming that all samples belong to the same condition and the read counts $y_1, y_2, \ldots, y_k$ in a bundle with k introns follow a Dirichlet-multinomial distribution with priors $\alpha_1, \alpha_2, \ldots, \alpha_k$: $y_1, y_2, \ldots, y_k \sim DM(\alpha_1, \alpha_2, \ldots, \alpha_k)$

Let $p(y_1, y_2, \ldots, y_k)$ be the probability density function. For intron read counts $(y_{1j}, y_{2j}, \ldots, y_{kj})$ in sample $j = 1, \ldots, n$, we define the log likelihood function:

$$L(\theta) = \log p(y_1, y_2, \ldots, y_n) = \sum_{j=1}^{n} \log p(y_j)$$

Similar to the discussion in the previous subsection, to extend to the case where samples belong to multiple conditions we define a Dirichlet-multinomial distribution with prior $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik}$ for each condition $i \in \{1, \ldots, m\}$.

$$y^i{}_1, y^i{}_2, \ldots, y^i{}_k \sim DM(\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik})$$

Let $p^i(y^i{}_1, y^i{}_2, \ldots, y^i{}_k)$ be the probability density function for condition i. We define the log likelihood function

$$L(\theta) = \log p(y_1, y_2, \ldots, y_n) = \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \log p^i(y_j),$$

where $y_j = (y_{1j}, y_{2j}, \ldots, y_{kj}$ are the read counts of introns in the bundle in sample j, $z_{i,j} \in \{0, 1\}$ indicates whether sample j belongs to condition i or not, and θ represents the parameter set of the model.

With the two Bayesian models above, we formulate a log-likelihood ratio test as before: the null hypothesis assumes all samples belong to the same condition, and the alternative hypothesis assumes multiple conditions. Under the alternative hypothesis, the parameters $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik}$ for condition i can be used to define the splicing ratio, similar to Percent Splicing Inclusion [16, 38], Ψ_{il} of intron $l \in \{1, \ldots, k\}$

under condition i as:

$$\Psi_{il} = \frac{\alpha_{il}}{\sum_{l'=1}^{k} \alpha_{il'}}$$

3.2.1.4 Sequences and materials

Simulated data. We generated 25 control and 25 perturbed RNA-seq samples with 86 million 101 bp paired-end reads each, using the software Polyester with human GENCODE v.22 as reference annotation. For the control samples, we used a model of gene and transcript abundance inferred from lung fibroblasts (GenBank Accession: SRR493366). To simulate the perturbed condition, we randomly selected 2,000 annotated protein coding genes with two or more expressed isoforms and assigned them to four groups as follows [37, 53]: i) 500 genes were left unperturbed (NONE); ii) 500 genes had only expression changes (DE), where genes were randomly assigned one half or double the original FPKM value; ii) 500 genes had only splicing differences (DS), obtained by swapping the expression values of the top two isoforms; and iv) 500 genes had both expression and splicing changes (DE-DS). Thus, 1,500 genes underwent changes in splicing abundance, and 1,000 had differences in splicing, and were used as the gold reference for evaluating the tools under the DSA and DSR models, respectively.

Real data. Reads for 44 mouse hippocampus samples (24 cases and 20 controls) were obtained from GenBank (ProjectID: PRJEB18790). Tissue RNA-seq samples for comparative analyses (121 cortex, 105 frontal cortex, 132 cerebellum, and 196 lung samples) were obtained from the GTEx collection [27]. Lastly, RNA-seq data from differentiating mouse taste organoids [54] (14 samples, 7 stages) were obtained from the Sequence Read Archive (Accession: DRA005238).

3.2.1.5 Performance evaluation

Reads were mapped with the program STAR v.2.4.2a to the human genome GRCh38 or mouse genome GRCm38 (mm10), as applicable. Alignments were analyzed with the programs MntJULiP v1.0, LeafCutter v0.2.8, MAJIQ v1.1.7a, rMATS v3.2.5 and Cuffdiff2 v2.2.1 to determine changes in alternative splicing profiles. For the simulated tests, transcripts were reconstructed across each sample with StringTie v2.1.4 then merged across samples with StringTie(ST)-merge and the GENCODE transcripts as reference, to create a set of gene annotations to be used with all programs. To evaluate the programs' accuracy in predicting differentially spliced genes from the simulated data, the 1000 (DS, DE-DS) gene set and the 1,500 (DS, DE, DE-DS) gene set were used as the gold standard for DSR and DSA prediction, respectively. Any other program predictions were deemed false positives. Standard sensitivity ($Sn = TP/(TP+FN)$), precision ($Pr = TP/(TP+SP)$), and the $F1 = Sn*Pr/(Sn+Pr)$ value were used to measure accuracy. To assess the programs' fidelity in quantifying alternative splicing for the DSR test, reference Percent Splice Inclusion (PSI) values for all reference introns were calculated from the simulated data, as the ratio between the intron abundance and that of its bunch. Similarly, for the DSA test, reference log fold change values were calculated for each intron as the log fold change of the cumulative expression levels of all splice isoforms containing that intron.

3.2.2 Evaluation

We assess the performance of MntJULiP and other programs on simulated and real RNA-seq data, with varying degrees of splice variation and different dataset sizes. We illustrate MntJULiP's ability to detect more types of alternative splicing variation in the comparison of hippocampus samples from healthy and epileptic mice. We then demonstrate MntJULiP's unique capability for simultaneous multi-condition comparisons in a 7-point time series experiment on differentiating mouse taste organoids, and its ability to handle large data sets on a large collection of RNA-seq samples from four human tissues obtained from the GTEx project. We include in the comparisons, as feasible, the state-of-the-art intron-based tools LeafCutter, MAJIQ, JunctionSeq, and the event-based rMATS, and Cuffdiff2 as the only tool among them compatible with the DSA test.

3.2.2.1 Performance on simulated data

In a first, controlled experiment we used simulated data, namely 25 control and 25 perturbed samples, to evaluate MntJULiP (DSR), MAJIQ, LeafCutter, and rMATS in detecting differences in splicing ratios, and MntJULiP (DSA) and Cuffdiff2 in detecting differences in splicing abundance (see Methods and Figure 3-2). On the DSR experiment, MntJULiP(DSR) achieved sensitivity 74.5%, which was 8.0-60.0% higher than its competitors, at very high and comparable precision, 97.4%. Notably, Cuffdiff2, which was not designed as a DSR method, had the highest sensitivity at 94.9%, however at a very significant drop in precision, to 46.4%. On the DSA experiment, MntJULiP(DSA) had very high 97.9% sensitivity and 95.3% precision, to Cuffdiff2's values of 95.9% and 70.3%, respectively. Sensitivity of MntJULiP's DSA test was also significantly higher than any of the DSR programs', which ranged between 31.7-50.3%, illustrating the fact that methods developed for DSR detection are in general not suitable to detect changes in splicing abundance. We further examined in more details the programs' results by gene class. While true positives for all programs were fairly uniformly distributed across the constituent gene categories, false positives for MAJIQ, rMATS and Cuffdiff2 were dominated by genes outside of the simulated gene set, underscoring the difficulty for these programs to effectively distinguish and filter paralogs and other alignment and assembly artifacts.

We further assessed the methods' accuracy in quantifying the amount of change in splicing of individual introns. For the DSR experiment, MntJULiP predictions most closely aligned with the reference annotation ($R^2 = 0.935$, Pearson correlation coefficient) between predicted and reference dPSI values, compared to 0.879 for LeafCutter and 0.847 for MAJIQ. For the DSA experiment, MntJULiP had the higher correlation (0.991 versus 0.848) between predicted and reference log fold change values of the two methods. Therefore, MntJULiP predicted values are strongly indicative of the amount of change, and can be used reliably to inform event selection, for instance to select candidate events for experimental validation.

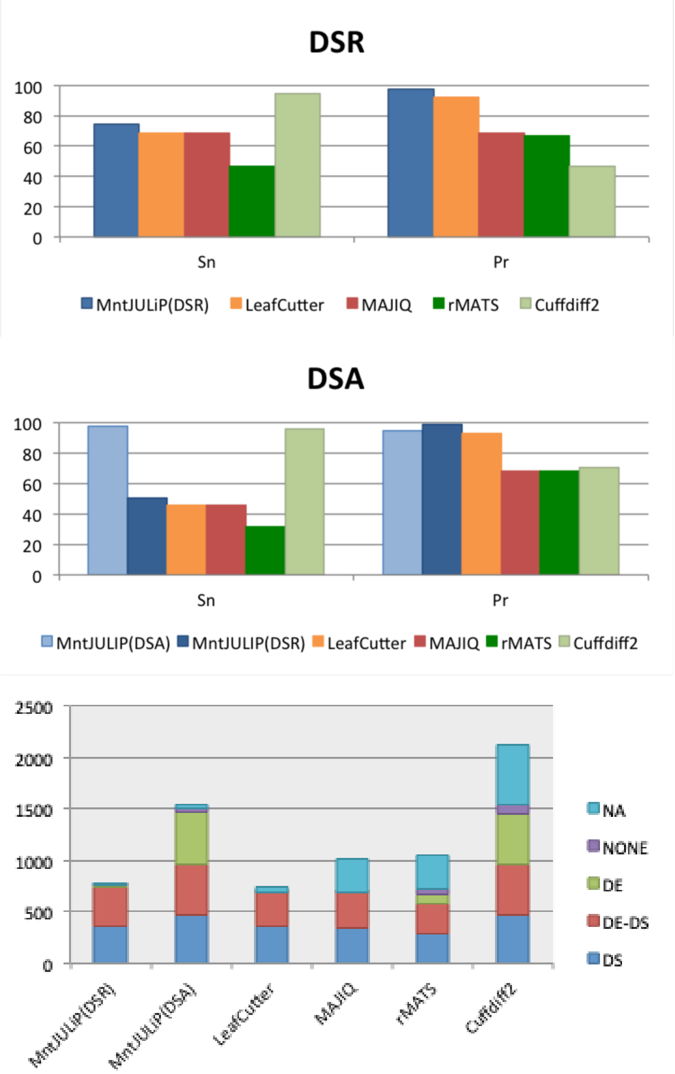


Figure 3-2. Comparative evaluation of differential splicing methods on 25 control and 25 perturbed simulated RNA-seq data sets: (top) DSR, (middle) DSA, and (bottom) breakdown of programs' predictions by the four gene categories (DS, DE, DE-DS and NONE), and novel (NA), i.e. not in the simulation set.

3.2.2.2 Performance on real data

We next applied the methods to RNA-seq samples from hippocampus tissue of 24 healthy mice and 20 mice with pilocarpine induced epilepsy, illustrating a typical RNA-seq experiment. Programs MntJULiP(DSR), LeafCutter, MAJIQ and rMATS predicted between 700 and 1,137 DSR genes (Figure 3-3, left). While it is not possible to precisely measure the prediction accuracy in the absence of a ground truth reference, we deem genes predicted by multiple tools as being more reliable. A majority of DSR genes (974 out of 1,878) were predicted by two or more tools. Importantly, MntJULiP had the smallest number and proportion of uniquely predicted genes, 84 (9.7% of its predictions), compared to 350 genes (35.5%) for rMATS, 367 genes (32.3%) for LeafCutter and 103 genes (14.7%) for MAJIQ, and therefore potentially reported the smallest number of putative false positives.

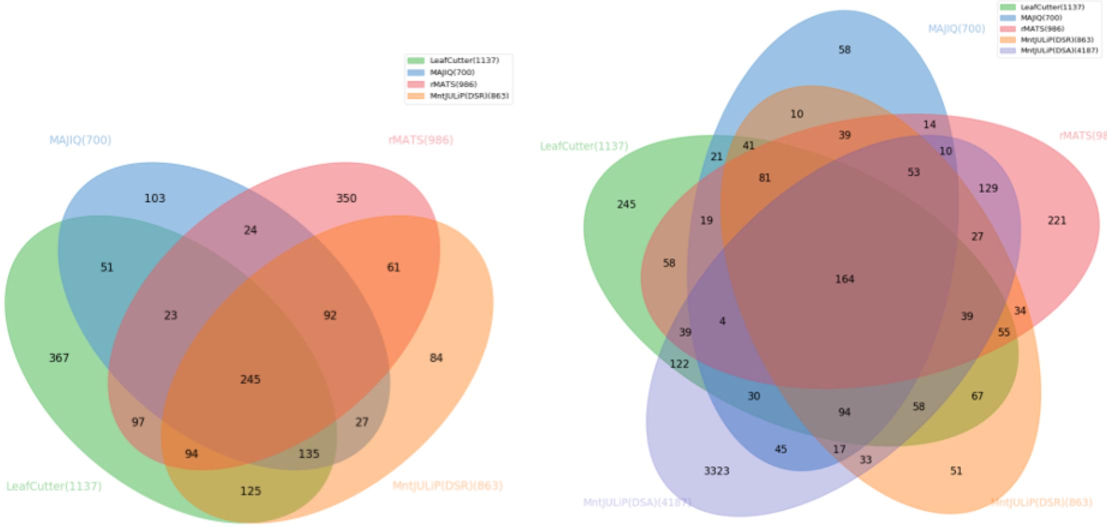


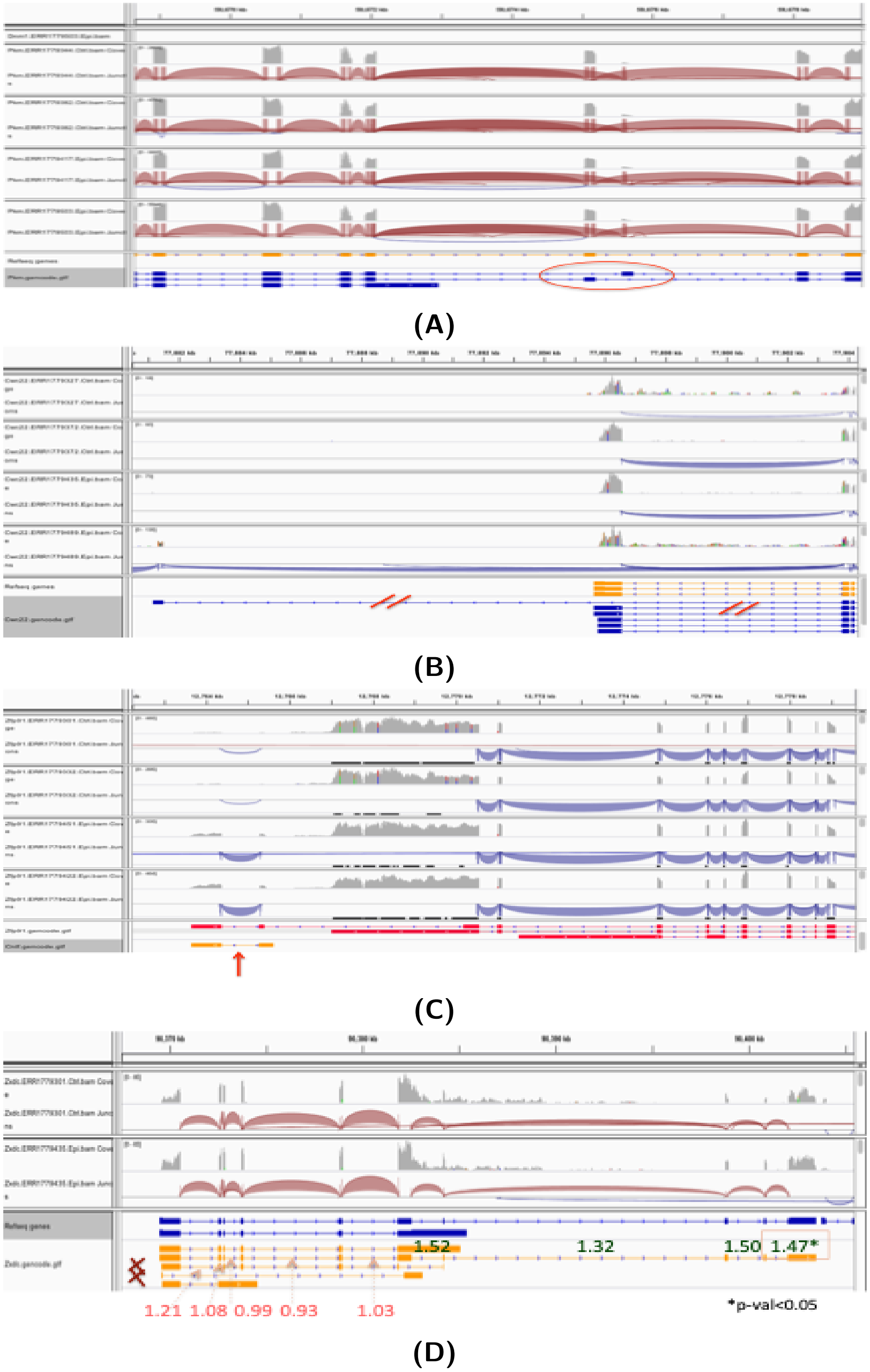
Figure 3-3. Venn diagram of methods' gene-level predictions on samples from 24 healthy and 20 epileptic mice.

