2017

Transcript assembly, quantification and differential alternative splicing detection from RNA-Seq

Ruolin Liu
Iowa State University

Transcript assembly, quantification and differential alternative splicing detection from RNA-Seq

by

Ruolin Liu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Bioinformatics and Computational Biology

Program of Study Committee:
Julie A. Dickerson, Co-major Professor
Steven Cannon, Co-major Professor
Karin Dorman
Peng Liu
Amy Toth

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

DEDICATION

I dedicate this thesis to my parents for nursing me with affections and love and their dedicated partnership for success in my life.

TABLE OF CONTENTS

LIST OF TABLES v

LIST OF FIGURES vii

ACKNOWLEDGEMENTS xviii

ABSTRACT xix

CHAPTER 1. GENERAL INTRODUCTION 1

1.1 Alternative splicing 2

1.2 Next-gen sequencing of transcriptome 3

1.3 Problem formulations 4

1.4 Structure of this thesis 8

CHAPTER 2. COMPARISONS of COMPUTATION METHODS for DIFFERENTIAL ALTERNATIVE SPLICING USING RNA-SEQ in PLANT SYSTEMS . . 14

CHAPTER 3. STRAWBERRY: FAST and ACCURATE GENOME-GUIDED TRANSCRIPT RECONSTRUCTION and QUANTIFICATION from RNA-SEQ 71

CHAPTER 4. RSTRAWBERRY: DIFFERENTIAL ALTERNATIVE SPLICING from MULTIPLE SAMPLES 118

4.1 Introduction 118

4.2 Differential alternative splicing detection model 120

4.2.1 Definitions and notations 120

4.2.2 Likelihood function and priors 121

4.2.3 Bias correction 124

4.2.4 Model estimation 125

4.2.5 Implementation details 127

4.3 Result 129
4.3.1 Correcting RNA-Seq Coverage Bias 129
4.3.2 Controlling false discovery rate 130
4.3.3 A sensitivity analysis 132
4.4 Conclusions and Discussion 134
CHAPTER 5. GENERAL CONCLUSIONS 142
5.1 Conclusions 142
5.2 Future works 144
APPENDIX A. ADDITIONAL MATERIAL 146
A.1 STAN model 146
A.2 Overdispersed RNA-Seq read counts 147

LIST OF TABLES

Table 1.1 Abbreviations and acronyms. 7

Table 2.1 Area under the ROC curve (AUC) and relative ranking measured under all simulation studies. Larger values of AUC indicate better performance. 43

Table 2.2 Recall and precision at $P_{adj} = 0.05$ measured under all simulation studies. Recalls were shown as the numbers in the left column, precisions in the right column. Larger values of both metrics are better. Under a sample size of 3, SeqGSEA found no genes at $P_{adj} = 0.05$ and therefore no values were reported. 44

Table 2.3 The number of shared differentially spliced genes detected by the selected methods for the HeatT1 data set. 45

Table 2.4 The evaluation of the methods on the seven PCR validated genes. . 45

Table 2.5 Summary of the main observation for selected methods 46

Table 2.6 MATS result using junction reads only versus result using both junction reads and exon body reads in simulation study $RD100^{H}_{D}$. The Pearson correlation of the p-values in these two results is as high as 0.978. 62

Table 2.7 Total computational time in CPU-seconds 63

Table 3.1 Averaged Spearman correlation, Proportional correlation, Mean Absolute Relative Difference (MARD) for the 6 samples in *GEU*, which is a simulated Human data. These statistics are calculated based on the predicted FPKM values of 1) all reconstructed transcripts 2) only transcripts that match the known, and the true FPKM values used in the simulation. 80

Table 3.2 Correlation of FPKMs and probe counts on real RNA-Seq data HepG2. NanoString counts were compared to the FPKM values reported for three programs. The number of probes which have matching transcripts is reported on the last line. 81

Table 4.1 Multiple testing adjusted p values cumulative table. rStrawberry, Cuffdiff 2 and DEXSeq were compared using 6 sample GEUVADIS data as a negative control, where no differential alternative splicing are expected. 132

Table 4.2 Area under the ROC curve (AUC) of the 4 methods were compared using simulated Arabidopsis data. Different methods have a different test units and filters which leads to different number of tested genes. 133

LIST OF FIGURES

Figure 2.1 Quantification schema. A simplified gene model consists of two expressed isoforms (Top). Exons are colored according to the isoform of origin. Two model types used for quantification purpose (Bottom). In the count-based models (left), reads are assigned to counting units (shown by dash lines) without ambiguity. For each counting unit the model can be viewed as a test on two possible outcomes (spliced in or spliced out). The isoform resolution model is shown on the right where two ends of a read pair (show as dark solid boxes connected by curly dash line) align upstream and downstream of an alternative donor site. $l_{i1}(f)$ is the length of alignment of fragment f to isoform $i1$, and is shorter than $l_{i2}(f)$. Therefore if the fragment size distribution is known, it is possible to infer which isoform is more likely to generate f. Note that transcript effective length, i.e. $l_{i1}(f)$, $l_{i2}(f)$ and other parameters (depends on model you use) might also affect the probability of assigning reads to isoforms. Usually a maximum likelihood based approach is used to optimize this probability. . . . 37

Figure 2.2 ROC curves evaluation for three levels of AS ratio when two groups of samples have the different dispersion pattern. ROC curves for eight selected methods in simulation studies $\text{High}^{\text{Diff}}_{100\text{x}}$ (left panel), $\text{Medium}^{\text{Diff}}_{100\text{x}}$ (middle panel), $\text{Low}^{\text{Diff}}_{100\text{x}}$ (right panel). These ROC curves are obtained at a simple size of 3 for each condition. When the level or degree of DS across conditions become smaller (panel left-right), the power of discrimination of true-DS and non-DS drops significantly. However the relative ranking of each methods tend to be unchanged. DEXSeq perform consistently the best with respect to all three simulation studies. 38

Figure 2.3 ROC curves evaluation for accurate and incomplete annotation. ROC curves for eight selected methods using simulation study $\text{High}^{\text{Diff}}_{100\text{x}}$ with complete annotation (left panel) and incomplete annotation (right panel). Isoform resolution model methods, such as Cufflinks, are more robust to incomplete annotation compared with count-based models methods. 39

Figure 2.4 ROC curves evaluations for three splicing classes. ROC curves of eight selected methods based on 1755 genes containing single splicing event from simulation study $\text{High}^{\text{Diff}}_{100\text{x}}$. These 1755 genes were further divided into three splicing event classes: 803 genes with alt. donor/acceptor sites (left panel), 850 genes with intron retention (middle panel), 102 genes with exon skipping (right panel). 39

Figure 2.5 Venn digram of heat shock data set. Overlap among the set of DS genes found by 5 methods. SplicingCompass is not included because it almost shares nothing with other methods based on table 3. . . 40

Figure 2.6 Heat Map for correlation of the gene ranking scores obtained by the different methods for heat shock data set. The correlations are generally low for any two methods, indicating the methods are very different. Two methods both using NB statistics (DSGseq and SeqGSEA) achieve the highest Spearman rank correlation of 0.52. . 41

Figure 2.7 SR45a. Heat-induced differential splicing of Arabidopsis gene SR45a (AT1G07350) encoding an RNA-binding protein involved in splicing. Tracks labeled Hot and Cool contain exon-exon junction features inferred from spliced read alignments from heat-treated (hot) and control samples (cool). Junctions with fewer than five supporting reads are not shown. Two annotated gene models for SR45a are shown in the track labeled TAIR 10 mRNA. Taller blocks indicate translated regions of the gene model. Note that inclusion of an internal exon introduces a premature stop codon that interrupts translation and the exon-skipped form likely encodes the full-length protein. The gene is on the minus strand of chr1 and so transcription proceeds from right to left. 42

Figure 2.8 A two-step simulation pipeline. SAM files from real data are used as input for this pipeline. In the first step biological replicates are simulated by using Negative Binomial (NB) models. The raw fragment counts mean μ_g and variance σ_g^2 are calculated from the input. A regression function f is fitted on the set of points (μ_g, σ_g^2). Then the fitted variances are used as parameters in the NB models to generate three replicates, e.g. a, b, c. In the second step. The updated gene-level fragment counts are separate onto transcript levels based on the relative abundances and desired read depth. Finally, Flux Simulator is used to generated simulated RNA-seq reads. 55

Figure 2.9 Comparison between real (left panels) and synthetic data (right panels). The 2 panels on top are scatter plots of mean-variance relationship across replicates. The blue lines are LOWESS regression lines. The orange lines are $variance = mean$ lines. It is clear that the real data is overdispersed with respect to what we would expect from a Poisson distribution and that it was well captured by a negative binomial distribution using in the simulated data. The two panels at the bottom compare the fragment counts distribution. 56

Figure 2.10 LHY. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red whereas reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 64

Figure 2.11 SR45. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 65

Figure 2.12 SR1/SR34. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 66

Figure 2.13 SR30. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 67

Figure 2.14 P5CS1. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 68

Figure 2.15 FLM. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model. 69

Figure 2.16 ROC curves evaluation for three different AS ratios when two groups of samples have the same dispersion pattern. ROC curves for simulation studies $\text{High}_{100\text{x}}^{\text{Same}}$ (left panel), $\text{Medium}_{100\text{x}}^{\text{Same}}$ (middle panel), $\text{Low}_{100\text{x}}^{\text{Same}}$(right panel). These ROC curves are obtained at a simple size of 3 for each condition. 69

Figure 2.17 ROC curves evaluation for the two different samples sizes. Left panel shows ROC curves in the baseline simulation study $\text{High}_{100\text{x}}^{\text{Diff}} RD100_D^H$ which contained three replicates for each condition. The right panel shows the ROC curves when the sample size was increased to 8. . . 70

Figure 2.18 ROC curves evaluation for three different read depths, simulation studies $100\text{x}_{\text{High}}^{\text{Diff}}$ (left panel), $60\text{x}_{\text{High}}^{\text{Diff}}$ (middle panel), $25\text{x}_{\text{High}}^{\text{Diff}}$(right panel). 70

Figure 3.1 Overview of the algorithm of Strawberry, compared to StringTie and Cufflinks. All methods begin with a set of RNA-Seq alignments and output transcript structures and abundances in GFF/GTF format. Strawberry uses a min-flow algorithm for solving Constrained Minimum Path Cover(CMPC) problem on splicing graph, followed by assigning subexon paths to compatible assembled transcripts. In quantification step, all of the RNA-Seq read alignments on each subexon path as a whole are the subject of the EM algorithm. . . . 95

Figure 3.2 recall and precision at the nucleotide, exon, intron and transcript level. StringTie, Cufflinks and Strawberry were run on data *RD100*, which is a simulated Arabidopsis RNA-Seq data set. 96

Figure 3.3 Box plots of F1 scores at the transcript and loci level. StringTie, Cufflinks and Strawberry were evaluated on data *GEU*, which is a simulated Human RNA-Seq data set. 97

Figure 3.4 Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 10 replicates in *RD100*, which is a simulated Arabidopsis data. 98

Figure 3.5 Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 6 samples in *GEU*, which is a simulated Human data. These statistics are calculated based on the predicted FPKM values of all reconstructed transcripts and the true FPKM values used in the simulation. . . . 99

Figure 3.6 Read alignments and reconstructed transcripts at gene NAT14 using HepG2 data. A new isoform, transcript.14285.3 (shown as the middle one), has been identified by Strawberry. The junction reads that support the new AS event (alternative 3 splice site) are highlighted. The two ends of a read-pair are in the same color. A total 7 uniquely mapped read-pairs supports the novel junction. This figure is made by IGV (http://software.broadinstitute.org/software/igv/) 100

Figure 3.7 Running time in minutes of Cufflinks, Strawberry, linux word count and StringTie(ordered by slowest to fastest) on textitRD25(2.5 million reads), *RD100*(10 millions reads), and *HepG2* data(100 millions reads). 101

Figure 3.8 Translation of read alignments into a splicing graph. (a) Eleven imaginary aligned paired-end reads (or read-pairs) are represented by light blue boxes intersected by solid lines, which indicate splicing junctions, and broken lines, which indicates gap sequences. Above the read-pairs, the coverage plot is shown. The white regions have zero coverage. Below the read-pairs, three primitive exons are shown as purple boxes and five subexons in dark blue, numbered from 1-5. (b) The splicing graph constructed from part (a). The numbered nodes in the splicing graph are subexons from part (a). Dashed Arrows represent the non-intron edges and solid arrows indicate the intron edges. The numbers next to edges are the weights(number of read-pairs supports). A read-pair that contributes to an edge weight is stressed using an asterisk near its upper-left corner. All the arrows also indicate the transcription direction. The source node and target node in the splicing graph are not shown. 102

Figure 3.9 An input flow network with a subpath constraint {2-4-7}. (a),the number next to an edge is the edge cost. For every edge e, the edge constraint implies $1 \leq f(e) \leq \inf$. (b), the transformed min-flow circulation network. The 2-tuple (a,b) next to each edge indicates the optimal flow on the edge and the edge cost respectively. After Step 3, the path constraints set is $P^{\text{sub}} = \{(1,2),(1,3),(2,4,7),(4,5),(4,6),(5,8),(6,8),(7,8)\}$. Two edges no longer in the constraint set are shown in green. For these two edges, the minimum flow requirement is 0; for the rest of edges, it is 1. Two dummy nodes, s and t, are added to complete the circulation. The number of flows after decomposition is equal to the minimum flow which is 3. 103

Figure 3.10 (a), a gene with three subexons and two isoform are shown. The length of i1 is 260 bp, i2 200 bp. A paired-end read (or read-pair) is represented by light blue boxes intersected by broken lines, which indicates gap sequences. The read length is 50x2 bp. (b) A subexon path $\{s_1, s_3\}$ applies to both isoform. When on i1, this subexon path implies three subexons with the one in middle shown in gray. Consider a fixed size fragment with gap size 75 bp(shown in gray) and total fragment length 175 bp. This particular fragment can arise from 16 different positions from subexon path $\{s_1, s_3\}$ on i1 and 26 different positions from subexon path $\{s_1, s_3\}$ on i2. 104

Figure 4.1 Graphical model representation of rStrawberry alternative splicing detection model, where $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, a_0, b_0 are fixed parameters and $\boldsymbol{W}$, $\boldsymbol{Y}$, bX are observed variables. Also, $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ are transformed parameters and thus their functions are deterministic. Here, $\boldsymbol{\theta}$ is the bias parameter. The inference is focused on $\boldsymbol{\beta}$. 126

Figure 4.2 GC Bias in GEUVADIS. Sashimi plot of ERR188021 and ERR188114 on USF2 gene. Sashimi plots quantitatively visualize splice junctions for multiple samples from RNA-Seq alignments. This plot is produced by IGV (https://software.broadinstitute.org/software/igv/Sashimi). The bottom track is the genomic coordinates and the USF2 gene annotation (only 3 exons). The middle track (shown in light blue) is the junction alignments of ERR188114 (from center 1) and the top track (shown in red) is for ERR188021 (from center 2). This plot shows that samples from center 1 suffer more coverage drops for high GC exons. 129

Figure 4.3 Predicted isoform fractions of gene USF2 before and after bias correction. Totally 6 samples are used for comparison, ERR188021, ERR188052, and ERR188088 from center 2 and ERR188204, ERR188 -317 and ERR188453 from center 1. The colors sea green and chocolate represent short isoform and long isoform respectively. The x-axis is a 2 by 2 factorial table of isoforms and centers. Thus a total of 4 x-axis ticks are displayed, short isoform on the first two ticks and long isoform on the last two ticks. For each tick, there are three samples and a total of 6 points. The y-values of them represent the predicted isoform fractions before and after bias correction. The open circle indicates before bias correction and the open triangle represents after bias correction. The point-up or point-down of the triangles indicate the relative isoform fraction is increased or decreased, respectively, after bias correction. It is clear that the fraction of long isoform increases after bias correction and the amount of increase is larger for center 1 than center 2. The paired t-test of the FPKM values before and after bias correction for the long isoform is 0.02607 vs. 0.04808 for center 1 and center 2 respectively. (8) has pointed out that the samples from center 1 suffer a more dramatic loss of coverage than center 2 when it comes to high-GC exons. 135

Figure 4.4 Predicted subexon coverage on NUP107 gene of ERR188297 sample using rStrawberry. The x-axis is the transcript position. And y-axis is the density so that the area under the curve is 1. The coverage is predicted only using the sequence signals, such as GC-content. And we can see these signals can explain, to some extent, the coverage variabilities of RNA-Seq. 136

Figure 4.5 Full ROC curves of the differential alternative splicing detection results of 4 methods. 137

Figure 4.6 Partial ROC curves (False positive rate 0 - 0.2) of the differential alternative splicing detection results of 4 methods. 138

Figure A.1 Gene-wise dispersion on biological sample HG00117. Each data point represents a gene. Mean and variance of TPM is calculated using all 8 replicates. The red line is the best linear regression of variance on mean and the blue line is the 45-degree angle straight line which indicates no overdispersion. 148

Figure A.2 Transcript-wise dispersion on biological sample HG00117. Each data point represents a transcript. And only the transcripts from two-isoform genes are used. Mean and variance of TPM is calculated using all 8 replicates. The red line is the best linear regression of variance on mean and the blue line is the 45-degree angle straight line which indicates no overdispersion. 149

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Julie A. Dickerson for her guidance, patience and support throughout this research and the writing of this thesis. Her insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and guidance of this work. I would additionally like to thank Dr. Karen S. Dorman for her guidance throughout the last stages of my graduate career and her inspirational teaching style.

ABSTRACT

This dissertation is focused on improving RNA-Seq processing in terms of transcript assembly, transcript quantification and detection of differential alternative splicing. There are two major challenges of solving these three problems. The first is accurately deriving transcript-level expression values from RNA-Seq reads that often align ambiguously to a set of overlapping isoforms. To make matter worse, gene annotation tends to misguide transcript quantification as new transcripts are often discovered in new RNA-Seq experiments. The second challenge is accounting for intrinsic uncertainties or variabilities in RNA-Seq measurement when calling differential alternative splicing from multiple samples across two conditions. Those uncertainties include coverage bias and biological variations. Failing to account for these variabilities can lead to higher false positive rates.

To addressed these challenges, I develop a series of novel algorithms which are implemented in a software package called Strawberry. To tackle the read assignment uncertainty challenge, Strawberry assembles aligned RNA-Seq reads into transcripts using a constrained flow network algorithm. After the assembly, Strawberry uses a latent class model to assign reads to transcripts. These two steps use different optimization frameworks but utilize the same graph structure, which allows a highly efficient, expandable and accurate algorithm for dealing large data. To infer differential alternative splicing, Strawberry extends the single sample quantification model by imposing a generalized linear model on the relative transcript proportions. To account for count overdispersion, Strawberry uses an empirical Bayesian hierarchical model. For coverage bias, Strawberry performs a bias correction step which borrows information across samples and genes before fitting the differential analysis model.

A serious of simulated and real data are used to evaluate and benchmark Strawberry's

result. Strawberry outperforms Cufflinks and StringTie in terms of both assembly and quantification accuracies. In terms of detecting differential alternative splicing, Strawberry also outperforms several state-of-the-art methods including DEXSeq, Cuffdiff 2 and DSGseq. Strawberry and its supporting code, e.g., simulation and validation, are freely available at my github (https://github.com/ruolin).

CHAPTER 1. GENERAL INTRODUCTION

Proteins are the basic building blocks of cells. The central dogma of molecular biology states that the genetic information are passed from gene to transcripts and then to proteins. The process of making transcripts from DNA is called gene expression. Gene expression dictates what kinds of proteins are being made and the amount of it. Different cells in a multicellular organism may express very different sets of genes, even though they contain the same DNA. Thus studying gene expression is important to understand cell function. The thesis covers transcript identification, transcript quantification and detection of differential alternative splicing from RNA-Seq. I first give a quick overview of RNA-Seq and alternative splicing. Over the years, RNA-Seq has become the state-of-the-art assay to study alternative splicing from transcriptome level. With this prior knowledge, the core of this monograph, transcript assembly, quantification, and differential alternative splicing problems, will be defined and classes of different bioinformatics approaches will be introduced. All of these methods have their merits and Strawberry i) borrows and combines their strengths, 2) extends and improves their ideas, and iii) innovates based on what they have not done.

i Borrowing. Strawberry borrows the idea of doing transcript assembly before quantification to avoid annotation bias from Cufflinks. Also, Casper is the first to use read counts on a set of exons (called exon path) to speed up the EM algorithm. Strawberry borrows the notion of exon path.

ii Extending. Strawberry improves the idea of exon path by modeling it in a parametric latent class model. Strawberry also extends Alpine's idea of correcting coverage bias via a generalized linear model into this exon path model. In addition, Strawberry improves on Traph's flow network algorithm to better serve for pair-end reads in assembly.

iii Innovating. To my knowledge, Strawberry is the first to simultaneously estimate transcript abundances and detect differential alternative splicing. Strawberry is also the first differential splicing analysis method that is not restricted to gene-by-gene detection.

1.1 Alternative splicing

In the late 1970s, (2; 4) showed that infected cells produce several pre-mRNAs which are much larger than any of the mRNAs present later. Part of pre-mRNA sequences are removed and the remaining sequences are joint together. Their studies also revealed that the pre-mRNA produced by adenovirus was spliced in many different ways, leading to different viral proteins. Since the first dawn of this phenomenon, recognized as Alternative Splicing (AS), it was quickly found in every eukaryotic cell. AS is a post-transcriptional regulation mechanism that allows a single gene to produce multiple mRNA transcripts. AS occurs as a normal phenomenon in eukaryotes and is more abundant in higher eukaryotes than in lower eukaryotes (10). Researchers have found more than 95% of human genes and 60% of Drosophila multi-exon genes are alternatively spliced (8). In plants, 61% of intron-containing genes undergo alternative splicing (25). The ubiquitousness of AS implies its functional importance. AS generates protein isoforms which have different biological properties, including protein-protein interaction, subcellular localization, or catalytic ability (23). In addition to contributing to protein diversity and regulation, some variants of AS may be nonfunctional and quickly degraded, providing cells another mechanism to regulate gene expression after transcription but before translation. Like many biological regulatory processes, however, The complete picture and full roles of AS are still not clear. Some well known examples include sex-specific splicing in Drosophila (19; 1; 26), regulating gene expression in response to environmental stimuli and developmental changes in Arabidopsis (3; 13; 25), and hallmarks of cancer and cancer related regulation in human (17; 11; 29). Another interesting example to humans is that the stress of exams on a medical student induces an alternative splice variant of SMG-1 which lacks exon 63 in peripheral leukocytes

(12). There are several different types of alternative splicing (AS) events and some are more common than the others. The common events include exon skipping (ES), alternative 3 splice site (A3SS), alternative 5 splice site (A5SS), and intron retention (IR). ES happens when a cassette exon is spliced out of the transcript together with its flanking introns; A3SS and A5SS occur when two or more splice sites are recognized at one end of an exon; IR refer to an intron that remains in the mature mRNA transcript (10). Animals and plants differ in their most common types of AS events. ES is the most common AS type in humans ($> 40\%$), but the least common type in plants (5%) (10). Intron retention is the most prevalent AS type in plants ($\sim 40\%$) but the least prevalent type in humans (21; 15). Alternative 3'SS and 5'SS account for 18% and 8% of all AS events in higher eukaryotes, respectively (10). Less frequent, complex events involving the concurrence of same or different simple events (e.g. mutually exclusive exons). Other less frequent AS events include alternative promoter usage and alternative polyadenylation (10).

1.2 Next-gen sequencing of transcriptome

Sequencing is a process of digitalizing biological genetic materials and converting them to human readable strings. The so-called Next-Gen Sequencing is a high-throughout and low-cost alternative to the first generation Sanger Sequencing. Although various Next-Gen sequencing platforms or instruments exist in today's market, Illumina dominates over 90% of the market. Without mentioning a specific sequencing technology, I refer to Illumina in this monograph. Illumina sequencer outputs short nucleotide strings (called reads), ranging from 100bp - 300bp. These reads can appear in pairs if two reads are sequenced from the same underlying DNA (or cDNA) fragments but from different ends.

RNA-Seq is the short term for RNA-Sequencing which sequences the transcriptome. One field in RNA-Seq is the studying of gene expression, i.e., the amount of mRNA copies a gene produces. Before the NGS era, DNA microarray is a powerful tool for analyzing gene expression and alternative splicing (16). However, one of the drawbacks of microarray

technology is that it requires prior knowledge of the genomic sequencing and splicing models of the organism of interest. This makes it difficult to study alternative splicing since it must require probes that span unique splicing junctions. Applying Next-Gen Sequencing on RNA, named RNA-Seq (31), has changed the game completely as it allows single base resolution of a complete transcriptome and is also applicable to non-model organisms, which are difficult for microarrays. RNA-Seq has made many problems more accessible to study, e.g. complete assembly of transcriptome, identification of new splicing variants, compared to hybridization-based technologies such as microarrays.

However, analyzing RNA-Seq data is difficult and not straightforward. This creates a huge demand to develop bioinformatics tools or pipelines which can analyze and interpret these data in a straightforward and useful way. Since 2008, we have witnessed a boom in RNA-Seq bioinformatics tools. A summary of RNA-Seq bioinformatics tools on a Wikipedia page has listed more than 400 tools, covering almost every aspect of RNA-Seq data analysis, e.g. quality controls, alignments, assembly, expression and differential analysis, visualization and etc. In this monograph, I focus on transcript reconstruction (which is assembly + quantification), and differential alternative splicing detection. The success of transcript reconstruction depends on the success of a series of upstream bioinformatics steps such as raw read processing and alignment as well. However, they are not the focus in this monograph. Note the word isoform is different from transcript in this monograph. I use isoform to refer to gene isoform(s) which are the transcript(s) that come from a single locus. The sets of isoforms form a *partition* of the set of transcripts.

1.3 Problem formulations

Problem 1. Transcript reconstruction from single sample RNA-Seq Given a set of RNA-Seq reads from single biological sample and optionally other supplementary inputs (for example, genome sequences, annotation models), the problem is to identify all expressed transcripts from this sample, i.e., transcript identification, and associate each

transcript with a positive expression value that can be used to compared within the sample, i.e., transcript quantification.

This problem describes an algorithm that takes a sample of RNA-Seq reads and outputs a file that defines transcripts and their expression such that a transcript can be compared to other transcripts in terms of the expression. How a transcript is defined biologically is out of the scope of this thesis. From an algorithmic point of view, I consider a transcript as an array of genomic intervals. Each interval has information such as chromosome, start position, end position and associated meta data such as the nucleotide sequences, and whether it is an exon, UTR and etc.. (24) uses the word transcript reconstruction to refer to both assembly and quantification and I follow their standard in this dissertation.

Based on whether an algorithm takes additional inputs, several workflows exist. Firstly, if no additional inputs are given and the raw reads are the only inputs, (7; 32; 22) can perform de-novo assembly using the unmapped reads into transcriptome, but they usually do not estimate transcript expression. However, people can always use other quantification tools to calculate the expression of de-novo assembled transcripts. The second type of workflow uses only reference genome. These workflows usually align RNA-seq reads to the reference genome using splice aware aligners. After the alignment, methods may assemble transcripts and quantify the expression in a sequential manner or simultaneously. This kind of methods will be called genome-dependent methods in this monograph. The third type of workflow uses either reference genome plus annotation, or equivalently a reference transcriptome. This type of workflows can not detect new transcripts and I refer to this kind of methods as annotation-dependent.

Another important aspect from problem 1 is the expression metric. A well-known expression metric for transcript is Reads Per Kilobase of transcript per Million mapped reads (RPKM). To my knowledge, RPKM was first proposed in (18). RPKM was soon adopted by the community but was later extended as Fragments Per Kilobase of transcript per Million mapped reads (FPKM) by (28) to adjust for pair-end reads. FPKM or RPKM of any

transcript is calculated as FPKM $= \frac{c \cdot 10^9}{l \cdot d}$, where c is the read counts for that transcript, and l is the transcript length and d is the total number of mapped reads in a sample. Almost at the same time, (14) proposed another metric called transcripts per kilobase million (TPM). TPM has been considered as a better metric for comparing transcript expression across samples than FPKM or RPKM (30). TPM can be calculated through FPKM or RPKM, by $\text{TPM}_i = \text{FPKM}_i \cdot 10^6 / \sum_i \text{FPKM}_i$. This extra normalization makes TPM a better metric for comparison across samples since the sum of TPMs in each sample are the same. For a single sample, TPM and FPKM would be equivalent.

The difficulty of solving transcript reconstruction problem arises from the ambiguity of reads assignment to isoforms uniquely. This read assignment challenge is twofold: statistically, it often requires high-dimensional mixture models, and computationally, it needs to process datasets that commonly consist of tens of millions of fragments (20). To make matters worse, the assignment problem can not faithfully rely on gene annotation as transcripts are often discovered in new RNA-Seq experiments. In the next chapter, I will give a brief but comprehensive overview of the existing methods of tackling the transcript quantification problem.

Problem 2: Differential alternative splicing from multiple RNA-Seq samples. Consider a RNA-Seq experiment that involves two experimental conditions. For each condition, replicate RNA-Seq libraries are generated and sequenced. The goal is to identify differentially spliced transcripts and/or differential AS events across the two conditions. The set of genes and isoforms structures, i.e., exon-intron structures, known as gene annotation or just annotation, is known a priori.

The problem 2 identifies changes in the relative abundances of transcripts between two experimental conditions This concept should be distinguished from differential transcript expression (DTE). DTE compares the expression level of a transcript in each condition and calls significance if the change of seeing such a change is small enough under an appropriate statistical model (27). DTE can be considered as a natural progression of differential gene

Table 1.1 Abbreviations and acronyms.

AS	Alternative Splicing
ASM	Alternative Splicing Module
DTE	Differential Transcript Expression
DGE	Differential Gene Expression
DAS	Differential Alternative Splicing
FPKM	Fragments Per Kilobase of transcript per Million mapped reads
RPKM	Reads Per Kilobase of transcript per Million mapped reads.
TSS	Transcript Start Site
MPC	Minimum Path Cover
GLM	Generalized Linear Model
CMPC	Constrained Minimum Path Cover
NB	Negative Binomial
TPM	Transcripts Per kilobase Million
PSI	Percent of Splice In
FDR	False Discovery Rate
DAG	Directed Acyclic Graph
TSS	Transcription Start Site
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
PSI	Percent Splice in
AUC	Area Under the Curves
ROC	Receiver Operating Characteristic

expression (DGE) problem. Both DTE and DGE are important, well-studied topics in RNA-Seq. However, I will not discuss them in this monograph. Differential alternative splicing (DAS) is, instead, interested in a group of transcripts (>= 2) and the change of relative abundances across conditions. In the case of a group of two isoforms, DAS is equivalent to the simple notion of *isoform switching* which is well-defined as the predominant isoform switches from one to another when the condition is changed.

DAS can be analyzed at the level of full-length transcripts or at the level of single splicing events (for example inclusion or exclusion of a particular cassette exon) (5). Therefore, methods for detecting DAS may be categorized into exon-centric models and transcript-centric models. Event-centric models focus on individual AS events across conditions while transcript-centric models seek to identify alternatively spliced isoforms. The prototype of

event-centric versus transcript-centric nomenclature was first introduced in (9) and they used *exon-centric analyses* to refer the inference on single alternative exon and *isoform-centric analyses* for detection of significant changes in isoform composition. (6) extended *exon-centric methods* to *event-centric methods* to refer to methods that can detect one or many AS events. Due to the limitation of read length of short read technologies, such as Illumina, event-centric models are designed to apply differential analysis directly on unambiguous counting units (i.e., exons or exon-exon junctions) rather than the whole transcripts. If a counting unit is differentially expressed (usually in terms of read count), it can be further translate to AS events. Although the event-centric models do not directly address the issue of quantifying isoform abundances, the reads at counting units can fully reflect isoform expression as long as there is no isoform that can be composed by the combination of other isoforms (26). Instead of transforming the question into detecting differential usage of counting units, transcript-centric methods seek to directly compare the relative transcript abundance across samples and/or conditions.

1.4 Structure of this thesis

The rest of this dissertation is organized in the following way. Chapter 2 includes a published literature review of relative bioinformatics methods for differential alternative splicing detection using RNA-Seq. Although tremendous success has been made in this field, I have witnessed same weakness and things that can be improved from the status quo. Therefore, chapter 3 introduces the transcript assembly and quantification method of Strawberry for a single sample. In chapter 3, both real and simulated data are used to benchmark Strawberry's result against other state-of-the-art methods. The differential alternative splicing detection model is given in chapter 4, which also includes results of benchmarking against other state-of-the-art methods using both simulated and real data. Chapter 5 contains the conclusion and future works.

Bibliography

[1] Bell, L. R., Horabin, J. I., Schedl, P., and Cline, T. W. (1991). Positive autoregulation of sex-lethal by alternative splicing maintains the female determined state in Drosophila. *Cell*, 65(2):229–239.

[2] Berget, S. M., Moore, C., and Sharp, P. A. (1977). Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proc. Natl. Acad. Sci. U.S.A.*, 74(8):3171–3175.

[3] Black, D. L. (2003). Mechanisms of alternative pre-messenger RNA splicing. *Annu. Rev. Biochem.*, 72:291–336.

[4] Chow, L. T., Gelinas, R. E., Broker, T. R., and Roberts, R. J. (1977). An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell*, 12(1):1–8.

[5] Dvinge, H., Kim, E., Abdel-Wahab, O., and Bradley, R. K. (2016). RNA splicing factors as oncoproteins and tumour suppressors. *Nat. Rev. Cancer*, 16(7):413–430.

[6] Goldstein, L. D., Cao, Y., Pau, G., Lawrence, M., Wu, T. D., Seshagiri, S., and Gentleman, R. (2016). Prediction and quantification of splice events from rna-seq data. *PLOS ONE*, 11(5):1–18.

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

[8] Graveley, B. R., Brooks, A. N., Carlson, J. W., Duff, M. O., Landolin, J. M., Yang, L., Artieri, C. G., van Baren, M. J., Boley, N., Booth, B. W., Brown, J. B., Cherbas,

L., Davis, C. A., Dobin, A., Li, R., Lin, W., Malone, J. H., Mattiuzzo, N. R., Miller, D., Sturgill, D., Tuch, B. B., Zaleski, C., Zhang, D., Blanchette, M., Dudoit, S., Eads, B., Green, R. E., Hammonds, A., Jiang, L., Kapranov, P., Langton, L., Perrimon, N., Sandler, J. E., Wan, K. H., Willingham, A., Zhang, Y., Zou, Y., Andrews, J., Bickel, P. J., Brenner, S. E., Brent, M. R., Cherbas, P., Gingeras, T. R., Hoskins, R. A., Kaufman, T. C., Oliver, B., and Celniker, S. E. (2011). The developmental transcriptome of Drosophila melanogaster. *Nature*, 471(7339):473–479.

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

[10] Keren, H., Lev-Maor, G., and Ast, G. (2010). Alternative splicing and evolution: diversification, exon definition and function. *Nat. Rev. Genet.*, 11(5):345–355.

[11] Klinck, R., Bramard, A., Inkel, L., Dufresne-Martin, G., Gervais-Bird, J., Madden, R., Paquet, E. R., Koh, C., Venables, J. P., Prinos, P., Jilaveanu-Pelmus, M., Wellinger, R., Rancourt, C., Chabot, B., and Abou Elela, S. (2008). Multiple alternative splicing markers for ovarian cancer. *Cancer Res.*, 68(3):657–663.

[12] Kurokawa, K., Kuwano, Y., Tominaga, K., Kawai, T., Katsuura, S., Yamagishi, N., Satake, Y., Kajita, K., Tanahashi, T., and Rokutan, K. (2010). Brief naturalistic stress induces an alternative splice variant of SMG-1 lacking exon 63 in peripheral leukocytes. *Neurosci. Lett.*, 484(2):128–132.

[13] Lareau, L. F., Green, R. E., Bhatnagar, R. S., and Brenner, S. E. (2004). The evolving roles of alternative splicing. *Curr. Opin. Struct. Biol.*, 14(3):273–282.

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

[15] Marquez, Y., Brown, J. W., Simpson, C., Barta, A., and Kalyna, M. (2012). Transcriptome survey reveals increased complexity of the alternative splicing landscape in Arabidopsis. *Genome Res.*, 22(6):1184–1195.