DSR tests capture only a fraction of the alternative splicing variation in an experiment. To showcase the potential of MntJULiP to expand upon the classes of alternative splicing events detected, we assessed the outcomes of MntJULiP's DSA test.

Of the 4,187 genes predicted, 485 were also reported by MntJULiP's DSR test and an additional 379 by other tools, representing genes with traditional splicing patterns (Figure 3-3, right). An additional 2,510 genes were determined to be differentially expressed by the DESeq2 14 method, a category that is captured by the DSA test. The remaining 813 genes represent a combination of genes with traditional event patterns that could not have been discovered by other tools and putative complex or non-conventional splicing events.

Figure 3-4 illustrates some of these cases. The Pyruvate Kinase M 1/2 (Pkm) gene has two isoforms resulting from the use of mutually exclusive exons (Figure 3-4A). Pkm1 is expressed in the adult stage where it promotes oxidative phosphorylation, whereas Pkm2 is prevalent during embryogenesis and promotes aerobic glycolysis. Splicing dysregulation at this gene has been identified as an oncogenic driver and passenger factor in brain tumors [55]. While the difference in the isofoms' splicing ratio is low (0.05) and may have contributed to being missed by other tools, introns flanking both exons yielded positive MntJULiP DSA tests. Most importantly, MntJULiP can detect classes of events that cannot be detected by other methods. In one example at the CWC22 Spliceosome Associated Protein Homolog (Cwc22) gene, the two overlapping and mutually exclusive introns at the 3' end of the gene (chr2:77881490-77903814 and chr2:77896578-77903796) do not share an endpoint and therefore could not have been interrogated by other methods (Figure 3-4B). Similarly, none of the traditional methods can capture variation that results when one isoform's intron chain is entirely subsumed by another, where the 'extension' introns do not share endpoints with others. The ZXD Family Zinc Finger C (Zxdc) gene illustrates this example with its 3' most terminal introns. The GENCODE annotation for this gene lists five isoforms, of which two can be eliminated based on the fact that their unique introns do not appear in any of the 44 samples. Of the remaining isoforms, two have their intron chains entirely subsumed by the longest isoform. In Figure 3-4D, the distribution and average fold change abundance differs significantly between the shared (average 1.03) and isoform specific (average 1.45) intron sets, which can only be explained by a difference in the proportion of splice isoforms in the gene's output. Lastly, further case analyses revealed other intriguing scenarios, such as at the Zfp91-Cntf gene locus (Figure 3-4C). The two genes have in common the only intron in the Ciliary Neurotrophic Factor (Cntf) gene (chr19:12.764.380-12,765,281), which shows a significant six-fold increase in abundance in the epileptic mice, whereas all other introns for Zfp91 show a slight decrease within statistical error. While the event can be at first sight attributed to the differential splicing of Zfp91, careful observation of the expressed introns reveals that the sole Zfp91 isoform containing the intron is present at residual levels or not at all in both conditions. Therefore, the increase in abundance appears to be due to the change in the expression of Cntf, which owing to the special sharing of gene structure was missed by DESeq2. Cntf is a survival factor for multiple neuronal cell types, and an increase in its levels was shown to be involved in attenuating epilepsy-related brain damage [56, 57].

True accuracy cannot be assessed in analyses on real data. However, to evaluate robustness and reproducibility in the tools' predictions as an alternative measure of performance 9, we divided and analyzed the data into two sets of 10 healthy and 12 epileptic mouse samples. The graphs in Figure 3-5 show the scatterplots of the estimated difference in percent splicing inclusion (dPSI) between the two replicated experiments. MntJULiP has the highest correlation between the runs (0.579), followed closely by MAJIQ (0.577) and LeafCutter (0.460), and therefore its results are the most robust with the sample set.



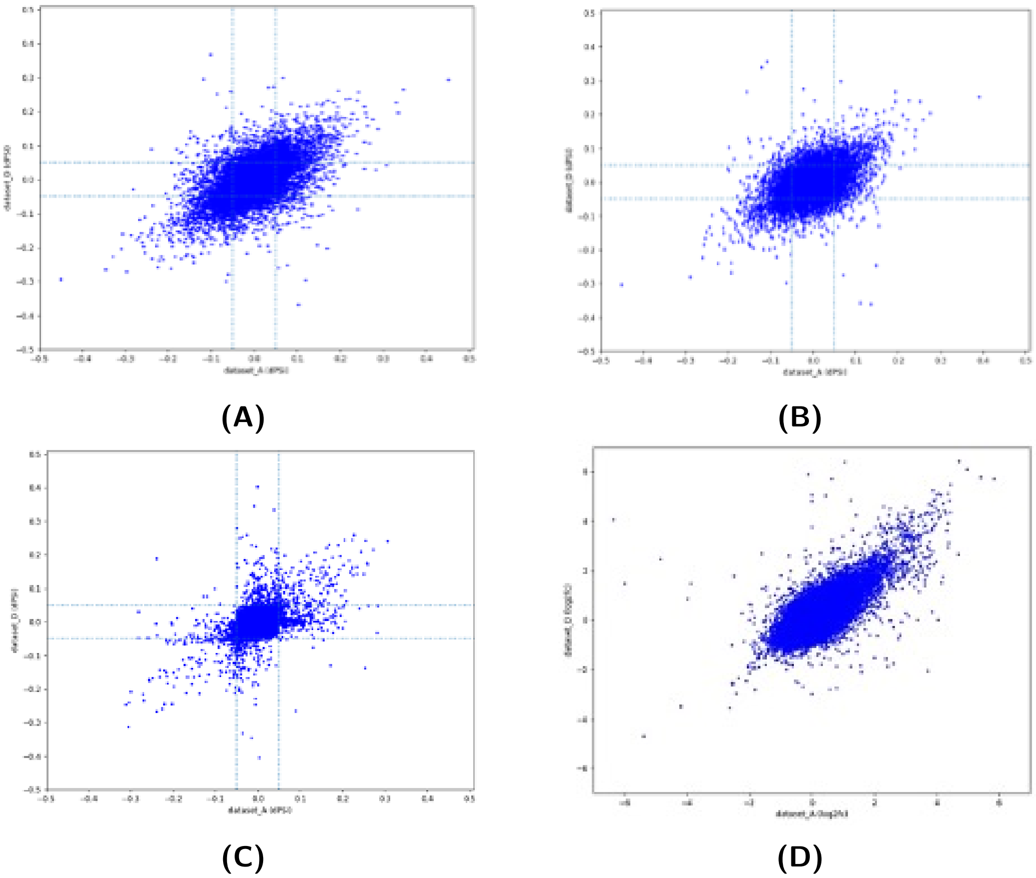
(A)

(B)

(C)

(D)

Figure 3-4. Examples of MntJULiP DSA predictions not identified by other tools (mouse hippocampus data set), at the (A) Pkm, (B) Cwc22, (C) Cntf-Zfp91 and (D) Zxdc gene loci. At Zxdc, introns are annotated with the fold change values in the comparison of healthy and epileptic mice. Note that the genes in (B), (C) and (D) do not have any endpoint sharing introns, and therefore are not identifiable by any of the other tools.



(A)

(B)

(C)

(D)

Figure 3-5. Reproducibility plots for MntJULiP, LeafCutter and MAJIQ (DSR test) and MntJULiP (DSA test) on the mouse hippocampus data (A-D). Mouse hippocampus samples were divided randomly into two sets of 10x12 samples (healthy versus epileptic) each, and the per intron dPSI values (log2fc) predicted by each program are plotted between the two comparisons. Number of introns represented: 16,607 for MntJULiP (DSR), 30,738 for LeafCutter, 93,642 for MAJIQ, and 132,289 for MntJULiP (DSA). Correlation coefficients for the 4 comparisons are 0.579 for MntJULiP (DSR), 0.460 for LeafCutter, 0.577 for MAJIQ, and 0.665 for MntJULiP (DSA).

3.2.2.3 Performance on large data sets

To demonstrate the scalability of MntJULiP and its unique capability to perform simultaneous multi-way comparisons, we applied it to four tissue datasets (frontal cortex, cortex, cerebellum, and lung; 554 samples total) extracted from the GTEx RNA-seq collection. We performed pairwise comparisons as well as three-way comparisons among tissues. In a first experiment comparing the three brain tissues, the multi-way comparison largely recapitulated the individual pairwise comparisons, detecting 99.0%

(1,070) of the 1,081 genes and 11 additional genes (Figure 3-6A). The test also revealed highly similar splicing profiles between cortex and frontal cortex, with only one gene differentiating the samples. The robustness of the method was confirmed in a second test, comparing the cortex, cerebellum and lung samples (Figure 3-6C). All but 14, 18 and 21 of the genes reported from the three pairwise comparisons were selected by the multi-way test, and 37 genes were unique to the three-way comparison, for a 99.3% (5,324 out of 5,364 predicted genes) recovery rate. Figure 3-7 shows the heatmaps of PSI values, reiterating these observations. Similar results can be observed for the DSA test, where the multi-way comparison discovered 97.1% (15,090 out of 15,491) of all genes detected by pairwise comparisons, and only 36 (0.02%) unique genes among the 15,126 predicted (Figure 3-6D). Importantly, the comparisons highlighted thousands of differential splicing events that distinguish among the tissues [58]. Experiments took between 18-44 minutes per comparison on a 24 CPU Intel processor, thereby demonstrating the ability of MntJULiP to handle large-scale applications.

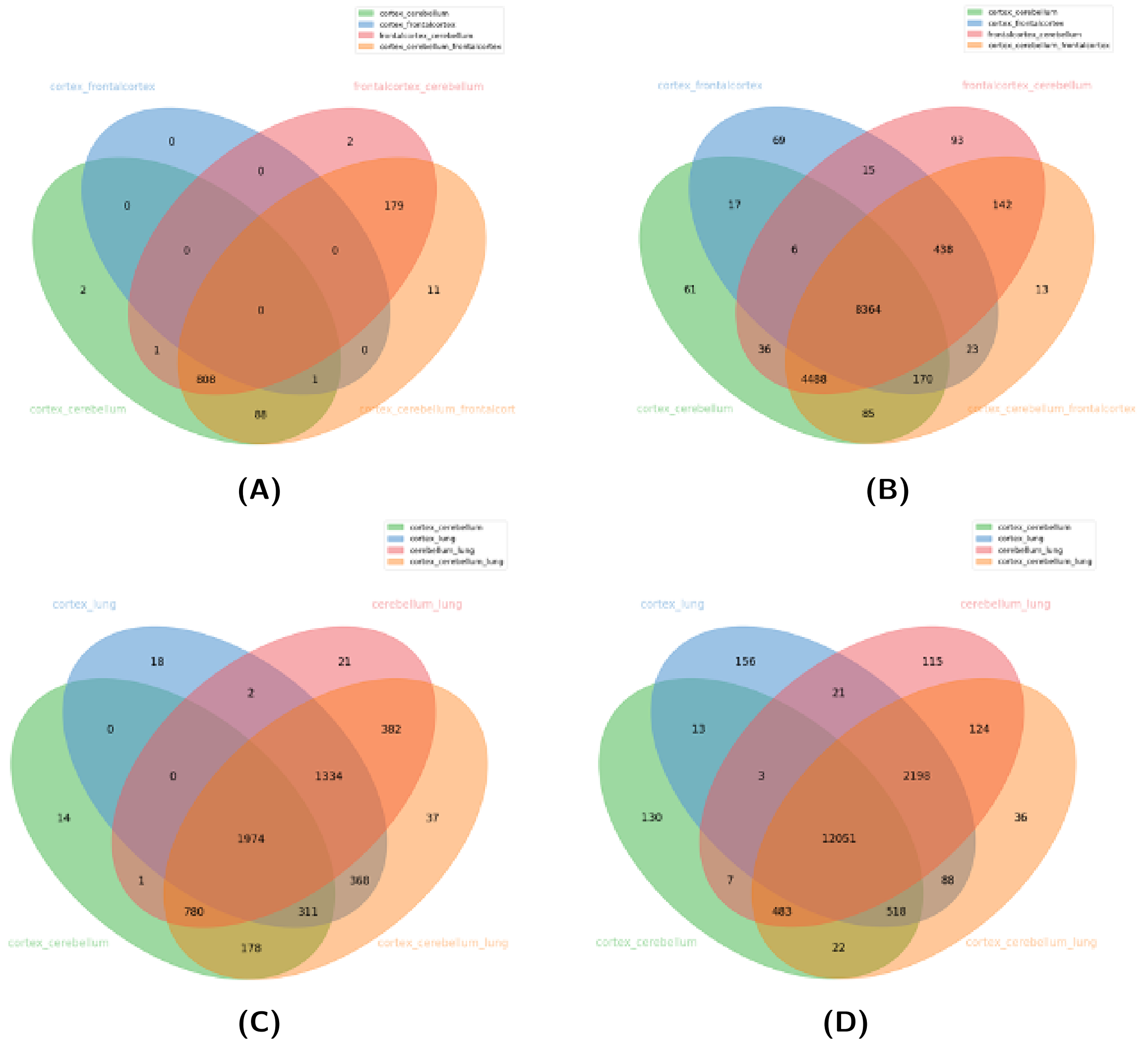


Figure 3-6. Multi-way versus all-against-all pairwise comparisons on GTEx tissue samples - gene sets. (A-B) Venn diagram of gene sets predicted by MntJULiP DSR (DSA) in comparisons of frontal cortex (105 samples), cortex (121 samples) and cerebellum (132 samples) RNA-seq collections. (C-D) Venn diagram of gene sets predicted by MntJULiP DSR (DSA) in comparisons of cortex (121 samples), cerebellum (132 samples) and lung (196 samples) RNA-seq collections (A p-value cutoff of 0.05 was used for all comparisons).