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

[17] Misquitta-Ali, C. M., Cheng, E., O'Hanlon, D., Liu, N., McGlade, C. J., Tsao, M. S., and Blencowe, B. J. (2011). Global profiling and molecular characterization of alternative splicing events misregulated in lung cancer. *Mol. Cell. Biol.*, 31(1):138–150.

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

[19] Nagoshi, R. N., McKeown, M., Burtis, K. C., Belote, J. M., and Baker, B. S. (1988). The control of alternative splicing at genes regulating sexual differentiation in D. melanogaster. *Cell*, 53(2):229–236.

[20] Patro, R., Duggal, G., and Kingsford, C. (2015). Salmon: Accurate, versatile and ultrafast quantification from rna-seq data using lightweight-alignment. *bioRxiv*.

[21] Reddy, A. S., Rogers, M. F., Richardson, D. N., Hamilton, M., and Ben-Hur, A. (2012b). Deciphering the plant splicing code: experimental and computational approaches for predicting alternative splicing and splicing regulatory elements. *Front Plant Sci*, 3:18.

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

[23] Stamm, S., Ben-Ari, S., Rafalska, I., Tang, Y., Zhang, Z., Toiber, D., Thanaraj, T. A., and Soreq, H. (2005). Function of alternative splicing. *Gene*, 344:1–20.

[24] Steijger, T., Abril, J. F., Engstrom, P. G., Kokocinski, F., Hubbard, T. J., Guigo, R., Harrow, J., Bertone, P., Abril, J. F., Akerman, M., Alioto, T., Ambrosini, G., Antonarakis, S. E., Behr, J., Bertone, P., Bohnert, R., Bucher, P., Cloonan, N., Derrien, T., Djebali, S., Du, J., Dudoit, S., Engstrom, P., Gerstein, M., Gingeras, T. R., Gonzalez, D., Grimmond, S. M., Guigo, R., Habegger, L., Harrow, J., Hubbard, T. J., Iseli, C., Jean, G., Kahles, A., Kokocinski, F., Lagarde, J., Leng, J., Lefebvre, G., Lewis, S., Mortazavi, A., Niermann, P., Ratsch, G., Reymond, A., Ribeca, P., Richard, H., Rougemont, J., Rozowsky, J., Sammeth, M., Sboner, A., Schulz, M. H., Searle, S. M., Solorzano, N. D., Solovyev, V., Stanke, M., Steijger, T., Stevenson, B. J., Stockinger, H., Valsesia, A., Weese, D., White, S., Wold, B. J., Wu, J., Wu, T. D., Zeller, G., Zerbino, D., and Zhang, M. Q. (2013). Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods*, 10(12):1177–1184.

[25] Syed, N. H., Kalyna, M., Marquez, Y., Barta, A., and Brown, J. W. (2012). Alternative splicing in plants–coming of age. *Trends Plant Sci.*, 17(10):616–623.

[26] Telonis-Scott, M., Kopp, A., Wayne, M. L., Nuzhdin, S. V., and McIntyre, L. M. (2009). Sex-specific splicing in Drosophila: widespread occurrence, tissue specificity and evolutionary conservation. *Genetics*, 181(2):421–434.

[27] Trapnell, C., Hendrickson, D. G., Sauvageau, M., Goff, L., Rinn, J. L., and Pachter, L. (2013). Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nat. Biotechnol.*, 31(1):46–53.

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

[29] Venables, J. P., Klinck, R., Koh, C., Gervais-Bird, J., Bramard, A., Inkel, L., Durand, M., Couture, S., Froehlich, U., Lapointe, E., Lucier, J. F., Thibault, P., Rancourt,

C., Tremblay, K., Prinos, P., Chabot, B., and Elela, S. A. (2009). Cancer-associated regulation of alternative splicing. *Nat. Struct. Mol. Biol.*, 16(6):670–676.

[30] Wagner, G. P., Kin, K., and Lynch, V. J. (2012). Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci.*, 131(4):281–285.

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

[32] Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., Huang, W., He, G., Gu, S., Li, S., Zhou, X., Lam, T. W., Li, Y., Xu, X., Wong, G. K., and Wang, J. (2014). SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics*, 30(12):1660–1666.

CHAPTER 2. COMPARISONS of COMPUTATION METHODS for DIFFERENTIAL ALTERNATIVE SPLICING USING RNA-SEQ in PLANT SYSTEMS

Authors: Ruolin Liu, Ann E Loraine, and Julie A Dickerson

BMC Bioinformatics. 2014; 15(1): 364. Published 2014 Dec 16.

doi: 10.1186/s12859-014-0364-4

Author's contributions

RL leads this study. RL, JD and LA contribute to the design of the study, the interpretation of the results. RL and JD together write the manuscript. LA provides the real RNA-seq data and performed the RT-PCR validation. RL writes the programs and does all the data analysis. All the authors read and approved the final manuscript.

Abstract

Alternative Splicing (AS) as a post-transcription regulation mechanism is an important application of RNA-seq studies in eukaryotes. A number of software and computational methods have been developed for detecting AS. Most of the methods, however, are designed and tested on animal data, such as human and mouse. Plants genes differ from those of animals in many ways, e.g., the average intron size and preferred AS types. These differences may require different computational approaches and raise questions about their effectiveness on plant data. The goal of this paper is to benchmark existing computational differential splicing (or transcription) detection methods so that biologists can choose the most suitable tools to accomplish their goals. This study compares the eight popular public available software packages for differential splicing analysis using both simulated and real Arabidopsis

thaliana RNA-seq data. All software are freely available. The study examines the effect of varying AS ratio, read depth, dispersion pattern, AS types, sample sizes and the influence of annotation. Using a real data, the study looks at the consistences between the packages and verifies a subset of the detected AS events using PCR studies. No single method performs the best in all situations. The accuracy of annotation has a major impact on which method should be chosen for AS analysis. DEXSeq performs well in the simulated data when the AS signal is relatively strong and annotation is accurate. Cufflinks achieve a better tradeoff between precision and recall and turns out to be the best one when incomplete annotation is provided. Some methods perform inconsistently for different AS types. Complex AS events that combine several simple AS events impose problems for most methods, especially for MATS. MATS stands out in the analysis of real RNA-seq data when all the AS events being evaluated are simple AS events.

Background

Alternative splicing (AS) is a post-transcriptional regulation mechanism that allows a single gene to produce multiple mRNA transcripts. Some of the roles of AS include regulating gene expression in response to environmental stimuli and developmental changes (1; 2; 3). In addition to contributing to protein diversity and regulation, some variants of AS may be nonfunctional and quickly degraded, providing gives cells another mechanism to regulate gene expression after transcription but before translation. AS occurs as a normal phenomenon in eukaryotes and is more abundant in higher eukaryotes than in lower eukaryotes (4). More than 95% of human genes and 60% of Drosophila multi-exon genes are alternatively spliced. In plants, 61% of intron-containing genes undergo alternative splicing(3).

Although there is no consensus classification of AS types, the five standard types are skipped exon (SE), alternative 3 splice site (A3SS), alternative 5 splice site (A5SS), mutually exclusive exons (MXE), and intron retention (IR) (6). Animals and plants differ in their most common types of AS events. SE is the most common AS type in humans ($> 40\%$), but the least common type in plants (5%) (4). Intron retention is the most prevalent AS type in plants ($\sim 40\%$) but the least prevalent type in humans (7; 8). This difference suggests plants and animals may recognize exons and introns in different ways (7). Also, AS does not always occur as one of the simple events described above; combinations of multiple simple AS events are common. In Arabidopsis, multiple exons may be skipped together and/or exon skipping occurs in the company of alternative 5' and/or 3' splice sites (8). Such complex AS events are abundant in Arabidopsis latest annotation version, TAIR 10 (9).

Some evidence also suggests that plants and animals may regulate AS in different ways. For examples, plants possess nearly double the number of SR proteins as compared to nonphotosynthetic organisms(10). SR stands for serine(S)-arginine(R)-rich proteins, a conserved family of pre-mRNA splicing factors. Interestingly, most SR proteins (14 of the 18 Arabidopsis SR protiens) (11) are themselves alternatively spliced and some studies have linked the AS of several SR proteins (e.g., SR45,SR45a,SR1/SR34, SR30) to environmental signals. AS is believed to play a critical role in helping plants adapt to their environment and may increase our understanding of plant and crop phenotypes (3).

The advent of RNA-seq has increased the observed frequency of AS in plants from 30% (12; 13; 14) in the pre-NGS era to 61% (8). As RNA-seq becomes the new standard for studying gene and transcription expression, a key problem is to detect condition-specific differences, such as differential expression and differential alternative splicing. To date, dozens of methods for detecting differential AS using RNA-seq have been published. Most of the methods are designed for and tested on human, mouse and other mammals. Their performance on RNA-seq data from plants remains in question due to the differences in AS

machinery between animals and plants. Recent review papers (15; 16; 17) compare differential alternative splicing detection methods with respect to methodology but do not evaluate performance under realistic conditions. Another two publications (18; 19) benchmark methods and algorithms for transcript reconstruction and quantification. To our knowledge, this study is the first to systematically compare differential alternative splicing methods using RNA-seq in plant systems.

Selection criteria and limitation of this study

This work benchmarks eight popular methods for differential AS according to the three criteria given below: effectiveness, biological replicates and software engineering.

- Effectiveness: the method should detect differential AS across samples. Note that this is not necessarily equivalent to isoform quantification problem as changes in the absolute isoform expression do not necessarily imply differential alternative splicing (15).

- Biological replicates: the selected method should be able to take advantage of biological replicates in the RNA-seq data sets.

- Software engineering: the method has to be implemented as a usable and robust program so that a scientist with limited computational skills can run the program regardless of understanding the theory behind it.

For example, under these criteria, some methods are ruled out for inclusion in this study. E.g., SpliceTrap (20) only quantifies alternative splicing within a single condition and MISO (21) and PSGInfer (22) do not support biological replicates. Our list of programs is not exhaustive; however, we have selected a set of programs which represent a variety of approaches. Due to our limited human resources and computational power, the current versions of FDM(23) and JuncBase(24) met our criteria but were excluded from this study. FDM uses a splice graph representation of aligned RNA-seq data and Jensen Shannon

Divergence (JSD) to measure the difference in relative transcript abundances. JuncBase uses exclusively reads spanning exon-exon junctions. These concepts are well represented by the other methods we have compared in this study. Importantly, our testing pipeline and the input data needed to run the simulation are available in a Github repository, https://github.com/ruolin/ASmethodsBenchmarking. The whole pipeline is documented, interested readers can repeat the study and test the results with their preferred differential AS detection tools.

Method Classification

Methods for detecting AS may be categorized into two quantification schemas: count-based models and isoform resolution models (Figure 1). These two terms are based on the classification nomenclature defined by Pachter in (17). We selected eight methods and evaluated them based on simulated and real data. Six of them are from count-based models: DEXSeq (25), DSGseq (26), SplicingCompass(27), MATS(28), rDiff-parametric(29) and SeqGSEA(30). The remaining two, Cufflinks (31) and DiffSplice (32), use isoform resolution models. A brief overview of the eight methods follows.

Count-based models

The count-based models are based on the methods used to quantify transcripts with single isoforms. The number of reads falling on a transcript (adjusted for transcript length and the total number of mapped reads), like RPKM (Reads Per Kilobase per Millions of reads mapped), is used as an estimate for abundance (17). Count-based models are commonly used in differential gene expression. For differential splicing, the count-based models are modified to count reads in smaller counting units (i.e., exons) rather than the whole transcript regions. Also the focus changes to the differential expression of the counting units. Count-based models usually configure each gene into a single representation consisting of counting units. Counting units can be full or truncated exonic regions (e.g., DEXSeq and

DSGseq), or junction regions (MATS). Although the count-based model does not directly address the issue of quantifying isoform abundances, the DSGseq authors prove that the reads at counting units can fully reflect isoform expression as long as there is no isoform that can be composed by the combination of other isoforms (26). The count-based model can be seen as testing of two possible splicing outcomes, inclusion and/or exclusion, of each counting unit. Some papers refer to this model as an event-based model (15). Methods using the count-based model are usually dependent on existing annotation on the gene structure and typically employ Poisson, generalized Poisson or Negative Binomial (NB) distributions to model the read counts on counting units. For RNA-seq, the NB distribution is considered better suited for the analysis of biological replicates than the Poisson distribution, as it is able to account for overdispersion in replicate counts (33; 34).

SeqGSEA(30) and DSGseq (26) are examples of count-based models. These two methods are similar in many ways. Given a known set of transcripts at a locus, they both flatten these transcripts into a union transcript consisting of counting units (called mathematical exons in DSGseq and sub-exons in SeqGSEA). Both DSGseq and SeqGSEA model the number of reads that fall on the counting units as NB random variables after adjusting for overall gene expression. For a given gene, they calculate $\hat{p}_{ij}$ as the expected read count fraction of counting units i in group j and variance of $\hat{p}_{ij}$. Both methods define a gene-wise statistic to measure the difference in the expected read count fraction across two conditions by averaging over all counting units and adjusting for variance. Both methods mention that the null distribution is hard to obtain based on such statistics. SeqGSEA uses a permutation based approach to calculate the p-values while DSGseq just reports the statistics and does not calculate the p-values. Both DSGseq and SeqGSEA report which gene is alternatively spliced. A novel AS gene can be predicted only if an annotated constitutive exon is found to be a skipped exon. DSGseq can also tell you where the skipped exon may actually occur.

Like SeqGSEA and DSGseq, DEXSeq (25) transforms known gene models to sets of counting units (called counting bins in DEXSeq) based on any possible splice sites. The

difference is that DEXSeq uses a generalized linear model (GLM) to detect the differential usage of counting units. The GLM in DEXSeq assumes a NB model for the counts. DEXseq reports which counting unit is alternatively used across conditions and, like SeqGSEA and DSGseq, a novel skipped exon can be predicted only on an annotated constitutive exon.

The rDiff (29) package consists of two methods: rDiff-parametric and rDiff-nonparametric. rDiff-parametric is a count-based model. Unlike other count-based methods it only makes inference on regions that are not shared among all isoforms (called alternative regions). rDiff-parametric uses the NB distribution to model the number of reads on counting units to account for biological variance. Unlike SeqGSEA and DSGseq, the variance is calculated from an empirical variance-mean relationship (29). A p-value is calculated on each alternative region within a gene, and Bonferroni(BF) correction is used to obtain a genewise p-value. As a result, rDiff-parametric reports which gene is a significant AS gene but no novel AS gene can be found. The BF correction is known to be very stringent,which could explain why rDiff-parameteric has very low recall but high precision (see results section).

MATS (28) first retrieves all AS events from input gene models and annotates the identified AS events with the corresponding AS types (e.g. SE, IR, A3SS). More specifically, it cannot detect novel AS events and only retrieves the simple AS events, not complex ones. MATS calculates a statistical metric called exon inclusion level, ψ, which is the proportion of the reads that exclusively support one outcome of the events to reads that exclusively support another outcome of the identified events. The exon inclusion level is always between 0 and 1. Then, the posterior probability of the difference of exon inclusion level across two samples which is larger than a user-defined cutoff, denoted $p(|\psi_1 - \psi_2| > c \,|\, data)$, is calculated. MATS reports which AS event is significant rather than which gene is alternatively spliced. MATS differs from other count-based model methods in that it uses Bayesian approaches. It is also the only method that does not assume independence of two biological conditions. A bivariate uniform prior is used to model the dependence. Information across genes is borrowed in the process of estimating the common prior. Although the method in

MATSs original paper is only designed for a two sample comparison, the latest version of MATS (3.0+) accepts multiple replicates. However, it is unclear how the program models biological variability.

Like DEXSeq and DSGseq, SplicingCompass (27) uses a union transcript model for each gene. However, it does not utilize any statistical model based on the counting process. SplicingCompass first constructs vectors of read counts on exons as well as on splicing junctions for each gene and sample, then calculates pairwise geometric angles between two vectors. Finally, a one-sided t-test comparing the within condition angles and between condition angles is carried out for each gene. SplicingCompass reports which gene is AS gene based on the t-test. Therefore a novel AS gene can be found if the aforementioned test turns out to be significant. Again only SE can be detected.

Isoform resolution models

Isoform resolution models (also called multi-read models (17)) are multi-isoform models. Instead of transforming the question into detecting differential usage of counting units, they seek to directly solve this problem by comparing the relative isoform abundance across samples and/or conditions. The estimation of the isoform proportion vector q is usually done by maximizing a likelihood function $L(q|observing\ a\ set\ of\ reads\ alignments)$. Maximizing this likelihood function is equivalent to maximizing the likelihood of selecting a read or fragment from a transcript (31). Isoform resolution models try to assign reads or fragments to the transcripts they came from at the cost of introducing additional uncertainty in read assignments due to the overlap between isoforms. In count-based models there is no ambiguity in assigning reads toward counting units. It is worth mentioning that this question is also connected to the question of transcriptome assembly as novel transcripts are found in nearly every RNA-seq study (17).

Cufflinks (31) and DiffSplice (32) are examples of the isoform resolution models. Cufflinks contains three independent but connected programs: Cufflinks, Cuffmerge and Cuffd-

iff. Cufflinks assembles and quantifies the aligned reads while Cuffdiff performs differential testing. Cufflinks uses a linear model (31) which includes a specific parameter for fragment length. This differentiates Cufflinks from other methods by allowing Cufflinks to take advantage of insert size information in paired-end data. In this sense, Cufflinks is more appropriate for paired-end reads. The estimate of relative abundance of a transcript is reported in the form of FPKM (fragments per kilobase per million mapped fragments) which is equivalent to RPKM in the single-end case. Cuffdiff performs tests for relative isoform abundance changes (called post-transcriptional overloading in the Cufflinks paper) using a one-sided t-test of the Jensen-Shannon Divergence metric (31). Cufflinks is able to assemble transcriptomes and is thus less dependent on the accuracy of gene annotation.

Rigorously speaking, DiffSplice(32) is not "Isoform resolution" but "alternative paths resolution". In DiffSplice, the alternative paths stand for the paths from the Alternative Spliced Module (ASM) in spliced graphs and each ASM has at least two alternative paths. An ASM is a region in splice graphs where isoforms differ from each other. ASM seeks to minimize the ambiguity in isoform resolution by only considering regions that are not shared by all isoforms. DiffSplice tests differential splicing on each ASM instead of whole transcripts. The relative abundances of alternative paths are estimated using the maximum likelihood method. The difference of the relative abundances compositions is measured using Jensen-Shannon Divergence metric (JSD). Both the DiffSplice and Cufflinks models are extensions of the model of (35). Cufflinks extends the model to the paired-end case while DiffSplice restricts it to ASMs. Like Cufflinks, DiffSplice is also capable of assembling the aligned reads onto the transcriptome. Therefore, both programs are able to detect novel AS events that are not in the annotation. However, the Cuffmerge from Cufflinks packages can merge the assembly with annotations to provide gene models with higher confidence while no previous knowledge of gene models is used by DiffSplice. In other words, annotation is not used in DiffSplice.

Results and discussion

These differential AS detection methods were first evaluated using simulated data with known ground truth, where we could control the level of differential splicing across conditions and other factors that may affect detection. The NB distributions were used to simulate read counts on genes. The mean and dispersion parameters for the NB distributions were estimated from heat shock data (36). The 5885 genes that are known to have at least two splice variants in the Arabidopsis TAIR 10 reference annotation were focused on in the simulation studies. Using our custom simulation pipeline (see Additional file 1), a set of 2000 genes was randomly chosen from the overlaps between the 5885 known AS genes and genes that have non-zero expression in real data sets. These 2000 genes were simulated to be alternatively spliced and are referred to as "true AS genes". Details about the simulation settings and procedures can be found in the Methods section.

In the simulation study, we evaluated the robustness of the methods by varying the degree of differential splicing, read depths, sample sizes and dispersion setting in different conditions. We set High, Medium and Low levels for AS ratio, two dispersion patterns and three levels of read depth (100x, 60x and 25x). In addition, we have compared the computational time required for running the analysis (Additional file 1: Table S1). We used two dispersion settings in the simulation. One allows the two conditions to use two different dispersion parameters in the NB distributions which are estimated from two replicated real RNA-seq data sets, whereas the other forces both conditions to have the same dispersion parameter which is estimated from the pooled RNA-seq data sets. We call these two settings different dispersion pattern versus same dispersion pattern (denoted by Diff vs Same). We also investigated the effect of sample size, from 3 to 8 samples per conditions. A simple notation $\text{High}^{\text{Diff}}_{\text{100x}}$ means a condition of read depth at 100, different dispersion pattern and high AS ratio across conditions.

All of these evaluations were carried out in terms of the Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) metric. The ROC curve depicts the

true-positive rate (TPR) of a method for different false-positive rates (FPR) by varying the threshold for given scores. TPR is defined as the proportion of the events that are known to be differentially spliced that test as positives. Similarly the FPR is the proportion of the events that are known not to have differential splicing that test as positives. As almost all AS detection software packages tightly control FPR, we restricted the ROC curves to the range of $0 - 0.2$ (Figure 2-4). The area under the ROC curve, or AUC, is the numerical measurement that summarizes the ROC curves. Here we calculated the AUC under the restricted ROC curves. Methods with larger AUC have better performance. The results of all simulation studies under the measurement of AUC are summarized in Table 1.

As ROC curves and AUC measure the discrimination power between non-differentially spliced (non-DS) gene and differentially spliced (DS) gene over an interval, scientists are often interested in the discrimination power at a single cutoff point. Therefore the recall and precision at a $P_{adj} = 0.05$ cutoff were used as a additional set of evaluation metrics (P_{adj} stands for multiple testing corrected p-value). Recall is equivalent to TPR while precision is the proportion of the events that test as positives that are actually true discoveries. Precision is also known as $1 - false\ discovery\ rate$. Evaluating on precision examines whether the methods are able to control the FDR at the claimed 0.05 level. DSGseq does not return p-values and was excluded from this evaluation and SeqGSEA did not report any gene under $FDR = 0.05$ when the sample size was 3. The results of other seven methods under the measurement of recall and precision or FDR at $P_{adj} = 0.05$ are summarized in Table 2.

For the real data, we first compared the results obtained by the different methods in terms of absolute number of significant AS gene calling, the overlap of results across software and the concordance of gene rankings. We further compared these results to a list of experimentally validated genes that are known to be alternatively spliced in response to ambient temperature changes. Finally we carried out an semi-RT-PCR study and compared the results of the computational methods using RNA-seq to the results from RT-PCR.

The effect of different levels of AS ratio in conjunction with dispersion pattern

Since the difference required between two isoform compositions to be biologically significant enough to call as differential splicing is an open question, we defined a parameter $PALT$ (Percentage of ALTernative isoform) to control the level of differential splicing in our simulation. $PALT$, whose range is from $0-1$, simply represents the relative abundances of alternative isoforms for given genes. For multi-transcript genes, we randomly chose one transcript as an alternative isoform while the rest of isoforms remained as standard isoforms across conditions. For each of given genes, all standard isoforms have relative abundances which summed to $1-PALT$. The PALT for 2000 true AS genes was set to 0.2 in the control group and 0.4, 0.6, 0.8 in the three treatment groups, corresponding to low, medium and high AS ratio levels. We investigated the effect of varying the AS ratio level under two dispersion patterns. As a result we carried out 6 simulation studies and denoted them in the format of $\text{High}^{\text{Diff}}_{\text{100x}}$, representing the situations for high AS ratio, different dispersion patterns for two conditions and 100x read depth.

The restricted ROC curves of the 8 selected methods based on 3 simulation studies on different dispersion patterns are shown in Figure 2. As $PALT$ changed from 0.8 to 0.4, the difference between the isoform compositions under the two simulated conditions became smaller. All methods lost their discrimination power as the signal of differential splicing became weaker. The results from simulation studies with the same dispersion pattern were similar and are shown in the (Additional file 1: Figure S9). When two simulated conditions had different dispersion patterns, DEXSeq performed well in high and medium AS ratio situations but not in the low AS ratio situation.(Figure 2 and Table 1). When two conditions had the same dispersion pattern, DSGseq consistently performed the best out of the 8 methods (Table 1). As we focused on the low AS ratio in both dispersion situations, Cufflinks performed the best.

Both AUC and recalls were affected by the change of the AS ratio but the effect on recalls seemed to be larger. Taking Cufflinks as an example, the recall rates were 57%,

40% and 3% at high, medium and low levels of differential splicing respectively (Table 2). However the AUC dropped only 14% from high to low alternative splicing ratio (Table 1). It is not surprising that AUC is a more robust measurement than recall and precision. But it is not uncommon for people to use a single cutoff point, e.g. declare significance at $FDR = 0.05$. In this sense, the low AS ratio has a severe impact on the discrimination power (Table 2). DiffSplice achieved the highest recall in both $\text{Low}_{100\text{x}}^{\text{Diff}}$ and $\text{Low}_{100\text{x}}^{\text{Same}}$. However, its performance under the measurement of AUC (Table 1) was far from satisfactory since many AS events were not detected by using ASM and some detected ASMs were simply artifacts. In the baseline simulation study $\text{High}_{100\text{x}}^{\text{Diff}}$, 2123 ASMs were reported by DiffSplice and 94 of them resided at least 1kb away from coding regions. 4 ASMs were even longer than the longest gene (which is 31257 nt long) in Arabidopsis TAIR 10 model.

When considering the ability to control for false discoveries, all methods except MATS performed more poorly when the AS ratio became smaller (Table 2). Only MATS was able to control the FDR at all levels of AS ratio and dispersion pattern. SplicingCompass and rDiff-parametric could control the FDR at the desired 0.05 level in the simulation studies with high AS ratio but failed at low AS ratio, low levels of coverage. DEXSeq and rDiff-parametric's abilities to control FDR improved if the data shared the same dispersion pattern across conditions. With same dispersion pattern, rDiff-parametric was able to perfect control the FDR in all three AS ratios while DEXSeq achieved the desired FDR level on low AS ratio but not on high AS ratio. Although DEXSeq had the best performance in terms of AUC, it did a poor job in controlling the FDR (Table 2).

Detecting novel splicing events

We simulated RNA-seq reads using the latest Arabidopsis TAIR 10 gene sequences and models. This implies that no AS event is novel to this annotation. Theoretically methods that use annotation information should be able to find all candidate AS regions provided the annotation is correct. However in a real RNA-seq study, even in model organisms,

there may be many novel splicing events. To simulate this case, we deliberately removed the mRNA model of the alternative transcripts from annotation for the set of true AS genes. The relative abundances of alternative transcripts are controlled by $PALT$ and are the dominant force in the simulated AS events. By running the software using this incomplete annotation, we evaluated their abilities to detect novel splicing events. This comparison was evaluated on the baseline simulation study $\text{High}^{\text{Diff}}_{\text{100x}}$ (Figure 3). Except for DiffSplice, the performances of all other methods were degraded. Because DiffSplice does not use annotation information, its performance did not change. Overall, Cufflinks was more robust to incomplete annotation than other methods. MATS and DEXSeqs performances dropped significantly, suggesting that these two methods are very dependent on accurate annotation.

The effect of different AS types

Based on the gene models and sequences of the 5885 annotated AS genes in TAIR 10 annotation, we simulated 2000 true AS genes to be differentially spliced. However, most of the genes (1335 out of 2000, 67%) have more than one AS type. This made testing the performance in terms of the effect of different AS types difficult. Also as some methods, e.g. MATS and DiffSplice, test on individual events or local regions while others work on the gene level, the previous comparisons were not based on common ground. To overcome these problems, we picked out 1755 genes that have exactly two transcripts and a single splicing event from the 5885 genes. We then reevaluated all methods on these 1755 genes in the baseline simulation study. This equated the detection on a gene level to the detection on a splicing event. We classified these 1755 genes into three new sets by their splicing event types which include exon skipping, intron retention and alternative donor/acceptor sites (Figure 4). We treated alternative donor sites and acceptor sites together as a single class because there is almost no difference in detecting them from mathematical and computational perspective. 803 genes had an alternative donor or acceptor event, 850 showed

intron retention and 102 demonstrated exon skipping and about one third of genes in each new set were pre-selected AS genes (274, 275 and 38 respectively). We evaluated the eight methods in each category. This is a simplified scenario where a gene has exactly one AS event.

DEXSeq achieved the highest AUC in two of the three simple event classes, IR and SE, (Table 1). In these two cases, the exons or introns are either included or excluded as a whole. However in the cases of A3SS and A533, the counting units could be as short as several bps. DEXSeq may not have enough read counts to perform reliable statistical tests in such short regions. We observed that Cufflinks which uses isoform-resolution models perform the best for A3SS and A5SS. When the complex AS events were excluded MATS's improvement was very significant. The averaged AUC for MATS was 0.5763 when complex AS events were included. It, however, averaged at 0.9143 in the simplified scenarios (Table 1). This agrees with our observation that MATS is not capable of discovering complex AS events. In the simple scenario MATS acquired the highest recall and lowest FDR at $P_{adj} = 0.05$ threshold in all simple AS events (Table 2). As we looked at the individual types of AS events, DSGseq performed well for detecting IR but not so well on other splicing types. Similarly, Cufflinks performed well at A3SS and A5SS but poorly with other AS types, indicating a bias in detecting different AS types.

The effect of sample sizes and read depth

The increase in sample size from 3 to 8 did not have a significant impact on the AUC statistics and the methods' rankings based on the AUC (Table 1). Even for the recall and precision statistics (Table 2), the increase in sample size had a small impact for all methods except for SplicingCompass and SeqGSEA. Recall for SplicingCompass increased from 14% to 50% when the sample size increased from 3 to 8. SeqGSEA was not statistically significant at $FDR = 0.05$ for a sample size of 3 but achieved a recall of 95% at the cost of having a low precision (58%) in a sample size of 8. However the ROC curves and AUC statistics

for SeqGSEA were almost the same for the different sample sizes (Additional file 1: Figure S10). A possible explanation is that the permutation-based approach used in SeqGSEA may scale the P_{adj} according to the sample size. Therefore, we would recommend a sample size between 4 to 7 for using SeqGSEA.

Most methods were robust to different read depths or coverage of RNA-seq with a minor drop of discrimination power as read depth decreased (Table 1 and Additional file 1: Figure S11). However it is interesting to note that Cufflinks achieves its best discrimination power at $RD60$ and ranked 1st among all methods at this read depth (Table 1). This may suggest that Cufflinks performs better when read depth is around 60.

Real RNA-seq data from Arabidopsis heat shock experiment

In addition to the simulated data, we also evaluated the methods on heat shock RNA-seq data sets (36). Three RNA-seq samples were generated from heat shock T1 group and two from control T1 group (See Methods for a description for the heat shock data sets). All the eight methods except for DiffSplice are able to handle the unbalanced design with different sample sizes. For DiffSplice, we took out one sample from the heat stress group to make it a balanced design. All genes found to be AS at the threshold of $FDR = 0.05$ were consider statistically significant. DSGseq does not report a p-value and therefore was not used for this comparison.

We first compared the number of significant AS events found by each method (Table 3). SeqGSEA did not find any gene with significant AS. This result was consistent with our simulation studies that SeqGSEA usually requires a sample size larger than 3 to declare significance at the $FDR = 0.05$ level. For the rest of the methods, the highest number of significant AS events was found by Cufflinks, followed by MATS and DEXSeq. The most conservative method was SplicingCompass as shown in Table 3.

We also examined the overlaps of the set of significant AS genes found by each methods (Figure 5, Table 3). From Table 3, we noted that SplicingCompass was very conservative

(having the smallest number of significant DS genes) and was also very "unique"in that it almost did not share any significant DS genes with other methods. The Venn diagram (Figure 5) did not include SplicingCompass. The results showed that the methods were very different from each other in that there was no gene that found by all five methods and that the proportion of genes that were found exclusively by each method was more than half. rDiff-parametric had 48.4% genes that were shared by at least one other methods. It was the only one that was close to 50% level. DEXSeq shared 40% of rDiff-parametric reported DS genes.

We further compared the results of all eight methods by investigating the correlation of gene ranking scores (computed as previously). We computed the Spearman rank correlations between all pairs of the eight methods and visualized it using a heat map (Figure 6). The correlations were calculated based on the ranking scores from 600 common genes that were reported by all methods. The highest correlation was observed between DSGseq and SeqGSEA as both methods use NB statistics (see Methods). Overall, the correlations were very low which indicated that these methods tended to rank genes differently with respect to alternative splicing.

A list of experimentally validated AS genes which are known to exhibit AS in response to temperature changes

Since there have been studies that have linked some genes to alternatively spliced variants in response to heat stress, we came up with a list of six experimentally validated AS genes based on a search of the literature. AT1G01060 encodes LHY, a transcription factor involved in regulation of circadian rhythm. An A3SS event, encoding a 3-nt difference, has been found to occur as the ambient temperature changes (37). This alternative splicing event has been confirmed by high resolution RT-PCR (37). AT1G16610 encodes SR45, a member of SR proteins. AT1G16610 has two splice variants which differ by a 21-nt sequence which is present in SR45.1 but absent in SR45.2 (38). It has been found that the relative

abundance of SR45.2 is increased as temperature goes up (38). Another two SR proteins, SR1/SR34 (AT1G02840) and SR30 (AT1G09140), have been reported to be alternatively spliced in response to heat stress (39; 40; 6; 41). In both cases, relevant transcripts differ by several hundred nts (337 nts in SR30 and 352 nts in SR1/SR34). All of the above AS events are A3SS. AT1G77080 encodes FLM, a MANS domain protein which regulates flowering. A mutually exclusive exon event has been found in this gene which is subject to temperature changes (42). The P5CS1 gene (AT2G39800) contains an exon-3 skipping event that is subject to temperature variation (43). The SR45a gene (AT1G07350) also contains an alternatively spliced internal exon and the proportion of exon-skipped transcript increases when exposure to heat stress. We illustrate the SR45a gene model and junction read alignments in different conditions using the Integrated Genome Browser (44) (Figure 7). Similar illustrations of the read pileups for the rest of genes are given in the Additional file 1.

At the cutoff $FDR = 0.05$, MATS identified all seven genes and successfully located the actual genomic regions. DEXseq found two of them (SR1/SR34 and SR30) and Cufflinks reported one (FLM). None of the other methods were able to find these genes. For LHY and SR45, the A3SS events encompass a range of nt differences from a few to tens. MATS's success in finding these events can probably be attributed to the exclusive use of junction reads. The small differences were easily overlooked by other methods that take into account of reads on full exonic regions. The junction reads that uniquely supported the A3SS events tend to be overwhelmed by the non-junction reads along the long exon (see the visualized read alignments in Additional file 1). DEXseq detect SR1/SR34 and SR30, with the differences in the A3SS events are several hundreds nt long. In DEXSeq, the junction reads are used as exon body reads.

PCR validation of the real data set

In a separate study, we used semi-quantitative PCR to characterize heat induced splicing changes in seven genes that were annotated in TAIR 10 as being alternatively spliced.

These seven genes thus provided a useful positive control for estimating the accuracy of the splicing analysis methods described here. These seven genes are AT1G77180, AT1G01490, AT2G02390, AT2G26670, AT3G19720, AT5G26780, AT1G09140. At the cutoff $FDR = 0.05$, MATS reported five genes, followed by Cufflinks and DEXSeq, both of which picked out four genes. DSGseq, DiffSplice and rDiff identified one gene. The details about which methods picked out which genes and which AS events are contained in the seven genes are provided in table 4.

Conclusions

In this paper, we have evaluated and compared eight methods for alternative/differential splicing analysis of RNA-seq data. The major observations for the AS methods are summarized in Table 5. These methods are classified into count-based models and isoform resolution models. Count-based models transform the question of AS analysis into the question of alternative usage of counting units while isoform resolution models seek to resolve the isoform relative abundances and in further compare the difference across conditions. Only Cufflinks and DiffSplice in our comparison belong to isoform resolution models. We've conducted both simulation studies and studies using real data to evaluate the methods. We created a customized simulation pipeline based on Flux Simulator. This pipeline allows users to repeat the simulation with different alternative splicing ratios, read depths and sample sizes.

From the perspective of AUC statistics, DEXSeq and DSGseq performed well in the simulation studies when the annotation is accurate and complete. DEXSeq was slightly better when two groups of samples were simulated using different dispersion parameters while DSGseq excelled when the same dispersion parameter is used. DSGseq is also more robust to changes in the AS ratio than DEXSeq. The drawback of DSGseq is that it does not calculate p-value. Both methods belongs to count based models. However, like other methods which depend on gene models, they performance was largely impaired when

incomplete annotation was used. This may impose problems when working on non-model species or simply any species that are not well annotated. Cufflinks and DiffSplice are capable of assembling reads into transcripts and are thereby able to detect novel AS events. Only Cufflinks can take advantage of established gene models and is not fully dependent on the prior knowledge. These attributes render Cufflinks the best combination of accuracy and robustness against incomplete annotation. Therefore it is recommended for non-model species. On the other hand, Cufflinks achieves a better tradeoff between precision and recall. It also performs the best in an median read coverage of 60. The change of AS ratio affected methods' discrimination power as well as the ability to control FDR. The rankings, however, were relatively stable as AS ratio changed, indicating that most methods is generally good enough to analyze real RNA-seq experiments where the splicing ratio might vary from gene to gene.

MATS uses a Bayesian framework to calculate the probability of a gene being alternatively spliced. Although MATS did not exhibit good performance under the evaluation of ROC curves and AUC, it was the best method under our comparison with respect to controlling the FDR at a proposed level. MATS excels in the precision of its results, which is very important for most biologists. The reason MATS had low recall and AUC is that MATS was only designed for detecting simple AS events. Therefore it was not satisfactory when the simulation included complex AS events. When only genes with simple AS events were involved, both recall and AUC improved dramatically for MATS. The superb performance of MATS in real data is boosted by the fact that all the 6 validated AS genes from the literature as well as for the 7 PCR validated AS genes are simple AS genes. rDiff-parametric also had a low FDR, however, but it appears to be due to its use of BF correction. In the analysis of heat shock RNA-seq data, MATS turned out to be the method that was the most consistent with the established experimental evidence as well as our PCR validations. The drawback of the MATS is that it is highly dependent on the goodness of annotation but it would be recommended for validating known AS events.

Large sample size (8 samples per condition) did not affect the discriminating power under ROC and/or AUC evaluation, but did improve several methods' recall at the cost of decrease in precision. The several methods include Cufflinks, DEXSeq, SplicingCompass and especially SeqGSEA. SeqGSEA uses a permutation based approach to calculate p-values for genes being alternative spliced. It is likely that the p-values are scaled in accordance with sample size and we may expect a optimal sample size around 5 or 6 for using SeqGSEA. The sets of significantly alternatively spliced genes at given FDR threshold (FDR = 0.05) varied considerably between methods for the analysis of heat shock data. SeqGSEA and DSGseq had the highest correlations of the gene ranking scores due to using the same test statistics.

Methods

Parameter choices of software

All of the selected methods in this paper allow users to specify certain parameters. We have mostly used the default parameters as this is how most users apply these software packages. The detailed command lines and parameter choices used in the baseline simulation study are given in the Additional file 1. The version of each program used for the evaluations in the main paper is also given. For those that are implemented in R, including DEXSeq, SeqGSEA and SplicingCompass, it contains sample R code to run the analysis. For more detailed information, e.g., the meaning of the parameters and/or the whole list of parameters, we refer to the original publications.

For MATS, we used the mapping results instead of fastq files as the program input. Starting with MATS (3.0+), the program outputs two types of results: analysis based on both exon body reads as well as junctions reads and analysis based on junction reads alone. For all the comparisons, we used the latter but we showed in the Additional file 1 that there are only negligible differences in these two results. For Cufflinks, we first assembled

each sample individually using Cufflinks and then merged the resultant transcripts with annotation using Cuffmerge. The merged transcripts was used in Cuffdiff to perform the analysis of differential splicing. We used the fragment bias correction option in Cufflinks. In the analysis of heat shock data, the minimum number of replicates were set to 2 because one of the conditions has only two samples.

SeqGSEA integrates analysis regarding differential gene expression (DE) with analysis regarding differential splicing (DS). We only performed the latter and calculated the DS permutation p-values for 1000 iterations.

Heat shock data sets

In the heat shock experiment (36), RNA was harvested from two experimental conditions (heat vs control) at two time points (T1 and T2). Previously grown in the same normal conditions, 3-week-old Arabidopsis plants were divided into 2 groups. In the heat shock group, plants were put into an incubator with temperature set to 38 °C during a 3 h treatment. The first set of plants were collected immediately after the 3 h treatment and the second set of plants were harvested 24 h after the treatment. The first time point was designated as heat shock period and the second time point was designated as recovery period. In the control groups, the incubator was set to 22 °C during the 3 h heat treatment and two sets of plants were collected from that incubator at T1 and T2 respectively. The RNA-Seq alignments used in this study are available for visualization in the Integrated Genome Browser via the IGB Quickload site http://www.igbquickload.org/abiotic. IGB is freely available from http://www.bioviz.org.

Simulated RNA-seq data sets

We generated Arabidopsis RNAseq data using Flux Simulator (45) with exact ground truth expression levels. Arabidopsis is chosen because of its relatively small genome size and detailed genomic annotation. Two real data sets, Heat shock T1 and Heat shock T2, each

with three replicates were used for generating simulated data. There was a good agreement between the simulated data by NB distributions and real data (Additional file 1: Figure S2).

We created a custom simulation pipeline (see Additional file 1) to create synthetic Arabidopsis RNA-seq data simulating different conditions. Flux Simulator is a single sample generator which carries out in-silico RNA-seq experiments. It starts with a random transcript population and then carries out library construction processes. Finally, it simulates the sequencing process including size selections, and platform-specific base calling errors. Our simulation pipeline extends the Flux Simulator capabilities to simulating differential splicing on two conditions with biological replicates. The simulation is a two-step workflow (Additional file 1: Figure S1). 1) First, we set empirical total transcript copy numbers for each gene and each sample based on real data and randomly choose genes for differential splicing across the conditions. The number of simulated replicates can be specified by the user. 2) Second, the transcript-level abundances are calculated based on the previous total transcript copy numbers, relative isoform proportions, and sequencing depth. Then, Flux Simulator can generate in-silico RNA-seq reads based on transcript-level abundances.

The custom simulation pipeline generated 100bp paired-end reads in fastq format. The relatively long read length(100bp) was deliberately chosen to produce more reads that cross exon-exon junctions. The generated synthetic reads were then mapped against the latest Arabidopsis genome TAIR 10 using the GMAP and GSNAP packages (version 2013-05-09) (46). To maximize GSNAP's ability to find spliced alignments, we used the RIKEN Arabidopsis full length cDNA sequences (47). These sequences were utilized by GMAP with an option "-f" that looked for all possible splice sites and reported them to GSNAP as a database of known splice sites. The alignment results were output in SAM/BAM format which can be used for the subsequent alternative splicing analysis.

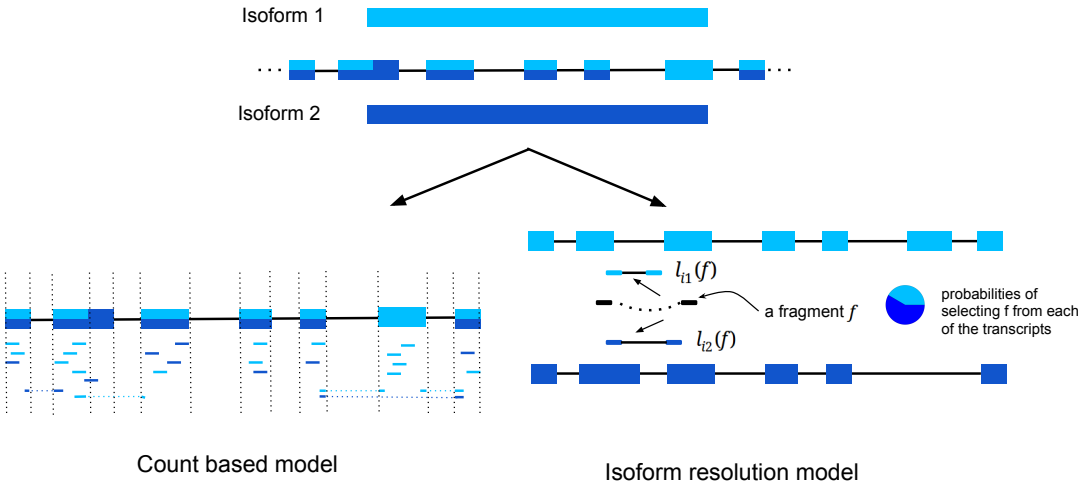


Figure 2.1 Quantification schema. A simplified gene model consists of two expressed isoforms (Top). Exons are colored according to the isoform of origin. Two model types used for quantification purpose (Bottom). In the count-based models (left), reads are assigned to counting units (shown by dash lines) without ambiguity. For each counting unit the model can be viewed as a test on two possible outcomes (spliced in or spliced out). The isoform resolution model is shown on the right where two ends of a read pair (show as dark solid boxes connected by curly dash line) align upstream and downstream of an alternative donor site. $l_{i1}(f)$ is the length of alignment of fragment f to isoform $i1$, and is shorter than $l_{i2}(f)$. Therefore if the fragment size distribution is known, it is possible to infer which isoform is more likely to generate f. Note that transcript effective length, i.e. $l_{i1}(f)$, $l_{i2}(f)$ and other parameters (depends on model you use) might also affect the probability of assigning reads to isoforms. Usually a maximum likelihood based approach is used to optimize this probability.

Evaluation of the software results

We defined ranking scores for each method directly from the output. This score is a direct reflection of significance or evidence for alternative splicing across two conditions. For the six methods that provide adjusted p-values after multiple testing correction, we defined the score as $1-P_{adj}$. Rdiff use Bonferroni correction while SplicingCompass, MATS, DEXSeq, SeqGSEA and Cufflinks-Cuffdiff use Benjamini-Hochberg correction. DiffSplice and DSGseq do not provide p-values, and so we used their test statistics as the ranking scores: square root of JSD for DiffSplice and NB statistics for DSGseq (see the method overview in Background).

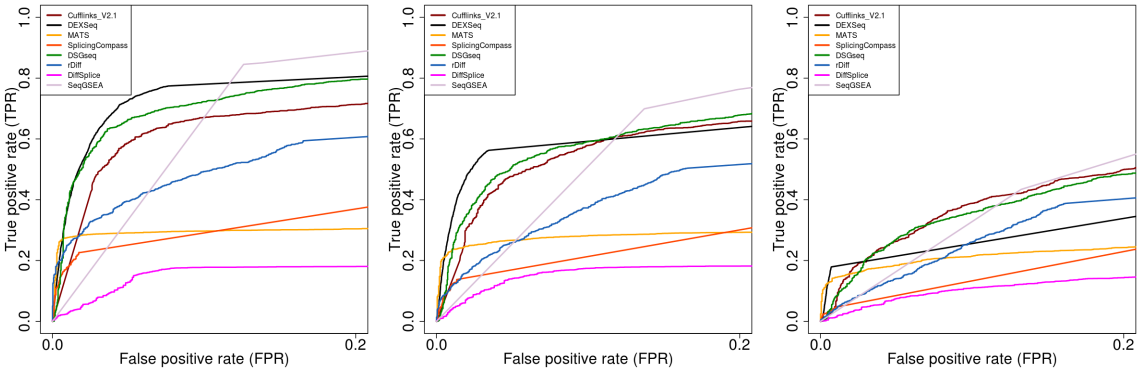


Figure 2.2 ROC curves evaluation for three levels of AS ratio when two groups of samples have the different dispersion pattern. ROC curves for eight selected methods in simulation studies $\text{High}^{\text{Diff}}_{100\text{x}}$ (left panel), $\text{Medium}^{\text{Diff}}_{100\text{x}}$ (middle panel), $\text{Low}^{\text{Diff}}_{100\text{x}}$ (right panel). These ROC curves are obtained at a simple size of 3 for each condition. When the level or degree of DS across conditions become smaller (panel left-right), the power of discrimination of true-DS and non-DS drops significantly. However the relative ranking of each methods tend to be unchanged. DEXSeq perform consistently the best with respect to all three simulation studies.

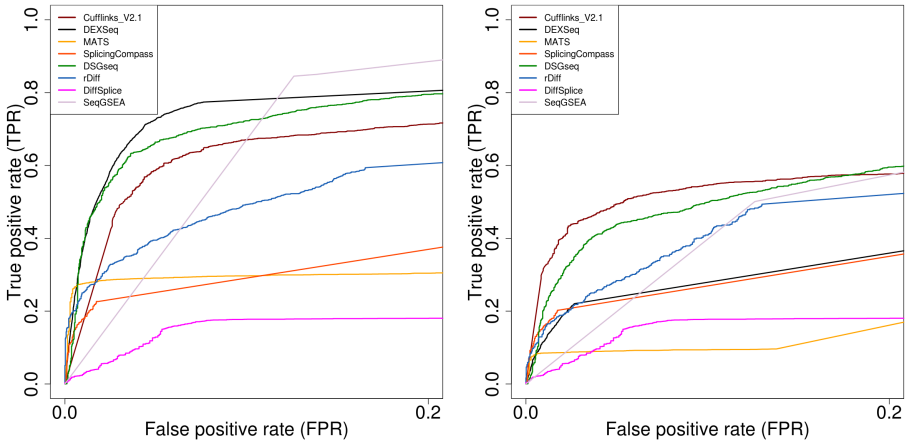


Figure 2.3 ROC curves evaluation for accurate and incomplete annotation. ROC curves for eight selected methods using simulation study $\text{High}^{\text{Diff}}_{100\text{x}}$ with complete annotation (left panel) and incomplete annotation (right panel). Isoform resolution model methods, such as Cufflinks, are more robust to incomplete annotation compared with count-based models methods.

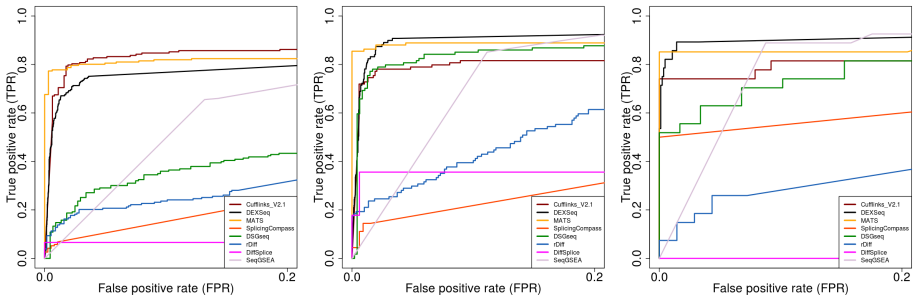


Figure 2.4 ROC curves evaluations for three splicing classes. ROC curves of eight selected methods based on 1755 genes containing single splicing event from simulation study $\text{High}^{\text{Diff}}_{100\text{x}}$. These 1755 genes were further divided into three splicing event classes: 803 genes with alt. donor/acceptor sites (left panel), 850 genes with intron retention (middle panel), 102 genes with exon skipping (right panel).

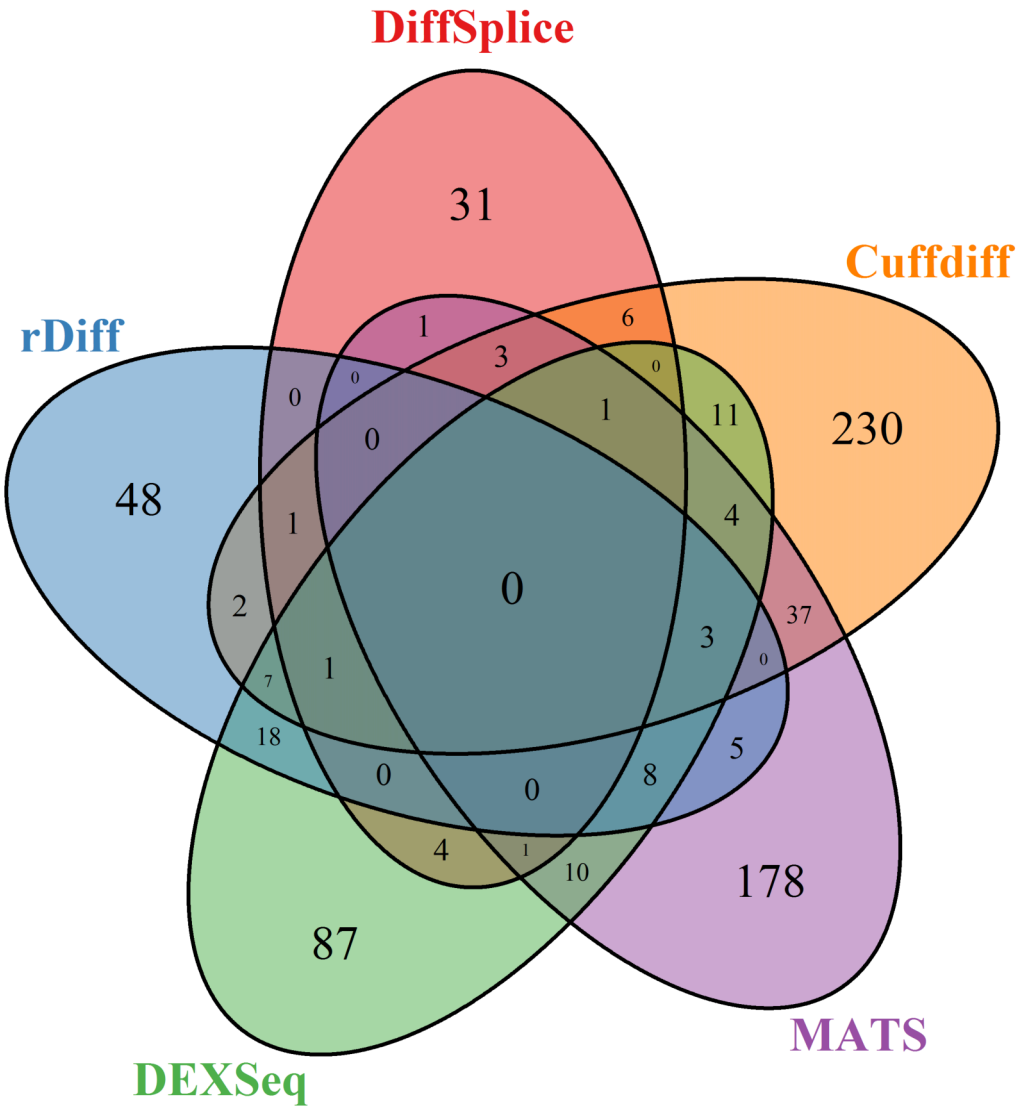


Figure 2.5 Venn digram of heat shock data set. Overlap among the set of DS genes found by 5 methods. SplicingCompass is not included because it almost shares nothing with other methods based on table 3.

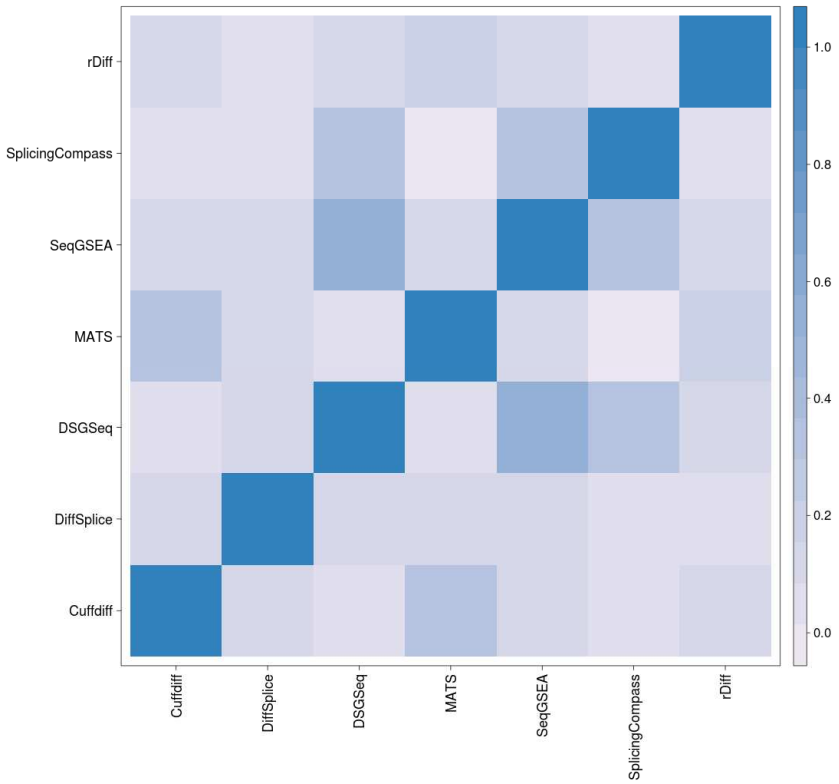


Figure 2.6 Heat Map for correlation of the gene ranking scores obtained by the different methods for heat shock data set. The correlations are generally low for any two methods, indicating the methods are very different. Two methods both using NB statistics (DSGseq and SeqGSEA) achieve the highest Spearman rank correlation of 0.52.

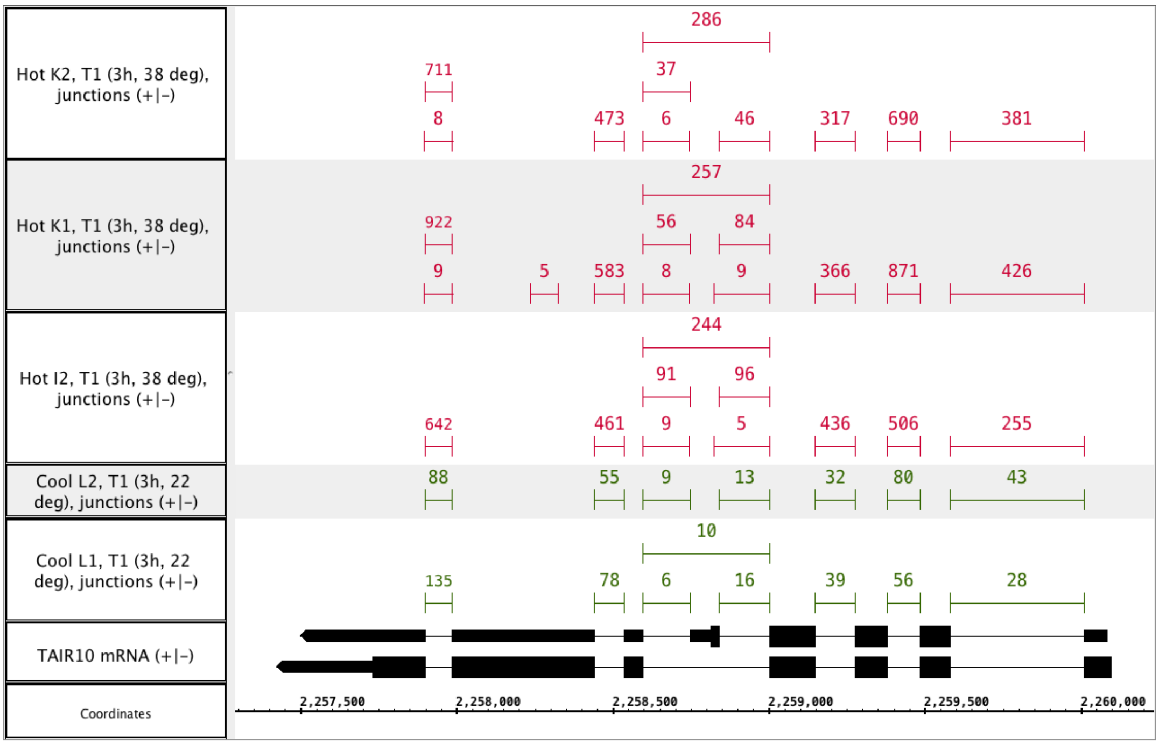


Figure 2.7 SR45a. Heat-induced differential splicing of Arabidopsis gene SR45a (AT1G07350) encoding an RNA-binding protein involved in splicing. Tracks labeled Hot and Cool contain exon-exon junction features inferred from spliced read alignments from heat-treated (hot) and control samples (cool). Junctions with fewer than five supporting reads are not shown. Two annotated gene models for SR45a are shown in the track labeled TAIR 10 mRNA. Taller blocks indicate translated regions of the gene model. Note that inclusion of an internal exon introduces a premature stop codon that interrupts translation and the exon-skipped form likely encodes the full-length protein. The gene is on the minus strand of chr1 and so transcription proceeds from right to left.

Table 2.1 Area under the ROC curve (AUC) and relative ranking measured under all simulation studies. Larger values of AUC indicate better performance.

	Cufflinks	**DEXSeq**	**MATS**	**SpComp**	**DSGseq**	**rDiff-param**	**DiffSplice**	**SeqGSEA**
$\text{High}^{\text{Diff}}_{\text{100x}}$	0.7765(3)	**0.8435(1)**	0.6066(7)	0.603(6)	0.8214(2)	0.704(5)	0.5262(8)	0.7699(4)
$\text{Medium}^{\text{Diff}}_{\text{100x}}$	0.7334(3)	**0.7583(1)**	0.5960(6)	0.5612(7)	0.7472(2)	0.6421(5)	0.5276(8)	0.7055(4)
$\text{Low}^{\text{Diff}}_{\text{100x}}$	**0.6369(1)**	0.5847(4)	0.5583(6)	0.518(7)	0.6288(2)	0.5807(5)	0.4982(8)	0.6155(3)
$\text{High}^{\text{Same}}_{\text{100x}}$	0.7751(4)	0.8351(2)	0.6046(6)	0.5998(7)	**0.8373(1)**	0.6871(5)	0.5371(8)	0.7797(3)
$\text{Medium}^{\text{Same}}_{\text{100x}}$	0.7357(4)	0.7407(2)	0.5914(6)	0.5582(7)	**0.7669(1)**	0.6201(5)	0.5341(8)	0.7374(3)
$\text{Low}^{\text{Same}}_{\text{100x}}$	0.6487(2)	0.5546(5)	0.5506(6)	0.5159(7)	**0.6496(1)**	0.5773(4)	0.5049(8)	0.6297(3)
$\text{100x}^{\text{Diff}}_{\text{High}}$	0.7765(3)	**0.8435(1)**	0.6066(7)	0.603(6)	0.8214(2)	0.704(5)	0.5262(8)	0.7699(4)
$\text{60x}^{\text{Diff}}_{\text{High}}$	**0.8687(1)**	0.7667(2)	0.5861(6)	0.5688(7)	0.7648(3)	0.6848(5)	0.5266(8)	0.7338(4)
$\text{25x}^{\text{Diff}}_{\text{High}}$	0.6807(4)	**0.7432(1)**	0.5607(6)	0.5479(7)	0.6967(2)	0.6659(5)	0.5001(8)	0.6815(3)
Complete annot.	0.7765(3)	**0.8435(1)**	0.6066(7)	0.603(6)	0.8214(2)	0.704(5)	0.5262 (8)	0.7699 (4)
Incomplete annot.	**0.7271(1)**	0.5939(5)	0.5012(8)	0.5930(6)	0.7033(2)	0.6561(3)	0.5262 (7)	0.6425 (4)
A3A5SS	**0.8990(1)**	0.8574(3)	0.8948(2)	0.5283(7)	0.6272(5)	0.5732(6)	0.4811(8)	0.6932(4)
IR	0.8810(4)	**0.9368(1)**	0.9360(2)	0.5639(8)	0.8990(3)	0.6696(6)	0.6391(7)	0.7940(5)
SE	0.8795(3)	**0.9407(1)**	0.9177(2)	0.7500(6)	0.8301(5)	0.5916(7)	0(8)	0.8334(4)
8samples	0.7408(5)	**0.8495(1)**	0.6078(7)	0.7450(4)	0.8301(2)	0.7196(6)	0.5030(8)	0.7656(3)

The table contains the AUC and relative ranking for the methods under all simulation study. The ranking position is shown in the parenthesis. A3A5SS stands for the joint class of alternative 3' splice site event and alternative 5' splice site event. IR stands for intron retention event and SE stands for skipping exon event.

Table 2.2 Recall and precision at $P_{adj} = 0.05$ measured under all simulation studies. Recalls were shown as the numbers in the left column, precisions in the right column. Larger values of both metrics are better. Under a sample size of 3, SeqGSEA found no genes at $P_{adj} = 0.05$ and therefore no values were reported.

	Cufflinks		DEXSeq		MATS		SpComp		rDiff-param		DiffSplice		SeqGSEA	
	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
$\text{High}_{100x}^{\text{Diff}}$	0.57	0.91	0.53	0.65	0.28	0.98	0.14	0.95	0.06	0.99	0.24	0.79	-	-
$\text{Medium}_{100x}^{\text{Diff}}$	0.40	0.91	0.31	0.71	0.22	0.98	0.08	0.90	0.02	0.95	0.24	0.76	-	-
$\text{Low}_{100x}^{\text{Diff}}$	0.03	0.77	0.06	0.59	0.1	0.99	0.02	0.82	0.002	0.833	0.20	0.66	-	-
$\text{High}_{100x}^{\text{Same}}$	0.58	0.90	0.49	0.71	0.27	0.98	0.13	0.94	0.05	1.0	0.26	0.84	-	-
$\text{Medium}_{100x}^{\text{Same}}$	0.42	0.91	0.25	0.80	0.21	0.99	0.07	0.92	0.01	1.0	0.25	0.81	-	-
$\text{Low}_{100x}^{\text{Same}}$	0.15	0.91	0.04	0.96	0.08	0.99	0.02	0.84	0.001	1.0	0.21	0.68	-	-
$100x_{\text{High}}^{\text{Diff}}$	0.57	0.91	0.53	0.65	0.28	0.98	0.14	0.95	0.06	0.99	0.24	0.79	-	-
$60x_{\text{High}}^{\text{Diff}}$	0.49	0.91	0.29	0.72	0.22	0.99	0.09	0.93	0.02	1.0	0.25	0.81	-	-
$25x_{\text{High}}^{\text{Diff}}$	0.39	0.92	0.22	0.75	0.15	0.98	0.06	0.93	0.008	0.94	0.17	0.79	-	-
A3A5SS	0.73	0.95	0.71	0.71	0.85	1	0.04	0.875	0.01	1	0.07	0.85	-	-
IR	0.69	0.95	0.43	0.8	0.76	0.99	0.09	0.8	0.09	1	0.36	0.82	-	-
SE	0.67	1	0.71	0.91	0.85	1	0.38	1	0.04	1	0	0	-	-
Complete annot.	0.57	0.91	0.53	0.65	0.28	0.98	0.14	0.95	0.06	0.99	0.24	0.79	-	-
Incomplete annot.	0.42	0.92	0.14	0.41	0.08	0.97	0.12	0.93	0.008	0.94	0.24	0.79	-	-
8samples	0.65	0.81	0.66	0.55	0.3	0.93	0.50	0.82	0.06	0.99	0.17	0.72	0.95	0.58

A3A5SS stands for the joint class of alternative 3' splice site event and alternative 5' splice site event. IR stands for intron retention event and SE stands for skipping exon event.

Table 2.3 The number of shared differentially spliced genes detected by the selected methods for the HeatT1 data set.

	DiffSplice	**Cuffdiff**	**DEXSeq**	**MATS**	**rDiff-param**	**SplicingCompass**
DiffSplice	48	12	7	6	2	0
Cuffdiff		306	27	48	14	1
DEXSeq			155	27	37	3
MATS				241	16	0
rDiff-param					93	0
SplicingCompass						31

The table contains the number of significant differential spliced genes that reported by each methods (number on the diagonal) and numbers that are shared with another method

Table 2.4 The evaluation of the methods on the seven PCR validated genes.

Gene	**Found by which methods**	**AS events**
AT1G77180	DEXSeq, DSGseq, MATS	alt acceptor in 5 UTR
AT1G01490	None	retained intron in 5 UTR
AT2G02390	Cufflinks, DEXSeq, DiffSplice, MATS	4th exon alt acceptor
AT2G26670	Cufflinks, MATS	1st exon alt donor in coding region
AT3G19720	Cufflinks, MATS	intron retention 3rd to last exon
AT5G26780	Cufflinks, DEXSeq	intron retention last exon 3 UTR
AT1G09140	DEXSeq, MATS, rDiff-param	next to last exon alt acceptor

Table 2.5 Summary of the main observation for selected methods

	Class	Novel AS	Det. Region	Comments
DiffSplice	IR	Any type	ASM	Assembles transcriptome based on graph theory. Does not rely on annotation but does not use annotation either. The goodness of ASM is questionable. Generally low AUC. Performs poorly when detecting SE events.
Cufflinks	IR	Any type	Gene	Assembled transcripts merge with annotation to provide a more confident reference. Is least affected by incomplete annotation. Model is designed for pair-end data. Performs better for medium read depth than both low and high read depth. Performs better when detecting A3SS and A5SS events than other types of AS events. Computationally slow, but allows parallelization.
DEXSeq	CB	Only SE	Exon	Uses a generalized linear NB model. Achieves the highest AUC in many cases using accurate annotation. However, incomplete annotation can impose considerable problems for it. Poor FDR control.
MATS	CB	NS	AS event	Uses a Bayesian model. Solely based on junction reads. Cannot detect complex AS events. Annotates splicing events with corresponding event types. Good FDR control in many simulation studies. Performs the best for real data.
rDiff-param	CB	NS	Gene	Conservative with default settings. Good FDR control but low AUC in many cases. Computationally fast.
SplicingCompass	CB	Only SE	Gene	Compares geometry angles of read count vectors. Generally poor FDR control and Medium AUC. Performs well when detecting SE events.
DSGseq	CB	Only SE	Gene	No p-value reported. Generally medium AUC. Performs well when detecting IR events and when using incomplete annotation. Computationally fast.
SeqGSEA	CB	Only SE	Gene	Integrates DE analysis with DS analysis. Generally high AUC. Requires a sample size around 5 to claim significance at a reasonable FDR level, i.e. $FDR = 0.05$. Computation time increases dramatically as permutation times increases.

IR: Isoform resolution models

CB: Count based models

NS: Not Supported

ASM: Alternative Spliced Module

Bibliography

[1] Black, D. L. (2003). Mechanisms of alternative pre-messenger RNA splicing. *Annu. Rev. Biochem.*, 72:291–336.

[2] Lareau, L. F., Green, R. E., Bhatnagar, R. S., and Brenner, S. E. (2004). The evolving roles of alternative splicing. *Curr. Opin. Struct. Biol.*, 14(3):273–282.

[3] Syed, N. H., Kalyna, M., Marquez, Y., Barta, A., and Brown, J. W. (2012). Alternative splicing in plants–coming of age. *Trends Plant Sci.*, 17(10):616–623.

[4] Keren, H., Lev-Maor, G., and Ast, G. (2010). Alternative splicing and evolution: diversification, exon definition and function. *Nat. Rev. Genet.*, 11(5):345–355.

[5] Graveley BR, Brooks AN, Carlson JW, Duff MO, Landolin JM, Yang L, Artieri CG, van Baren MJ, Boley N, Booth BW, Brown JB, Cherbas L, Davis CA, Dobin A, Li R, Lin W, Malone JH, Mattiuzzo NR, Miller D, Sturgill D, Tuch BB, Zaleski C, Zhang D, Blanchette M, Dudoit S, Eads B, Green RE, Hammonds A, Jiang L, Kapranov P, et al (2011). The developmental transcriptome of Drosophila melanogaster. *Nat.* 2011;471(7339):473479.

[6] Reddy, A. S. (2007). Alternative splicing of pre-messenger RNAs in plants in the genomic era. *Annu Rev Plant Biol*, 58:267–294.

[7] Reddy, A. S., Rogers, M. F., Richardson, D. N., Hamilton, M., and Ben-Hur, A. (2012b). Deciphering the plant splicing code: experimental and computational approaches for predicting alternative splicing and splicing regulatory elements. *Front Plant Sci*, 3:18.

[8] Marquez Y, Brown JW, Simpson C, Barta A, Kalyna M. (2012). Deciphering the plant splicing code: experimental and computational approaches for predicting alternative splicing and splicing regulatory elements. *Front Plant Sci*, 3:18.

[9] Lamesch, P., Berardini, T. Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., Muller, R., Dreher, K., Alexander, D. L., Garcia-Hernandez, M., Karthikeyan, A. S., Lee, C. H., Nelson, W. D., Ploetz, L., Singh, S., Wensel, A., and Huala, E. (2012). The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucleic Acids Res.*, 40(Database issue):D1202–1210.

[10] Richardson, D. N., Rogers, M. F., Labadorf, A., Ben-Hur, A., Guo, H., Paterson, A. H., and Reddy, A. S. (2011). Comparative analysis of serine/arginine-rich proteins across 27 eukaryotes: insights into sub-family classification and extent of alternative splicing. *PLoS ONE*, 6(9):e24542.

[11] Reddy, A. S., Day, I. S., Gohring, J., and Barta, A. (2012a). Localization and dynamics of nuclear speckles in plants. *Plant Physiol.*, 158(1):67–77.