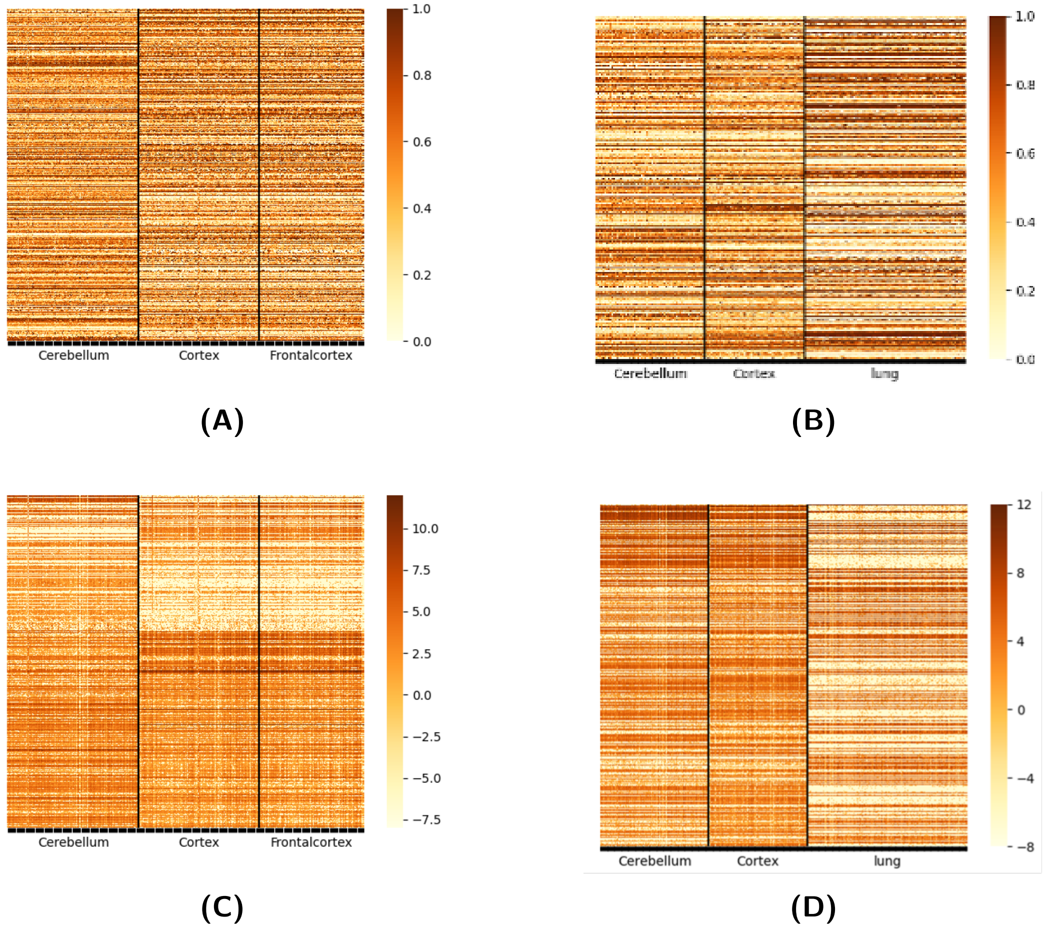


Figure 3-7. Heatmaps of differentially spliced introns from multi-way tissue comparisons. (A) DSR, frontal cortex-cortex-cerebellum comparison. (B) DSR, cortex-cerebellum-lung comparison. (C) DSA, frontal cortex-cortex-cerebellum comparison. (D) DSA, cortex-cerebellum-lung comparison. Note: Introns from genes with > 30 reads across all samples, with dPSI ≥ 0.2 (for DSR) and q-value < 0.05, were plotted. For DSA, additionally, only the intron with the largest log 2 fold change was chosen to represent the gene.

3.2.2.4 Application to complex and time-series experiments

All differential splicing methods to date are designed for comparing two conditions, typically 'cases' versus 'controls'. This simple framework is inadequate and impractical for scenarios that involve time-series or complex multi-condition experiments, which seek to determine features that vary across the full range of conditions. As an illustration, we applied both LeafCutter and MntJULiP to RNA sequencing data from mouse taste organoids) at seven growth stages [54] (Accession: DRA005238; two samples each at days 2D, 4D, 6D, 8D, 10D, 12D and 14D, for a total of 14 samples). LeafCutter predicted DSR events in 889 genes and MntJULiP in 3,285 genes when combining the results from all-against-all pairwise analyses. By comparison, MntJULiP's multi-way test predicted 204 differentially spliced genes across all conditions. While true accuracy cannot be measured, we deem features (genes) reported by multiple comparisons to have higher confidence than those predicted in a single comparison, on the basis that features that are differentiated between two stages will likely show variation in other comparisons involving one of the original conditions.

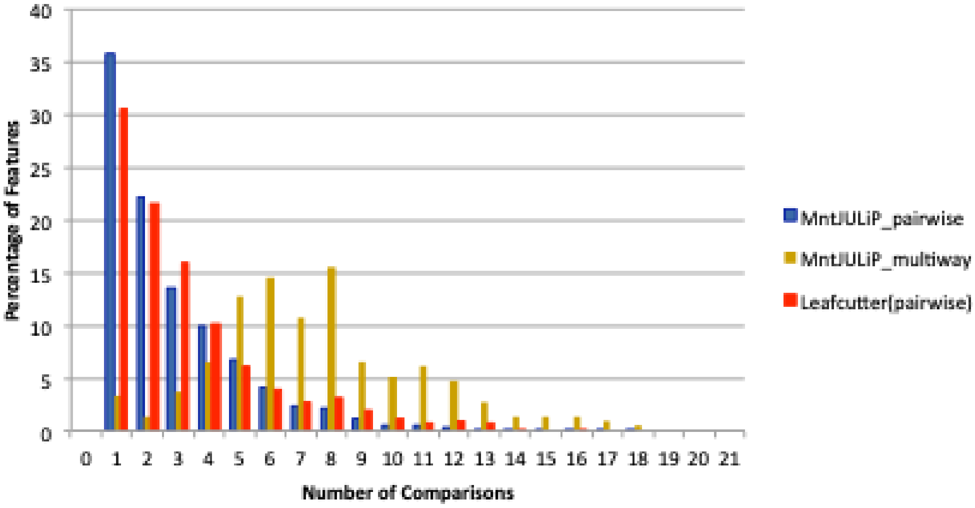


Figure 3-8. Assessment of program predicted features for the time-series taste organoid data set, based on the assumption of continuity of feature space. The distribution of program-predicted features by number of comparisons is shown for three methods: i) union of MntJULiP predicted features from all (21 total) pairwise comparisons, ii) MntJULiP multi-way predicted features, and iii) union of LeafCutter predicted features, from all pairwise comparisons (21 total). For MntJULiP multi-way predictions, features were traced back to the pairwise comparisons in which they were reported (from i).

As Figure 3-8 indicates, the distribution of genes according to the number of comparisons in which they are reported is very similar for the LeafCutter and MntJULiP pairwise protocols, with 31-36% of the genes found in only one comparison, pointing to potentially large numbers of false positives. In contrast, the distribution for MntJULiP multi-way predicted genes follows a Bell curve distribution with the mode at 8 comparisons, which provides a more realistic reflection of the experiment. Therefore, the multi-way comparison more accurately identified differences in splicing across the experimental range.

To further examine the landscape of alternative splicing variation during organoid differentiation, we generated heatmaps of the introns discovered with the MntJULiP all-pairwise and the MntJULiP multi-way comparison methods (Figure 3-9). Introns' PSI values show small variation in splicing between consecutive stages, but clear distinguishing characteristics when comparing across all experimental timepoints. In particular, features detected by the multi-way comparison clearly distinguish between the organoid growth stages, with a significant inflexion point between early (days 2D-6D) and late development and differentiation into taste cells (days 8D-14D), and facilitate more accurate clustering of samples. Interestingly, the visualizations point to distinguishing features separating stage 2D from the other non-differentiated stages, and the separation becomes even more apparent in the DSA visualizations (Figure 3-9B). The two visualizations provide complementary and overlapping views of the changes in the global transcriptome, with DSA reflecting changes in the expression level of features, and DSR reflecting changes in the relative contribution of isoforms. We note here that, from a technical standpoint, the DSR and DSA maps reflect different data types and characteristics; while the PSI values in the DSR maps are restricted to the [0,1] interval and are more limited in their variability, the feature expression levels in the DSA displays potentially range over 4-5 orders of magnitude (4-5 fold in logistic conversion) and exhibit over-dispersion, and therefore may reflect more clearly the changes in the global transcriptome, and be better suited to highlighting more subtle changes. (These observations have led us to develop the comprehensive and flexible visualization tools in section 3.3.) Importantly, these graphical representations highlight the ability of MntJULiP to detect even mild differences between conditions. We also note the ability of MntJULiP to work with very small numbers of samples per condition, as low as two samples per organoid stage.

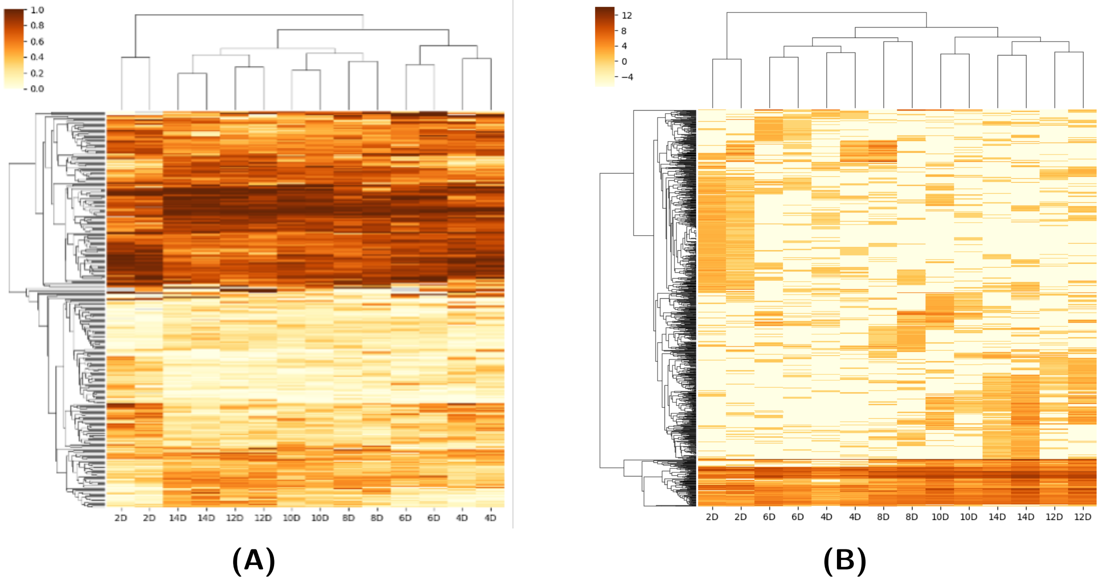


Figure 3-9. Heatmaps of differentially spliced features (introns) in the taste organoid data set. (A) MntJULiP DSR, features discovered via multi-way comparison; (B) MntJULiP DSA-predicted features. Heatmaps show PSI (A) or expression log fold change (B) values. Features were filtered at p-value<0.05 and dPSI>=0.2 (for DSR). Grouping was performed using weighted hierarchical clustering with the Bray-Curtis metric.

3.3 Jutils: A visualization toolkit for differential alternative splicing events

While multiple methods have been developed to determine differential splicing patterns from RNA-seq data (LeafCutter, MAJIQ, rMATS, MntJULiP [33, 34, 38, 59], there is a scarcity of tools to present the results to the user in a way that is intuitive and easy to explore. Moreover, most visualization tools are designed for a particular differential splicing method, such as rmats2sashimiplot (https://github.com/Xinglab/rmats2sashimiplot) and LeafViz (https://leafcutter.shinyapps.io/leafviz/), and are not adapted for general use. To fill this gap, we developed Jutils, a toolkit for visualizing alternative splicing differences that can be used across methods.

3.3.1 System design

Jutils works with the output of a differential splicing tool, converting it into a unified data file that contains the information necessary for the visualizations. (Additional information, such as the BAM files, can be optionally provided.) Metadata about experiment design, such as the condition associated with each sample, can be provided in a specification file. Jutils then extracts events to include in the visualizations based on user specified criteria. Lastly, it generates one of three types of visualizations: heatmap, sashimi plot, and Venn diagram. Details of each component are provided below.

3.3.1.1 The unified file format

Jutils uses an intermediate Tab Separated Values (TSV) file format to collect event information generated by a differential splicing program, which it then uses to create visualizations. Jutils has built-in output conversion modules for several analysis programs, including LeafCutter, MAJIQ, MntJULiP, and rMATS, and users can develop their own conversion scripts for other programs of interest.

The TSV file has 14 columns containing, in order: gene name, group id, feature id, feature type (e.g., intron, event skipping), feature label (derived from the chromosomal location), strand, p-value, q-value, dPSI (difference in Percent Splice In values), read count 1, read count 2, and PSI values per sample, estimated PSI per condition. The default feature for Jutils is introns, but the program can represent more complex events such as those reported by rMATS, for instance, exon inclusion and exon exclusion constituted as a single exon-skipping event. Programs may further aggregate features into groups, for instance LeafCutter and MntJULiP group introns that share a splice junction. Jutils supports identifiers and operations on individual features as well as groups. Read count 1, read count 2, and PSI represent vectors of per sample values. Read counts 1 and 2 correspond to the paired splice forms in a complex feature (e.g., exon skipping), whereas for simple features (e.g., introns) read count 2 is marked with ‘’..

3.3.1.2 Heatmaps

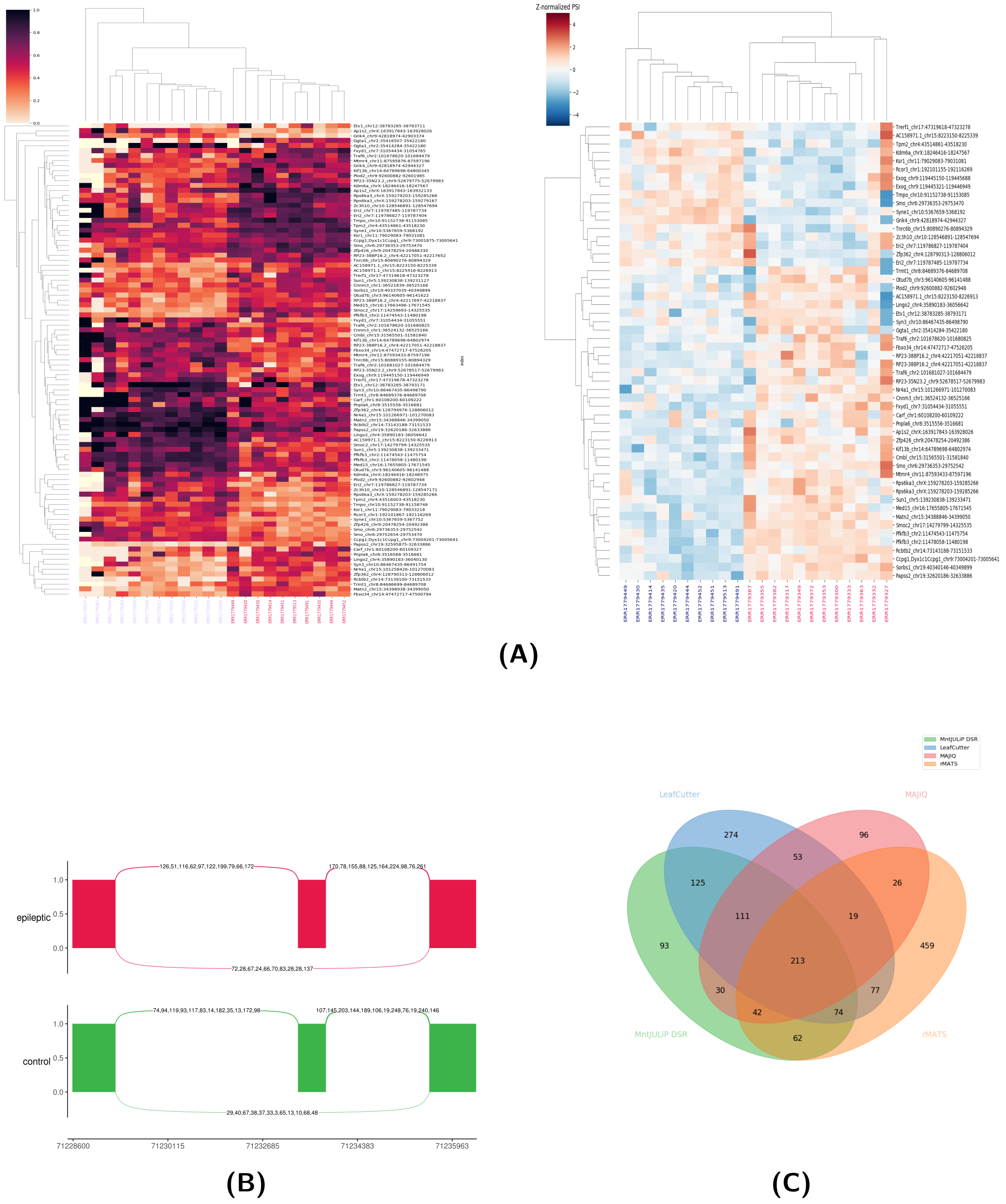
Jutils generates heatmaps of differential splicing events represented in the TSV file. A metadata file contains the classification of each sample. Jutils generates heatmaps of PSI values, either Z-score normalized or absolute values (Figure 3-10A). The software allows clustering by rows (events) and columns (samples), using different distance metrics and clustering methods. By default, the ‘cityblock’ (Manhattan distance) metric with the ‘weighted’; (weighted pair group with arithmetic mean) method is used. Events can be filtered at run time based on quality and confidence measures such as p-value, q-value and dPSI, and the user may choose to visualize all relevant features or select a representative feature per group or per gene. Lastly, while Jutils is intended to work primarily with the output of differential splicing tools, it can also be used to display the features with the highest variance (option ‘–unsupervised’).