[12] Wang, B.-B. and Brendel, V. (2006). Genomewide comparative analysis of alternative splicing in plants. *Proceedings of the National Academy of Sciences*, 103(18):7175–7180.

[13] Campbell, M. A., Haas, B. J., Hamilton, J. P., Mount, S. M., and Buell, C. R. (2006). Comprehensive analysis of alternative splicing in rice and comparative analyses with Arabidopsis. *BMC Genomics*, 7:327.

[14] Xiao, Y. L., Smith, S. R., Ishmael, N., Redman, J. C., Kumar, N., Monaghan, E. L., Ayele, M., Haas, B. J., Wu, H. C., and Town, C. D. (2005). Analysis of the cDNAs of hypothetical genes on Arabidopsis chromosome 2 reveals numerous transcript variants. *Plant Physiol.*, 139(3):1323–1337.

[15] Alamancos, G., Agirre, E., and Eyras, E. (2013). Methods to study splicing from high-throughput RNA Sequencing data. *ArXiv e-prints*.

[16] Chen, L. (2013). Statistical and Computational Methods for High-Throughput Sequencing Data Analysis of Alternative Splicing. *Stat Biosci*, 5(1):138–155.

[17] Pachter, L. (2011). Models for transcript quantification from RNA-Seq. *ArXiv e-prints*.

[18] Steijger, T., Abril, J. F., Engstrom, P. G., Kokocinski, F., Hubbard, T. J., Guigo, R., Harrow, J., Bertone, P., Abril, J. F., Akerman, M., Alioto, T., Ambrosini, G., Antonarakis, S. E., Behr, J., Bertone, P., Bohnert, R., Bucher, P., Cloonan, N., Derrien, T., Djebali, S., Du, J., Dudoit, S., Engstrom, P., Gerstein, M., Gingeras, T. R., Gonzalez, D., Grimmond, S. M., Guigo, R., Habegger, L., Harrow, J., Hubbard, T. J., Iseli, C., Jean, G., Kahles, A., Kokocinski, F., Lagarde, J., Leng, J., Lefebvre, G., Lewis, S., Mortazavi, A., Niermann, P., Ratsch, G., Reymond, A., Ribeca, P., Richard, H., Rougemont, J., Rozowsky, J., Sammeth, M., Sboner, A., Schulz, M. H., Searle, S. M., Solorzano, N. D., Solovyev, V., Stanke, M., Steijger, T., Stevenson, B. J., Stockinger, H., Valsesia, A., Weese, D., White, S., Wold, B. J., Wu, J., Wu, T. D., Zeller, G., Zerbino, D., and Zhang, M. Q. (2013). Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods*, 10(12):1177–1184.

[19] Hayer K, Pizzaro A, Lahens N, Hogenesch J, Grant G Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data. *BioRxiv2014*.

[20] Wu, J., Akerman, M., Sun, S., McCombie, W. R., Krainer, A. R., and Zhang, M. Q. (2011). SpliceTrap: a method to quantify alternative splicing under single cellular conditions. *Bioinformatics*, 27(21):3010–3016.

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

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

[23] Singh, D., Orellana, C. F., Hu, Y., Jones, C. D., Liu, Y., Chiang, D. Y., Liu, J., and Prins, J. F. (2011). FDM: a graph-based statistical method to detect differential transcription using RNA-seq data. *Bioinformatics*, 27(19):2633–2640.

[24] Brooks, A. N., Yang, L., Duff, M. O., Hansen, K. D., Park, J. W., Dudoit, S., Brenner, S. E., and Graveley, B. R. (2011). Conservation of an RNA regulatory map between Drosophila and mammals. *Genome Res.*, 21(2):193–202.

[25] Anders, S., Reyes, A., and Huber, W. (2012). Detecting differential usage of exons from RNA-seq data. *Genome Res.*, 22(10):2008–2017.

[26] Wang, W., Qin, Z., Feng, Z., Wang, X., and Zhang, X. (2013). Identifying differentially spliced genes from two groups of RNA-seq samples. *Gene*, 518(1):164–170.

[27] Aschoff, M., Hotz-Wagenblatt, A., Glatting, K. H., Fischer, M., Eils, R., and Kdonig, R. (2013). SplicingCompass: differential splicing detection using RNA-seq data. *Bioinformatics*, 29(9):1141–1148.

[28] Shen, S., Park, J. W., Huang, J., Dittmar, K. A., Lu, Z. X., Zhou, Q., Carstens, R. P., and Xing, Y. (2012). MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic Acids Res.*, 40(8):e61.

[29] Drewe, P., Stegle, O., Hartmann, L., Kahles, A., Bohnert, R., Wachter, A., Borgwardt, K., and Ratsch, G. (2013). Accurate detection of differential RNA processing. *Nucleic Acids Res.*, 41(10):5189–5198.

[30] Wang, X. and Cairns, M. J. (2014). SeqGSEA: a Bioconductor package for gene set enrichment analysis of RNA-Seq data integrating differential expression and splicing. *Bioinformatics*.

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

[32] Hu, Y., Huang, Y., Du, Y., Orellana, C. F., Singh, D., Johnson, A. R., Monroy, A., Kuan, P. F., Hammond, S. M., Makowski, L., Randell, S. H., Chiang, D. Y., Hayes,

D. N., Jones, C., Liu, Y., Prins, J. F., and Liu, J. (2013). DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Res.*, 41(2):e39.

[33] Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11:94.

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

[35] Salzman, J., Jiang, H., and Wong, W. H. (2011). Statistical Modeling of RNA-Seq Data. *Stat Sci*, 26(1).

[36] Gulledge, A. A., Roberts, A. D., Vora, H., Patel, K., and Loraine, A. E. (2012). Mining Arabidopsis thaliana RNA-seq data with Integrated Genome Browser reveals stress-induced alternative splicing of the putative splicing regulator SR45a. *Am. J. Bot.*, 99(2):219–231.

[37] James, A. B., Syed, N. H., Bordage, S., Marshall, J., Nimmo, G. A., Jenkins, G. I., Herzyk, P., Brown, J. W., and Nimmo, H. G. (2012). Alternative splicing mediates responses of the Arabidopsis circadian clock to temperature changes. *Plant Cell*, 24(3):961–981.

[38] Zhang, X. N. and Mount, S. M. (2009). Two alternatively spliced isoforms of the Arabidopsis SR45 protein have distinct roles during normal plant development. *Plant Physiol.*, 150(3):1450–1458.

[39] Yan, K., Liu, P., Wu, C. A., Yang, G. D., Xu, R., Guo, Q. H., Huang, J. G., and Zheng, C. C. (2012). Stress-induced alternative splicing provides a mechanism for the regulation of microRNA processing in Arabidopsis thaliana. *Mol. Cell*, 48(4):521–531.

[40] Barta, A., Kalyna, M., and Reddy, A. S. (2010). Implementing a rational and consistent nomenclature for serine/arginine-rich protein splicing factors (SR proteins) in plants. *Plant Cell*, 22(9):2926–2929.

[41] Tanabe, N., Yoshimura, K., Kimura, A., Yabuta, Y., and Shigeoka, S. (2007). Differential expression of alternatively spliced mRNAs of Arabidopsis SR protein homologs, atSR30 and atSR45a, in response to environmental stress. *Plant Cell Physiol.*, 48(7):1036–1049.

[42] Pose D, Verhage L, Ott F, Yant L, Mathieu J, Angenent GC, Immink RG, Schmid M (2013). Temperature-dependent regulation of flowering by antagonistic FLM variants. *Nature.*, 503(7476):414417.

[43] Kesari R, Lasky JR, Villamor JG, Des Marais DL, Chen YJ, Liu TW, Lin W, Juenger TE, Verslues PE. Intron-mediated alternative splicing of Arabidopsis P5CS1 and its association with natural variation in proline and climate adaptation. *Proc Natl Acad Sci.*, 109(23):91979202.

[44] Nicol, J. W., Helt, G. A., Blanchard, S. G., Raja, A., and Loraine, A. E. (2009). The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, 25(20):2730–2731.

[45] Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigo, R., and Sammeth, M. (2012). Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res.*, 40(20):10073–10083.

[46] Wu, T. D. and Nacu, S. (2010). Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7):873–881.

[47] Seki, M., Satou, M., Sakurai, T., Akiyama, K., Iida, K., Ishida, J., Nakajima, M., Enju, A., Narusaka, M., Fujita, M., Oono, Y., Kamei, A., Yamaguchi-Shinozaki, K., and

Shinozaki, K. (2004). RIKEN Arabidopsis full-length (RAFL) cDNA and its applications for expression profiling under abiotic stress conditions. *J. Exp. Bot.*, 55(395):213–223.

Appendix

Step 1: Simulating biological replicates

In order to approximate the situation in real RNA-seq experiment, we required two groups of empirical RNA-seq samples representing control and treatment groups respectively. First, the pipeline selected a random subset of genes that had more than one transcript based on annotation and that were expressed (have non-zero read counts in every replicate) in both input groups as true AS genes. The total transcripts copy number on a simulated gene was proportional to the number of reads counted on the real gene. We also introduced biological variance to gene expression by using Negative Binomial(NB) distributions. NB distribution is widely used for modeling variance across biological replicates. For each gene g we calculated mean μ_g and variance σ_g^2 of gene-level read counts across replicates and then performed a Loess regression f on the set of points (μ_g, σ_g^2). Thus we can borrow information across genes and do not rely on having large enough number of replicates to estimate variance. In the simulation studies with the same dispersion pattern we forced the regression function f to be the same under two conditions. For the simulation studies using different dispersion patterns the regression function f was learned from each of the two input groups and thus it differed for the two simulated conditions. The advantage of using Loess function is that Loess fitting does not make the same assumption of global homoscedasticity as general linear regression. Finally, the transcript counts for gene g were generated by NB distribution parameterized by mean μ_g and fitted variance $f(\mu_g)$.

Step 2: Simulating differential splicing

We defined a parameter, $PALT$, to control the relative transcript abundances across conditions. $PALT$ stands for Percentage of ALTernative form, ranging from 0 to 1. The

relative transcript abundances of a multi-isoform gene g which has i isoforms, denoted by $e_g = (e_g^1, ..., e_g^i)$, were decided through the following formulas.

- if g is a AS gene, then we set $e_g^j = PALT, if j = i$ and $e_g^j = \frac{1-PALT}{i-1}, if j \neq i$.
- if g is not a AS gene, then we draw the relative abundance from a standard uniform distribution $e_g^j \in uniform(0, 1)$ with a constraint $\sum_{j=1}^{i} e_g^j = 1$

In addition, we introduced another parameter Read Depth(RD) to allow user to control the mean per-based read depth which is defined as: $L * N/T$ Where L is the read length; N is the number of reads mapped to transcriptome; T is the transcriptome size.

Therefore the final absolute transcript abundance in the custom transcriptome expression profile are the product of gene-level transcript counts from step 1, relative transcript abundances and read depth tuner which makes sure the desired read depth is generated. Finally, the program, Flux Simulator calls this profile to generate RNA-seq reads.

Sanity check of synthetic data

To simulate biological replicates, we used Arabidopsis heat shock dataset (36) which contains three replicates for each of the two time points. The first time point was immediate after heat stress. The second was 24 h after recovery from the heat stress. The mean fragment counts across replicates and mean-variance relationship used in the simulation were estimated from the heat shock data set. Figure S2 shows the mean and variance of fragment counts in the log scale for synthetic data in baseline simulation study $RD100_D^H$ and heat shock data. There was a good agreement which indicated that the negative binomial model used in the simulation captured the mean-variance relationship or dispersion well. We further compared the distribution of the mean fragment counts in log scale. The simulation again captured the distribution in real data well.

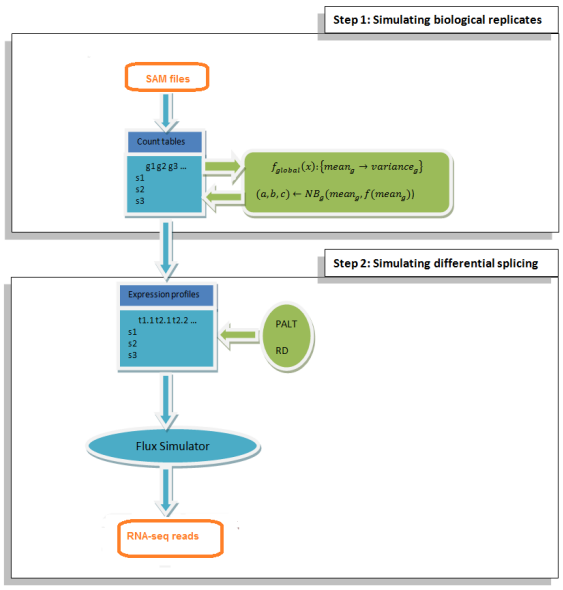


Figure 2.8 A two-step simulation pipeline. SAM files from real data are used as input for this pipeline. In the first step biological replicates are simulated by using Negative Binomial (NB) models. The raw fragment counts mean μ_g and variance σ_g^2 are calculated from the input. A regression function f is fitted on the set of points (μ_g, σ_g^2). Then the fitted variances are used as parameters in the NB models to generate three replicates, e.g. a, b, c. In the second step. The updated gene-level fragment counts are separate onto transcript levels based on the relative abundances and desired read depth. Finally, Flux Simulator is used to generated simulated RNA-seq reads.

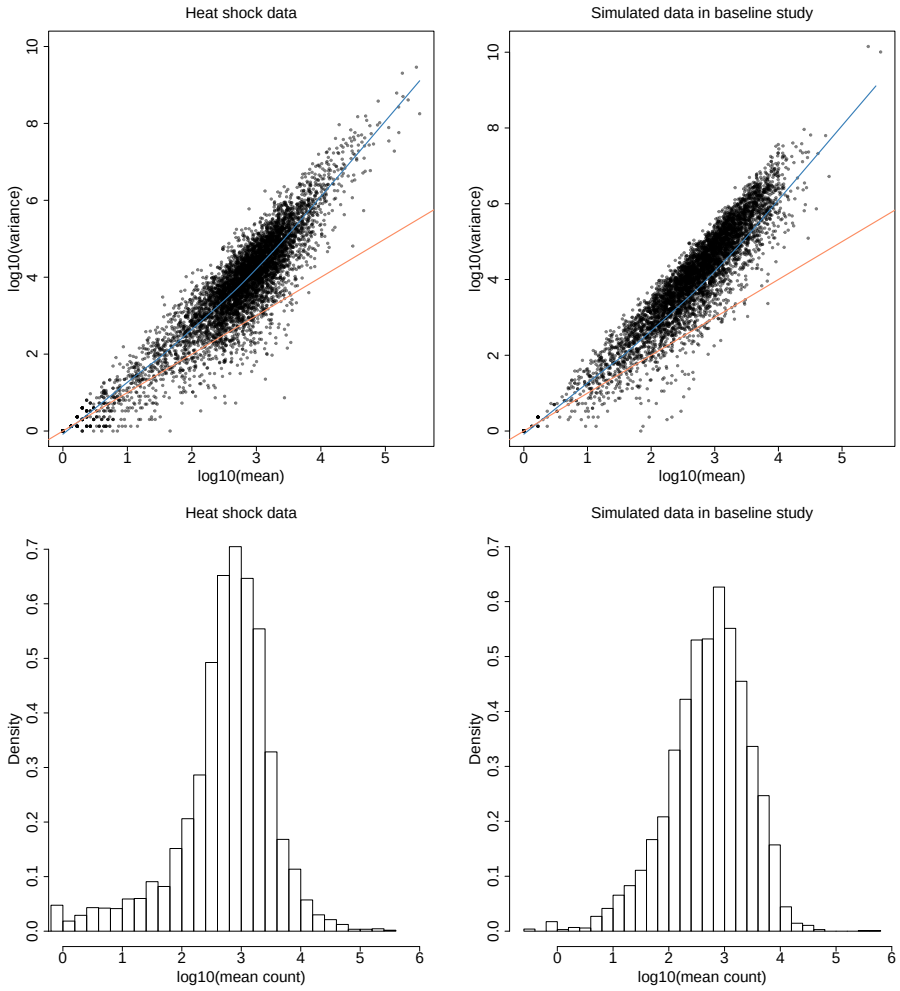


Figure 2.9 Comparison between real (left panels) and synthetic data (right panels). The 2 panels on top are scatter plots of mean-variance relationship across replicates. The blue lines are LOWESS regression lines. The orange lines are *variance* = *mean* lines. It is clear that the real data is overdispersed with respect to what we would expect from a Poisson distribution and that it was well captured by a negative binomial distribution using in the simulated data. The two panels at the bottom compare the fragment counts distribution.

Command lines and parameter choices

Cufflinks

Cufflinks was written in Python and C++. It can be downloaded from http://cufflinks.cbcb.umd.edu/. We used the version 2.1.1 in this study. A newer version 2.2.0 was release while we were writing the paper.

```
cufflinks -p 8 -o RD100.control_r1 -L RD100C1 RD100.control_r1.sam
cufflinks -p 8 -o RD100.control_r2 -L RD100C2 RD100.control_r2.sam
cufflinks -p 8 -o RD100.control_r3 -L RD100C3 RD100.control_r3.sam
cufflinks -p 8 -o RD100.high.diff_r1 -L RD100HDM1 RD100.high.diff_r1.sam
cufflinks -p 8 -o RD100.high.diff_r2 -L RD100HDM2 RD100.high.diff_r2.sam
cufflinks -p 8 -o RD100.high.diff_r3 -L RD100HDM3 RD100.high.diff_r3.sam
cuffmerge -g TAIR10_GFF3_genes.gff -s TAIR10_nucleus.fas -p 8 assemblies.txt
cuffdiff -o diff_out -b TAIR10_nucleus.fas -L treatment,control -p 8 -u
merged_asm/merged.gtf RD100.high.diff_r1.sam,RD100.high.diff_r2.sam,RD100.high.diff_r3.sam
RD100.control_r1.sam,RD100.control_r2.sam,RD100.control_r3.sam
```

DEXSeq

DEXSeq is a R package available in Bioconductor. We used the latest version 1.8.0 in this study.

```
library("DEXSeq")
inDir="countTables"
infile=c("RD100.high.diff_r1.count","RD100.high.diff_r2.count","RD100.high.diff_r3.count",
"RD100.control_r1.count","RD100.control_r2.count","RD100.control_r3.count")
setwd("countTables")
annotationfile=file.path("TAIR10_GFF3_genes_countingBin.gtf")
samples = data.frame(
condition = c(rep("treated", 3), rep("untreated", 3)),
replicate = c(1:3, 1:3),
row.names = c("g2_1","g2_2","g2_3","g1_1","g1_2","g1_3"),
stringsAsFactors = TRUE,
check.names = FALSE
)
samples$replicate=factor(samples$replicate)
ecs = read.HTSeqCounts(countfiles = file.path(inDir,infile),design = samples,
flattenedfile = annotationfile)
ecs <- estimateSizeFactors(ecs)
ecs <- estimateDispersions(ecs)
ecs <- fitDispersionFunction(ecs)
```

```
ecs <- testForDEU(ecs)
res1 <- DEUresultTable(ecs)
sigExon=subset(res1, res1$padjust<0.05)
```

DiffSplice

DiffSplice was written in C++. It can be downloaded from http://www.netlab.uky.edu/p/bioinfo/DiffSplice/. We used the latest version 0.1.1 in this study.

```
diffsplice settings.cfg datafile.cfg output

## parameters used in settings.cfg
thresh_junction_filter_max_read_support      2
thresh_junction_filter_mean_read_support         0
thresh_junction_filter_num_samples_presence 0
ignore_minor_alternative_splicing_variants   yes
thresh_average_read_coverage_exon    0
thresh_average_read_coverage_intron 0
balanced_design_for_permutation_test         no
false_discovery_rate                  0.05
thresh_foldchange_up                  0.5
thresh_foldchange_down            0.5
thresh_sqrtJSD                0.1
```

DSGseq

DSGseq consists of a set of R scripts but is not a standard R packages. It can be downloaded from http://bioinfo.au.tsinghua.edu.cn/software/DSGseq/. We used the latest version 0.1.0.

```
bamToBed -i RD100.high.diff_r1.bam > RD100.high.diff_r1.bed
bamToBed -i RD100.high.diff_r2.bam > RD100.high.diff_r2.bed
bamToBed -i RD100.high.diff_r3.bam > RD100.high.diff_r3.bed
bamToBed -i RD100.control_r1.bam > RD100.control_r1.bed
bamToBed -i RD100.control_r2.bam > RD100.control_r2.bed
```

```
bamToBed -i RD100.control_r3.bam > RD100.control_r3.bed

SeqExpress count RD100.high.diff_r1.bed TAIR10.merge.refFlat RD100.high.diff_r1.count
SeqExpress count RD100.high.diff_r2.bed TAIR10.merge.refFlat RD100.high.diff_r2.count
SeqExpress count RD100.high.diff_r3.bed TAIR10.merge.refFlat RD100.high.diff_r3.count
SeqExpress count RD100.control_r1.bed TAIR10.merge.refFlat RD100.control_r1.count
SeqExpress count RD100.control_r2.bed TAIR10.merge.refFlat RD100.control_r2.count
SeqExpress count RD100.control_r3.bed TAIR10.merge.refFlat RD100.control_r3.count

Rscript DSGNB.R 3 RD100.high.diff_r1.count RD100.high.diff_r2.count RD100.high.diff_r3.count 3
RD100.control_r1.count RD100.control_r2.count RD100.control_r3.count RD100_high_diff.DSGresult
```

MATS

MATS was written Python. It can be downloaded from http://rnaseq-mats.sourceforge.net/. We used the latest version 3.0.8 in this study.

```
python RNASeq-MATS.py -b1 RD100.high.diff_r1.bam,RD100.high.diff_r2.bam,RD100.high.diff_r3.bam -b2
RD100.control_r1.bam,RD100.control_r2.bam,RD100.control_r3.bam -gtf TAIR10_GFF3_genes.gtf -t paired
-len 100 -o MATS_OUT
```

SeqGSEA

SeqGSEA is a R package available in Bioconductor. We used the version 1.2.1. A newer version 1.5.0 was release while we were writing the paper.

```
library(SeqGSEA)
rm(list=ls())
case.pattern <- "^RD100.high"
ctrl.pattern <- "^RD100.control"
case.files <- dir("RD100.high.dm/seqgsea", pattern=case.pattern, full.names = TRUE)
control.files <- dir("RD100.control/seqgsea", pattern=ctrl.pattern, full.names = TRUE)
output.prefix <- "SeqGSEA.result"
library(doParallel)
cl <- makeCluster(2)
registerDoParallel(cl)
perm.times <- 1000
RCS <- loadExonCountData(case.files, control.files)
RCS <- exonTestability(RCS, cutoff=5)
geneTestable <- geneTestability(RCS)
RCS <- subsetByGenes(RCS, unique(geneID(RCS))[ geneTestable ])
geneIDs <- unique(geneID(RCS))
RCS <- estiExonNBstat(RCS)
```

```
RCS <- estiGeneNBstat(RCS)
permuteMat <- genpermuteMat(RCS, times=perm.times)
RCS <- DSpermutePval(RCS, permuteMat)
```

SplicingCompass

SplicingCompass is a R package. We used the latest version 1.0.1.

```
library("SplicingCompass")
packageDescription("SplicingCompass")
expInf=new("ExperimentInfo")
expInf=setDescription(expInf,"Group1 vs Group2")
expInf=setGroupInfo(expInf,
groupName1="ControlGroup1",sampleNumsGroup1=1:3,
groupName2="CaseGroup2",sampleNumsGroup2=4:6)
covBedCountFilesControl=c(
"RD100.control_r1.covBed",
"RD100.control_r2.covBed",
"RD100.control_r3.covBed")
covBedCountFilesCase=c(
"RD100.high.diff_r1.covBed",
"RD100.high.diff_r2.covBed",
"RD100.high.diff_r3.covBed")
junctionBedFilesControl=c(
"RD100.control_r1.juncBed",
"RD100.control_r2.juncBed",
"RD100.control_r3.juncBed")
junctionBedFilesCase=c(
"RD100.high.diff_r1.juncBed",
"RD100.high.diff_r2.juncBed",
"RD100.high.diff_r3.juncBed")
expInf=setCovBedCountFiles(expInf,c(covBedCountFilesCase,covBedCountFilesControl))
expInf=setJunctionBedFiles(expInf, c(junctionBedFilesCase,junctionBedFilesControl))
expInf=setReferenceAnnotation(expInf,"TAIR10_TableUnion.gtf")
referenceAnnotationFormat=list(IDFieldName="geneSymbol",idValSep=" ")
expInf=setReferenceAnnotationFormat(expInf,referenceAnnotationFormat)
checkExperimentInfo(expInf)
countTable=new("CountTable")
countTable=setExperimentInfo(countTable,expInf)
countTable=constructCountTable(countTable,printDotPerGene=TRUE)
sc = new("SplicingCompass")
sc = constructSplicingCompass(sc, countTable, minOverallJunctionReadSupport=3)
sc = initSigGenesFromResults(sc, adjusted=TRUE, threshold=0.05)
sigGenes = getSignificantGeneSymbols(sc)
resTab = getResultTable(sc)
```

rDIff-parametric

rDiff can be downloaded from http://cbio.mskcc.org/public/raetschlab/user/drewe/rdiff/. We used the latest version 0.3.

```
rdiff -o RD100HighDm -d data/ -a RD100.control_r1.bam,RD100.control_r2.bam,RD100.control_r3.bam
-b RD100.high.diff_r1.bam,RD100.high.diff_r2.bam,RD100.high.diff_r3.bam
-g data/TAIR10_GFF3_genes.gff -m param -L 100
```

Comparison of two different MATS results

Table 2.6 MATS result using junction reads only versus result using both junction reads and exon body reads in simulation study $RD100^{H}_{D}$. The Pearson correlation of the p-values in these two results is as high as 0.978.

EventType	NumEvents.JC.only	SigEvents.JC.only	NumEvents.JC+ readsOnTarget	SigEvents.JC+ readsOnTarget
SE	704	153	704	152
MXE	14	1	14	1
A5SS	556	165	556	165
A3SS	1106	314	1106	313
RI	983	311	985	311

SE: Skipped exon

MXE: Mutually exclusive exon

A5SS: Alternative 5' splice site

A3SS: Alternative 3' splice site

RI: Retained intron

NumEvents.JC.only: total number of events detected using junction reads only

SigEvents.JC.only: number of significant events detected using junction reads only

NumEvents.JC+readsOnTarget: total number of events detected using both junction reads and exon body reads

SigEvents.JC+readsOnTarget: number of significant events detected using both junction reads and exon body reads

Computational time requirement

We ran the code shown in the previous section in Iowa State University super cluster called Lightning. The code was all executed in a single node and a single core with

16GB RAM. Although we used a cluster, this amount of computational power can be easily obtained in a standard PC. All the programs were finished within a few hours. The computation time required for SeqGSEA is largely affected by the permutation times. In this study, we set it to 1000. The total required CPU time for each method in the baseline simulation study $RD100_D^H$ is given in the Table S1.

Table 2.7 Total computational time in CPU-seconds

Cufflinks	**DEXSeq**	**MATS**	**SpComp**	**DSGseq**	**rDiff-param**	**DiffSplice**	**SeqGSEA**
41172s	6096s	8371s	10408s	1256s	1038s	4415s	39539s

Visualization of read alignments in heat shock data for experimentally validated AS genes

We have examined a few Arabidopsis genes that are known to be differentially spliced in response to ambient temperature changes. The following figures are the visualization of reads alignment of these few known genes using Integrated Genome Browser (44). Solid bars represent reads, and thin lines indicate gaps in the alignment.

LHY

LATE ELONGATED HYPOCOTYL (LHY), circadian clock genes, are known to be differential spliced in response to temperature changes(37). 5 transcripts have been found (based on TAIR10) in gene AT1G01060 which belongs to LHY gene family. Transcript AT1G01060.4 differs from other transcripts by 3-nt difference in the 3' site.

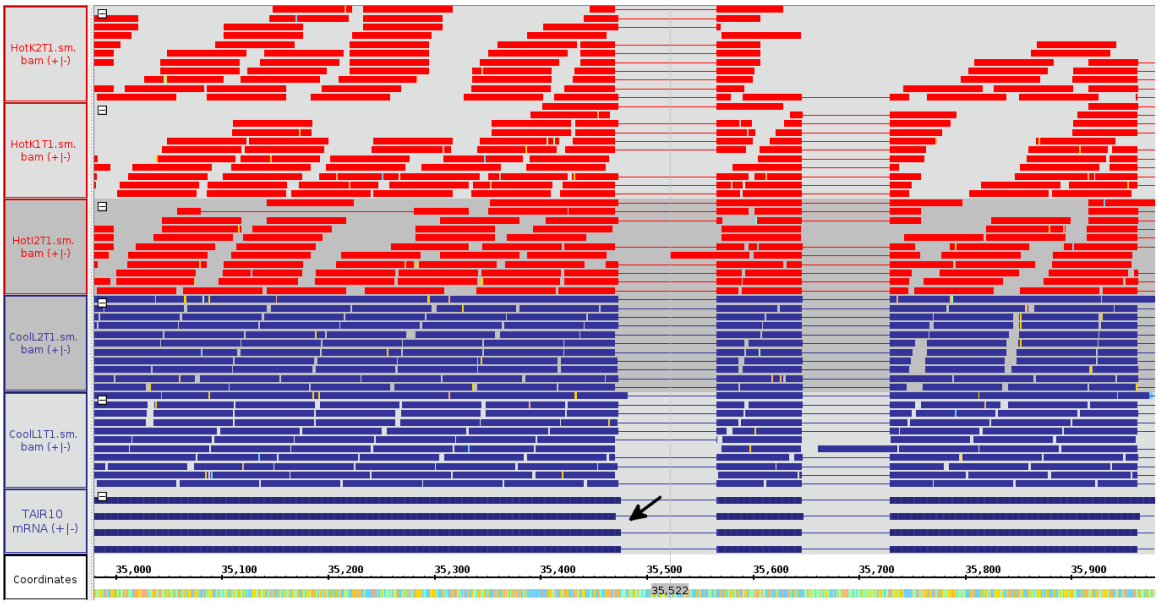


Figure 2.10 LHY. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red whereas reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

SR45

AT1G16610 encodes SR45 which is a member of SR protein family. A alternative 3'SS event differed by a 21-nt sequence has been found to occur as ambient temperature changes (38).

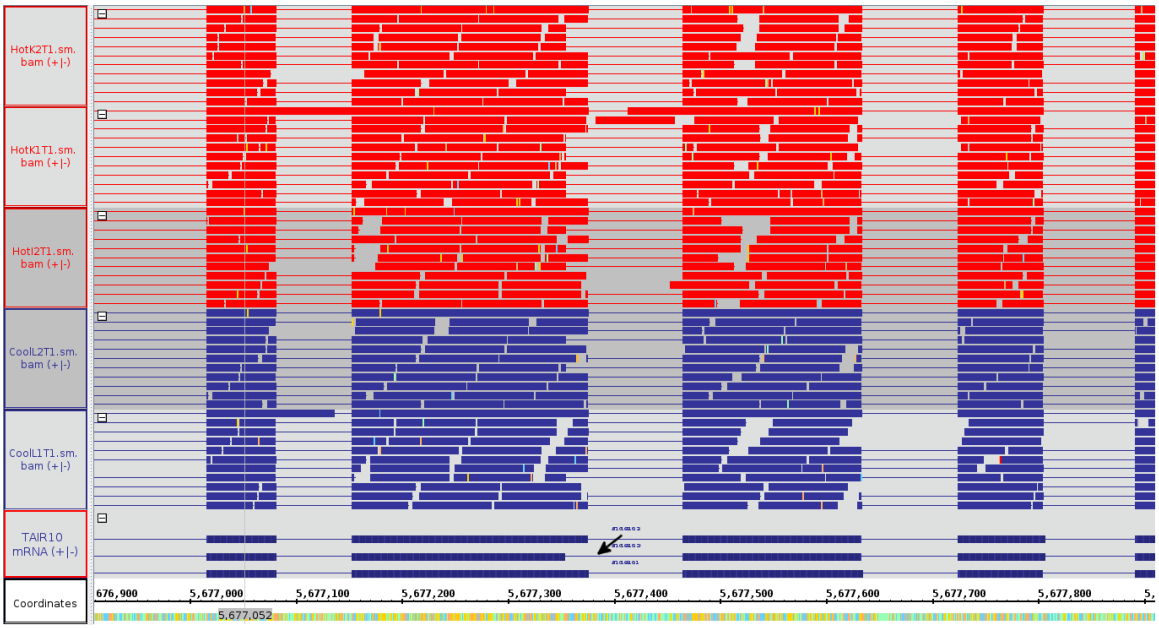


Figure 2.11 SR45. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

SR1/SR34

AT1G02840 encodes SR1/SR34 protein, a member of highly conserved family of spliceosome proteins. An alternative 3'SS event has been found as ambient temperature changes(39).

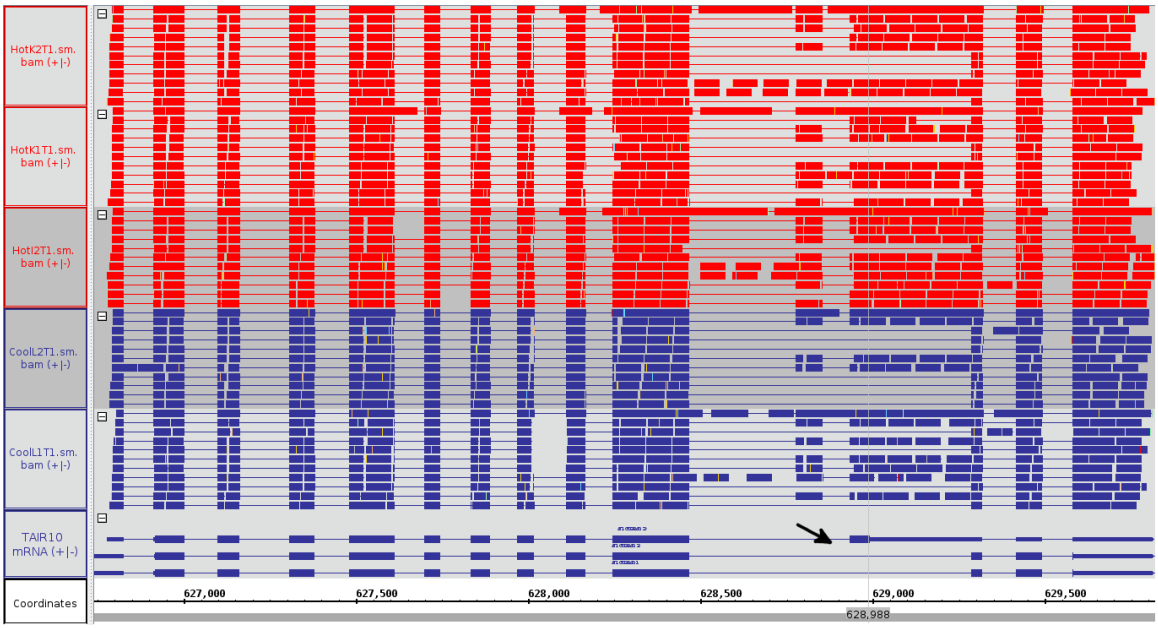


Figure 2.12 SR1/SR34. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

SR30

AT1G09140 encodes SR30, a member of highly conserved family of spliceosome proteins. An alternative 3'SS event has been found in response to heat stress (39).