3.3.1.3 Sashimi plots

Sashimi plots have been previously introduced to visually represent differences in splicing. A traditional sashimi plot shows raw RNA-seq densities along with exons and junctions for multiple samples. The Jutils sashimi visualization utilizes a modified version of the ggsashimi package [60] to display graphical representations of intron read counts within a specified genomic region, intron, or intron group (Figure 3-10B). By default, Jutils provides a lightweight repre-sentation based solely on the intron read counts provided in the TSV file, without the flanking exon read depth information. When alignment files are also provided, Jutils extracts alignments from the BAM files and provides full sashimi representations reflecting the accurate exonic coverage.

3.3.1.4 Venn diagrams

Methods for differential splicing detection employ a variety of models for features and objective functions. Therefore, it becomes desirable to compare the outputs of different programs to obtain a complete view of the predicted differential splicing and to gauge support from multiple tools. Jutils provides a Venn diagram visualization of the gene sets predicted by multiple programs (Figure 3-10C), along with a text file containing the list of genes in each category.



(A)

(B)

(C)

Figure 3-10. Jutils visualization of differential splicing events from the comparison of hippocampus samples of 12 healthy and 10 epileptic mice (GenBank ProjectID: PRJEB18790). (A) Heatmaps of absolute (left) and Z-score normalized (right) PSI values generated with MntJULiP shown, respectively, at the intron and group level. Left: darker red indicates PSI values closer to 1; Right: blue colors mark values lower than the row average, and red ones values higher than the row average. (B) Sashimi plot of events at the Dync1i2 gene, predicted by LeafCutter. (C) Venn diagram of gene predictions from four analysis methods.

3.4 Discussion

Detecting differences in splicing patterns between cellular conditions is a critical task. Multiple methods have been developed that differ in their target features, objective function, and employed techniques, which makes selecting a tool and comparing the results across methods daunting [50]. To answer the need for highly accurate methods combined with a systematic approach to problem definition, we developed MntJULiP. MntJULiP provides a comprehensive view of alternative splicing variation, by representing it at the most granular level (intron) and by implementing two objective functions, aimed at determining differences in the absolute and relative (ratios) intron splicing levels. In comparisons on simulated and real data, we demonstrated that MntJULiP identifies more alternative splicing variation and more classes of variation than other tools, and across a spectrum of experimental conditions, dataset sizes and degrees of variation.

Traditionally, differential analyses have targeted gene expression changes that could lead to the discovery of causative or marker genes for diseases or other cellular conditions. The advent of deep RNA sequencing has made it possible to uncover finer grained changes in the gene's output, at the level of the transcripts that it expresses. Both changes in the expression level of a given transcript, often referred to as isoform specific regulation, and changes in the relative proportion of splice isoforms of a gene, lead to biological and phenotypic changes. Therefore, it is important to design tools that address each of these problems. The differential splicing abundance (DSA) and differential splicing ratio (DSR) define the two problems in computational terms, and methods for differential splicing detection have adopted either of these models. However, there has been no software that implements them both, in a unified framework, to allow comprehensive discovery and the unbiased comparison of their results. Our DSA and DSR models, implemented in MntJULiP using a unified bayesian architecture and feature selection filters, are individually powerful, and taken together provide a comprehensive view of transcriptomic variation.

MntJULiP introduces several technical innovations, including its zero-inflated negative binomial and multinomial Dirichlet models to account for low count genes and splice junctions, and the mixture distributions that allow for modeling multiple conditions, thus facilitating multi-way differential analyses. The ability to perform multi-way comparisons, in both its DSA and DSR modes, is unique to MntJULiP, and is desirable when characterizing complex time series or multi-condition experiments, to identify a global set of features that distinguish among subgroups or stages.

MntJULiP also has limitations. While it benefits from being able to extract the splicing event information directly from the RNA-seq read alignments, without the need for reference gene annotations, MntJULiP relies on the splice junction information gathered from the sequence alignments. Therefore, the performance of MntJULiP largely depends on the reliability of the alignments. Spliced alignment is a highly complex computational problem, with heuristic solutions that may not accurately capture all biological patterns, for instance non-canonical splice sites or introns flanking very short exons. Further, sequencing errors as well as genuine differences between the RNA-seq sample and the reference genome can lead to difficulty in mapping to the correct location, if multiple potential matches exist. Furthermore, aligners could introduce biases to the alignments. For example, sequence-specific bias due to GC-content and dinucleotide frequencies and motif content in hexamer primer regions could reduce the ability of MntJULiP and other alternative splicing analysis tools to accurately quantify the events. Moreover, the existing aligners that utilize a standard reference genome as a template sometimes have difficulty in mapping reads that carry rare genomic variants, which can lead to allelic ratio biases and could hamper MntJULiP's ability to accurately assess the differences in splicing patterns.

Lastly, to complement our effort with designing differential splicing tools, and to meet the needs for intuitive and easy to use visual representations, we also developed Jutils, to make the results accessible and easy to interpret by the users. Jutils is a lightweight toolkit for visualizing differential alternative splicing between cellular conditions. It can be used automatically with the popular differential splicing analysis tools LeafCutter, MAJIQ, MntJULiP, and rMATS, and can be easily adapted to any other program, and thus represents a useful and practical tool to explore the landscape of alternative splicing.

Overall, our tools are highly efficient and scalable, providing an effective platform for comprehensive differential splicing analyses of RNA sequencing data from a wide range of experiments and data collections.

Chapter 4

A Deep Learning (DL) splicing model

As we discussed in the previous Chapters, alternative splicing of pre-mRNA represents an important regulatory mechanism that contributes to protein diversity in the cellular environment. However, its mechanisms are still incompletely understood. While a plethora of regulatory *cis*- and *trans*-factors are known to interact to determine the splice site selection and alternative splicing outcome, the mechanisms remain to be revealed. Splicing signals can be very complex, and most alternative splicing outcomes involve the competition among candidate splice sites. Therefore, splicing patterns can be controlled by any mechanism that alters the relative rates of splice site recognition.

In this chapter, we focus on the RNA sequences around the splice sites, and build a computational model aimed at revealing the splicing regulation. More specifically, we describe a probabilistic deep learning model to predict and quantify alternative splicing events from the information on *cis*-regulatory sequence elements and *trans*-splicing factors in different tissues.

4.1 Background

4.1.1 Regulation of alternative splicing

In the pre-mRNA splicing process, relatively small exons (∼100 nt) are generally ligated while much longer introns (>1000 nt) are excised. Pre-mRNA splicing is carried out by spliceosome, a large complex consisting of several small nuclear ribonucleoproteins (snRNPs) and auxiliary protein factors. The spliceosome removes introns from a transcribed pre-mRNA, a type of primary transcript. A spliceosome is either recruited or assembled at the correct 5' splice site (donor) and 3' splice site (acceptor), in part through recognition of conserved sequences spanning the intron-exon junctions [61]. A canonical splice site contains the dinucleotides 'GU' and 'AG' at the 5' end and 3' end of an intron, respectively, and a branch point containing a conserved 'A' upstream of the 3' end in the intron. The snRNPs recognize these splicing signals, bind to the pre-mRNA sequence and help assemble the spliceosome. The splice site signals are typically only a few nucleotides long and very common in pre-mRNA sequences, resulting in a large number of candidate splice sites. In reality, however, only a handful of splice sites are recognized by the spliceosome, and the splicing process is conservative and highly regulated, especially for the protein coding genes. These suggest the existence of a global and local regulated machinery that precisely controls the recognition of splice sites.

While splice site consensus sequences are necessary for splicing, starting with the formation of the spliceosome, they are not sufficient in many eukaryotic systems. Changes in splice site choice (alternative splicing) arise from combinatorial interactions among *cis*- and *trans*-regulatory factors mediated by the spliceosome. Outside of the core splice signals, the bulk of the information required for splicing is thought to be contained in exonic and intronic *cis*-regulatory elements. Those *cis*-regulatory elements function by recruiting sequence-specific RNA-binding proteins that either activate or repress the use of adjacent splice sites [2]. The RNA sequences that act positively to stimulate spliceosome assembly are called splicing enhancers. Conversely, the RNA sequences that act negatively to block the spliceosome assembly and certain splicing choices are called splicing silencers or repressors.

Besides their mode of action, the location of *cis*-regulatory sequences is also important. Exons often contain enhancer or silencer elements that affect their ability to be spliced. There are many exonic splicing enhancers (ESEs) and most of them interact with a family of splicing regulatory proteins known as SR proteins. The SR proteins form a group of highly conserved proteins that are required for constitutive splicing and also influence alternative splicing regulation. SR proteins bound to ESEs can promote U2AF recruitment to the polypyrimidine tract and activate an adjacent 3' splice site[62–64] (Figure 4-1). Moreover, other non-SR proteins may also interact with the ESEs to regulate splicing. In one example, proteins YB-1 and p72 are found to mediate an AC-rich splicing enhancer at the human CD44 gene [65, 66]. To contrast the positive action of exonic splicing enhancers, exonic splicing silencers (ESSs) have also been identified. The best characterized of these are bound by the hnRNP proteins [67]. As an example, the hnRNP A1 protein can bind to an ESS to create a zone of the RNA where spliceosome assembly is repressed (Figure 4-2A).

Many splicing regulatory sequences are present in introns rather than exons. Binding sites for regulators are often found within the polypyrimidine tract or immediately adjacent to the branch point or the 5' splice site. Similarly to exonic regulation, positive- and negative-acting sequences make up intronic splicing enhancers and silencers, respectively (ISEs and ISSs). In the example in Figure 4-2B, the hnRNP A1 protein binds to an ISS adjacent to the branch point to block U2 snRNP binding to the branch point. Further, the uridine-rich sequence immediately downstream of the 5' splice sites was found to bind the protein TIA-1 (Figure 4-1), and TIA-1 stimulates U1 snRNP binding to 5' splice sites [68].

Therefore, alternative splicing is regulated by the complex interactions of *cis*-acting regulatory sequences, which can act to enhance or suppress splicing, and the *trans*-acting proteins that bind to them.

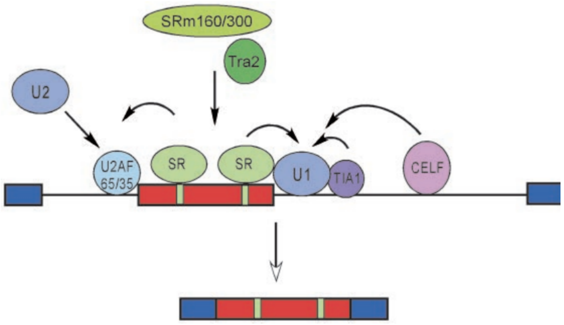


Figure 4-1. Mechanisms of splicing activation with SR proteins as splicing activators (image from [4]).

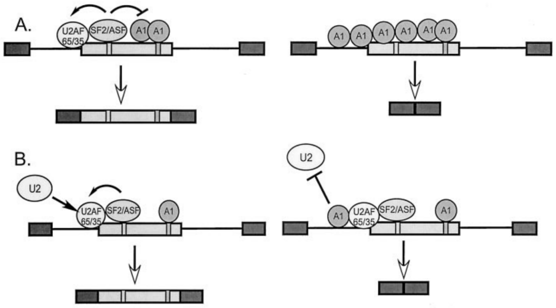


Figure 4-2. Models for splicing repression by hnRNP A1 (image from [4]).

4.1.2 RNA binding proteins

RNA binding proteins (RBPs) are a specialized type of proteins that can bind to RNA sequences and potentially play a role in the regulation of gene expression. Each alternative splicing event is controlled by multiple RBPs, the combined action of which creates a distribution of alternatively spliced products in a given cell type. Therefore, the interpretation of regulatory information on a given RNA target is exceedingly dependent on the cell type. Current estimates are that more than 1,500 proteins have the capacity to bind the human RNA sequences, and as many as 690 proteins are mRNA-binding [69]. Some RBPs are key regulators of post-transcriptional regulation. The functions of RBPs are largely dependent on the binding location, either activating or repressing the splice site choice.

As an example of RBPs, SR proteins are a conserved family of proteins involved in RNA splicing. An SR protein has one or several RNA recognition domains (RRM domains) to bind the RNA sequence, and an arginine and serine residues domain (SR domains) to interact with other proteins. SR proteins promote the binding of U1 snRNP, and U2AF auxiliary factor, and stimulate the formation of the spliceosome. Therefore, SR proteins contribute to splice site recognition by directing the splicing machinery to different splice sites under different circumstances.

Another class of important splicing factors is that of hnRNP proteins. hnRNP proteins are less conserved compared to the SR proteins. hnRNP proteins contain the RRM domains but lack an RS domain. hnRNP proteins can bind to RNA sequences but are unable to interact and recruit the snRNPs. Therefore, they play a role in repressing splicing by directly antagonizing the recognition of splice sites, blocking the SR proteins' binding to splicing enhancers, or hindering communication between splicing factors.

The major families of RBPs contain canonical RNA-binding protein domains, such as the RNA-recognition motif (RRM), CCCH zing finger, K homology (KH) and cold shock domain (CSD). Recently developed technologies such as crosslinking of RNA to proteins followed by sequencing (CLIP-seq) [70], SELEX [71], RNAcompete [72], RNA Bind-n-Seq (RNBS) [73] have been used to determine the binding locations and specificities of a growing number of RBPs [74–76].

4.1.3 Sequence motifs

Deciphering the mechanisms behind splice site choice requires a comprehensive list of splicing regulatory RNA-binding proteins (RBPs) and their *cis*-acting binding sites. Recent technologies mentioned above (CLIP-seq, RNAcompete, RBNS) can help identify the binding preferences of RBP, *in vitro* or *in vivo*.

To represent RBP binding affinities to an RNA sequence computationally, so called Position Weight Matrices (PWMs), as designed for transcription factor binding sites, have been adopted [77]. A typical PWM is a 4 × W matrix in which position (j, w) gives the probability of observing the nucleotide j at position w of a motif of length W. PWMs are most often derived from collections of known binding sites for a given protein, or through computational methods applying pattern discovery algorithms to functional genomics data.