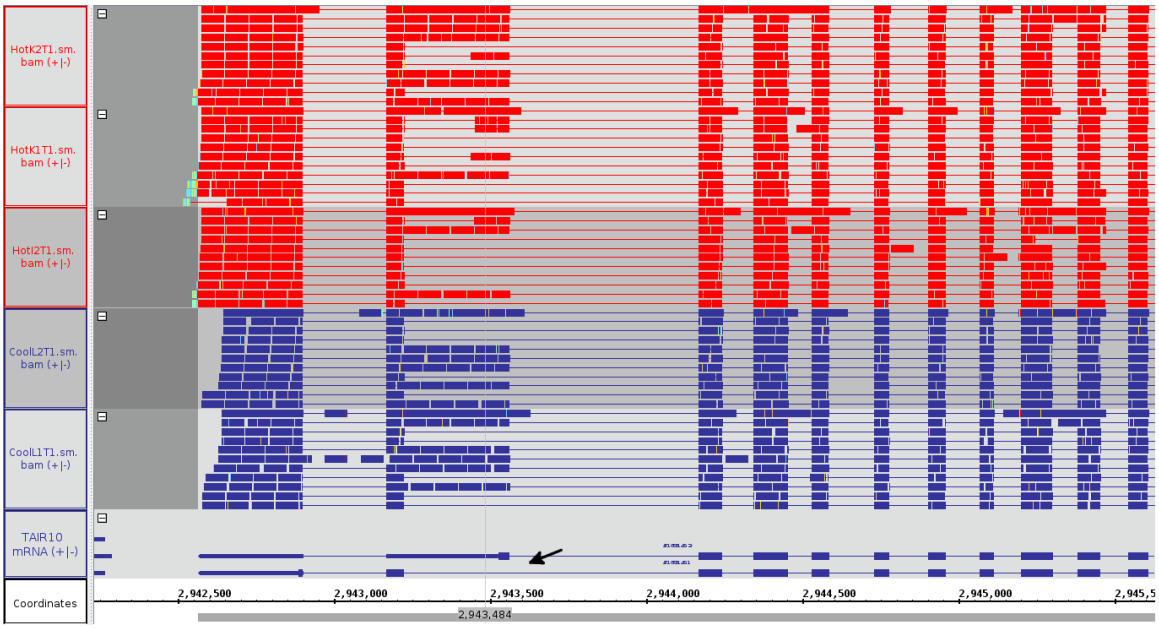


Figure 2.13 SR30. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

P5CS1

P5CS1 gene (AT2G39800) contains an exon-3 skipping event which is subject to temperature variation (43).

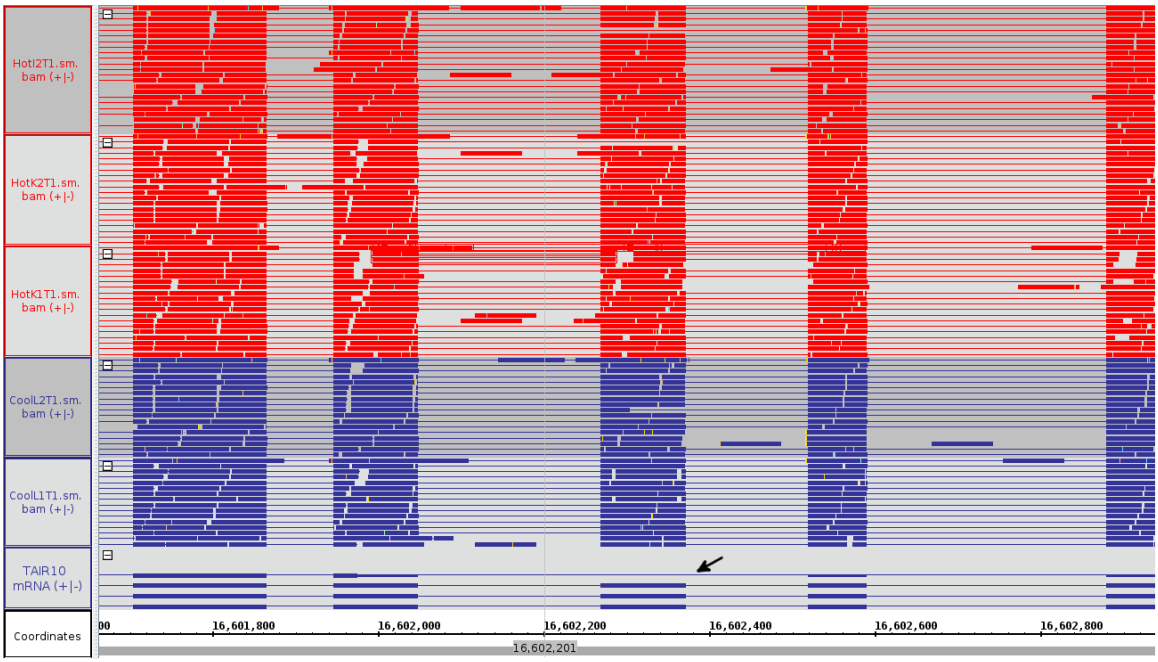


Figure 2.14 P5CS1. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

FLM

AT1G77080 encodes FLM, a protein which regulates flowering. An mutually exclusive exon event has been found in subject to temperature changes (42)

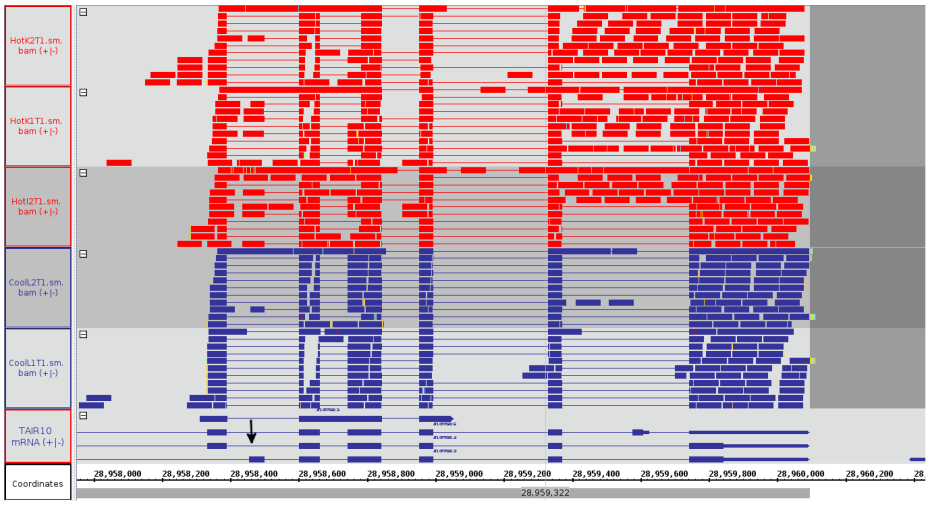


Figure 2.15 FLM. Visualization of read alignments in heat shock data. Reads from heat stress group are colored in red and reads from control groups are colored in blue. The black arrow indicates where the AS event happens in the gene model.

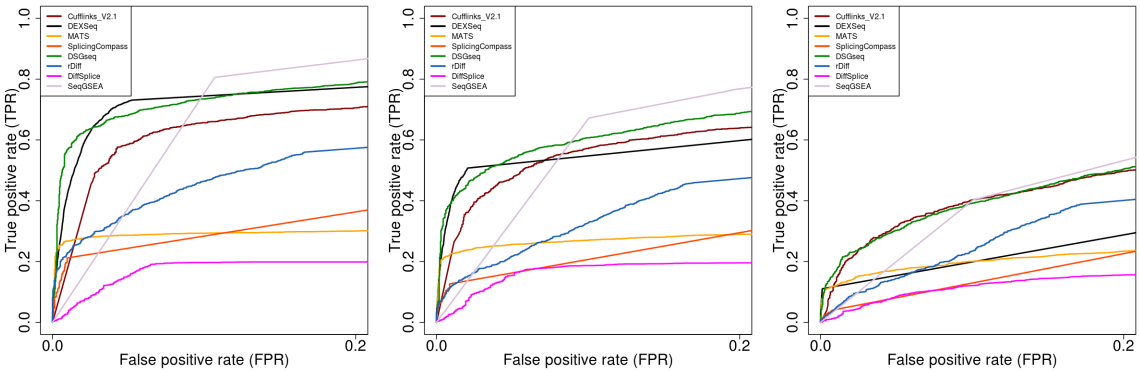


Figure 2.16 ROC curves evaluation for three different AS ratios when two groups of samples have the same dispersion pattern. ROC curves for simulation studies $\text{High}_{100x}^{\text{Same}}$ (left panel), $\text{Medium}_{100x}^{\text{Same}}$ (middle panel), $\text{Low}_{100x}^{\text{Same}}$(right panel). These ROC curves are obtained at a simple size of 3 for each condition.

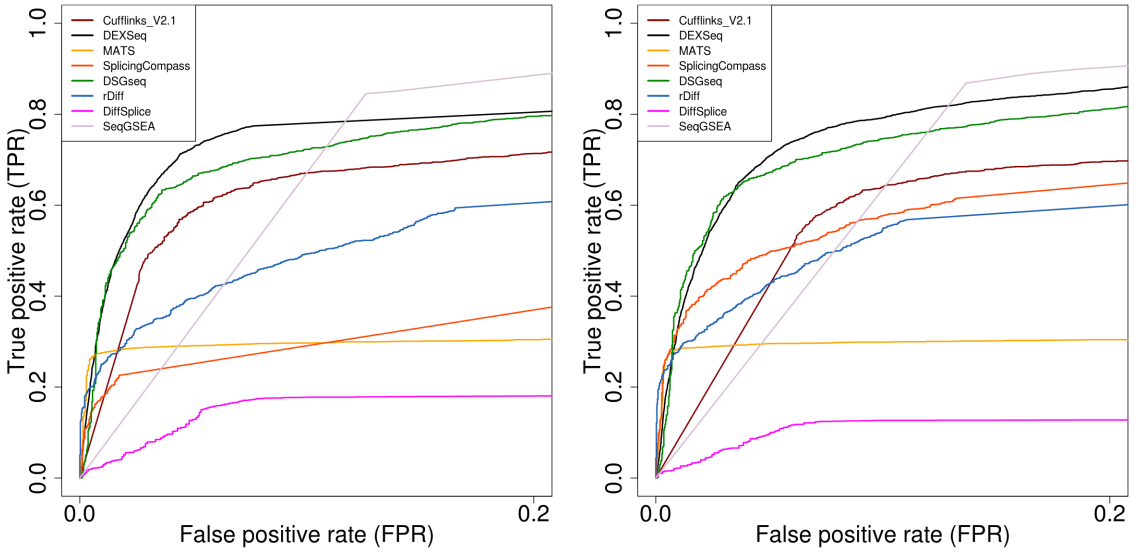


Figure 2.17 ROC curves evaluation for the two different samples sizes. Left panel shows ROC curves in the baseline simulation study $\text{High}^{\text{Diff}}_{100\text{x}}RD100^{H}_{D}$ which contained three replicates for each condition. The right panel shows the ROC curves when the sample size was increased to 8.

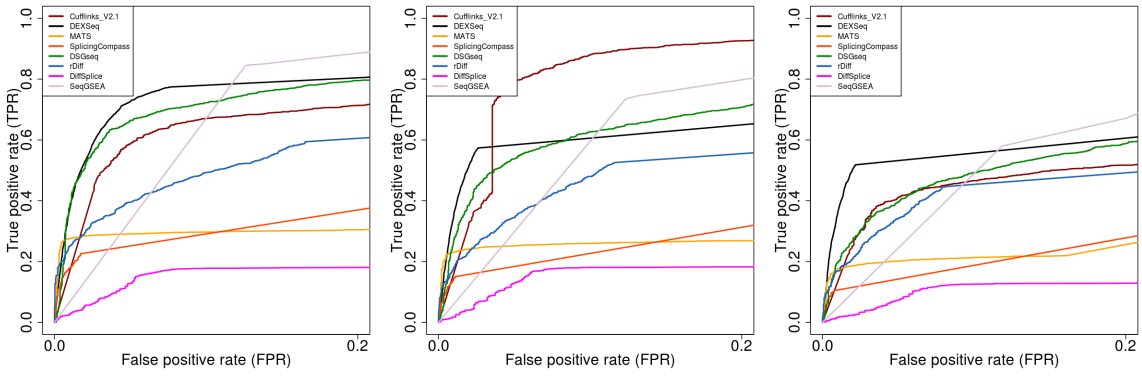


Figure 2.18 ROC curves evaluation for three different read depths, simulation studies $100\text{x}^{\text{Diff}}_{\text{High}}$ (left panel), $60\text{x}^{\text{Diff}}_{\text{High}}$ (middle panel), $25\text{x}^{\text{Diff}}_{\text{High}}$(right panel).

CHAPTER 3. STRAWBERRY: FAST and ACCURATE GENOME-GUIDED TRANSCRIPT RECONSTRUCTION and QUANTIFICATION from RNA-SEQ

Authors: Ruolin Liu and Julie A Dickerson

PLoS Comput Biol 13(11): e1005851. Published Nov 2017.

https://doi.org/10.1371/journal.pcbi.1005851

Author's contributions

RL leads this study. RL and JD contribute to the design of the study, the interpretation of the results. RL writes the manuscript and is supervised by JD. RL writes the programs and does all the data analysis. All the authors read and approved the final manuscript.

Abstract

We propose a novel method and software tool, Strawberry, for transcript reconstruction and quantification from RNA-Seq data under the guidance of genome alignment and independent of gene annotation. Strawberry consists of two modules: assembly and quantification. The novelty of Strawberry is that the two modules use different optimization frameworks but utilize the same data graph structure, which allows a highly efficient, expandable and accurate algorithm for dealing large data. The assembly module parses aligned reads into splicing graphs, and uses network flow algorithms to select the most likely transcripts. The quantification module uses a latent class model to assign read counts from the nodes of splicing graphs to transcripts. Strawberry simultaneously estimates the transcript abundances and corrects for sequencing bias through an EM algorithm. Based on simulations, Strawberry outperforms Cufflinks and StringTie in terms of both assembly and

quantification accuracies. Under the evaluation of a real data set, the estimated transcript expression by Strawberry has the highest correlation with Nanostring probe counts, an independent experiment measure for transcript expression.

Availability: Strawberry is written in C++14, and is available as open source software at https://github.com/ruolin/strawberry under the MIT license.

Author summary

Transcript assembly and quantification are important bioinformatics applications of RNA-Seq. The difficulty of solving these problem arises from the ambiguity of reads assignment to isoforms uniquely. This challenge is twofold: statistically, it requires a high-dimensional mixture model, and computationally, it needs to process datasets that commonly consist of tens of millions of reads. Existing algorithms either use very complex models that are too slow or assume no models, rather heuristic, and thus less accurate. Strawberry seeks to achieve a great balance between the model complexity and speed. Strawberry effectively leverages a graph-based algorithm to utilize all possible information from pair-end reads and, to our knowledge, is the first to apply a flow network algorithm on the constrained assembly problem. We are also the first to formulate the quantification problem in a latent class model. All of these features not only lead to a more flexible and complex quantification model but also yield software that is easier to maintain and extend. In this method paper, we have shown that the Strawberry method is novel, accurate, fast and scalable using both simulated data and real data.

Introduction

Transcript-level quantification is a key step for detecting differential alternative splicing and differential gene expression. A number of computational methods have been developed for estimation of transcript abundances (1; 2; 3; 4; 5; 6; 7; 8; 9). However, many of the methods (4; 5; 6; 7; 8; 9) rely on existing gene annotations and limits the use of such

methods because even for the model organisms like *Drosophila melanogaster* new isoforms are discovered all the time under different tissues and/or conditions (*Pachter, 2011, Models for transcript quantification from RNA-Seq*). In addition, Liu et al. has shown that incomplete annotation is a major factor that negatively affects quantification accuracy for detecting alternative splicing (10). Thus, transcript-level quantification should be coupled with transcript assembly when dealing with RNA-Seq data. Pure de novo assembly of raw RNA-Seq is very challenging. Genome-guided methods, instead, assemble aligned RNA-Seq reads into transcripts, taking advantage of (if possible) a finished and high quality genome assembly and the-state-of-art spliced alignment algorithms.

Two strategies have evolved for tackling transcript assembly and quantification after RNA-Seq reads have been aligned to reference genome: simultaneous transcript construction and expression quantification vs. sequential transcript construction then expression quantification. Clearly, transcript reconstruction and quantification are closely related and many methods try to solve both simultaneously (3; 11; 12; 13). These methods usually exhaustively enumerate all possible transcripts and then use regularization to get rid of unlikely transcripts when calculating their expression. The $L1$ penalty is commonly used to favor sparse transcript solutions (13). Another strategy involves breaking the problem up in a step-by-step manner, like Cufflinks. First, reconstruct a set of transcripts, and then performs quantification on the transcripts. The latter is a more conservative strategy and usually leads to "maximum precision vs. maximum sensitivity" (14) compared to the former.

Method overview

Strawberry consists of two modules: assembly module and quantification module. The two modules work in a sequential manner (**Fig1**). Strawberry is a genome-guided transcript-level assembler and quantification tool. It takes aligned RNA-Seq data in BAM format and output a gene annotation file in gff format with estimated transcript abundances. Using

alignment format as input allows Strawberry to take advantages of the latest reference genome (if possible, a finished and high-quality one) and stat-of-the-art splice-awareness aligners. Strawberry is designed for Illumina pair-end reads. To be clear in this article, a read-pair refers to aligned paired-end reads with sequences observed at both ends and unknown sequence in between and a read refers to either the upstream or downstream observed sequence of a read-pair. For single-end reads, replace the terminology "read-pair" with "read" and proceed.

The assembly module of Strawberry seeks a parsimonious representation of transcripts which best explains the observed read-pairs with the aid of flow network algorithms. The read-pairs are converted to splicing graphs where the nodes are subexons and edges are splice alignments. FlipFlop (3), StringTie (2) and Traph (11) also use network flow algorithm, but for different purposes. StringTie and Traph renounce the likelihood-based approach and solve transcript assembly and quantification as optimization problems and solve the two problems simultaneously in a flow network framework build upon on splice graph. The difference is that Traph uses a min-flow algorithm to find a set of flows that minimize the difference between the flows and the observed coverages, while StringTie uses an iterative algorithm to harvest the heaviest path and then uses maximum flow to estimate their expression. Here, a flow can be understood as a transcript with uniform coverage along it. Although also using flow network, FlipFlop constructs a penalized likelihood model. The penalized likelihood model is carefully designed to be convex and the estimation problem can be cast into a convex-cost min-flow. Different from all of them, Strawberry uses a min-cost circulation flow to solve a parsimonious assembly problem. If the underlying sequence of a read-pair contains an unsequenced portion, such as the insert, this read-pair might indicate necessary paths that are usually neglected by other methods (15), while Strawberry explicitly converts them to graph constraints. In a nutshell, StringTie uses a flow network to calculate transcript expression; Traph and FlipFlop use flow networks to concurrently solve transcript identification and quantification. Strawberry is the only one that applies a

flow network to an assembly problem. The assembly problem that Strawberry is solving is also unique. It is a constrained assembly problem that is tailored for paired-end reads by converting them to graph path constraints (see method section).

The quantification model of Strawberry is based on a latent class model with an effective data collapsing mechanism, which utilizes the same graph topology used in assembly to reduce the individual reads to subexon path counts. A subexon is a maximal portion of covered region (covered by reads) without any splice junctions. And subexon path is regarded a set of ordered subexons. The subexon path representation allows Strawberry to save computational cost and model nonuniform reads distribution along transcripts. To the best of our knowledge, the concept of subexon path was first proposed in (6). However, it can be seen as a modification/extension of the idea of *maximum collapsing* in (16). Although using same data collapsing mechanism, Rossell et al. uses a Bayesian framework and does not have a joint estimation of transcript proportion and coverage bias effect (6). While Strawberry applies a conditional multinomial distribution for the subexon paths and estimates the transcript proportion and coverage effect simultaneously in the mixture model. The change from a non-parametric model in (6) to a multinomial model in Strawberry permits better model expandability.

Strawberry is designed to be versatile and modular. It is possible to skip the assembly step and just run quantification module against an external set of transcripts, e.g. those from gene annotations. In this case, Strawberry reduces any overlapping set of isoforms to a splicing graph consisting of subexons and subexon paths. The external set of transcripts can also be used by Strawberry to help with assembly. Finally, Strawberry reports the calculated transcript expression in the units of FPKM (Fragments Per Kilobase of transcript per Million mapped reads) and TPM (Transcripts Per Kilobase Million).

Results

Ground truth simulated data and programs to compare

We compare Strawberry to two state-of-the-art programs, Cufflinks v2.2.1 (1) and StringTie v1.3.3 (2), on three simulated data sets, *RD25 RD60* and *RD100*. The only difference among these three data sets is the average sequencing depth. Roughly speaking, RD25 contains $\sim$ 2.5 million, RD60 $\sim$ 6 million and RD100 $\sim$ 10 million reads. These data were generated by the procedure used in (10)—100bp paired-end reads generated from 5800 multi-isoform Arabidopsis genes on genome version TAIR10 (29) using Flux Simulator (30). This simulation was repeated 10 times so that each data set consists of 10 RNA-Seq libraries. Those simulated reads were then mapped onto the Arabidopsis TAIR10 genome assembly using Tophat2 (31) and HISAT2 (32). Since plant genomes have shorter introns than mammals, all the programs ran on the default parameters except for the maximum intron length, which was set to 5000 bp.

To evaluate Strawberry's performance on higher eukaryotes, we also compare the three programs using simulated human RNA-seq data. To avoid possible simulation bias, we choose a different simulator called Polyester (40). Polyester requires a count matrix, where each row represents a transcript and each column contains the read counts for a sample, as an input. To generate this count matrix, we downloaded 6 samples from the GEUVADIS data base (41) and aligned them with HISAT2. Then Cufflinks was used to estimate transcript expression. All transcripts were selected from loci which have at least two isoforms with FPKM $>$ 1.0 for all six samples. This human simulation is referred to as *GEU* (see **S1 Data**). Compared to *RD100*, *GEU* has relatively longer read length (150 bp paired-end) and longer fragment length (700 bp in average). This read length and fragment size are intended for the latest illumina sequencer NextSeq.

Comparing assembly accuracy

We use a Cufflinks module called Cuffcompare http://cole-trapnell-lab.github.io/cufflinks/cuffcompare/index.html to compare the assembled transcripts or transfrags to the reference transcripts since the reads are all simulated based on the reference transcripts. We use Cuffcompares evaluation algorithm which implements typical gene finding measures of recall and precision (39). For example, the recall of an exon is the percentage of number of corrected exons divided by the number of actual exons and precision is the number of correct exons divided by the number of predicted exons. Determination of transcription start and end sites is a known weakness of RNA-Seq and impairs its application on identification of transcript boundaries (34). Thus, Cuffcompare defines a correct transcript as the chain of introns that match with the reference, leaving possible variances in the first and last exon.

We first assessed the genome-guided assembly accuracy of the three programs using simulated Arabidopsis data set. The degree to which transcripts reported by each method matched the reference annotation at the nucleotide, exon, intron and transcript level for three different sequencing depths are shown in (**Fig 3.2** , **S1 Fig** and **S2 Fig**). In all comparisons, Strawberry has higher recall as well as precision. In *RD100* data, for example, Strawberry averages 71.78%, 80.36%, 52.35% on recall at exons, intron, and full transcripts level respectively, followed StringTie, 67.03%, 74.41%, 46.65% and then Cufflinks, 65.51%, 74.09%, 42.76%. For all the methods, the recall decreases as sequencing depth decreases while the precision remains at a high level and doesn't change much. This indicates that although lower read depths make it harder for these methods to recover the true signal, the results are still very reliable. Correct detection of full transcripts using RNA-Seq data is still a very challenging task for all assemblers. Given sufficient sequencing depth (*RD100*), all methods can correctly retrieve more than 65% exons, and 75% intron but only around 50% of the full transcripts. On the other hand, precision for exons and intron detection are very high for all methods, averaging 98-99%. For transcript detection, Strawberry's

average precision is 81.62%, while StringTie is at 80.46% and Cufflinks at 74.68% . For the methods that parsimoniously assemble reads into transcripts, this may indicate some room for improvement—although the individual exons and introns are correctly recovered, the ways to stitch them together are still not optimal. We further conducted a paired t-test to evaluate the statistical significance of the difference in F1 score (the harmonic mean of recall and precision) between Strawberry and the other tools (p value = 7.02e-12 when compared to StringTie, and p value = 1.947e-14 when compared to Cufflinks).

Next, we evaluated the methods using *GEU*. Overall, we observe that the F1 values at transcript level are roughly at the same level as in *RD100*, and Strawberry clearly maintains its lead, followed by StringTie (**Fig. 3.3**). However, the gap between Strawberry and StringTie is smaller compared to *RD100*. Again, a paired t-test of F1 scores is used, yielding p value = 5.614e-03 when compared to StringTie, and p value = 2.965e-09. Strawberry also achieves the best F1 score at gene level (**Fig. 3.3**), and Cufflinks performs better than StringTie at gene level. When it comes to exon and intron levels comparison, however, StringTie clearly performs better than Strawberry and Cufflinks, see **S5 Fig**. This suggests Strawberry still has room to improve the detection on exon and intron level for human, which can lead to higher transcript reconstruction rate.

Comparing quantification accuracy

Let x_i be the true value of the FPKM for transcript i based on ground truth simulated data and y_i be the predicted FPKM. If log transformation is taking, these FPKM values were incremented by 1 before log transformation to avoid infinite numbers. We adopt the metrics defined in *Patro et.al 2017* (4).

1. Proportionality correlation

$$\rho_p = \frac{2\mathrm{Cov}\{\log x, \log y\}}{\mathrm{Var}\{\log x\} + \mathrm{Var}\{\log y\}} \tag{3.1}$$

2. Spearman correlation of between the true FPKM values and predicted FPKM values.

3. Mean Absolute Relative Difference (MARD), which is computed using the absolute relative difference ARD_i for each transcript i:

$$\text{ARD}_i = \frac{|x_i - y_i|}{0.5|x_i + y_i|}, \tag{3.2}$$

MARD is the mean value of the $\{\text{ARD}_i | i \in 1, ..., I\}$. ARD is bounded above by 2 and below by 0 and smaller value indicates a better prediction. *Patro et al.* (4) computes MARD on the number of reads deriving from each transcript which is commensurable to FPKM values.

Again, we first evaluate the methods using simulated Arabidopsis data. **Fig 3.4** , **S3 Fig**, **S4 Fig** show the histogram of the three measures over 10 replicates for all three read depth data sets *RD100*, *RD60* and *RD25* respectively. In these simulations, It can be seen that these methods are all well separated in terms of the all evaluation metrics except for only one case in which StringTie and Cufflinks are virtually tied over Spearman correlation in *RD60* data (**S3 Fig**). In the case of *RD100* data, Strawberry averaged 0.911, 0.912, and 0.370 on Proportional correlation, Spearman correlation and MARD respectively, followed by StringTie, 0.866, 0.869, 0.385 and then Cufflinks, 0.834, 0.876, 0.450. Cufflinks outperforms StringTie in terms of Spearman correlation but not the other two metrics. Like the assembly results, the sequencing depth seems to have a uniform impact on the quantification accuracy and all methods favor the highest read depth. It is worth mentioning that our enumeration of read depths only focuses on down sampling. Overall, Strawberry outperforms the other methods under all evaluation metrics and sequencing depth and StringTie performs better than Cufflinks. However, the distance between the second and third place is less than the distance between the first and second place. We also observe that Strawberry and StringTie have less variability in results than Cufflinks did, suggesting they are more consistent in terms of their estimates.

When evaluated on simulated human RNA-Seq data, all three methods have lower correlations and higher relative differences compared with the true FPKM values. The order of the methods' performances slightly changes based on different evaluation metrics (**Fig. 3.5**). Strawberry has the lowest average MARD across the 6 samples compared to StringTie and Cufflinks (**Table 3.1**). When the methods are compared using Spearman correlation, the differences among the three methods are the smallest. Cufflinks performs poorly under proportionality correlation (averaged at 0.3573). StringTie achieves the highest average proportionality while Strawberry is the second. **Fig 3.5** compares the FPKM value of each predicted transcript against its best possible matched known transcript's true FPKM value. **S6 Fig** removes the predicted transcripts that are partially matched and only keeps the transcripts that fully match the known transcripts, i.e., class code equal to "=" in the Cuffcompare result. In this "match only" case, all statistics improved significantly for all the methods, and Strawberry performs the best in every comparisons (**Table 3.1**).

Table 3.1 Averaged Spearman correlation, Proportional correlation, Mean Absolute Relative Difference (MARD) for the 6 samples in *GEU*, which is a simulated Human data. These statistics are calculated based on the predicted FPKM values of 1) all reconstructed transcripts 2) only transcripts that match the known, and the true FPKM values used in the simulation.

	Method	**Avg. Sp.**	**Avg. Prop.**	**Avg. MARD**
All transcripts	Strawberry	0.7272	0.7430	0.4801
	StringTie	0.7476	0.7759	0.5392
	Cufflinks	0.7631	0.3573	0.5287
Match only	Strawberry	0.8706	0.8706	0.3144
	StringTie	0.8517	0.8704	0.4068
	Cufflinks	0.8614	0.6621	0.4561

Real RNA-Seq data

To demonstrate Strawberry utility on real data, we tested all three programs on the Homo sapiens HepG2 data from *Steijger et al.* (34). The data was downloaded from `http:`

//www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-1730/, which includes alignment results from a library of 100 million 76bp paired-end Homo sapiens RNA-Seq reads and a total of 140 NanoString probe counts. These 140 probes targeted 109 genes, designed against specific transcripts. NanoString counts were then compared to the highest FPKM value reported for transcripts consistent with the probe design (34). We followed the same procedure used in *Steijger et al.* except for using the Tophat2 alignment result and Cuffcompare for finding the best matching transcripts. Correlations between the log-transformed FPKM reported by each method and NanoString count was calculated. Strawberry again is clearly the front-runner, correlation increased by 10.3%, 5.26% compared to Cufflinks and StringTie respectively (Table 3.2). The number of probes having matched transcripts were very close for all three methods.

It's worth mentioning that the numbers reported here may not be directly comparable to the numbers in *Steijger et al.* because we use a different aligner. In *Steijger et al.*, STAR (35) was used as the default aligner. However the STAR alignment result, as a supplementary file in their paper, does not contain *XS*, which is used in the BAM format to suggest the transcription orientation from splice site dinucleotides, such as GT-AG.

Table 3.2 Correlation of FPKMs and probe counts on real RNA-Seq data HepG2. NanoString counts were compared to the FPKM values reported for three programs. The number of probes which have matching transcripts is reported on the last line.

	Strawberry	Cufflinks	StringTie
Spearman Corr.	0.640	0.580	0.608
Num. of matches	82	82	83

Fig 3.6 shows an example of a novel isoform discovered by Strawberry in the HepG2 data. At locus ENSG0000009097, Strawberry reconstructs three isoforms. Two of them matches known isoforms ENST00000591590 and ENST00000205194 based on GRCh37 Ensemble annotation. The third isoform, transcript.14285.3, contains a novel splicing junction

which is supported by 7 uniquely aligned read-pairs. Strawberry predicts the new isoform at a fraction of 0.277 in all the three predicted isoforms.

Running Time

The running time of all three program plus a simple linux word count program on RD25, RD100, and HepG2 are plotted in **Fig 3.7**. For the HepG2 data, Cufflinks tooks 62.2 min, Strawberry 12.35 min and StringTie 4.05 min. All programs were run using 8 threads on a Dell Precision T1650, equipped with Intel Core i7-3770 CPU and 16 GB RAM. Each program was given the aligned data in BAM format and the time spent on alignment is not included. To see how well these programs scale when input grows in size, we ran a simple single thread linux word count program *wc* (which is known to have linear complexity) on the SAM format of the same data. Surprisingly, StringTie is even faster than *wc*(which uses 8.69 min), and it demonstrates the simplicity of StringTie algorithm. Strawberry also scales well compared to *wc*. Cufflinks running time shoots up when the number of RNA-Seq reads grows to 100 million. Cufflinks and Strawberry both use the EM algorithm for assigning ambiguous reads to transcripts. The EM algorithm is a time consuming algorithm but the reduced data representation used in Strawberry makes it almost 5 times faster than Cufflinks.

Discussion

Strawberry adopts a step-by-step approach for transcript assembly and quantification of expression levels. We believe it is critical to assemble the transcriptome before carrying out quantification since every eukaryotic RNA-Seq experiment is likely to generate unknown transcripts even for the well-annotated species. Our previous study of alternative splicing has shown that an incomplete genome annotation can have a huge negative impact on the detection accuracy of alternative splicing events (10). Strawberry avoids strictly using gene annotations for quantification and is able to assemble novel isoforms. However, with high-

quality annotation, Strawberry can take advantage of the annotation and yield a better assembly result. The genome guided assembly is enabled by "-g" option.

Strawberry's transcriptome assembly takes advantage of the latest genome assembly and state-of-art splice-awareness aligners and is usually more accurate than the de novo assemblers. However, this makes Strawberry reliant on alignment results. Another limitation of current Strawberry's assembly is the lack of detection of alternative promoter usage and alternative polyadenylation. Unlike other alternative splicing events, de-novo detection of alternative promoter usage and alternative polyadenylation can not be inferred from junction alignments and requires some sophisticated read depth models because of the intrinsic noisiness around transcription start and end sites introduced by RNA-Seq.

Compared to current approaches such as FlipFlop, Strawberry's assemble-then-quantify procedure cannot best utilize the quantification information in the assembly step. This is because for short-read technologies, such as Illumina, the local estimates of relative abundance are the only information available for phasing distant exons during assembly. However, Strawberry's flow network algorithm is able alleviate this phasing problem by converting the exon and junction coverage into the weighs of the flows. As a result, for example, the exons and exon-exon junctions which have similar coverages will tend to form one path by the optimization algorithm.

Both Cufflinks and Strawberry use the EM algorithm for optimizing the likelihood functions. However, because of a reduced data representation, Strawberry is 10 around times faster than Cufflinks. StringTie uses a flow algorithm for quantification which is very fast compared to the EM algorithm used by Strawberry and Cufflinks. This makes it unlikely for Strawberry to outrun StringTie. Like StringTie and Cufflinks, Strawberry implements the thread-level parallellism which can process several loci at a time to greatly speed up the program.

The lack of gold standard data for the assessment of RNA-Seq applications is still a major problem for the community. The comparisons used in this paper are primarily based

on simulated data where we know the ground truth. However, the simulation programs can fall short of resembling real data in various ways, including sequencing bias, read errors, etc. Numerous studies have shown that bias can be caused by local sequences (e.g., hexamer bias) around the reads (36), position of the reads (23), GC content bias (37), etc. Lahens et al. points out the bias in RNA sequencing is highly unpredictable and might be more complicated than the few reasons aforementioned (38). Interestingly enough, using the bias correction features in Cufflinks does not lead to an increase in performance even in the real data, all probes Pearson's r 0.672 vs. 0.670 without/with bias correction (-b option). By allowing different subexon bins to have different conditional probabilities, Strawberry model has more flexibility than models assuming uniform distributions of reads along transcripts and thus may be able to account for the bias problem to some extent. However, the bias problem is still a big problem for RNA-Seq and its application. The solution to this will require effort from both the sequencing and bioinformatics communities.

Materials and methods

Assembly problem formulation

Strawberry formulates the assembly problem as an optimization problem, trying to find a parsimonious representation of transcripts which best explains the read alignments. Cufflinks is one of the pioneers which formulates the assembly problem as an optimization problem. Thus, we start with a brief review of the Cufflinks assembly algorithm and use it to introduce Strawberry's assembly algorithm.

The set of all read-pairs at a locus $\mathcal{R} = \{r_1, \ldots r_m\}$ forms a partially ordered set in which $r_i \leq r_j$ if and only if the start position, in the transcription direction, of r_i is less than or equal to r_j and the two are compatible (can arise from the same transcript). In brief, two read-pairs are incompatible if they imply two different introns and the two introns overlap (cannot arise from the same isoform) (1). Cufflinks defines a read-pair path p as a subset of $\mathcal{R}$, an ordered set of read-pairs $\{r_{a_1}, \ldots, r_{a_k}\}$ with $r_{a-1} \leq r_a$ for all $1 < a \leq k$. Then, the

assembly problem is equivalent to finding the read-pair path cover $C = \{p_1, \ldots, p_n\}$, where $\|C\|_0$ is minimized and

$$\forall r \in \mathcal{R}, \exists p \in C \wedge p \neq \emptyset, \text{ such that } r \text{ is in } p.$$

The final estimated path cover $\hat{C}$ corresponds to the set of assembled transcripts. This is a canonical computer science problem known as the Minimum Path Cover (MPC) problem (17). Cufflinks uses a maximum matching algorithm in bipartite graphs to solve the MPC problem (1).

Instead of working with individual read-pairs, Strawberry uses a sparse representation called splicing graphs, a common feature of genome-guided methods. Heber et al. defines a splicing graph $G = (V, E)$ as a directed acyclic graph (DAG) on the set of transcribed positions V and edge set E (18). G contains an edge from v_i to v_j if and only if $v_i < v_j$ and they have consecutive positions in at least one transcript. The graph G can be refined by collapsing consecutive vertices if all of them have only one outgoing edge and one ingoing edge. When doing so, the vertices V become exons (or subexons) and edges E become introns (18). We use the term,subexons, to refer to such entities throughout this paper to avoid confusion with real biological exons. Note that subexons are ordered such that $v_i < v_j$ if subexon v_i starts upstream of subexon v_j. Furthermore, a read-pair path can be mapped to an ordered collection of subexons, which we call a subexon path.