4.1.4 Other splicing regulatory mechanisms

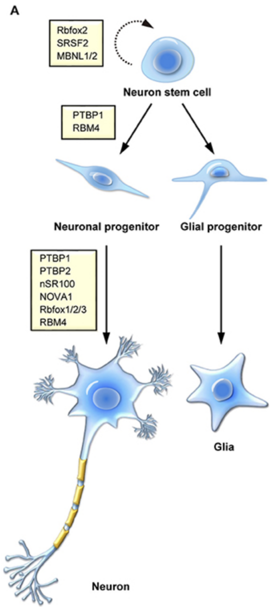
SR proteins and hnRNP proteins can explain a large variety of splicing decisions, but the control mechanisms of alternative splicing are diverse and complex. In one scenario, binding of SR proteins to the ESEs can antagonize the activity of hnRNP proteins recognizing ESS elements [78]. Therefore, the relative abundance of SR proteins and hnRNP proteins could be important in regulating the patterns of alternative splicing in a tissue-specific or developmentally regulated manner [79].

Although a single critical factor (the binding proteins) has not been shown to determine the tissue specificity of splicing in any system, the expression of some splicing regulatory proteins is restricted to certain cells. NOVA-1, a neuron-specific RNA binding protein regulates neuron-specific alternative splicing (Figure 4-3A) and is essential for neuronal viability [80]. CELF proteins bound to muscle-specific enhancers (MSE) in the cardiac troponin-T gene (cTNT) can promote inclusion of the developmentally regulated exon [81].

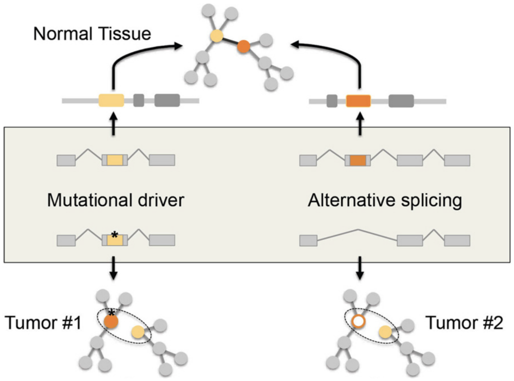
Different binding regions and concentration of a factor may also alter its functional activity. SR proteins bound to an intronic sequence near a branch point would block the use of this 3' splice site and shift splicing to an adjacent site [82]. Excess of hnRNP A/B proteins has the opposite effect, promoting the selection of intron-distal 5' splice sites.

4.1.5 Mutations affect alternative splicing

Point mutations, such as base substitutions, in genes may alter the target sequences by changing a codon for one amino acid into one coding for another or into a premature termination codon (PTC). Additionally, point mutations could change the sequence around the splice site, and therefore have a more subtle influence on alternative splicing. Synonymous single-nucleotide polymorphisms (SNPs) located in coding regions can disrupt (or create) exonic splicing enhancers and silencers [83, 84]. The mutations located in non-coding regions, such as those affecting 5' and 3' splice sites, branch sites or polyadenylation signals, can also alter the splicing pattern [85, 86]. Other types of mutations, for example, nonsense and missense mutations as well as exonic deletions or insertions, can affect alternative splicing in similar ways [87]. Mutations, as a consequence, can lead to the appearance of truncated proteins or to the lack of the correct gene product, and are frequently the cause of hereditary disease (Figure 4-3B). However, the incomplete understanding of the mechanisms for splice site choice hampers our ability to accurately predict the effects of mutations and to identify splice altering variants around the splice sites.



(A) Splicing regulatory proteins regulate neuronal cell differentiation through alternative splicing (image from [55]).



(B) Mutations lead to abnormal splicing patterns and result in tumor proteins.

4.1.6 The ‘splicing code’

To understand the complexity of pre-mRNA splicing, we must study how changes in splice site choice come about. For several decades now, the molecular components have been characterized and evidence for protein-nucleic acid interactions have been accumulating. A long list of splicing regulatory *cis*-elements have already been identified, and their associated *trans*-acting factors determined. Hence, it has become possible to assemble the available information into a computational framework to predict the splicing patterns of different cells and developmental stages. A long-term goal in the area of alternative splicing is to determine a set of rules (or *code*) for splicing [88]. Such a code would also be instrumental in predicting the consequence of mutations on splicing. It would be of great value not only to molecular biologists and geneticists, to enable better understanding of splicing events and the effect of mutations on mRNA splicing, but also to clinical researchers, to design new therapeutic approaches based on splicing interference [89].

Efforts to understand the splicing code have previously entailed statistical or machine learning methods that extensively integrate combinations of diverse RNA features. Barash *et al.* [90] adopted hundreds of RNA features to predict tissue-dependent changes in alternative splicing for thousands of exons. Zhang *et al.* [91] used Bayesian networks to probabilistically model diverse datasets and predict the target networks of specific regulators. With the advance of next-generation sequencing technologies, new tools have been designed to predict the splicing pattern largely based on the DNA/pre-mRNA sequences. Xiong *et al.* [92] and Zijun *et al.* [93] used deep learning to derive a computational model that takes DNA sequences as input and applies general rules to predict splicing in human tissues. Given a variant, Xiong’s model computes a score that predicts how much the variant disrupts splicing. Jaganathan *et al.* [94] developed a deep residual neural network (spliceAI) to predict cryptic splice sites and the effects of mutations on splice site usage, using pre-mRNA sequences as inputs. Unlike other computational models, this model works on long-range primary genomic sequences (>5k bp), and therefore can include multiple candidate splice sites and nonfunctional regions into consideration.

4.2 A Deep Learning (DL) splicing model

In this Chapter, we describe a novel probabilistic deep learning splicing model, using as input sequence elements and trained on intron-supporting read counts extracted from RNA-seq data. The model predicts intron splicing ratios from *cis*-acting sequence elements around the splice sites and the *trans*-acting splicing factors, represented by learned motifs.

4.2.1 Model design

Our deep learning splicing model consists of a sequence of individual networks that learn increasingly deeper hidden relationships among the data. The input to the model are sequences surrounding the splice sites of introns along with tissue information, in combination with splicing ratio derived from RNA-seq data using MntJULiP, and the output is, for a given intron, its splicing ratio in a specified tissue.

4.2.1.1 The learning model

Over-dispersion is a known characteristic of RNA-seq data and needs to be accounted for in modeling count data. To model over-dispersion, similarly to our approach implemented in MntJULiP, we use a Dirichlet-Multinomial distribution. Let $x = (x_1, x_2, \ldots, x_K)$ be the counts of sequencing reads mapped to the K alternative splice junctions (introns) in a 'bunch' that shares a splice site. Then, we assume x follows a Dirichlet-Multinomial distribution with unknown parameter $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$:

$$x_1, x_2, \ldots, x_K \sim Dirichlet - Multinomial(\alpha_1, \alpha_2, \ldots, \alpha_K)$$

Unlike MntJULiP, which estimates the parameter α for each individual ‘bunch’ of introns using a Bayesian model, here we design a neural network that takes as input of sequence elements, learns hidden relatoinships, and reports, as the output, the α for the Dirichlet-Multinomial distribution. More specifically, we propose a probabilistic deep learning model (PDL) to predict the introns’ splicing ratios given the sequence elements around the splice sites (Figure 4-4). The PDL model consists of several convolutional neural networks (CNNs) to extract features from the sequence elements around the splice sites, a fully connected layer to convert the feature representations to scores for different tissue types, and a Dirichlet-Multinomial layer that learns and predicts the splicing ratios as PSI values. Each individual network component is either a specifically designed convolutional neural network, a fully connected network, or a probabilistic module to extract information from its input. The details of each component are described below.

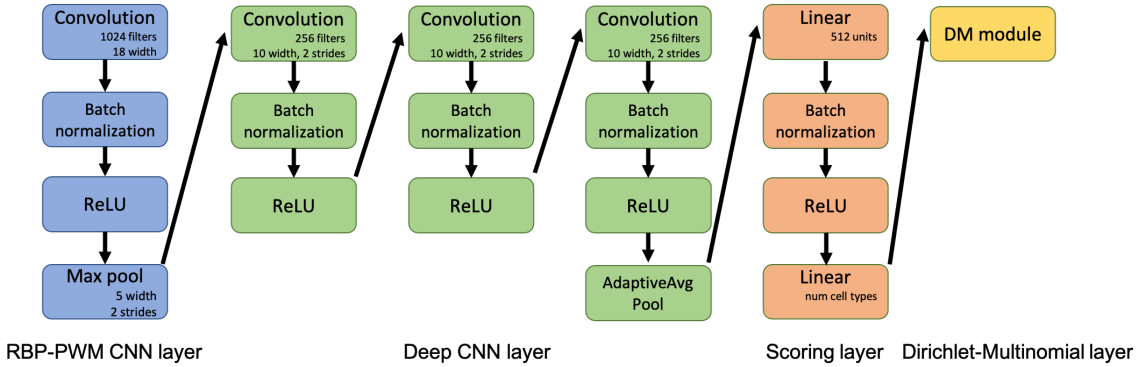


Figure 4-4. The architecture of the probabilistic deep learning splicing model.

4.2.1.2 The RBP-PWM CNN layer

A key functionality of convolutional neural network (CNN) is to filter and summarize latent features from its inputs. The CNN functions by multiplying the CNN weight matrix (called filter) with the input matrix to generate one value to summarize the input. This process is very similar to how a position weight matrix (PWM) function. For the first CNN layer, we optionally initialize the CNN filter by the RBP PWMs collected from the public RBP databases. Given the sequence inputs, the first CNN layer outputs the scores representing the binding affinity of RBPs to particular sequence positions.

4.2.1.3 The Deep CNN layer

Followed by the RBP-PWM CNN layer is a deep CNN layer. The purpose of this layer is to extract and combine the latent information from the RBP PWM score vectors, forming a deep representation of the sequence elements that can be used as a key latent feature for a splice junction (intron).

4.2.1.4 The Scoring layer

Further, we design a fully connected layer, taking as input the sequence representation, summarizing it, and generating scores to represent a splice junction. For example, for a model with 7 tissues, the model will generate 7 scores to represent a splice junction under the 7 different tissues.

4.2.1.5 The Dirichlet-Multinomial layer

The scores generated by the scoring layer can be used as the hyper parameter α for our Dirichlet-Multinomial distribution. In the training step, the Dirichlet-Multinomial layer receives the inputs of α, sampling the probabilities $p = p_1, p_2, \ldots, p_K$ that will be used in the loss function. Following training, the Dirichlet-Multinomial layer can predict the splicing ratio for each intron, and for each tissue.

4.2.1.6 The loss function

For training, the model requires two types of inputs: the first is the sequence elements $s = (s_1, s_2, \ldots, s_K)$ of the candidate splice junctions, and the second is the RNA sequence read counts $x = (x_1, x_2, \ldots, x_K)$ that mapped to the splice junctions. A log likelihood estimate is used to train the model. In detail, $f_{nn}(s; \theta)$ is the neural network (with parameters θ), receiving the sequence elements s and generating as output the parameter α of the Dirichlet-Multinomial distribution. The probability mass function of the K-categories Multinomial distribution of N trials (sum of read counts) is given by:

$$f_{MN}(x; N, p) = \frac{N!}{\prod_{k=1}^{K} x_k!} \prod_{k=1}^{K} p_k^{x_k},$$

Where the probabilities p follow a prior Dirichlet distribution, and the probability mass function is as follows:

$$f_{Dir}(p; \alpha) = \frac{\Gamma(A)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1},$$

Where $\Gamma(x)$ is the gamma function and $A = \sum_{i=1}^{K} \alpha_i$. Putting together the two functions, the probability mass function of the Dirichlet Multinomial distribution is derived:

$$f_{Dir}(x : N; \alpha) = \frac{N!}{\prod_{k=1}^{K} x_k!} \frac{\Gamma(A)}{\Gamma(A + N)} \prod_{k=1}^{K} \frac{\Gamma\left(\alpha_k + x_k\right)}{\Gamma\left(\alpha_k\right)}$$

Removing terms only containing N and x, and taking the logarithm, we have the loss function with respect to the parameters θ,

$$L(\theta; s, x) = log\Gamma(A) - log\Gamma(A + N) \sum_{k=1}^{K} (log\Gamma(f_{nn}(s_k; \theta) + x_k) - log\Gamma(f_{nn}(s_k; \theta))$$

After defining the loss function, the PDL model can be trained with the Adam optimization algorithm [95].

4.2.1.7 Input and Output

To allow investigating how the sequence elements and associated *trans*-splicing factors affect the selection of introns, the model implements three types of inputs. The first input is the RNA sequence (Figure 4-5). The RNA sequence around the splice site contains the exonic splicing enhancers (ESEs), exonic splicing silencers (ESSs), intronic splicing enhancers (ISEs), and intronic splicing silencers (ISSs), which are the binding motifs for RBPs. The second type of input consists of the splice junction read counts to represent the 'expression' level of an intron. The third type of input is the RBP RNA recognition motif (RRM), the RNA binding domain of the RBP. For this type of input, we downloaded position weight matrices (PWMs) of RBPs from online databases, such as the ATtRACT database [96]). Due to the structure similarity between PWMs and filters in the neural net, we can initialize and fix the neural filters in the RBP-PWM CNN layer by the downloaded PWMs (Figure 4-4). In this way, we incorporate the biological knowledge in the RNA binding proteins and their associated motifs into the PDL model, which would potentially improve the model performance.

To investigate the differential usage of the introns, we group introns into 'bunches' by their sharing splice sites, as described in Chapter 3. Given a set of introns that share either a 5' splice site or a 3' splice site, and $I_1, I_2, \ldots, I_n$ their (expected) read counts, we define the *splicing ratio* Ψ_j for intron j as: $\Psi_j = I_j / \sum_{k=1}^{n} I_k$. It is easy to see that $\Psi_j \in [0, 1]$, and can be viewed as a probability. See Figure 4-6 for a simple example of Ψ value.

To implement the model, we used an engineering solution, feeding into the PDL model batch inputs grouped by the intron 'bunches'. For example, assume we have 256 bunches, where each bunch contains 2 introns that share a splice site. We merge the one-hot encoding inputs (4×200 matrix) of the 2 introns, forming a $2 \times 4 \times 200$ tensor. We group those $2 \times 4 \times 200$ tensors into a batch, forming a $256 \times 2 \times 4 \times 200$ tensor that we feed into the PDL model.

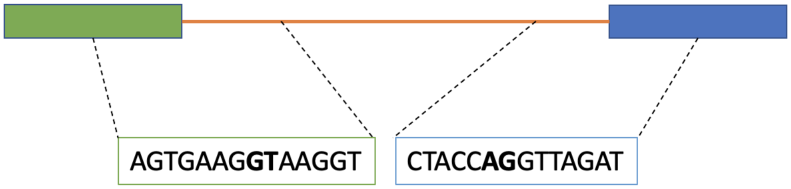


Figure 4-5. Two example sequences of 14 bp each, centered by the left and right splice sites of the intron. These two sequences are concatenated together to form a sequence element to describe the intron.