The splicing graph can be constructed from either a set of transcripts or from read-pairs. Under the assembly mode, Strawberry builds splicing graphs from read-pairs and then assembles the nodes (subexons) into transcripts. Under the splicing graph representation, a similar MPC problem arises on the subexon level. Since the splicing graph is a sparse representation of the read-pairs, assembly on the splicing graph is more time efficient than assembly with the read-pairs. This subexon representation also has a positive impact on quantification, since read-pair counts on subexons can be seen as compact sufficient statistics for our quantification model. The idea of quantification is discussed in more detail in the quantification section.

Our flow network algorithm requires some modifications on the splicing graph. A source node v_s connecting to all subexon(s) at the 5' end site(s), and a target node v_t connecting to the subexon(s) having at the 3' end site(s) are added to the splicing graph. We use the word (s,t)-path (in order to reserve the use of subexon path for quantification) to refer to an ordered set of subexons from v_s to v_t, inclusive. Notice that v_s and v_t are not real exons. Our new MPC problem on the splicing graph can be defined as finding a minimum set of (s,t)-paths which can cover every subexon at least once. The purpose of including nodes, v_s and v_t, is to remove partial or incomplete transcripts. In other words, each full transcript corresponds to a (s,t)-path which flows from a promoter region (v_s) to a terminator (v_t).

Constructing a weighted splicing graph

To define nodes and edges in the splicing graph, Strawberry separately retrieves primitive exons from the coverage data and retrieves introns from junction alignments. A primitive exon is defined as a continuous stretch of genomic positions covered by reads. An intron is defined as a unique junction alignment. The introns are then used to cut the primitive exons into subexons which are the final nodes defined in the splicing graph (**Fig 3.8**). However, in simulated data, many inferred introns are not real because of false junction calls by aligners. There is evidence these false calls also appear in real data (19). Strawberry uses the same criteria to pre-filter introns as in Cufflinks (1). The thresholds are arbitrary but work well in practice. Putative introns are discarded if any of the following apply.

- More than 70% of the reads supporting an intron are not uniquely aligned.
- If two introns overlap and one's expression is less than 5% of the other, then the one with lower expression is removed. Intron expression is defined as the total number of junction reads.
- The number of small overhang reads supporting a junction is likely to be low under the assumption that reads are distributed uniformly along their parent transcripts.

A small overhang read is a particular junction read where one end of the read is mapped within a small distance (we use 6 bp) of a subexon-intron boundary. The expected number of small overhang reads is calculated from a binomial distribution, $\text{Bin}(n, p)$, where n is the total junction reads and $p = \frac{2s}{l-1}$, s being the small overhang distance and l being read length. When n is large (e.g., > 100), we use the normal approximation $N(np, np(1-p))$.

Next, nodes (subexons) are connected in the splicing graph. Each subexon is either fully contained or excluded in any transcript. Two subexons are connected by an edge, which does not necessarily represent an intron, when they are consecutive in their genomic coordinates (see **Fig 3.8**). For the non-intron edges, the number of reads covering at least 6 bp of both subexons is used as the edge weight representing the support for these two subexons being in the same transcript. For the intron edges, the weight is simply the total junction read number. In the implementation, Strawberry negates the weight and adds the maximum weight to make all weights positive. The algorithm, described next, will solve for the minimum total weight.

Optimization with flow network

We have reformulated the problem on a splicing graph G, where (s, t)-paths (full-length transcripts) are ordered collections of subexons, and we seek a minimum path cover (MPC) of G. The ordinary MPC problem is not a good fit for the splicing graph since it only requires that every node (subexon) is covered at least once, leaving the possibility that some edges (indicating two subexons are consecutive in the transcriptome) might not be covered. Also, a read-pair (due to the unsequenced proportion) can span two non-consecutive nodes. These non-consecutive nodes (if they exist) constitute a subpath (**Fig 3.9**), denoted by p^{sub}, that also must be covered by at least one (s, t)-path in the cover. All the edges and subpaths constitute the constraints in a Constrained MPC (CMPC) problem. An efficient algorithm for solving the Constrained MPC (CMPC) problem has been advanced (20).

Definition 1 *CMPC problem. Given a DAG G with nodes $V(G)$ and edges $E(G)$, and a weight $w(e)$ for each $e \in E(G)$, and a set of subpaths $\{p_j^{sub} | j \in 1, ..., t\}$ the task is to find a minimum number of k directed paths $\{p_i | i \in 1, ..., k\}$ in G such that*

- *Every node in $V(G)$ occurs at least once in some p_i.*
- *Every edge in $E(G)$ occurs at least once in some p_i.*
- *Every path p_j^{sub} is entirely contained in some p_i.*
- *Every path p_i starts in v_s and ends in v_t, where v_s and v_t are the source and target nodes of G.*
- *$\sum_{i=1}^{k} \sum_{e \in p_i} w(e)$ is minimum among all solutions of k paths.*

Rizzi el at. showed that the CMPC problem can be reduced to the MPC problem with node constraints (20). The MPC with node constraints can be found using one of the well established flow network algorithms, e.g., the min-cost circulation flow algorithm (17), where a strong polynomial time solution is guaranteed. In a nutshell, a flow network is a DAG $G = (V, E)$ with source $v_s \in V(G)$ and target $v_t \in V(G)$, where every edge $e \in E(G)$ has an upper $u(e)$ and lower $l(e)$ capacity limit and flow $f(e)$ associated with it. The solution to a flow network problem is to construct a map, $f : E \rightarrow R$, which maps an edge to a real number or an integer, called a flow. The flow decomposition theorem (see, e.g., (17)) guarantees the flow network can be used to solve the MPC problem. It says that the flow $f(e)$ on edge e can be decomposed into a set of flows on the (s, t)-path. However the decomposition is not unique, which we overcome using a greedy algorithm.

Algorithm 1 *Constrained Minimum Path Cover Algorithm (CMPC) (20)*

1. *Add edges to the subpath constraints. Let $P^{sub} = \{p_i^{sub}\}$ denote the set of subpath constraints. Grow the P^{sub} to include all edges as subpath constraints.*
2. *Drop duplicates. For every pair of path constraints p_i^{sub} and p_j^{sub}, set P^{sub} to $P^{sub} \backslash p_i^{sub}$, if p_i^{sub} is contained in p_j^{sub}.*

3. *For every original path constraint* p_i^{sub} *which starts at node* u *and ends at node* v *and* $(u, v) \notin E(G)$*, do:*

   - $E(G) := E(G) \cup \{(u, v)\}$*. Add a new edge* (u, v) *directly from the start node of the subpath to the end node of the subpath.*
   - *Set the lower and upper bounds for this new edge:* $lower(u, v) = 1$ *and* $upper(u, v) = \inf$*.*
   - *The weight of the new edge is the sum of weights of the original subpath:* $w(u, v) = \sum_{e \in p_i^{sub}} w(e)$*.*

4. *For each* $e \in E(G)$ *and* $e \notin P^{sub}$*, set* $lower(e) = 0$ *and* $upper(e) = \inf$*.*

5. *Add an edge* (v_t, v_s) *from sink node* v_t *to start node* v_s *to complete the circle. Set lower and upper bounds for this edge as well:* $lower(t, s) = 0$ *and* $upper(t, s) = \inf$*.*

6. *Compute a min-weight min-flow circulation on this transformed input* G *with the following properties.*

   - G *is a flow network which satisfies capacity constraints and flow conservation constraints.*
   - *Min flow:* $\sum_{e \in E(G)} f(e)$ *is minimum.*
   - *Min weight:* $\sum_{e \in E(G)} w(e)$ *is minimum.*

7. *Finally, the integer flow on edge* (v_t, v_s) *equals to the achieved min-flow. We decompose the flow network into this number of paths and each path corresponds to an assembled transcript.*

Fig 3.9 demonstrates a toy example of this algorithm.

Quantification with latent class model

Strawberry's quantification model is based on the generative model proposed in (1). As in *Salzman et al. 2011* (16), Strawberry collapses data into sufficient statistics, but to match

the assembly, Strawberry collapses the data into subexon paths, defined on the splicing graph. In theory, for a gene with w subexons, Strawberry produces $2^w - 1$ equivalent classes independent of the number of isoforms. In contrast, the number of classes in (16) depends on the number of isoforms. Although *Salzman et al. 2011* achieves greater collapsing, Strawberry has a richer parameterization and is able to account for nonuniform distribution of the reads along a transcript. Either way, the idea of collapsing greatly reduces the number of observations and speeds up the calculation.

To describe the Strawberry model, we start with the definition of subexon path. A read-pair can be reduced to a unique set of ordered subexons, called a subexon path. The map from read-pair space $\mathcal{R}$ to subexon path space $\mathcal{S}$ is surjective. Strawberry's data reduction strategy creates an equivalency between the subexon paths $\mathcal{S}$ and a partition of fragments $\mathcal{F}$ (and hence reads $\mathcal{R}$). It collapses read-pairs based on the set of subexons they cross. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_L\}$ be the collection of subexon paths. Subexon paths are equivalent to sets of genomic intervals $\{[G_{sx}, G_{sy}] \mid \forall s \in S_l\}$, where G_{sx} and G_{sy} are the smallest and largest genomic positions in subexon s. Each observable read-pair $\boldsymbol{r}$ can be represented as a 4-tuple, $(u_{5'}, u_{3'}, d_{5'}, d_{3'})$, where u and d represent the upstream and downstream reads, $5'$ and $3'$ their respective ends, both along the transcription direction. Then we can partition (or project) the $\mathcal{R}$ onto S, so that a read pair r is assigned to a subexon path S_l if and only if r overlaps with only subexons in S_l and all subexons forming S_l have been hit by this r, i.e., $r \in S_l \Leftrightarrow \text{cond.1} \wedge \text{cond.2}$, where $\text{cond.1} = \forall s \in S_l, [G_{sx}, G_{sy}] \cap [u_{3'}, u_{5'}] \neq \emptyset \vee [G_{sx}, G_{sy}] \cap [d_{5'}, d_{3'}] \neq \emptyset$ and $\text{cond.2} = \forall j \in [u_{3'}, u_{5'}] \cup [d_{5'}, d_{3'}], \exists s$ such that $j \in [G_{sx}, G_{sy}]$. This definition ensures each r is uniquely assigned to a S_l. Notice, if a read pair contains an unsequenced portion, such as the insert, the subexon path of the read-pair is an incomplete observation of the unobserved set of subexons. However, when conditioning on the isoform, a subexon path can become a complete observation of the fragment from which the read-pair is generated. Therefore, an subexon path can be included or excluded from an isoform just like the read-pair. For each gene g, we derive a binary matrix C with L_g rows and K_g

columns, where we assume gene g has L_g subexon paths and K_g isoforms and $C_{kl} = 1$ if isoform k contains subexon path l, otherwise 0. If there are total n_g read-pairs observed for gene g, we derive our observation $\{\boldsymbol{y}_i\}_{n_g}$, where each element $\boldsymbol{y}_i$ identifies the subexon path of the read-pair i, i.e., $\boldsymbol{y}_i$ is an L_g-dimensional vector, one of the standard basis vectors of L_g-dimensional Euclidean space. In practice, Strawberry only uses the observed subexon path whose number so L_g is smaller than the theoretical number.

Like the assembly, this model handles one locus from a single sample at a time, allowing maximum parallelization. Our generative model for RNA-Seq is as follows. Transcripts from isoform k make up a proportion η_k in the sample. Transcripts are randomly fragmented, and long isoforms produce more fragments than short isoforms. Isoform k fragments constitute approximately proportion $\pi_k \approx l_k \eta_k$ in the sample. Having estimated $\hat{\pi}_k$ and knowing l_k, we can later retrieve η_k (1). Given the isoform of origin k, the fragment is considered as generated from the underlying subexon path as a one-trial multinomial experiment $\text{Mult}(1, \boldsymbol{\theta}_{k\cdot})$, where θ_{kl} is a conditional probability of the fragment generating from subexon path l. For a given read set $\mathcal{R} = \{\boldsymbol{y}_i\}_n$, the likelihood can be written as

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathcal{R}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \prod_{l=1}^{L} \theta_{kl}^{y_{il}} \tag{3.3}$$

Following the line of generative model of sampling transcripts first and then the fragments conditioning on the transcript and accounting for the read-isoform assignment uncertainty using a mixture model. Strawberry simultaneously estimates the class probability $\boldsymbol{\pi}$ and the conditional probability $\boldsymbol{\theta}$ under a EM algorithm (21) framework, while other models (1; 6; 22; 16) assume fixed conditional probability when estimating the class probabilities. Strawberry has a richer set of parameters which allow it to account for the non-uniform distribution of reads along transcripts often observed in real data (23; 24). Jiang et al. also proposed a model that simultaneously estimates the class probabilities and conditional probabilities for robust estimation of isoform expression (25) . However, their model has far more parameters than ours and uses a penalized likelihood. Because they don't publish their program, the actual performance of their model is unknown.

Estimation

We use the EM algorithm proposed for basic latent class models (26) and summarize in algorithm 2:

Algorithm 2

- *Initialize*

$$\pi_k = 1/K,$$
$$\theta_{kl} = \sum_t \frac{q(t) \cdot n_{klt}}{l_k - t + 1},$$

  where we sum over possible fragment length t conditioning on subexon bin l and isoform k. Here, $q(\cdot)$ is the empirical fragment distribution and n_{klt} is number of possible fragments with length t and l_k is the isoform length.

- *repeat EM steps until convergence.*

  - *E-step:*

$$\hat{n_{kl}}^{m+1} = \frac{n_l \hat{\pi_k}^m \hat{\theta_{kl}}^m}{\sum_{k=1}^K \hat{\pi_k}^m \hat{\theta_{kl}}^m}.$$

  - *M-step:*

$$\hat{\pi_k}^{m+1} = \frac{\sum_{l=1}^L \hat{n_{lk}}^{m+1}}{n},$$
$$\hat{\theta_{kl}}^{m+1} = \frac{\hat{n_{kl}}^{m+1}}{\sum_{l=1}^L \hat{n_{kl}}^{m+1}}.$$

The parameter $\boldsymbol{\theta}$ is initialized using the concept of *read type* (same as our read-pair concept) and *sample rate* $\boldsymbol{\alpha}$ in (16). The probability of observing a read pair r is $\sum_{k=1}^K \pi_k \alpha_{kr}$ where

$$\alpha_{kr} = \begin{cases} \frac{q(t_k)}{l_k - t_k + 1}, & \text{if } r \text{ is compatible to isoform k.} \\ 0, & \text{if } r \text{ is not compatible to isoform k.} \end{cases}$$

We use t_k to denote the fragment length of a read-pair under the isoform k. Note that Salzman et al.'s model assumes reads are generated uniformly when its isoform of origin

is known. Strawberry learns an empirical fragment size distribution $q(\cdot)$ from a place in genome ($> 2kb$) where no alternative splice sites exist according to the read alignments. If the input is single end reads, Strawberry relies on the users to define a Gaussian distribution for the fragment length. We assume the random fragmentation step in sample preparation leads to a nearly Gaussian distribution (24), but it is common to approximate the distribution using an empirical one (1).

Strawberry calculates the initial estimate of θ_{kl} for each pair of isoform k and subexon path l by summing α_{kr} over all potential read-pairs on subexon path l including the ones that are not observed:

$$\theta_{kl} = \begin{cases} \sum_{r \in S_l} \alpha_{kr}, & \text{if } C_{kl} = 1. \\ 0, & \text{if } C_{kl} = 0. \end{cases} \tag{3.4}$$

The summation in Eq 3.4 requires summing over all possible fragment lengths and conditioning on a fragment length, the possible 5' end which r can be generated from a given subexon path and transcript combination (**Fig 3.10**).

Implementation

Strawberry was written in C++14 and utilizes features such as threading library for parallelization. *Lemon* (27), a C++ graph template library, was used in assembly and *Eigen3* (http://eigen.tuxfamily.org), a C++ template library for linear algebra, was used in quantification. Strawberry is available as a free software at https://github.com/ruolin/strawberry under the MIT license.

Conclusion

This paper introduced Strawberry, a fast, accurate genome-guide assembler and quantification tool for RNA-Seq data. It facilitates transcriptome assembly and calculation of transcript-level expression. Based on our simulation, Strawberry not only recovers more true transcripts while achieving the same false discovery rate in assembly compared to two

other leading methods but also outperforms them in terms of the quantification accuracy. Using the real data from a highly cited method comparison study, we again show that Strawberry beats Cufflinks and StringTie by convincing margins. The other advantage of Strawberry is its speed and good scalability, makes it an intriguing candidate when processing large dataset (e.g., $>$ 100 million reads). It takes 12.35 min for Strawberry to process 100 million input RNA-Seq reads while a simple Linux program *wc* takes 8.69 min. Strawberry achieves this level of speed and accuracy through applying the min-cost, min-flow algorithm to assembly, a reduced data representation to subexon path counts which arise naturally from the splicing graph and latent class model used in the quantification step. Strawberry is written in C++14 and is fully self-contained. The installation does not require any pre-installation packages except for *g++ compiler* and *CMake*. Strawberry applies to both single-end and paired-end libraries, and also supports strand-specific protocols.

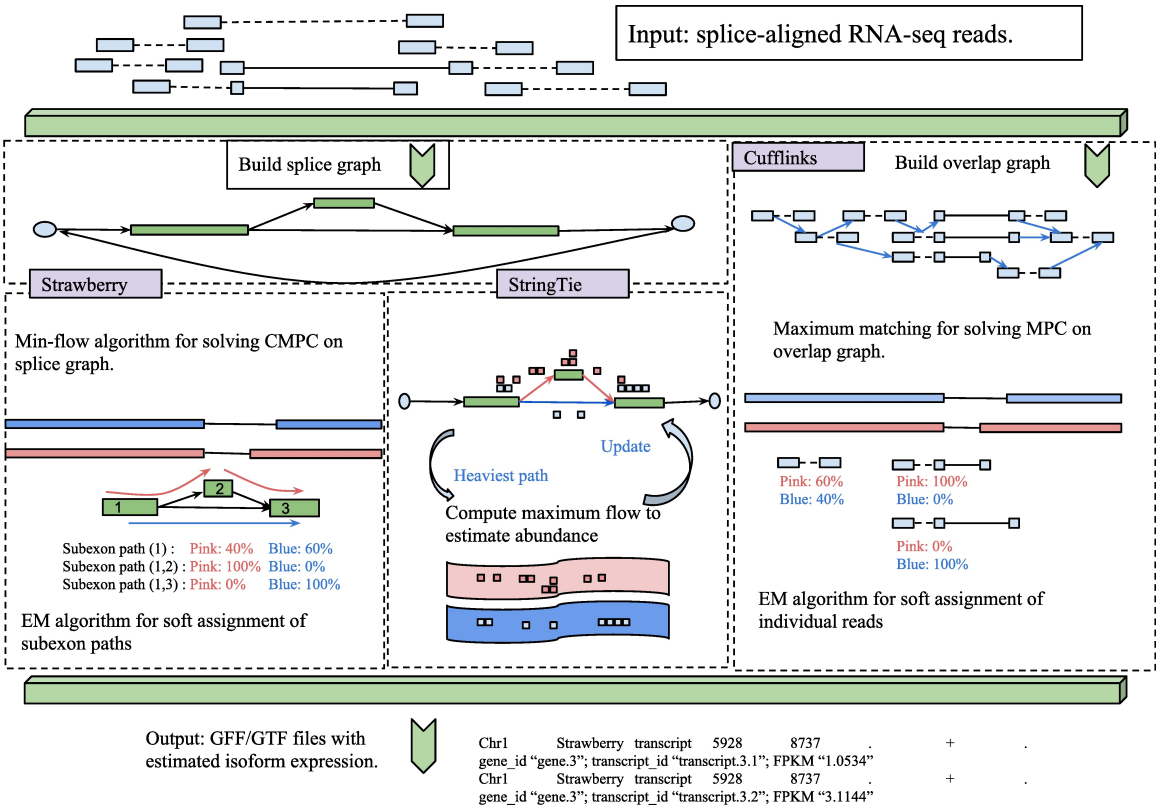


Figure 3.1 Overview of the algorithm of Strawberry, compared to StringTie and Cufflinks. All methods begin with a set of RNA-Seq alignments and output transcript structures and abundances in GFF/GTF format. Strawberry uses a min-flow algorithm for solving Constrained Minimum Path Cover(CMPC) problem on splicing graph, followed by assigning subexon paths to compatible assembled transcripts. In quantification step, all of the RNA-Seq read alignments on each subexon path as a whole are the subject of the EM algorithm.

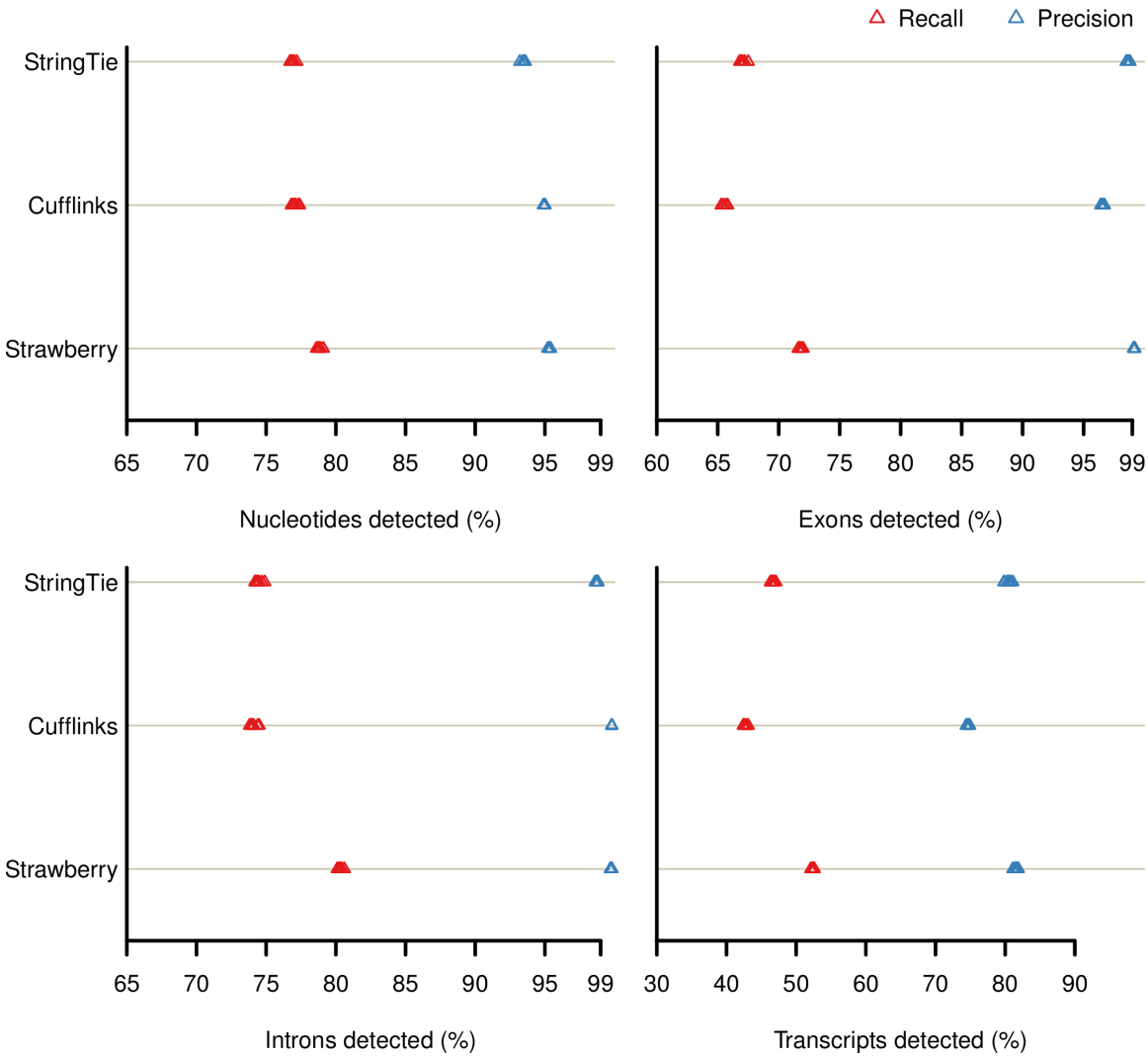


Figure 3.2 recall and precision at the nucleotide, exon, intron and transcript level. StringTie, Cufflinks and Strawberry were run on data *RD100*, which is a simulated Arabidopsis RNA-Seq data set.

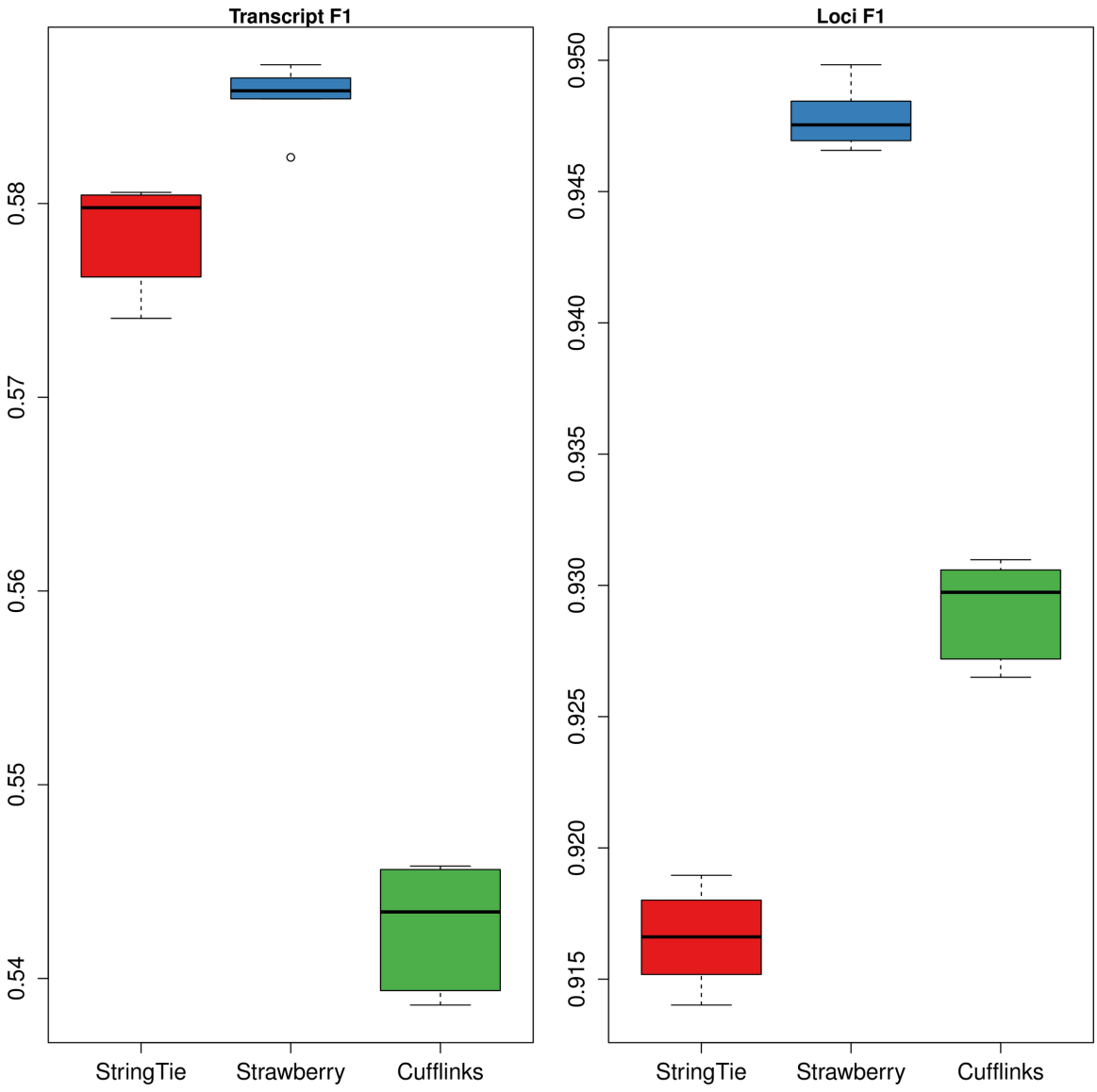


Figure 3.3 Box plots of F1 scores at the transcript and loci level. StringTie, Cufflinks and Strawberry were evaluated on data *GEU*, which is a simulated Human RNA-Seq data set.

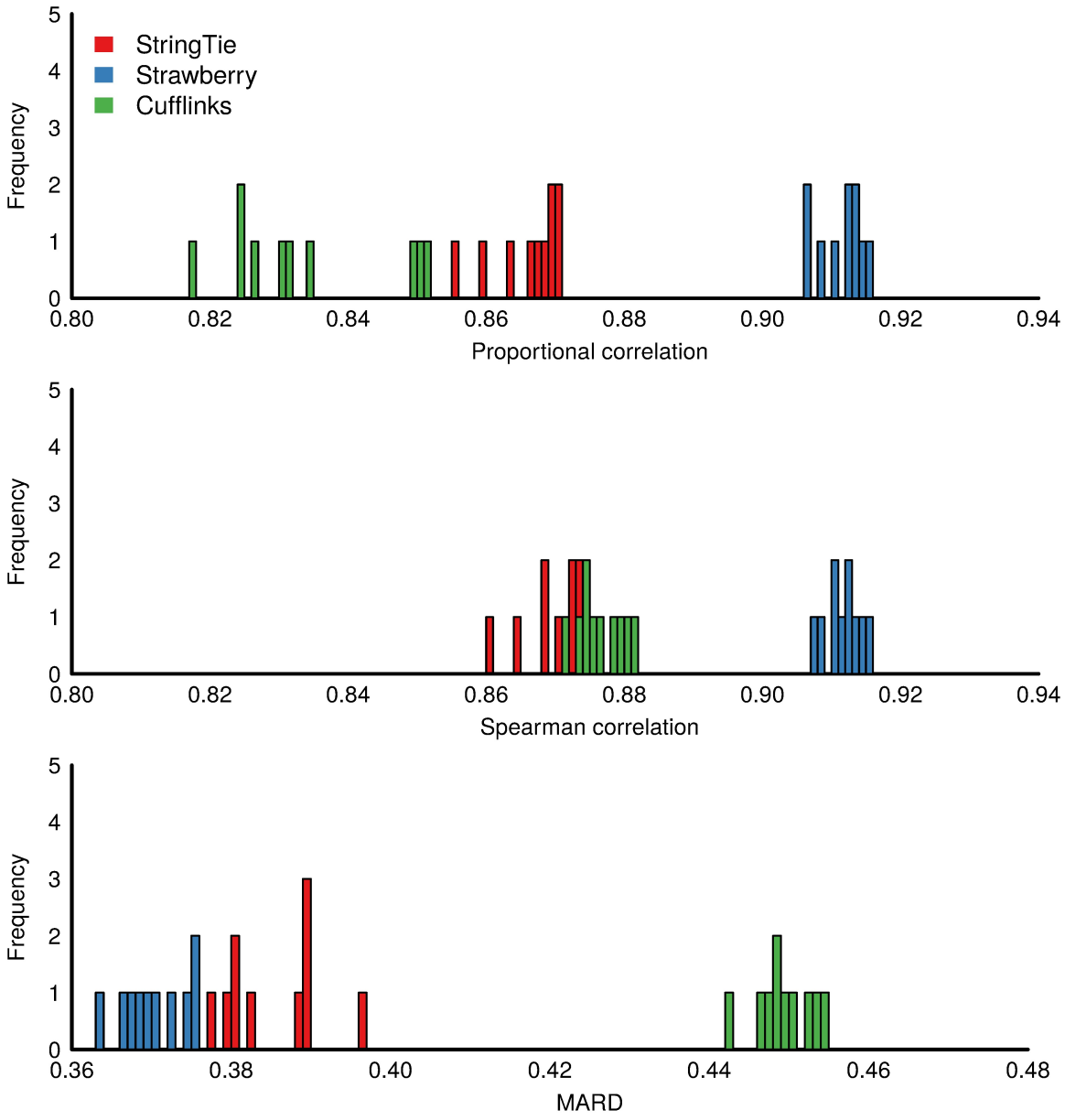


Figure 3.4 Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 10 replicates in *RD100*, which is a simulated Arabidopsis data.

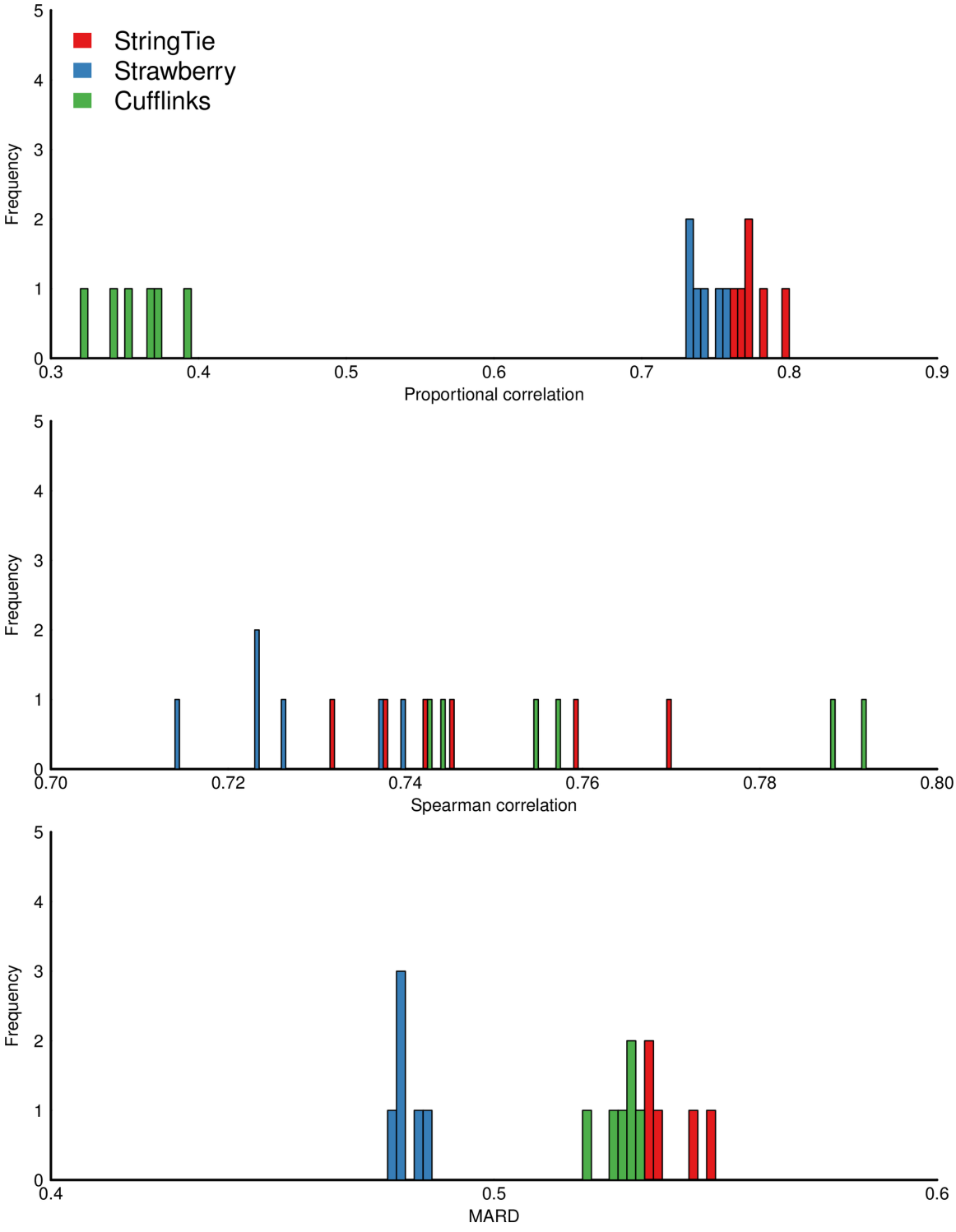


Figure 3.5 Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 6 samples in *GEU*, which is a simulated Human data. These statistics are calculated based on the predicted FPKM values of all reconstructed transcripts and the true FPKM values used in the simulation.