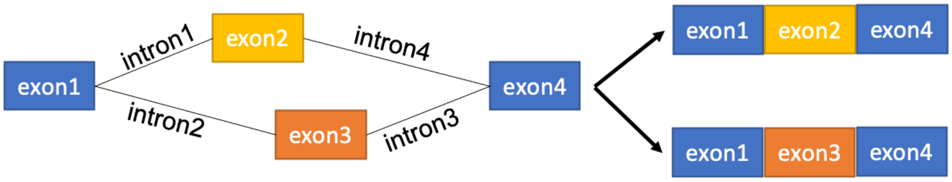
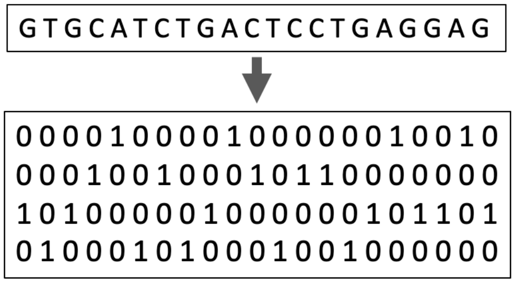


Figure 4-6. An example of a mutually exclusive splicing event. Let I_1 be the (expected) read counts of intron 1, I_2 be the (expected) read counts of intron2, then $\Psi_1 = I_1/(I_1+I_2)$ and $\Psi_2 = I_2/(I_1+I_2)$.

4.2.1.8 The sequence element and one-hot encoding

200 bp sequences are extracted for each splice site for training, and the 'ATCG' sequences are converted to a one-hot 4 × 200 encoding matrix for each splice site. An intron contains two splice sites, hence the input is a 4 × 400 matrix (Figure 4-7).



```
G T G C A T C T G A C T C C T G A G G A G

0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0
0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0
1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 1
0 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0
```

Figure 4-7. An example of the one-hot encoding of a given sequence.

4.2.2 Experiment setup and results

We evaluated our model for its ability to learn splicing information, in several ways. First, we tested the program to predict intron splicing ratios learned from RNA-seq data in 7 tissues. Second, we evaluated the ability of the model to learn sequence features that are important for splicing, such as binding sites of core and auxiliary splicing regulatory proteins. Third, we assessed the potential to predict the impact of sequence variants (mutations) on splicing using an *in silico* mutagenesis experiment.

4.2.2.1 Sequences and reference splice junctions

We obtained 144 RNA-seq samples from 7 tissues (Table 4-I) from the NIH Roadmap Epigenomics Mapping Consortium [97]. After downloading the FASTQ files, we aligned the reads to the reference genome (hg38) with STAR and extracted the splice junction read counts from the alignments with the tool junc [18]. To build a reliable reference set to train the deep learning model, we filtered out splice junctions with low read counts (i.e., average read count ≤ 4 across all 144 samples). We considered the remaining splice junctions that were not found in the GENCODE v.37 gene annotation as novel introns.

Tissue	Sample
Adipose	29
Esophagus	19
Lung	16
Ovary	4
Pancreas	20
Gastric	28
Spleen	28

Table 4-I. The 7 tissues.

Next, we extracted the splice sites in protein coding genes from the GENCODE annotation v.37 [41], and used them to group the introns (splice junctions) into 'bunches' as defined earlier. Hence, two or more introns that share an annotated splice site were kept (including the novel introns that were not in the reference annotation). We did not consider the novel splice sites present in the data, because they may be problematic and therefore not of a quality that would be suitable for training the model.

4.2.2.2 DNN models

We build two PDL models for training and evaluation, which largely share the same architecture, but with a key difference. For the first model, herein named DNN1, we initialize the filters of the RBP PWMs CNN Layer by RBP PWMs downloaded from the ATtRACT database [96]. A quality value of 0.1 was used to filter the PWMs, and 1041 PWMs for 151 human RBPs were selected. For the second model, named DNN2, we randomly initialized 1024 (4×1024) filters for the RBP PWMs CNN Layer.

4.2.2.3 Training and testing data

We split the data into a training set and a test set. We randomly selected 10% of the non-homologous protein coding genes as testing genes, and extracted the alternative splice sites and introns from these genes for testing; all the remaining alternative splice sites and introns were used for training. In our case, 51,165 splice sites were used for training, and 4,931 splice sites from 1,079 genes were used for testing.

4.2.2.4 Performance in predicting splicing ratios

Building a ground truth splicing ratio data set for evaluation is a daunting task, as experimentally validated data are not available at genome-wide scale. Instead, we used MntJULiP on the testing splice sites and introns to generate the splicing ratios to be used as reference. Note that an exact splicing ratio match between the PDL and MntJULiP is impossible, because MntJULiP and PDL are two different types of machine learning models and receive different inputs (read counts versus sequence elements). Therefore, we adopt the evaluation metrics previously proposed in the literature that classify the splicing ratio Ψ into low ($0 \leq \Psi < 0.33$), median ($0.33 \leq \Psi < 0.66$) and high ($0.66 \leq \Psi \leq 1$). With this convention, we define a true positive (TP) if the predictions of MntJULiP and PDL for a given intron lie within the same category, and otherwise we define a false positive (FP). The AUC metric is then used to measure the performance, and the statistical results are listed in Table 4-II. Surprisingly, DNN2's performance appears better than DNN1's, despite the fact that DNN1 adopted the referenced RBP PWMs as additional information in its RBP PWMs CNN Layer. We set to investigate this observation in the following sections.

Tissue	Method	Low	Medium	High
Adipose	DNN1	0.6940	0.5375	0.7013
	DNN2	0.6968	0.5506	0.7043
Esophagus	DNN1	0.6917	0.5263	0.6998
	DNN2	0.6959	0.5388	0.7043
Lung	DNN1	0.6879	0.5286	0.6971
	DNN2	0.6936	0.5598	0.7010
Ovary	DNN1	0.6825	0.5593	0.6896
	DNN2	0.6884	0.5358	0.6963
Pancreas	DNN1	0.6906	0.5243	0.6991
	DNN2	0.6942	0.5414	0.7020
Gastric	DNN1	0.7160	0.5609	0.7185
	DNN2	0.7285	0.6112	0.7305
Spleen	DNN1	0.7001	0.4901	0.7104
	DNN2	0.7090	0.5935	0.7171

Table 4-II. Comparison of the two models AUC performance on RoadMap Epigenomics data.

4.2.2.5 The model learns RBP binding motifs

We hypothesize that DNN2 automatically discovers known and novel sequence motifs. To evaluate this hypothesis, we extracted the 1024 filters from DNN2's RBP PWMs CNN Layer. We view the filters as PWMs and search for them in the CISBP-RNA database [72] and the RNAcompete RBP database [72]. We then used the TomTom motif comparison tool [98], with a q-value threshold of 0.1 to filter the results. We found that our PWMs matched 62 human RBP PWMs in the RNAcompete database (102 in total), and 51 PWMs in CISBP-RNA database (97 in total). Further, the predicted motifs matched annotated motifs for important RBPs, such as the SR

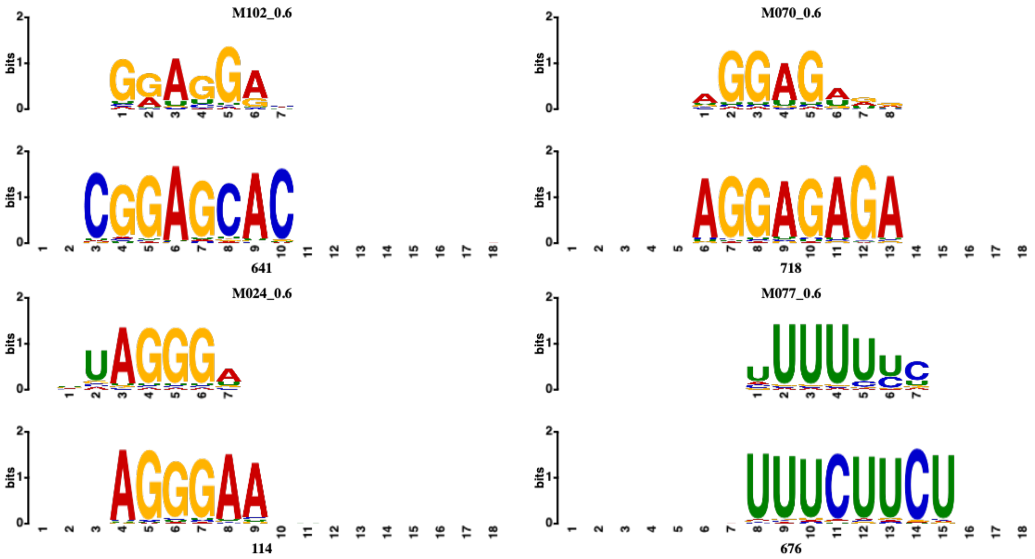


Figure 4-8. Model predicted motifs versus the annotated motifs of key splicing factors. Top left: SRSF1, top right: SRSF2, bottom left: hnRNPA2B1, bottom right: U2AF2.

protein family, the hnRNP protein family, and key splicing auxiliary factors (e.g., U2AF) (Figure 4-8). Hence, DNN2 can be used to model the sequence characteristics of the core splicing motifs, and can potentially characterize exonic and intronic splicing enhancers and silencers.

4.2.2.6 Performance robustness with different test data

Next, we tested whether the performance of the PDL models generalize to RNA-seq data from a different experiment. We obtained RNA-seq data from 7 human tissues (adipose, esophagus, lung, ovary, pancreas, stomach and spleen) from the GTEx project, with 5 samples for each tissue, where 'stomach' matches the 'gastric' in RoadMap project. As before, we aligned the reads to the hg38 human genome with STAR, and used the tool junc to calculate the splice junction read counts. Using the same test genes, we extracted the alternative splice sites and their associated introns, and predicted the splicing ratios for specific tissues with DNN1, DNN2, and with MntJULiP as reference. The AUC results are reported in Table 4-III.

Tissue	Method	Low	Medium	High
Adipose	DNN1	0.6878	0.5442	0.6958
	DNN2	0.6944	0.5442	0.7031
Esophagus	DNN1	0.6827	0.5559	0.6883
	DNN2	0.6975	0.5460	0.7045
Lung	DNN1	0.6846	0.5188	0.6950
	DNN2	0.6987	0.5460	0.7082
Ovary	DNN1	0.6831	0.5398	0.6928
	DNN2	0.6964	0.5426	0.7059
Pancreas	DNN1	0.6827	0.5205	0.6913
	DNN2	0.6982	0.5453	0.7050
Stomach	DNN1	0.6834	0.5406	0.6908
	DNN2	0.7015	0.5511	0.7078
Spleen	DNN1	0.6852	0.5346	0.6961
	DNN2	0.6961	0.5563	0.7055

Table 4-III. Comparison of the two models' AUC performance on GTEx data.

4.2.2.7 Mutations affect splice site choice

Single nucleotide polymorphisms (SNPs) occurring in splicing enhancers and silencers could affect RBP binding affinity and result in differential intron selection. To investigate the PDL model's awareness of sequence changes and whether it could take the SNP into consideration for prediction, we established an *in silico* mutagenesis experiment as follows. We chose alternative splice sites from the Cystic Fibrosis Conductance Regulator (CFTR) gene, mutations in which have been associated with cystic fibrosis through dysregulation of splicing. Specifically, we selected 15 'bunches' (30 introns, 2 introns per 'bunch'), and mutated the nucleotides (one by one) in the 200 bp sequence centered on each splice site. We then measured how the changes in sequence elements affected the PDL model's prediction. Figures 4-9, 4-10, 4-11 show the 3 splice sites in a 'bunch', namely the shared splice site and the two alternate ones, at three individual loci ('bunches'). We observe that mutations within the 25 bp around the splice site could potentially affect alternative splicing. Also, the mutations on the shared splice site, as opposed to the alternative splice site, have a less pronounced effect, suggesting that it plays a smaller part in the alternative splicing outcome.

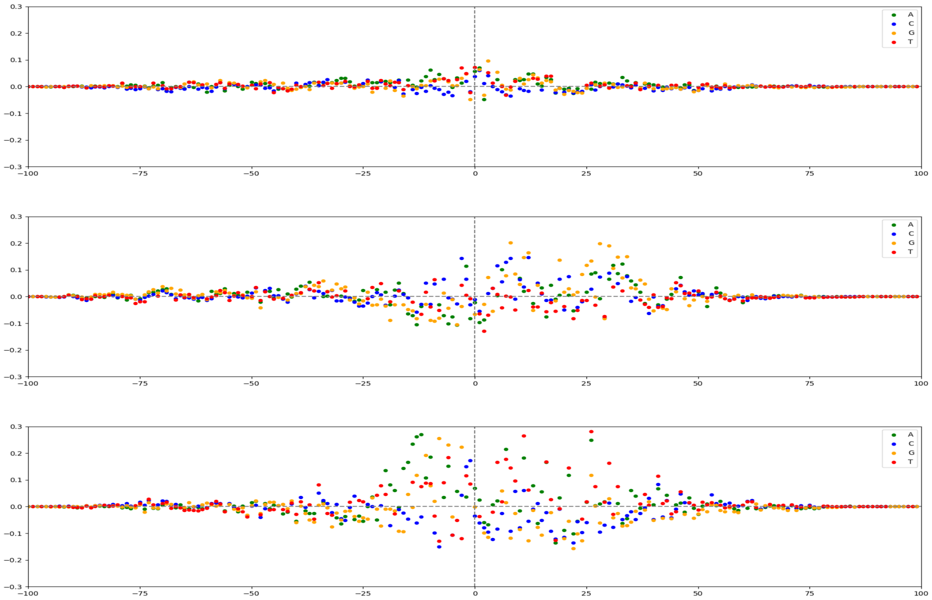


Figure 4-9. PDL predicts splicing ratio changes affected by point mutations. The top sub-graph is the RNA sequence around the shared splice site, the bottom two sub-graphs are the sequences of the other splice sites in the 'bunch'. The sequences have 200 nucleotides, centered by the splice site. Colors represent the nucleotide to which the wild-type nucleotide is mutated. The height of a nucleotide indicates the predicted splicing ratio change which affected by this point mutation. Splice sites: chr7:117301030 (top), chr7:117322058 (middle), chr7:117465747 (bottom).

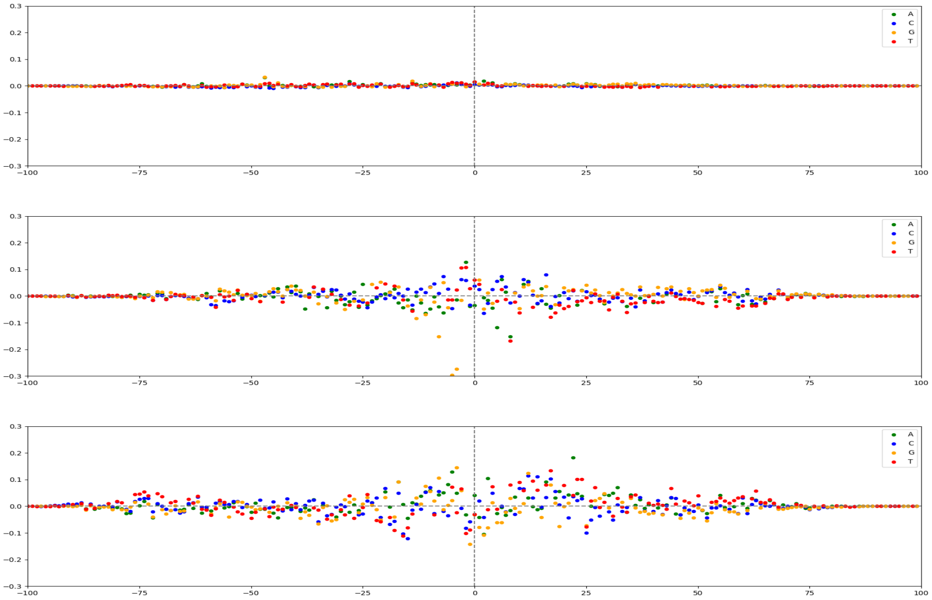


Figure 4-10. The point mutations on the shared splice site have a less pronounced effect. Splice sites chr7:117542108 (top), chr7:117548641 (middle), 117559464 (bottom).