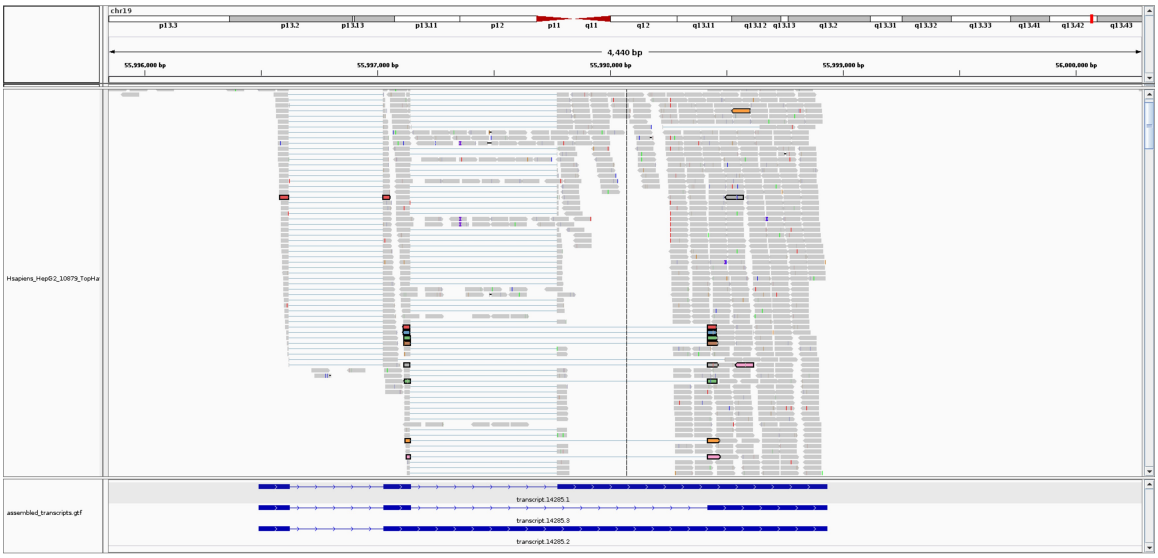


Figure 3.6 Read alignments and reconstructed transcripts at gene NAT14 using HepG2 data. A new isoform, transcript.14285.3 (shown as the middle one), has been identified by Strawberry. The junction reads that support the new AS event (alternative 3 splice site) are highlighted. The two ends of a read-pair are in the same color. A total 7 uniquely mapped read-pairs supports the novel junction. This figure is made by IGV (http://software.broadinstitute.org/software/igv/)

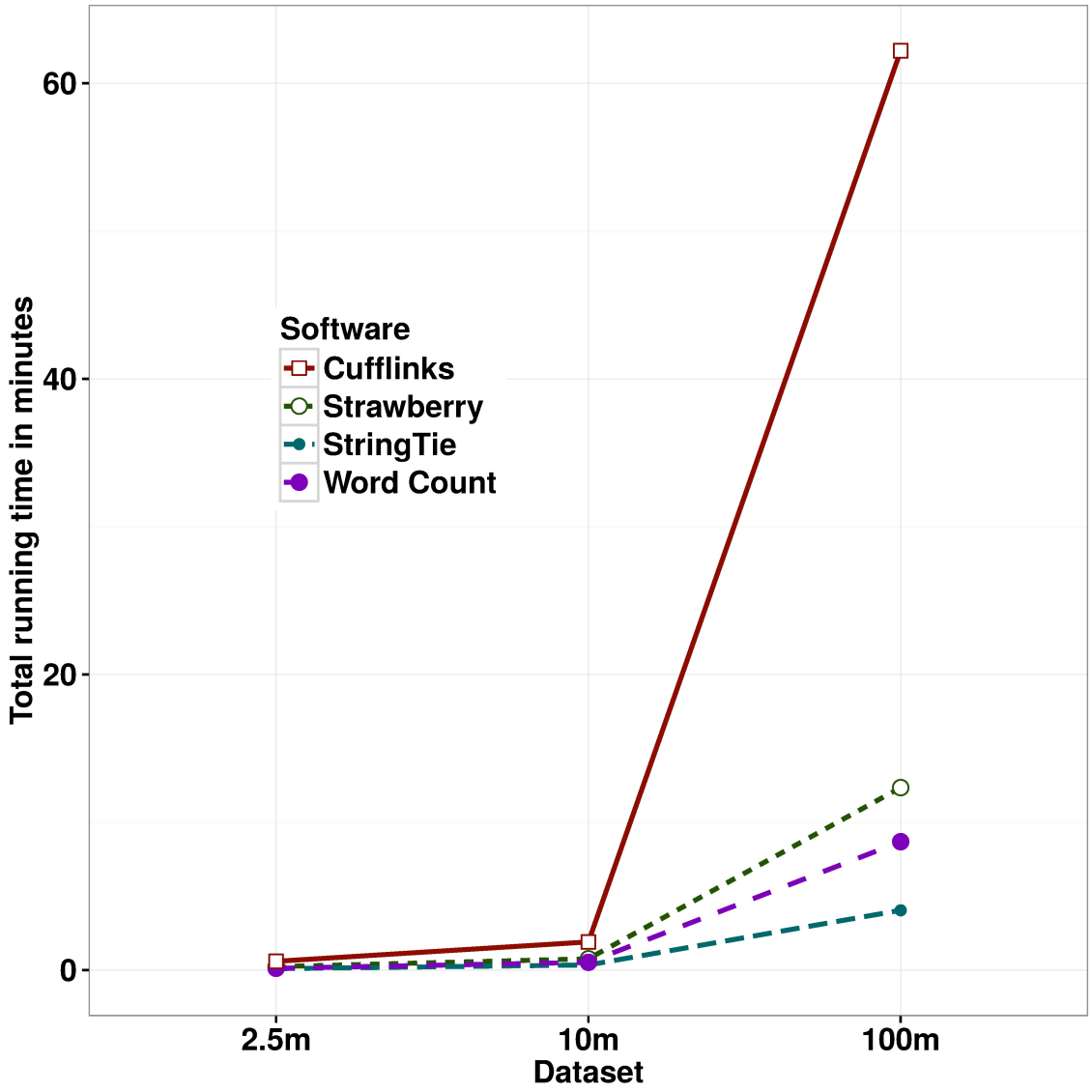


Figure 3.7 Running time in minutes of Cufflinks, Strawberry, linux word count and StringTie(ordered by slowest to fastest) on textitRD25(2.5 million reads), *RD100*(10 millions reads), and *HepG2* data(100 millions reads).

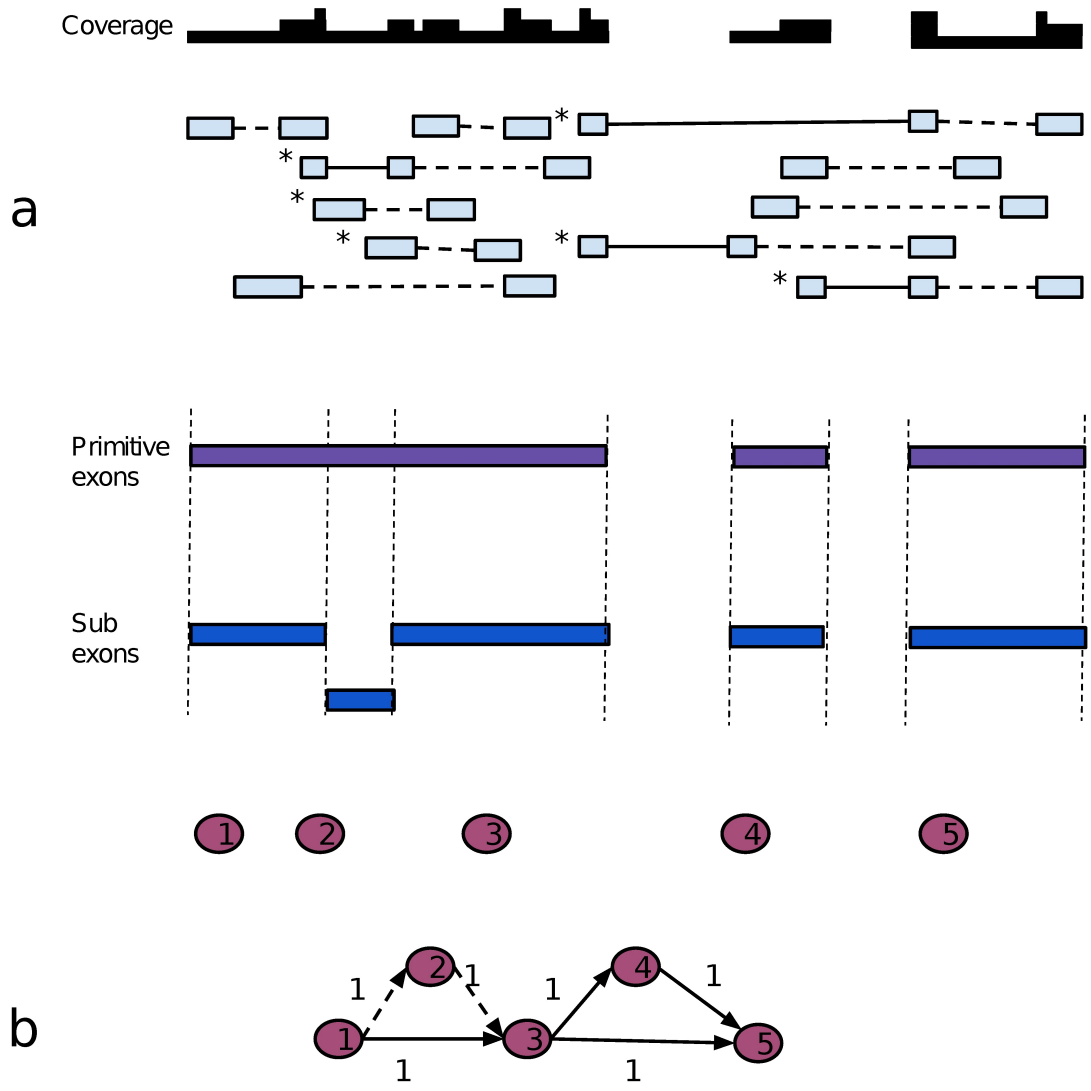


Figure 3.8 Translation of read alignments into a splicing graph. (a) Eleven imaginary aligned paired-end reads (or read-pairs) are represented by light blue boxes intersected by solid lines, which indicate splicing junctions, and broken lines, which indicates gap sequences. Above the read-pairs, the coverage plot is shown. The white regions have zero coverage. Below the read-pairs, three primitive exons are shown as purple boxes and five subexons in dark blue, numbered from 1-5. (b) The splicing graph constructed from part (a). The numbered nodes in the splicing graph are subexons from part (a). Dashed Arrows represent the non-intron edges and solid arrows indicate the intron edges. The numbers next to edges are the weights(number of read-pairs supports). A read-pair that contributes to an edge weight is stressed using an asterisk near its upper-left corner. All the arrows also indicate the transcription direction. The source node and target node in the splicing graph are not shown.

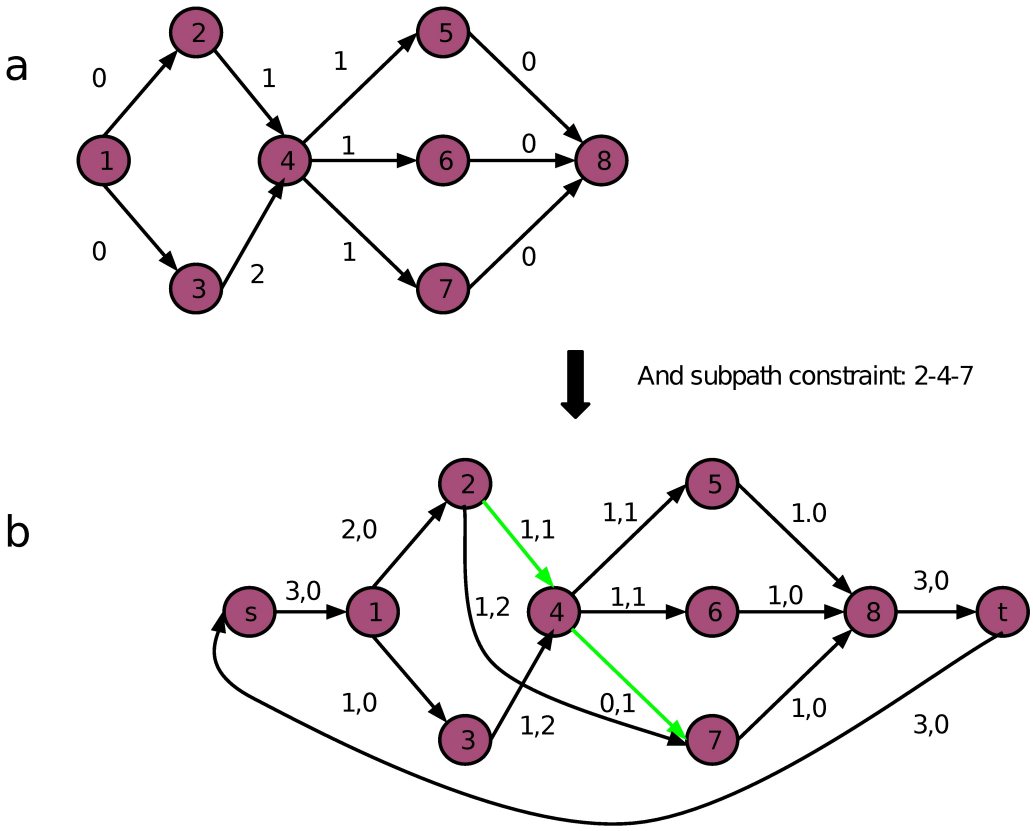


Figure 3.9 An input flow network with a subpath constraint {2-4-7}. (a),the number next to an edge is the edge cost. For every edge e, the edge constraint implies $1 \leq f(e) \leq$ inf. (b), the transformed min-flow circulation network. The 2-tuple (a,b) next to each edge indicates the optimal flow on the edge and the edge cost respectively. After Step 3, the path constraints set is $P^{\text{sub}} = \{(1,2),(1,3),(2,4,7),(4,5),(4,6),(5,8),(6,8),(7,8)\}$. Two edges no longer in the constraint set are shown in green. For these two edges, the minimum flow requirement is 0; for the rest of edges, it is 1. Two dummy nodes, s and t, are added to complete the circulation. The number of flows after decomposition is equal to the minimum flow which is 3.

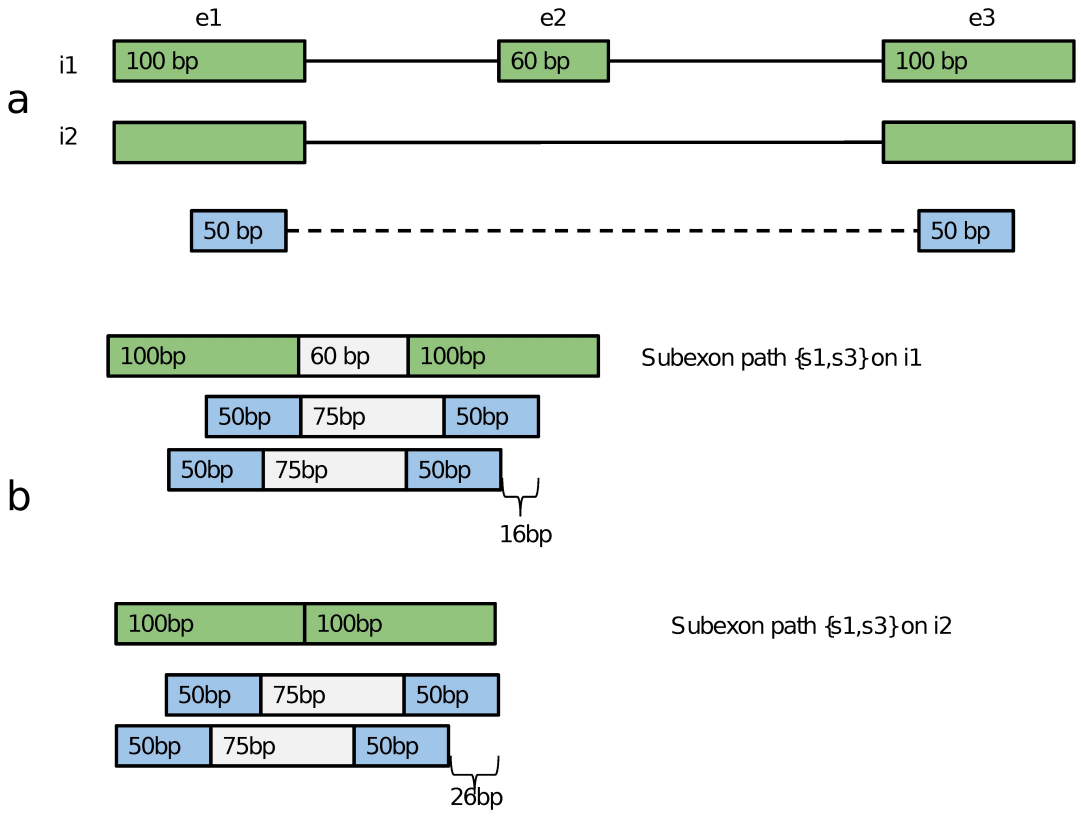


Figure 3.10 (a), a gene with three subexons and two isoform are shown. The length of i1 is 260 bp, i2 200 bp. A paired-end read (or read-pair) is represented by light blue boxes intersected by broken lines, which indicates gap sequences. The read length is 50x2 bp. (b) A subexon path $\{s_1, s_3\}$ applies to both isoform. When on i1, this subexon path implies three subexons with the one in middle shown in gray. Consider a fixed size fragment with gap size 75 bp(shown in gray) and total fragment length 175 bp. This particular fragment can arise from 16 different positions from subexon path $\{s_1, s_3\}$ on i1 and 26 different positions from subexon path $\{s_1, s_3\}$ on i2.

Bibliography

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

[2] Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T. C., Mendell, J. T., and Salzberg, S. L. (Mar, 2015) StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.*, **33**(3), 290–295.

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

[4] Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., and Kingsford, C. (2017) Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods*, **14**, 417–419.

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

[6] Rossell, D., Stephan-Otto Attolini, C., Kroiss, M., and Stocker, A. (Mar, 2014) Quantifying alternative splicing from paired-end rna-sequencing data. *Ann Appl Stat*, **8**(1), 309–330.

[7] Roberts, A. and Pachter, L. (Jan, 2013) Streaming fragment assignment for real-time analysis of sequencing experiments. *Nat. Methods*, **10**(1), 71–73.

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

[9] Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (May, 2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**(5), 525–527.

[10] Liu, R., Loraine, A. E., and Dickerson, J. A. (Dec, 2014) Comparisons of computational methods for differential alternative splicing detection using RNA-seq in plant systems. *BMC Bioinformatics*, **15**(1), 364.

[11] Tomescu, A. I., Kuosmanen, A., Rizzi, R., and Makinen, V. (2013) A novel min-cost flow method for estimating transcript expression with RNA-Seq. *BMC Bioinformatics*, **14 Suppl 5**, S15.

[12] Mezlini, A. M., Smith, E. J., Fiume, M., Buske, O., Savich, G. L., Shah, S., Aparicio, S., Chiang, D. Y., Goldenberg, A., and Brudno, M. (Mar, 2013) iReckon: simultaneous isoform discovery and abundance estimation from RNA-seq data. *Genome Res.*, **23**(3), 519–529.

[13] Li, W., Feng, J., and Jiang, T. (Nov, 2011) IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *J. Comput. Biol.*, **18**(11), 1693–1707.

[14] Garber, M., Grabherr, M. G., Guttman, M., and Trapnell, C. (Jun, 2011) Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat. Methods*, **8**(6), 469–477.

[15] Song, L. and Florea, L. (2013) CLASS: constrained transcript assembly of RNA-seq reads. *BMC Bioinformatics*, **14 Suppl 5**, S14.

[16] Salzman, J., Jiang, H., and Wong, W. H. (Feb, 2011) Statistical Modeling of RNA-Seq Data. *Stat Sci*, **26**(1).

[17] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993) Network flows : theory, algorithms, and applications, Prentice Hall, Upper Saddle River (N.J.).

[18] Heber, S., Alekseyev, M., Sze, S. H., Tang, H., and Pevzner, P. A. (2002) Splicing graphs and EST assembly problem. *Bioinformatics*, **18 Suppl 1**, S181–188.

[19] Engstrom, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Ratsch, G., Goldman, N., Hubbard, T. J., Harrow, J., Guigo, R., Bertone, P., Alioto, T., Behr, J., Bertone, P., Bohnert, R., Campagna, D., Davis, C. A., Dobin, A., Engstrom, P. G., Gingeras, T. R., Goldman, N., Grant, G. R., Guigo, R., Harrow, J., Hubbard, T. J., Jean, G., Kahles, A., Kosarev, P., Li, S., Liu, J., Mason, C. E., Molodtsov, V., Ning, Z., Ponstingl, H., Prins, J. F., Ratsch, G., Ribeca, P., Seledtsov, I., Sipos, B., Solovyev, V., Steijger, T., Valle, G., Vitulo, N., Wang, K., Wu, T. D., and Zeller, G. (Dec, 2013) Systematic evaluation of spliced alignment programs for RNA-seq data. *Nat. Methods*, **10**(12), 1185–1191.

[20] Rizzi, R., Tomescu, A. I., and Makinen, V. (2014) On the complexity of Minimum Path Cover with Subpath Constraints for multi-assembly. *BMC Bioinformatics*, **15 Suppl 9**, S5.

[21] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, Series B*, **39**(1), 1–38.

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

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

[24] Roberts, A. Ambiguous fragment assignment for high-throughput sequencing experiments PhD thesis EECS Department, University of California, Berkeley (Oct, 2013).

[25] Jiang, H. and Salzman, J. (2015) A penalized likelihood approach for robust estimation of isoform expression. *Statistics and Its Interface*, (4), 437–445.

[26] McCutcheon, A. L. (1987) Latent class analysis, Sage Publication .

[27] Porkolb, Z., Pataki, N., Dezs, B., Jttner, A., and Kovcs, P. (2011) Proceedings of the Second Workshop on Generative Technologies (WGT) 2010 LEMON an Open Source C++ Graph Template Library. *Electronic Notes in Theoretical Computer Science*, **264**(5), 23 – 45.

[28] Guennebaud, G., Jacob, B., et al. Eigen v3.

[29] Lamesch, P., Berardini, T. Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., Muller, R., Dreher, K., Alexander, D. L., Garcia-Hernandez, M., Karthikeyan, A. S., Lee, C. H., Nelson, W. D., Ploetz, L., Singh, S., Wensel, A., and Huala, E. (Jan, 2012) The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucleic Acids Res.,* **40**(Database issue), D1202–1210.

[30] Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigo, R., and Sammeth, M. (Nov, 2012) Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res.,* **40**(20), 10073–10083.

[31] Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S. L. (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.,* **14**(4), R36.

[32] Kim, D., Langmead, B., and Salzberg, S. L. (Apr, 2015) HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods,* **12**(4), 357–360.

[33] Quinlan, A. R. and Hall, I. M. (Mar, 2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics,* **26**(6), 841–842.

[34] Steijger, T., Abril, J. F., Engstrom, P. G., Kokocinski, F., Hubbard, T. J., Guigo, R., Harrow, J., Bertone, P., Abril, J. F., Akerman, M., Alioto, T., Ambrosini, G., Antonarakis, S. E., Behr, J., Bertone, P., Bohnert, R., Bucher, P., Cloonan, N., Derrien, T., Djebali, S., Du, J., Dudoit, S., Engstrom, P., Gerstein, M., Gingeras, T. R., Gonzalez, D., Grimmond, S. M., Guigo, R., Habegger, L., Harrow, J., Hubbard, T. J., Iseli, C., Jean, G., Kahles, A., Kokocinski, F., Lagarde, J., Leng, J., Lefebvre, G., Lewis, S., Mortazavi, A., Niermann, P., Ratsch, G., Reymond, A., Ribeca, P., Richard, H., Rougemont, J., Rozowsky, J., Sammeth, M., Sboner, A., Schulz, M. H., Searle, S. M., Solorzano, N. D., Solovyev, V., Stanke, M., Steijger, T., Stevenson, B. J., Stockinger, H., Valsesia, A., Weese, D., White, S., Wold, B. J., Wu, J., Wu, T. D., Zeller, G., Zerbino, D., and Zhang, M. Q. (Dec, 2013) Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods,* **10**(12), 1177–1184.

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

[36] Hansen, K. D., Brenner, S. E., and Dudoit, S. (Jul, 2010) Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Res.,* **38**(12), e131.

[37] Benjamini, Y. and Speed, T. P. (May, 2012) Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.,* **40**(10), e72.

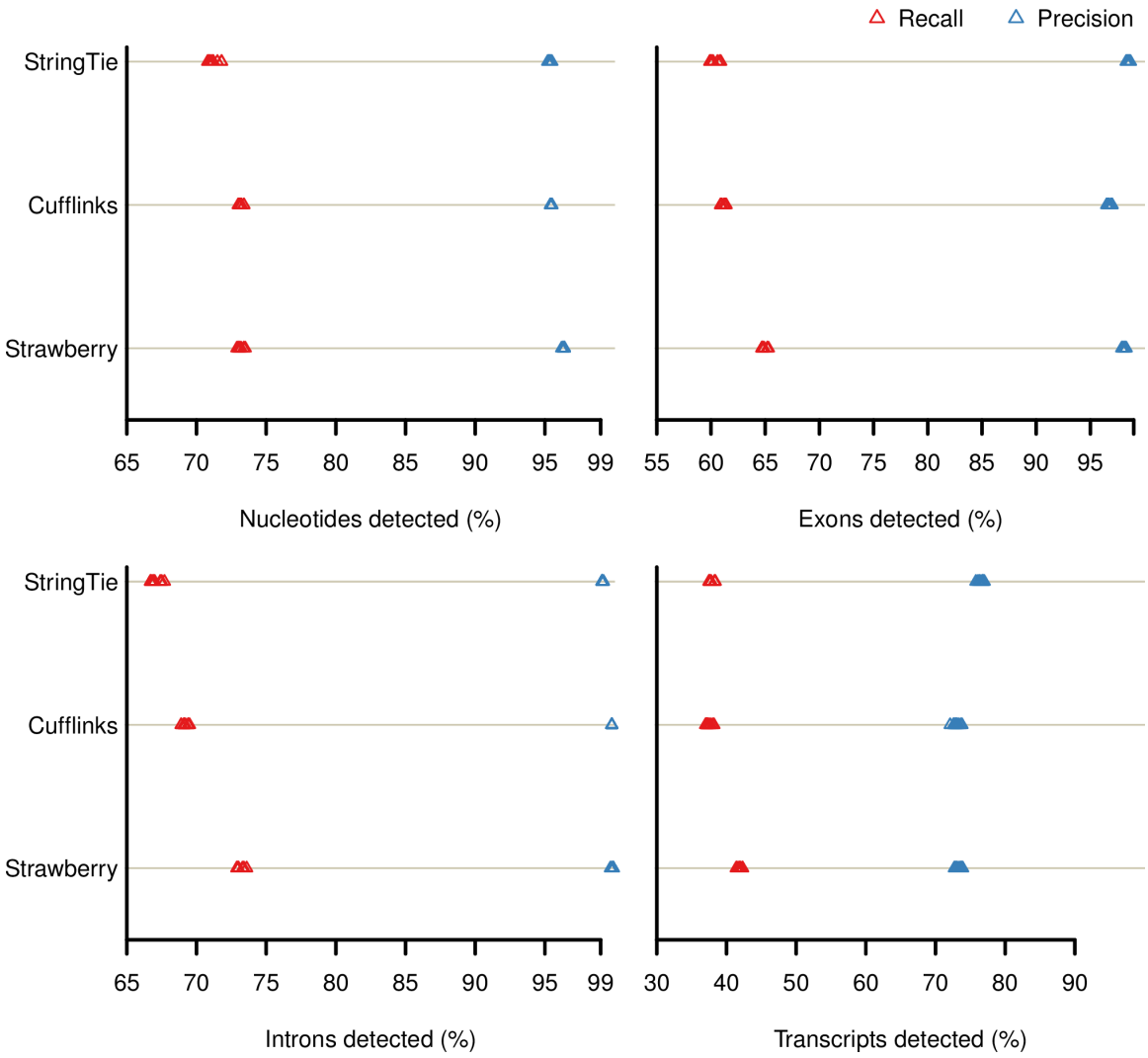
[38] Lahens, N. F., Kavakli, I. H., Zhang, R., Hayer, K., Black, M. B., Dueck, H., Pizarro, A., Kim, J., Irizarry, R., Thomas, R. S., Grant, G. R., and Hogenesch, J. B. (2014) IVT-seq reveals extreme bias in RNA sequencing. *Genome Biol.,* **15**(6), R86.

[39] Burset, M., Guig, R. (1996) Evaluation of gene structure prediction programs. *Genomics,* **34**(3), pp. 353-367.

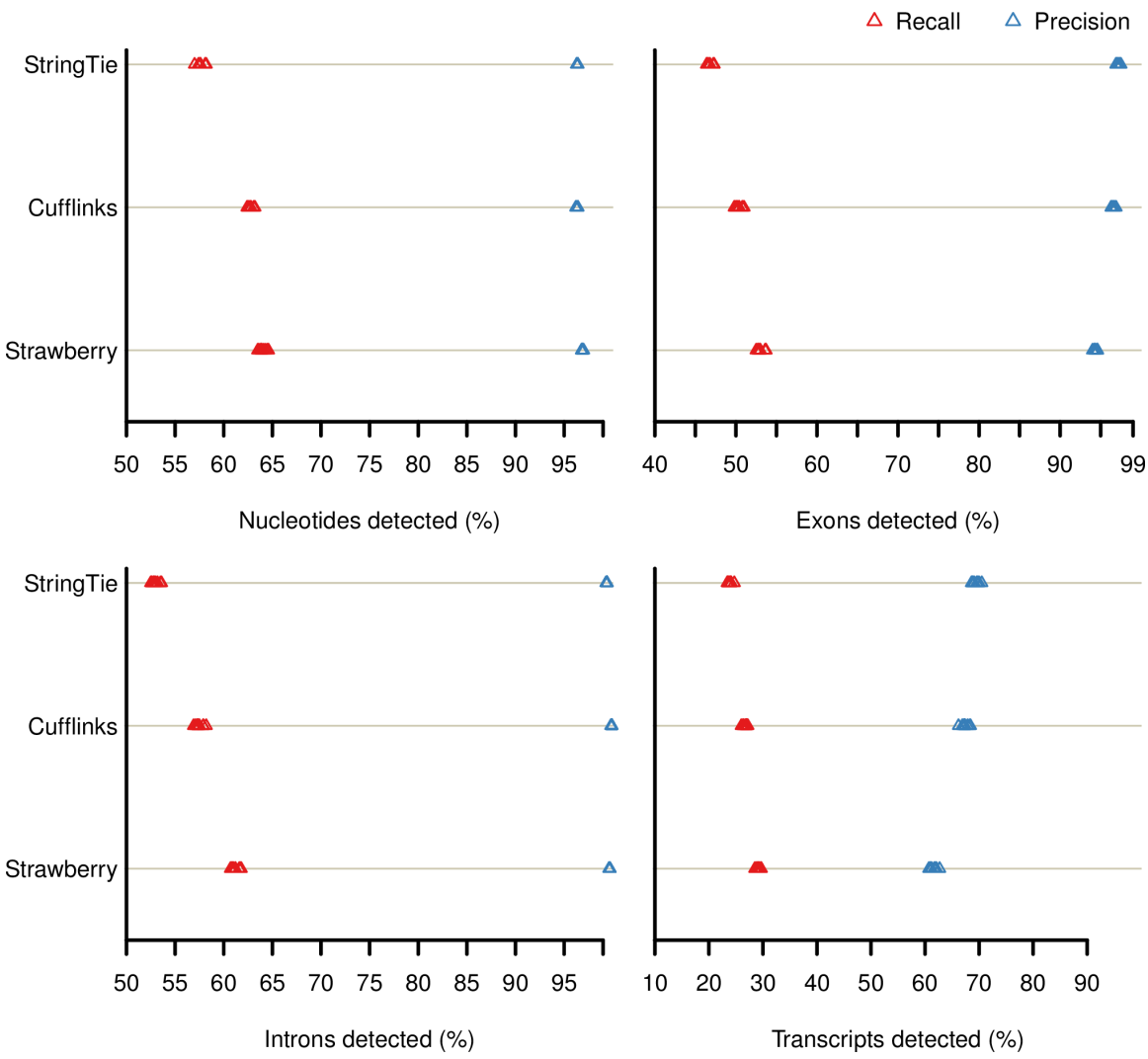
[40] Frazee, A. C. and Jaffe, A. E. and Langmead, B. and Leek, J. T. (2015) Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics,* **31**(17), 2778–2784.

[41] Lappalainen, T., Sammeth, M., Friedlander, M. R., 't Hoen, P. A., Monlong, J., Rivas, M. A., Gonzalez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P. G., Barann, M., Wieland, T., Greger, L., van Iterson, M., Almlof, J., Ribeca, P., Pulyakhina, I., Esser, D., Giger, T., Tikhonov, A., Sultan, M., Bertier, G., MacArthur, D. G., Lek, M., Lizano, E., Buermans, H. P., Padioleau, I., Schwarzmayr, T., Karlberg, O., Ongen, H., Kilpinen, H., Beltran, S., Gut, M., Kahlem, K., Amstislavskiy, V., Stegle, O., Pirinen, M., Montgomery, S. B., Donnelly, P., McCarthy, M. I., Flicek, P., Strom, T. M., Lehrach, H., Schreiber, S., Sudbrak, R., Carracedo, A., Antonarakis, S. E., Hasler, R., Syvanen, A. C., van Ommen, G. J., Brazma, A., Meitinger, T., Rosenstiel, P., Guigo, R., Gut, I. G., Estivill, X., Dermitzakis, E. T., Estivill, X., Guigo, R., Dermitzakis, E., Antonarakis, S., Meitinger, T., Strom, T. M., Palotie, A., Deleuze, J. F., Sudbrak, R., Lerach, H., Gut, I., Syvanen, A. C., Gyllensten, U., Schreiber, S., Rosenstiel, P., Brunner, H., Veltman, J., Hoen, P. A., van Ommen, G. J., Carracedo, A., Brazma, A., Flicek, P., Cambon-Thomsen, A., Mangion, J., Bentley, D., and Hamosh, A. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511.