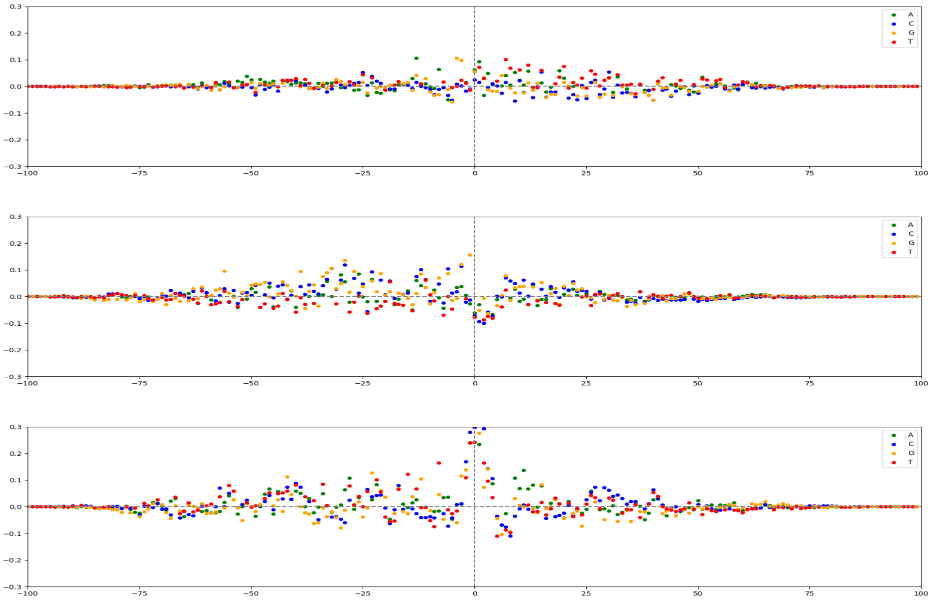


Figure 4-11. Mutations within the 25 bp around the splice site could potentially affect alternative splicing. For this 'bunch', mutations in Exon regions brings more affection to the splicing ratio change, compared to the intron regions. Splice site chr7:117714171 (top), introns: chr7:117710701 (middle), chr7:117712404 (bottom).

4.3 Discussion

The mechanisms of alternative splicing as a gene regulatory process are still incompletely understood. The splice site selection and the abundance of each splice variant are determined by the combinatorial action of multiple regulatory factors, which act in *cis*- or in *trans* to enhance or repress a particular outcome. Their signatures, and the signatures of their interactions, are encoded in the sequence proximal to the splice sites on the pre-mRNA molecule.

We described a probabilistic deep learning model (DPL model) to learn and reveal how the interactions of *cis*-regulatory elements and *trans*-factors affect alternative splicing, from the pre-mRNA sequence. The DPL model is trained on the reference human genome sequences and splice junction read counts, extracted from spliced RNA-seq read alignments generated by a 'splicing-aware' aligner. This presents several practical challenges. For instance, depending on the stringency of the alignment in the presence in the reads of sequencing errors or other differences from the genome reference, the aligner may report a number of false spliced alignments. Even when the aligner uses existing gene annotations to inform the spliced-read placement, some aligners will seek and force a false positive alignment that involves an annotated exon boundary. Also, aligners may incorrectly place a read that matches to multiple loci, altering the read counts. Prioritizing alignments in which read pairs map concordantly may alleviate this problem, but only partly. In general, aligners have different advantages and report spliced alignments leading to potential new and/or different splice variants, either real or artifactual, according to different protocols. For example, STAR produced substantially fewer spliced mappings when the alignment was not guided by known splice sites [99]. A direct consequence to our model, when aligners report numerous splices not corresponding to known introns, more effort is required to construct a reliable set of introns for training from the novel splices.

Moreover, to craft an accurate training and testing dataset for the DPL model, more effort is needed to select alignment filters to fit the metrics and overall objectives of the RNA-seq study. One particular challenge is the imbalanced data. For example, in most cases where an alternative splicing event involves two or more introns sharing a splice site, only one intron is highly expressed while the others have low expression values and PSI ratios. Therefore, the data is biased towards introns with high and low PSI ratios, resulting in higher performance of the models on these categories and poorer performance in the cases when the intron is expressed at a moderate level.

Despite significant progress, the interplay between alternative splicing (AS) and other RNA and DNA processes is poorly characterized. Recent evidence also indicates that alternative splicing might be regulated not only by the concentrations of the splicing factors or the relative concentrations of available splicing activators and repressors, but also by a more complex process involving the transcription machinery. In fact, transcription and pre-mRNA processing are not independent events. Rather, they are highly coordinated in both time and space because splicing occurs in close association with the transcript elongation by RNA polymerase II (Pol II). The 5'-terminal exons can be switched through the use of alternative promoters and alternative splicing. Similarly, the 3'-terminal exons can be switched by combining alternative splicing with alternative polyadenylation sites [100].

Apart from the relationships that arise from the co-transcriptional splicing, DNA methylation has more recently been reported to play a role in pre-mRNA splicing. As evidence, human exons are more highly methylated than introns, and methylation differences are stronger at the exon–intron boundaries [101].

Addressing these challenges requires integrative methods that combine data from multiple technologies and convert them into biologically and clinically meaningful insights. Particularly relevant, a deep neural network-based integration model learns a joint representation of multiple datasets, preserving the structure of data and merging them during the analysis stage to generate diverse types of predictions. As two examples, the program DeepCode was designed to predict splicing patterns and their changes during hESC differentiation [102], and [103] propose a deep learning framework to integrate RNA-seq and CLIP-seq data to analyze RBP-RNA interactions. As a future research direction, we will investigate combining the PDL model with a deep learning based methylation prediction module, to jointly measure the co-transcriptional splicing.

To conclude, our sequence-based model is a first and important step in constructing a deep learning based framework for alternative splicing regulatory and functional inference, in the wider context of RNA processing pathways, from heterogeneous omics resources.

Conclusions

In this thesis, we introduced several computational tools, aimed at quantitatively measuring the splicing events, revealing the splicing regulation and understanding the splicing process. Our methods leverage large scale RNA-seq data, reference RNA sequences and RNA binding motifs. We first describe our contributions, then outline limitations and possible extensions of the tools.

RNA-seq analyses aimed at determining differential splicing require a collection of features, such as exons, introns, local events or full transcripts, based on which to identify differences between two or more conditions. The accuracy of this set is critical for the downstream quantification and differential analyses. These features are extracted from a reference database, generated *de novo* by assembling the reads, or simply enumerated from the input alignments where possible. Each of these methods has drawbacks: reference features do not contain novel events, *de novo* ones may be inaccurate and may miss low expression isoforms, and raw features contain artifacts. So far there has been no judicious approach to selecting an optimal set of features. In JULiP and JULiP2, we present two mathematically rigorous methods for selecting an accurate set of introns that is as complete as possible. JULiP and JULiP2 extract introns from large collections of RNA-seq data, directly from the spliced alignments, and then refine them into a high-confidence set. JULiP and JULiP2 use similar generalized linear models only with different read count models, based on the Poisson and the negative binomial distributions, respectively. Additionally, an important technical innovation of the tools is that they work by simultaneously modeling splice junction information across all samples, taking advantage of the latent information in the set of samples to extract a highly accurate set of introns. Our evaluations showed that JULiP and JULiP2 were superior methods for feature (intron) selection, being able to reconstruct almost the entire set of features (90%) present in a collection of RNA-seq samples and >10% more compared to assembly-based methods, with very high precision. Additionally, JULiP2 builds on the selection step by incorporating a statistical model for testing for differential splicing, including covariates from condition-, sample- and gene-specific contributions. Overall, JULiP and JULiP2 are powerful tools that can be used to select a nearly complete and highly accurate feature set for downstream RNA-seq analyses to characterize splicing variation within and between conditions.

Our second effort addresses a core problem in transcriptomics, namely identifying differences in splicing between conditions. Currently, a variety of methods for differential splicing analysis are available, which differ in their selection of target features, objective functions, and technical approaches, leading to poor consistency among the results they produce. So far, there has been no tool that comprehensively addresses the wide range of differential splicing patterns, and that allows for unbiased comparisons. We designed MntJULiP to fill this gap and to computationally provide a comprehensive view of alternative splicing variation. In MntJULiP, we implemented two Bayesian models with two different objective functions: for determining the differences in intron splicing abundance (DSA) and differences in intron splicing ratios (DSR) relative to the local gene output, the two primary objective functions targeted by differential splicing methods. MntJULiP represents splicing variation at intron level, inferring introns directly from the alignments, thus capturing most splicing variations and discovering new unannotated candidates while avoiding the pitfalls of assembly. A unique capability of MntJULiP is its ability to perform multi-way comparisons, which is desirable when characterizing complex time series or multi-condition experiments, to identify a global set of features that distinguish among subgroups or stages. To complement our effort with designing differential splicing tools, and to meet the needs for intuitive and easy to use visual representations, we developed a lightweight toolkit Jutils. MntJULiP and Jutils collectively allow researchers to comprehensively survey and characterize the complexity of splicing variation across collections of samples.

Additionally, JULiP, JULiP and MntJULiP are highly scalable and efficient tools, and can process large collections comprising hundreds of RNA-seq data sets in a short amount of time, typically <1 hour.

Lastly, it is important to understand the complexity of pre-mRNA splicing and how changes in splice site choice come about. In recent years, a few machine learning based methods were developed that take DNA sequences as input and apply general rules or include results from state-of-the-art methods to predict splicing patterns [92, 93]. Such rules or features may include the lengths of donor/acceptor exons, the lengths of introns, motif scores, strength of donor/acceptor sites, whether exon can be translated without stop codon, junction conservation scores in multiple species, and others. While these early methods represent a modeling breakthrough, they are complicated, relying on heterogeneous collections of previously curated motifs but without providing insights into their role in the biological processes, and are designed for specific alternative splicing patterns or events, such as exon skipping and alternative 3'/5' site. To address this challenge and provide a flexible model to analyze any types of alternative splicing events, we developed a probabilistic deep learning method to predict alternative splicing through deep modeling of *cis*-regulatory elements and *trans*-splicing factors. Our model uses introns as the features, and therefore can capture a wide range of alternative splicing patterns. It also achieves similar performance when trained solely on the pre-mRNA sequences, without prior knowledge of regulatory motifs, which means it can be applied effectively to other systems and species where such knowledge is not available. Our PDL model can be trained rapidly with the aid of GPUs, thereby allowing it to have a large set of parameters and to deal with complex relationships present in the data. It is our first and important attempt to computationally reveal the tissue-regulated splicing code. Importantly, this DNN architecture can potentially be extended to include heterogeneous data from different sources, to characterize the cellular transcriptome and its variations along RNA processing pathways.

Overall, methods to characterize alternative splicing are structured around a set of features, used to characterize splicing variation in a sample or condition, which they quantify using read counts extracted from the alignment data, and then interrogate with statistical tests to detect differences between conditions. Each step in this process, including the selection of features, quantification and testing, is important to the outcome of the analysis. Our work contributes in a significant way to all of these computational problems. As an important design choice, our methods focus on introns as the features used to represent alternative splicing. This choice avoids the pitfalls of assembly, allows for the discovery of novel events, and captures most comprehensively the variety of alternative splicing patterns. Introns are extracted directly from the read alignments, thus ensuring the broadest collection of candidates, which is further curated with the selection methods implemented in JULiP, JULiP2 and MntJULiP before further analysis or modeling.

Nevertheless, some limitations, in particular related to the characteristics and biases of the aligner and the ’mappability’ of the sequences, remain. For instance, aligners may not be able to detect short exons, and shorter reads will be harder to align across splice junctions and may result in underestimating the abundance or even missing some low expression splice junctions entirely. Reads from repetitive or duplicated regions may not be unambiguously aligned, as sequencing errors and polymorphisms in the reads along with differences in the sequences of the genomic loci may make it impossible to determine the true location. Further, alignment at polymorphic loci may be biased against reads containing the non-reference allele.

Methods that alleviate these biases, such as the use of a pangenome reference [104] or accounting for read mappability [105] may be productive. Any further improvement in the alignment software will directly contribute to improvements in our and others' alternative splicing analysis tools.

Future extensions of these tools may target applicability to other sequencing technologies, in particular long RNA sequencing reads. Short read RNA-seq, such as Illumina RNA sequencing, has greatly expanded our ability to study the complexity of transcriptomes, including the computational prediction of alternative splicing events. However, short read RNA-seq reads have their shortages. Sufficient read depth is necessary to cover the alternative exon-exon boundaries to allow detection, which may hamper the identification of low expression isoforms and their unique splicing variations. Further, short reads mapping to multiple isoforms would confound alternative isoform identification and quantification. In contrast, long-read sequencing technologies are making it possible to sequence single full-length transcripts. With sufficient sequencing coverage, isoforms and the splice junctions can potentially be reconstructed unambiguously. However, analysis of long-read sequences has it own challenges, primarily the high error rates. To extend our tools to predict and analyze alternative splicing variation using long reads, several changes must be made to adapt to the inaccuracies in the sequencing data, while the statistical framework can be more readily adapted. First, we will error correct the reads to increase the accuracy of alignment [106]. Secondly, to correct for slight inaccuracies in the mapping of splice junctions of the error corrected reads, we can jointly analyze alignments of the reads along the genome, clustering candidate splice junctions within vicinities and selecting representative splice sites, as implemented for instance in [107]. Multiple sequencing runs per sample will produce sufficient reads to allow quantification and differential analyses. Lastly, a benefit of long reads is that they can be more accurately aligned to their correct location on the genome, thus improving the quantification. With these changes and other, we expect our JULiP family of tools to be able to accurately predict and analyze alternative splicing events from long read RNA sequencing data.

Also, as future work on our modeling of alternative splicing in the larger context of RNA processing pathways, we aim at designing an integrative machine learning framework, which incorporates heterogeneous data -including sequences, motifs, epigenomic data, tissue or condition labels, and expression data, examining the relationships between the different components in the RNA processing pathways - DNA methylation, RNA splicing and alternative splicing, and potentially RNA editing, and predicting the effects of sequence variation. The rapid and widespread adoption of deep learning in computational biology promises computational flexibility to effectively model and integrate data given enough context and scale. Currently, identification of causality and interaction in complex genotype-phenotype systems requires custom analyses and domain expert interpretation. However, one might envision a future of data accessibility, quality, and scale, that could enable near automated DL-based detection of the genetic and epigenetic events and their phenotypic effects.

In conclusion, predicting alternative splicing of the pre-mRNA is a fundamental and long standing problem in computational biology. Our work indicates that novel machine learning algorithms, high performance hardware and large amounts of data, be together, have the potential to enable drastically enhanced performance on such problems.

References

1. Wahl, M. C., Will, C. L. & Lührmann, R. The spliceosome: design principles of a dynamic RNP machine. *Cell* **136,** 701–718 (2009).
2. Berget, S. M. Exon recognition in vertebrate splicing. *Journal of biological Chemistry* **270,** 2411–2414 (1995).
3. Krämer, A. The structure and function of proteins involved in mammalian pre-mRNA splicing. *Annual review of biochemistry* **65,** 367–409 (1996).
4. Black, D. L. Mechanisms of alternative pre-messenger RNA splicing. *Annual review of biochemistry* **72,** 291–336 (2003).
5. Baker, B. S. Sex in flies: the splice of life. *Nature* **340,** 521 (1989).
6. Burgess, R. W., Nguyen, Q. T., Son, Y.-J., Lichtman, J. W. & Sanes, J. R. Alternatively spliced isoforms of nerve-and muscle-derived agrin: their roles at the neuromuscular junction. *Neuron* **23,** 33–44 (1999).
7. Grabowski, P. J. & Black, D. L. Alternative RNA splicing in the nervous system. *Progress in neurobiology* **65,** 289–308 (2001).
8. Modrek, B. & Lee, C. A genomic view of alternative splicing. *Nature genetics* **30,** 13 (2002).
9. Graveley, B. R. Alternative splicing: increasing diversity in the proteomic world. *TRENDS in Genetics* **17,** 100–107 (2001).
10. Black, D. L. Protein diversity from alternative splicing: a challenge for bioinformatics and post-genome biology. *Cell* **103,** 367–370 (2000).
11. Cartegni, L., Chew, S. L. & Krainer, A. R. Listening to silence and understanding nonsense: exonic mutations that affect splicing. *Nature reviews genetics* **3,** 285 (2002).
12. Philips, A. & Cooper, T. RNA processing and human disease. *Cellular and Molecular Life Sciences CMLS* **57,** 235–249 (2000).
13. Cáceres, J. F. & Kornblihtt, A. R. Alternative splicing: multiple control mechanisms and involvement in human disease. *TRENDS in Genetics* **18,** 186–193 (2002).
14. López-Bigas, N., Audit, B., Ouzounis, C., Parra, G. & Guigó, R. Are splicing mutations the most frequent cause of hereditary disease? *FEBS letters* **579,** 1900–1903 (2005).
15. Pan, Q., Shai, O., Lee, L. J., Frey, B. J. & Blencowe, B. J. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature genetics* **40,** 1413 (2008).

16. Wang, E. T. *et al.* Alternative isoform regulation in human tissue transcriptomes. *Nature* **456,** 470–476 (2008).

17. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* **28,** 511–515 (2010).

18. Song, L., Sabunciyan, S. & Florea, L. CLASS2: accurate and efficient splice variant annotation from RNA-seq reads. *Nucleic acids research* **44,** e98–e98 (2016).

19. Pertea, M. *et al.* StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature biotechnology* **33,** 290–295 (2015).

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

21. Canzar, S. & Florea, L. Computational methods for transcript assembly from RNA-seq reads. *METHODS FOR NEXT GENERATION SEQUENCING DATA ANALYSIS,* 247 (2016).

22. Steijger, T. *et al.* Assessment of transcript reconstruction methods for RNA-seq. *Nature methods* **10,** 1177–1184 (2013).

23. Hu, Y. *et al.* DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic acids research* **41,** e39–e39 (2013).

24. Shen, S., Wang, Y., Wang, C., Wu, Y. N. & Xing, Y. SURVIV for survival analysis of mRNA isoform variation. *Nature communications* **7,** 1–11 (2016).

25. Lappalainen, T. *et al.* Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501,** 506–511 (2013).

26. Battle, A. *et al.* Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individuals. *Genome research* **24,** 14–24 (2014).

27. Consortium, G. *et al.* The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science* **348,** 648–660 (2015).

28. Lin, Y.-Y. *et al. Cliiq: Accurate comparative detection and quantification of expressed isoforms in a population* in *International Workshop on Algorithms in Bioinformatics* (2012), 178–189.

29. Behr, J. *et al.* MITIE: Simultaneous RNA-Seq-based transcript identification and quantification in multiple samples. *Bioinformatics* **29,** 2529–2538 (2013).

30. Tasnim, M., Ma, S., Yang, E.-W., Jiang, T. & Li, W. *Accurate inference of isoforms from multiple sample RNA-Seq data* in *BMC genomics* **16** (2015), 1–12.

31. Singh, R. K. & Cooper, T. A. Pre-mRNA splicing in disease and therapeutics. *Trends in molecular medicine* **18,** 472–482 (2012).

32. Trapnell, C. *et al.* Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols* **7,** 562–578 (2012).

33. Shen, S. *et al.* rMATS: robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proceedings of the National Academy of Sciences* **111,** E5593–E5601 (2014).

34. Vaquero-Garcia, J. *et al.* A new view of transcriptome complexity and regulation through the lens of local splicing variations. *elife* **5,** e11752 (2016).

35. Hu, Y. *et al.* DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic acids research* **41,** e39–e39 (2012).

36. Anders, S., Reyes, A. & Huber, W. Detecting differential usage of exons from RNA-seq data. *Genome research* **22,** 2008–2017 (2012).

37. Hartley, S. W. & Mullikin, J. C. Detection and visualization of differential splicing in RNA-Seq data with JunctionSeq. *Nucleic acids research* **44,** e127–e127 (2016).

38. Li, Y. I. *et al.* Annotation-free quantification of RNA splicing using LeafCutter. *Nature genetics* **50,** 151 (2018).

39. Yang, G. & Florea, L. *JULiP: An efficient model for accurate intron selection from multiple RNA-seq samples* in *2016 IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)* (2016), 1–6.

40. Kim, D. *et al.* TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology* **14,** R36 (2013).

41. Harrow, J. *et al.* GENCODE: the reference human genome annotation for The ENCODE Project. *Genome research* **22,** 1760–1774 (2012).

42. Frazee, A. C., Jaffe, A. E., Langmead, B. & Leek, J. T. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics* **31,** 2778–2784 (2015).

43. Li, W., Feng, J. & Jiang, T. IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *Journal of Computational Biology* **18,** 1693–1707 (2011).

44. Frazee, A. C. *et al.* Flexible isoform-level differential expression analysis with Ballgown. *bioRxiv,* 003665 (2014).

45. Srivastava, P. K. *et al.* Genome-wide analysis of differential RNA editing in epilepsy. *Genome research* **27,** 440–450 (2017).

46. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).

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

48. Li, Y., Rao, X., Mattox, W. W., Amos, C. I. & Liu, B. RNA-seq analysis of differential splice junction usage and intron retentions by DEXSeq. *PloS one* **10,** e0136653 (2015).

49. Griffith, M. *et al.* Alternative expression analysis by RNA sequencing. *Nature methods* **7,** 843 (2010).

50. Liu, R., Loraine, A. E. & Dickerson, J. A. Comparisons of computational methods for differential alternative splicing detection using RNA-seq in plant systems. *BMC bioinformatics* **15,** 1–16 (2014).

51. Hilbe, J. M. *Negative binomial regression* (Cambridge University Press, 2011).

52. Liu, D. C. & Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical programming* **45,** 503–528 (1989).

53. Afsari, B. *et al.* Splice Expression Variation Analysis (SEVA) for inter-tumor heterogeneity of gene isoform usage in cancer. *Bioinformatics* **34,** 1859–1867 (2018).

54. Ren, W. *et al.* Transcriptome analyses of taste organoids reveal multiple pathways involved in taste cell generation. *Scientific reports* **7,** 1–13 (2017).

55. Su, C.-H., Tarn, W.-Y., *et al.* Alternative splicing in neurogenesis and brain development. *Frontiers in molecular biosciences* **5,** 12 (2018).

56. Bechstein, M. *et al.* CNTF-mediated preactivation of astrocytes attenuates neuronal damage and epileptiform activity in experimental epilepsy. *Experimental neurology* **236,** 141–150 (2012).

57. Moradi, P., Ganjkhani, M., Anarkooli, I. J. & Abdanipour, A. Neuroprotective effects of lovastatin in the pilocarpine rat model of epilepsy according to the expression of neurotrophic factors. *Metabolic brain disease* **34,** 1061–1069 (2019).

58. Florea, L., Song, L. & Salzberg, S. L. Thousands of exon skipping events differentiate among splicing patterns in sixteen human tissues. *F1000Research* **2** (2013).

59. Yang, G., Sabunciyan, S. & Florea, L. D. Comprehensive and scalable quantification of splicing differences with MntJULiP. *bioRxiv* (2020).

60. Garrido-Martin, D., Palumbo, E., Guigo, R. & Breschi, A. ggsashimi: Sashimi plot revised for browser-and annotation-independent splicing visualization. *PLoS computational biology* **14,** e1006360 (2018).

61. Burge, C. B., Tuschl, T. & Sharp, P. A. Splicing of precursors to mRNAs by the spliceosomes. *Cold Spring Harbor Monograph Series* **37,** 525–560 (1999).

62. Graveley, B. R. Sorting out the complexity of SR protein functions. *Rna* **6,** 1197–1211 (2000).

63. Fu, X.-D. The superfamily of arginine/serine-rich splicing factors. *Rna* **1,** 663 (1995).

64. Manley, J. L. *et al.* SR proteins and splicing control 1569. *Genes & development* (1996).

65. Stickeler, E. *et al.* The RNA binding protein YB-1 binds A/C-rich exon enhancers and stimulates splicing of the CD44 alternative exon v4. *The EMBO journal* **20,** 3821–3830 (2001).

66. Hönig, A., Auboeuf, D., Parker, M. M., O'Malley, B. W. & Berget, S. M. Regulation of alternative splicing by the ATP-dependent DEAD-box RNA helicase p72. *Molecular and cellular biology* **22,** 5698–5707 (2002).

67. Krecic, A. M. & Swanson, M. S. hnRNP complexes: composition, structure, and function. *Current opinion in cell biology* **11,** 363–371 (1999).

68. Förch, P. *et al.* The apoptosis-promoting factor TIA-1 is a regulator of alternative pre-mRNA splicing. *Molecular cell* **6,** 1089–1098 (2000).

69. Gerstberger, S., Hafner, M. & Tuschl, T. A census of human RNA-binding proteins. *Nature Reviews Genetics* **15,** 829–845 (2014).

70. Darnell, R. B. HITS-CLIP: panoramic views of protein–RNA regulation in living cells. *Wiley Interdisciplinary Reviews: RNA* **1,** 266–286 (2010).

71. Tuerk, C. & Gold, L. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *science* **249,** 505–510 (1990).

72. Ray, D. *et al.* A compendium of RNA-binding motifs for decoding gene regulation. *Nature* **499,** 172–177 (2013).

73. Lambert, N. *et al.* RNA Bind-n-Seq: quantitative assessment of the sequence and structural binding specificity of RNA binding proteins. *Molecular cell* **54,** 887–900 (2014).

74. Dominguez, D. *et al.* Sequence, structure, and context preferences of human RNA binding proteins. *Molecular cell* **70,** 854–867 (2018).

75. Ray, D. *et al.* Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nature biotechnology* **27,** 667–670 (2009).

76. Van Nostrand, E. L. *et al.* A large-scale binding and functional map of human RNA-binding proteins. *Nature* **583,** 711–719 (2020).

77. Stormo, G. D. DNA binding sites: representation and discovery. *Bioinformatics* **16,** 16–23 (2000).

78. Zhu, J., Mayeda, A. & Krainer, A. R. Exon identity established through differential antagonism between exonic splicing silencer-bound hnRNP A1 and enhancer-bound SR proteins. *Molecular cell* **8,** 1351–1361 (2001).

79. Zahler, A. M., Neugebauer, K. M., Lane, W. S. & Roth, M. B. Distinct functions of SR proteins in alternative pre-mRNA splicing. *Science* **260,** 219–222 (1993).

80. Polydorides, A. D., Okano, H. J., Yang, Y. Y., Stefani, G. & Darnell, R. B. A brain-enriched polypyrimidine tract-binding protein antagonizes the ability of Nova to regulate neuron-specific alternative splicing. *Proceedings of the National Academy of Sciences* **97,** 6350–6355 (2000).

81. Ladd, A. N., Charlet-B, N. & Cooper, T. A. The CELF family of RNA binding proteins is implicated in cell-specific and developmentally regulated alternative splicing. *Molecular and cellular biology* **21,** 1285–1296 (2001).

82. Blanchette, M. & Chabot, B. Modulation of exon skipping by high-affinity hnRNP A1-binding sites and by intron elements that repress splice site utilization. *The EMBO journal* **18,** 1939–1952 (1999).

83. Lynch, K. W. & Weiss, A. A CD45 polymorphism associated with multiple sclerosis disrupts an exonic splicing silencer. *Journal of Biological Chemistry* **276,** 24341–24347 (2001).

84. Boichard, A. *et al.* Two silent substitutions in the PDHA1 gene cause exon 5 skipping by disruption of a putative exonic splicing enhancer. *Molecular genetics and metabolism* **93,** 323–330 (2008).

85. Darman, R. B. *et al.* Cancer-associated SF3B1 hotspot mutations induce cryptic 3' splice site selection through use of a different branch point. *Cell reports* **13,** 1033–1045 (2015).

86. Niwa, M. & Berget, S. M. Mutation of the AAUAAA polyadenylation signal depresses in vitro splicing of proximal but not distal introns. *Genes & development* **5,** 2086–2095 (1991).

87. Liu, H.-X., Cartegni, L., Zhang, M. Q. & Krainer, A. R. A mechanism for exon skipping caused by nonsense or missense mutations in BRCA1 and other genes. *Nature genetics* **27,** 55 (2001).

88. Wang, Z. & Burge, C. B. Splicing regulation: from a parts list of regulatory elements to an integrated splicing code. *Rna* **14,** 802–813 (2008).

89. Dunckley, M. G., Manoharan, M., Villiet, P., Eperon, I. C. & Dickson, G. Modification of splicing in the dystrophin gene in cultured Mdx muscle cells by antisense oligoribonucleotides. *Human molecular genetics* **7,** 1083–1090 (1998).

90. Barash, Y. *et al.* Deciphering the splicing code. *Nature* **465,** 53 (2010).

91. Zhang, C. *et al.* Integrative modeling defines the Nova splicing-regulatory network and its combinatorial controls. *Science* **329,** 439–443 (2010).

92. Xiong, H. Y. *et al.* The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347,** 1254806 (2015).

93. Zhang, Z. *et al.* Deep-learning augmented RNA-seq analysis of transcript splicing. *Nature methods* **16,** 307–310 (2019).

94. Jaganathan, K. *et al.* Predicting splicing from primary sequence with deep learning. *Cell* **176,** 535–548 (2019).

95. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

96. Giudice, G., Sánchez-Cabo, F., Torroja, C. & Lara-Pezzi, E. ATtRACT—a database of RNA-binding proteins and associated motifs. *Database* **2016** (2016).

97. Bernstein, B. E. *et al.* The NIH roadmap epigenomics mapping consortium. *Nature biotechnology* **28,** 1045 (2010).

98. Bailey, T. L. *et al.* MEME SUITE: tools for motif discovery and searching. *Nucleic acids research* **37,** W202–W208 (2009).

99. Engström, P. G. *et al.* Systematic evaluation of spliced alignment programs for RNA-seq data. *Nature methods* **10,** 1185–1191 (2013).

100. Colgan, D. F. & Manley, J. L. Mechanism and regulation of mRNA polyadenylation. *Genes & development* **11,** 2755–2766 (1997).

101. Maor, G. L., Yearim, A. & Ast, G. The alternative role of DNA methylation in splicing regulation. *Trends in Genetics* **31,** 274–280 (2015).

102. Xu, Y., Wang, Y., Luo, J., Zhao, W. & Zhou, X. Deep learning of the splicing (epi) genetic code reveals a novel candidate mechanism linking histone modifications to ESC fate decision. *Nucleic acids research* **45,** 12100–12112 (2017).

103. Jha, A., Gazzara, M. R. & Barash, Y. Integrative deep models for alternative splicing. *Bioinformatics* **33,** i274–i282 (2017).

104. Kim, D., Paggi, J. M., Park, C., Bennett, C. & Salzberg, S. L. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature biotechnology* **37,** 907–915 (2019).

105. Pritt, J., Chen, N.-C. & Langmead, B. FORGe: prioritizing variants for graph genomes. *Genome biology* **19,** 1–16 (2018).

106. Abdel-Ghany, S. E. *et al.* A survey of the sorghum transcriptome using single-molecule long reads. *Nature communications* **7,** 1–11 (2016).

107. Florea, L. *et al.* Gene and alternative splicing annotation with AIR. *Genome research* **15,** 54–66 (2005).