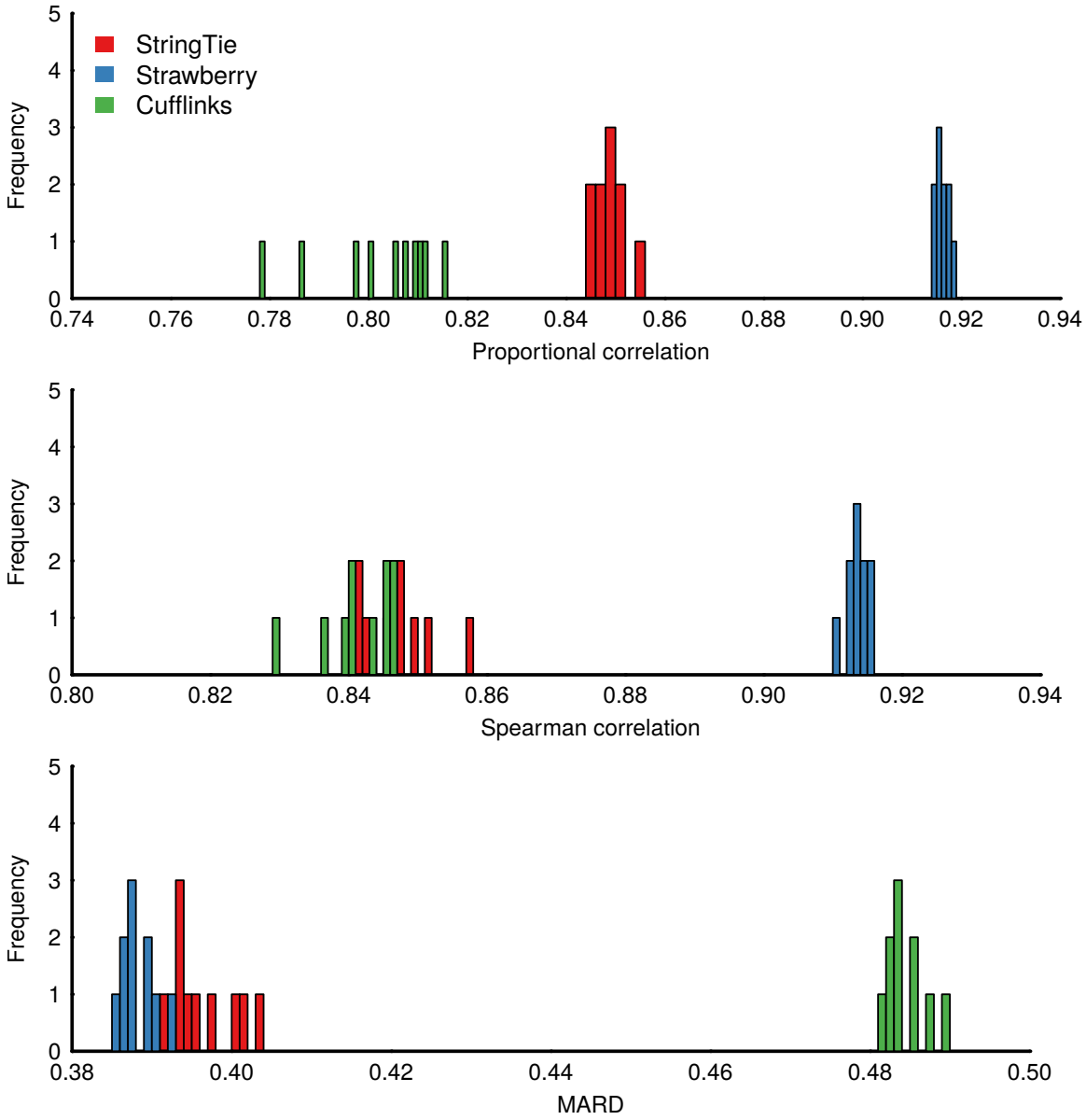
Supporting information



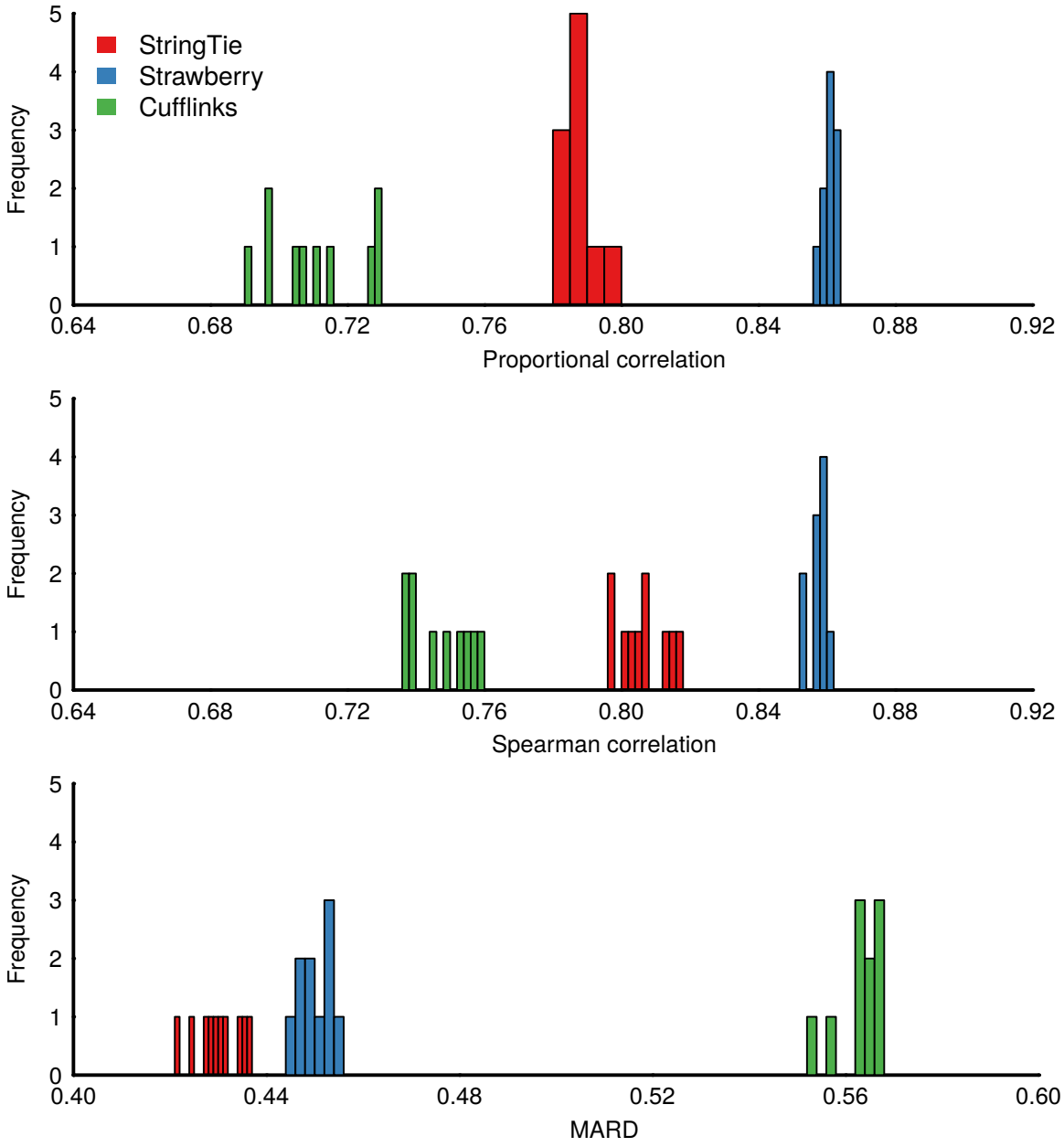
S1 Fig RD60 assembly result. recall and precision at the nucleotide, exon, intron and transcript level for StringTie, Cufflinks and Strawberry at RD60 data.



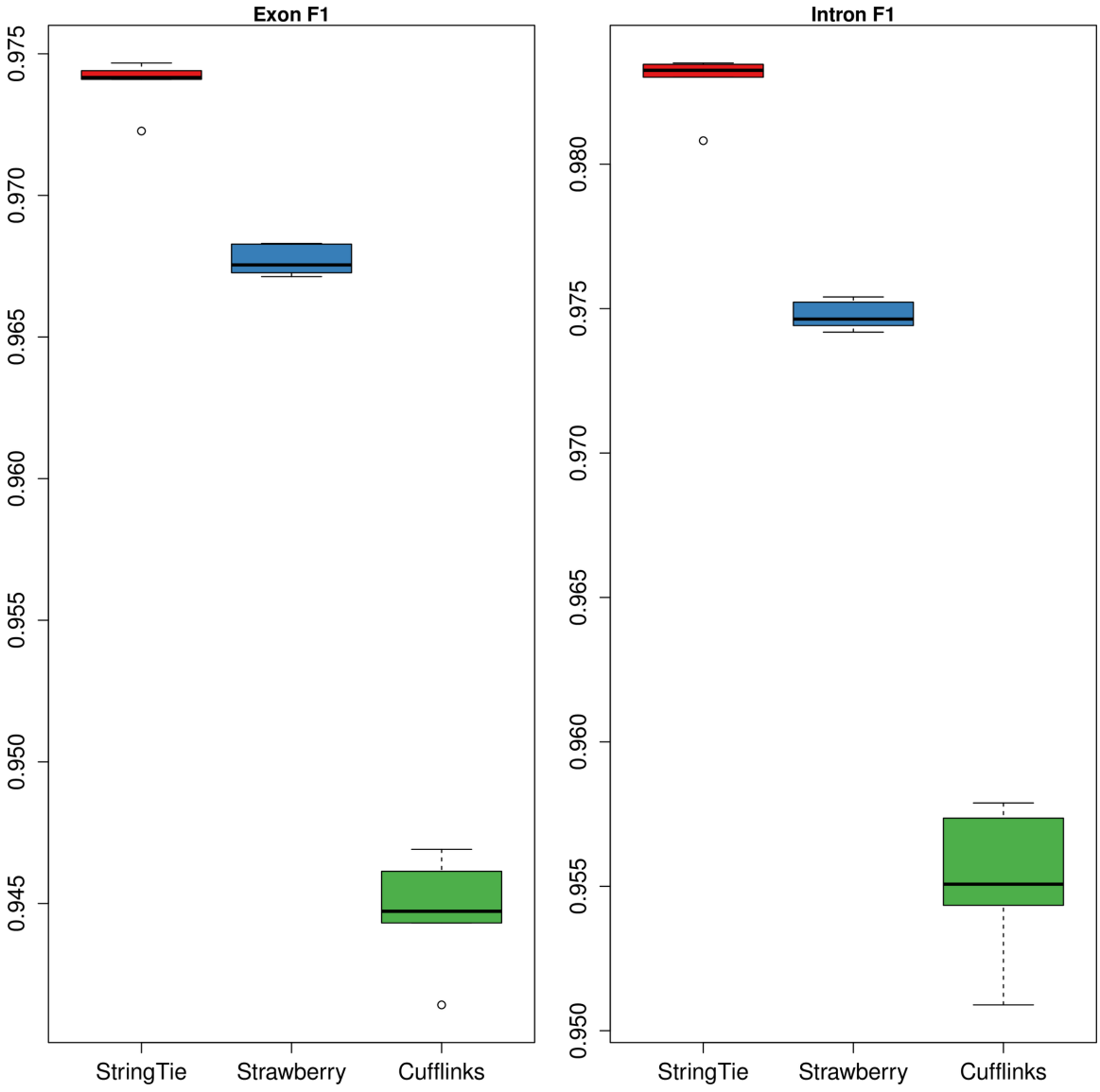
S2 Fig RD25 assembly result. recall and precision at the nucleotide, exon, intron and transcript level for StringTie, Cufflinks and Strawberry at RD25 data.



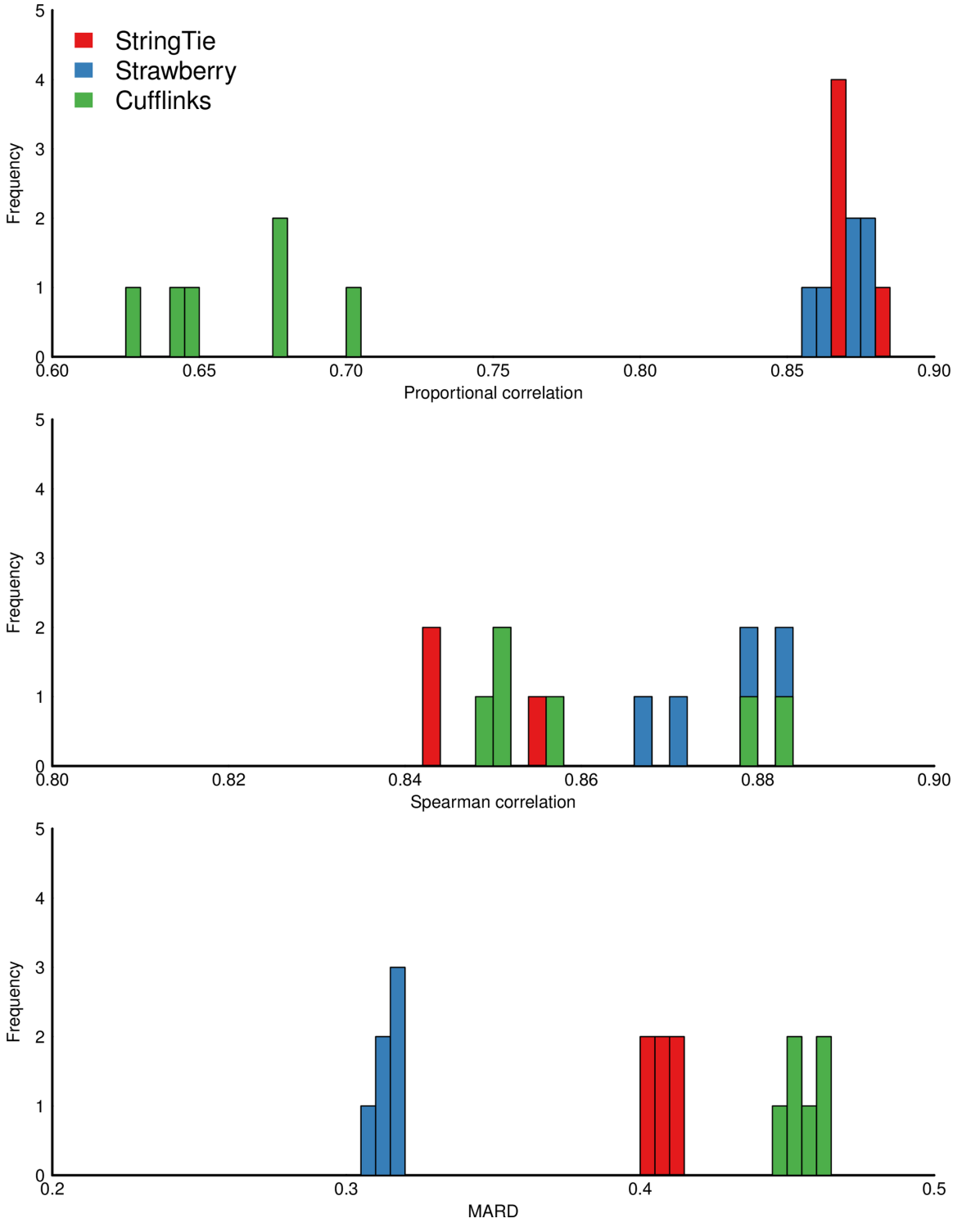
S3 Fig RD60 quantification result. Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 10 replicates in RD60 data.



S4 Fig RD25 quantification result. Frequency plot of Proportional correlation, Spearman correlation, Mean Absolute Relative Difference (MARD) for the 10 replicates in RD25 data.



S5 Fig Box plots of F1 scores at the exon and intron level. StringTie, Cufflinks and Strawberry were evaluated on data *GEU*, which is a simulated Human RNA-Seq data set.



S6 Fig Frequency plot of Proportional correlation, Spearman correlation,

Mean Absolute Relative Difference (MARD) for the 6 samples in *GEU*, which is a simulated Human data. These comparisons include only the reconstructed transcripts that fully match the known transcripts.

S1 Data. GEU simulation data. The GEU simulation pipeline and a step by step tutorial about how to generate the simulation and conduct the evaluation can be found at https://github.com/ruolin/strawberry_comp.

Acknowledgments

The material presented here is based upon work supported by the National Science Foundation under Grant IOS-1062546. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

CHAPTER 4. RSTRAWBERRY: DIFFERENTIAL ALTERNATIVE SPLICING from MULTIPLE SAMPLES

4.1 Introduction

A change in relative isoform expression can be measured when comparing samples from one condition to another. A significant change implies differential alternative splicing (DAS). Currently, two major paradigms exist for recognizing relative isoform expression from RNA-Seq samples. Transcript-centric methods seek to reconstruct isoform expression and then compare the relative expression across conditions. However, estimating isoform expression from short reads is not an easy problem and biased estimation can disrupt differential analysis. Thus, event-centric methods focus on comparing the direct evidence of read supports for AS (Alternative Splicing) events. For short read technologies, transcript-centric models need to deal with the extra uncertainty of phasing short read into overlapping isoforms. Although event-centric methods can avoid the "phasing" problem by localization, they do not address the fundamental question that which isoforms result in the expression pattern changes. This is because a single AS event can be shared by multiple isoforms, e.g., a cassette exon is spliced in two isoforms but spliced out in the other two isoforms.

We proposed rStrawberry, a transcript-centric method for differential alternative splicing detection. By extending the single sample quantification model introduced in (1), rStrawberry utilizes a multinomial logit regression which regresses transcript relative abundances on a set of covariates to explain replicate and condition effects. Also, the quantification model in (1) is unable to deal with two major issues that often adversely affect the differential splicing analysis, which are coverage bias and count variability.

Firstly, compared to DNA Sequencing, RNA-Seq is subject to a higher degree of coverage bias (5). It is well known that the distribution of read coverage along transcripts is generally

nonuniform. (3; 5; 8) have shown that bias is caused by local sequences (e.g., hexamer bias) around the read starts, positions of the reads, GC content bias, etc. (2) shows that the random hexamer priming is not random and creates read start bias on the Illumina platform regardless of organism and laboratory. Also, (5) shows that some RNA-specific library preparation step, e.g., reverse transcription PCR, introduce GC-bias. And (9) have confirmed there is a coverage bias caused either by the sequence of the underlying fragment, the RNA-Seq experiment itself or the interaction of these two. These biases bring an extra level of variations in addition to biological variations. To overcome the coverage bias, rStrawberry uses a dual-phase algorithm. During the bias correction phase, another multinomial logit regression model is trained on single-isoform loci across all samples to discover relationships between subexon path probability and the underlying local sequence information such as GC-content, existence of high GC-stretches and hexamer context. Then during the second phase, the differential splicing analysis algorithm uses the fitted path probabilities instead of the observed path probabilities.

Another challenge is due to the "count" nature of RNA-Seq data and the variability in count measurement across biological replicates. Unlike microarrays which measure the fluorescent intensity, researchers have to deal with the read counts when it comes to RNA-Seq. Because of the discrete count nature, we have rather limited model choices. To make matters worse, RNA-Seq counts are usually over-dispersed (see Appendix A.2). As a result of overdispersion, models such as Poisson, binomial/multinomial alone are not suitable for RNA-Seq. More generalized distributions, e.g., negative-binomial are more suitable for this kind of overdispersed count data. To account for the count overdispersion, rStrawberry employs an empirical Bayesian model which places shrinkage priors on the multinomial logit regression coefficients. These priors are analogous to the gene-specific negative-binomial dispersion parameter and are estimated by borrowing information across all loci.

To my knowledge, rStrawberry is the first to simultaneously estimate transcript abundances and identify differential alternative splicing at the transcript level. Existing methods

either measure at the level of splicing events (for example inclusion or exclusion of a particular cassette exon), e.g., DEXSeq (11), or detect at the level of genes, e.g., Cuffdiff 2 (13). rStrawberry, instead, calculates the significance of which transcripts are differentially spliced.

4.2 Differential alternative splicing detection model

4.2.1 Definitions and notations

rStrawberry deals with RNA-Seq experiments that compare two experimental conditions. For each condition, replicate RNA-Seq libraries are generated and sequenced. These reads are processed using specialized bioinformatics tools to align to a reference genome. The set of genes and transcripts, i.e., exon-intron structures, are given a priori for all samples. Given a set of locus Λ (which may contain only one locus (the place of a gene) or all loci in a species), let $g \in \{1, \ldots, G\}$ denote genes and $k \in \{1, \ldots, K\}$ index transcripts from Λ with $K \geq G$. In Illumina technology, reads can appear in pairs, with one read generated from each end of a sequenced template. Thus, a read-pair, denoted by r, refers to aligned paired-end reads with sequences observed at both ends and an unknown sequence in between. On the other hand, a read refers to either the upstream or downstream observed sequence of a read-pair. For single-end reads, replace the terminology "read-pair" with "read" and proceed. The word fragment is used to describe an actual DNA fragment that is being sequenced. There exists a surjective map from the fragments $\mathcal{F}$ to read-pairs $\mathcal{R}$. A read-pair may imply different fragments under different transcripts.

In addition, a transcript is a sequence of alternating exons and introns. A set of transcripts at a locus can be represented as a splice graph consisting of exons (or subexons) as vertices and edges connecting exons across introns. A subexon is a maximal portion of an exon that appears intact in all transcripts. An exon is split into two subexons, for example, if it contains an alternative splice site that is used in some transcripts. Edges have length zero when traversing subexons within an exon. A read-pair can be represented

as a unique set of ordered nodes, called a subexon path. Notice, if the underlying sequence of a read-pair contains an unsequenced portion, such as the insert, the subexon path of the read-pair is an incomplete observation of the unobserved set of subexons from which it is generated. However, when conditioning on the transcript, a subexon path becomes a complete observation. Also, a subexon path can be included or excluded from an transcript and thereby, I derive a sparse binary matrix C with L rows and K columns, where L is the total number of subexon paths. $C_{lk} = 1$ if transcript k contains subexon path l, otherwise 0. For each column, $L_k = \{l : C_{lk} = 1\}$ is a set of subexon path l that are included in the transcript k.

Let $j \in \{1, \ldots, J\}$ index samples, both conditions and all replicates, where $J = J_1 + J_2$ and J_1 and J_2 are the numbers of replicates for condition 1 and 2. I follow the standard that all the replicates from condition 1 have smaller indexes than all the replicates from condition 2. For sample j, let the number of mapped read-pairs for the set of locus Λ be n_j, and r_{ji} be the ith read-pair from the jth sample.

4.2.2 Likelihood function and priors

Let $\{r_{ji}\}$ denote the observations which are uniquely mapped fragments to the genome. The response variables are represented as a row-wise matrix $\boldsymbol{Y} = \{\boldsymbol{y}_{ji}\}$, where each row $\boldsymbol{y}_{ji}$ identifies the subexon path of the read-pair r_{ji}, i.e., $\boldsymbol{y}_{ji}$ is an L-dimensional vector, one of the standard basis vectors of L-dimensional Euclidean space. Specifically, if the observed subexon path is l, then $y_{jil} = 1$ and $y_{jil'} = 0$ for all $l' \neq l$. The predictor variable, $\boldsymbol{X} = \{\boldsymbol{x}_{ji}\}$, is also a row-wise matrix with the same number of rows as $\boldsymbol{Y}$. Each element $\boldsymbol{x}_{ji}$ indicates the experimental condition and replicate of read-pair r_{ji}. In addition, I encode $\boldsymbol{W} = \{\boldsymbol{w}_{kl}\}$, where $\boldsymbol{w}_{kl}$ are coverage bias covariates representing relevant sequence signals, such as GC content or existence of high GC stretches, that might affect the fragment selection during the RNA-seq experiment (8). To be specific, covariates $\boldsymbol{W}$ include a basis function of natural cubic spline for GC-content (knots at 0.4, 0.5, 0.6, and boundary knots at 0.3

and 0.7), a basis function of natural cubic spline for hexamer entropy (knots at 4,5,6 and boundary knots at 3 and 7) and 4 indicator variables for high-GC stretches, respectively, 80 GC-content or higher in a 20 nt stretch, 90 GC-content or higher in a 20 nt stretch, 80 GC-content or higher in a 40 nt stretch and lastly 90 GC-content or higher in a 40 nt stretch. These signals are known quantities given the transcript k and subexon path l. In addition, there is a unique mapping from r_{ji} to $\boldsymbol{w}_{kl}$ once the read-pair has been assigned to a transcript k with $C_{lk} = 1$. This formulation assumes all read-pairs from the same transcript and the same subexon path share the same coverage covariates.

Our generative model assumes transcript k with length τ_k makes up proportion η_k in the sample. Transcripts are randomly fragmented, and long transcripts produce more fragments than short transcripts. Transcript k fragments constitute approximately a proportion of $\pi_k \propto \tau_k \eta_k$ in the sample. Having estimated $\hat{\pi}_k$ and knowing τ_k, we can later derive η_k. Each read-pair is considered as generated from one, possibly unobserved, transcript (latent class). Given the latent class, observation $\boldsymbol{y}_{ji}$ is the realization of a one-trial multinomial experiment Mult$(1, \boldsymbol{\theta}_{k\cdot})$ where θ_{kl} is the probability of generating a fragment from subexon path l conditioned on transcript k. Note that θ_{kl} does not depend on sample j and read i, and is used to model transcript coverage effects, including possible coverage bias. Since the coverage biases are usually caused by library preparation steps and sequencing steps, we assume coverage biases are independent of replicates and experimental conditions. My model assumes that the entire effect of experimental conditions on the transcript expression is through changing the relative abundances of transcripts; reflected in the model, it is $\boldsymbol{\pi}_k$. Conditional on the transcript, all the samples share the same coverage biases encoded in θ_{kl}. In particular, the goal is to detect differential transcript expression, *i.e.*, differences in π_k across experimental conditions and θ_{kl} is a nuisance parameter.

To link $\boldsymbol{\pi}$ with the experimental conditions, a generalized linear (multinomial) logit link

function was used:

$$\pi_k(\boldsymbol{x}_{ji}) = \frac{\exp\left(\boldsymbol{x}'_{ji}\boldsymbol{\beta}_k\right)}{\sum_{t=1}^{K-1}\exp\left(\boldsymbol{x}'_{ji}\boldsymbol{\beta}_t\right)+1}, \text{ for } k \neq K. \tag{4.1}$$

Here, $\boldsymbol{\beta}_k$ is a J dimensional parameter vector, including β_{k0} the baseline level of transcript k, β_{k1} the effect of condition on transcript k, and $\beta_{k2}, \ldots, \beta_{kJ}$ the effect of replicates on transcript k, which represent isoform level dispersion parameters. In practice, the covariate $\boldsymbol{x}_{ji}$ includes a constant term 1, one dummy variable for the experimental condition, and $J-2$ dummy variables for replicates and only one of the $J-2$ dummy variables is 1 and the rest are 0. In addition, to ease the notation, I write $\boldsymbol{x}_j$ instead of $\boldsymbol{x}_{ji}$ because the read from the same sample have exactly same covariates, i.e., $\boldsymbol{x}_{ji} = \boldsymbol{x}_j, \forall i$.

The effect of GC-bias on coverage is believed to be non-linear (8). Multinomial logit regression is known for modeling non-linear relationships. Therefore, we use another multinomial logit regression for $\boldsymbol{\theta}$. Since our model assumes a constant bias effect across samples and conditions, a nontraditional formulation is needed. In our formulation, the coefficients do not vary across outcome categories. Instead, the same coefficient vector $\boldsymbol{\alpha}$ operates on different predictors $\boldsymbol{w}_{kl}$. Specifically, given a fragment from transcript k, the subexon path l is generated with probability

$$\theta_{kl}(\boldsymbol{w}_{kl}) = \begin{cases} \frac{\exp(\boldsymbol{w}'_{kl}\boldsymbol{\alpha})}{\sum_t \exp(\boldsymbol{w}'_{kt}\boldsymbol{\alpha})} & C_{lk} = 1, \\ 0 & C_{lk} = 0. \end{cases} \tag{4.2}$$

Putting the components together, the latent class regression model likelihood can be written as:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \{\boldsymbol{y}_{ji}, \boldsymbol{x}_j, \boldsymbol{w}_{kl}, \Lambda\}) = \prod_{j=1}^{J}\prod_{i=1}^{n_j}\sum_{k=1}^{K}\left[\pi_k(\boldsymbol{x}_j)\prod_{l=1}^{L}\theta_{kl}(\boldsymbol{w}_{kl})^{y_{jil}}\right]. \tag{4.3}$$

It simplifies as

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \{n_{jl}, \boldsymbol{x}_j, \boldsymbol{w}_{kl}\}, \Lambda) \\
&\qquad = \prod_{j=1}^{J} \prod_{i=1}^{n_j} \sum_{k=1}^{K} \prod_{l=1}^{L} [\pi_k(\boldsymbol{x}_j)\theta_{kl}(\boldsymbol{w}_{kl})]^{y_{jil}} \\
&\qquad = \prod_{j=1}^{J} \prod_{i=1}^{n_j} \prod_{l=1}^{L} \left[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}_j)\theta_{kl}(\boldsymbol{w}_{kl})\right]^{y_{jil}} \\
&\qquad = \prod_{j=1}^{J} \prod_{l=1}^{L} \left[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}_j)\theta_{kl}(\boldsymbol{w}_{kl})\right]^{n_{jl}},
\end{aligned}$$

where n_{jl} is the number of read-pairs from sample j with subexon path l. The second equation is valid because $y_{jkl} \in \{0, 1\}$ and $\sum_{l=1}^{L} y_{jil} = 1$.

In the Eq. 4.1, $\boldsymbol{\beta}_k$ contains parameters accounting for transcript-level overdispersion across samples. However, the per-transcript coefficient estimation is highly unstable. Also, to borrow information across genes, we use a hierarchical model and assume $\beta_{kj} \mid \gamma_j \overset{\text{iid}}{\sim} N(0, \gamma_j)$ such that γ_j (precision parameter) is analogous to the gene-specific negative-binomial dispersion parameter. Next, for model simplicity, a conjugate prior is placed on γ_j so that $\gamma_j \overset{\text{iid}}{\sim} \text{Gamma}(a_0, b_0)$. The hyperparameters a_0 and b_0 are estimated in the following way. The likelihood function 4.3 is first fitted for all loci, but 5 loci at a time. Then γ_j is estimated using a method of moments method based on the coefficients estimates $\hat{\boldsymbol{\beta}}_k$. Finally, a maximum likelihood estimation is obtained only through the 75% percent quantile (0.125-0.875) of all γ_j, to avoid the effect of outliers.

4.2.3 Bias correction

rStrawberry learns the bias coefficient $\boldsymbol{\alpha}$ in Eq. 4.2 from a subset of the input data where there is no ambiguity in assigning read-pairs to transcripts. Then $\boldsymbol{\alpha}$ is held constant when actually fitting the Hierarchical Bayesian model. Assume now drop the notation for transcript k and let l index all subexon paths in all single isoform genes, and y_l the read-pair count for path l. Note that $\boldsymbol{w}_l$ is the predictor vector of path l and $\boldsymbol{\alpha}$ is the universal coefficient vector. To calculate the subexon path probability $\boldsymbol{\alpha}$, an equivalent

Poisson regression model is used:

$$Y_l \overset{ind.}{\sim} \text{Pos}(\exp(\boldsymbol{w}_l'\boldsymbol{\alpha})).$$

This Poisson regression is fitted across all single-isoform genes to borrow information across genes to make the estimation more robust. To convert this Poisson regression fit to a multinomial probability vector $\boldsymbol{\theta}$, the following formula is applied:

$$\hat{\theta}_l = \frac{\exp(\hat{\boldsymbol{\alpha}}' w_l)}{\sum_j \exp(\hat{\boldsymbol{\alpha}}' w_j)}.$$

The bias model proposed here is different from (8) in that they calculates bias predictors from all fragments while rStrawberry reduces the fragment data to subexon paths which contain the underlying sequences of the fragments. A typical RNA-Seq sample often contains tens of millions reads and calculating bias predictors for all fragments is very time-consuming. Therefore, in both theory and practice, rStrawberry's model is orders of magnitude faster than (8).

4.2.4 Model estimation

Let $z_{ji} \in \{1, \ldots, K\}$ be the unobserved transcript source of the $(j,i)th$ read-pair from the set of locus Λ. The pair $(\boldsymbol{y}_{ji}, z_{ji})$ can be replaced by a new $L \times K$-dimension complete data matrix $\boldsymbol{h}_{ji}$ indicating the transcript and exon path. Note, the row sums of $\boldsymbol{h}_{ji}$ equal vector $\boldsymbol{y}_{ji}$. Let $n_{jkl} = \sum_{i=1}^{n_j} h_{jilk}$ be the hidden aggregate count of read-pairs from sample j that are assigned with transcript k and subexon path l. Note that we often observe n_{jl} but not n_{jkl} since the read-pairs are short can mapped to overlapping transcripts. The unnormalized posterior of all parameter and hidden variables is:

$$P(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{H} \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\theta}, a_0, b_0, \Lambda) \propto \prod_{j=1}^{J} \prod_{l=1}^{L} \prod_{k=1}^{K} [\pi_k(\boldsymbol{x}_j, \boldsymbol{\beta})\theta_{kl}]^{n_{jkl}} \prod_{k=1}^{K} \prod_{s=1}^{J} [\text{N}(\beta_{ks} \mid 0, \gamma_s)] \prod_{s=1}^{J} \text{Ga}(\gamma_s \mid a_0, b_0) \tag{4.4}$$

where $\mathrm{N}(x \mid \mu, \gamma)$ is the density of normal distribution with mean μ and precision γ and $\mathrm{Ga}(x \mid \alpha, \beta)$ is the density of gamma distribution with shape α and rate β. The selection of the normal distribution and gamma distribution makes the model estimation simpler so that we can avoid computationally expensive sampling procedure such as MCMC. The graph model representation of this posterior model is in Fig. 4.1.

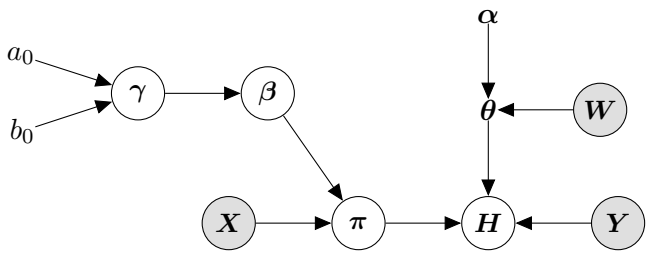


Figure 4.1 Graphical model representation of rStrawberry alternative splicing detection model, where $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, a_0, b_0 are fixed parameters and $\boldsymbol{W}$, $\boldsymbol{Y}$, bX are observed variables. Also, $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ are transformed parameters and thus their functions are deterministic. Here, $\boldsymbol{\theta}$ is the bias parameter. The inference is focused on $\boldsymbol{\beta}$.

The hyperprior a_0, b_0 and bias parameter $\boldsymbol{\theta}$ are estimated using the techniques described in section 4.2.2 and 4.2.3 before fitting the model 4.4. Instead of sampling from this posterior, I derive an EM-like iterative algorithm to obtain point estimates that maximize the unnormalized posterior density. During an iteration, the unnormalized posterior is maximized over $\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{H}$, one at a time while holding the other parameters constant. The posterior of n_{jkl} is a multinomial distribution and the MAP estimator is:

$$\mathbb{E}(n_{jlk}) = n_{jl} \frac{\hat{\pi}_{jk}\hat{\theta}_{kl}}{\sum_{v=1}^{K} \hat{\pi}_{jv}\hat{\theta}_{vl}}.$$

Secondly, for γ_s, $s \in \{1, \ldots, J\}$, we have

$$\begin{aligned} P(\gamma_s \mid \hat{\boldsymbol{\beta}}, \hat{a}_0, \hat{b}_0, \Lambda) \propto & \prod_{k=1}^{K} [\gamma_s^{1/2} e^{-\frac{\hat{\beta}_{ks}^2}{2}\gamma_s}] \gamma_s^{a_0 - 1} e^{-b_0 \gamma_s} \\ = & \gamma_s^{a_0 + \frac{J}{2} - 1} e^{-(b_0 + \frac{\sum_{k=1}^{K} \hat{\beta}_{ks}^2}{2})\gamma_s}. \end{aligned}$$

This is the kernel of a gamma distribution and the point estimate of γ_s as the posterior mode is:

$$\hat{\gamma}_s = \frac{2a_0 + J - 1}{2b_0 + \sum_{k=1}^{K} \hat{\beta}_{ks}^2}.$$

Finally for $\boldsymbol{\beta}$, we have:

$$P(\boldsymbol{\beta} \mid \boldsymbol{X}, \hat{\boldsymbol{n}}, \hat{\boldsymbol{\gamma}}, \Lambda) \propto \prod_{j=1}^{J} \prod_{k=1}^{K} \left[\left(\frac{e^{\boldsymbol{x}_j' \boldsymbol{\beta}_k}}{\sum_{t=1}^{K} e^{\boldsymbol{x}_j' \boldsymbol{\beta}_t}} \right)^{\hat{n}_{jk}} \right] e^{-\sum_{s=1}^{J} \frac{\hat{\gamma}_s}{2} \left(\sum_{k=1}^{K} \beta_{ks}^2 \right)},$$

where $\hat{n}_{jk} = \sum_{l=1}^{L} C_{lk} \hat{n}_{jkl}$. The object function for optimization is:

$$\log \mathcal{L}(\boldsymbol{\beta}) := \sum_{j=1}^{J} \sum_{k=1}^{K} n_{jk} \log(\pi_{jk}) - \sum_{s=1}^{J} \frac{\hat{\gamma}_s}{2} \left(\sum_{k=1}^{K} \beta_{ks}^2 \right),$$

where π_{jk} is a short form of $\pi_k(\boldsymbol{x}_j)$. Note that the last term acts as a L2 regularization term and γ control the degree of penalization. Let $\boldsymbol{I}$ be the identity matrix and I_{kp} are the elements of the identity matrix. To optimize this function, algorithms such as Newton-Raphson can be used. To derived the gradient and hessian, I utilized the fact that $\nabla_{\boldsymbol{\beta}_p} \pi_{jk} = \pi_{jk} \left(I_{kp} - \pi_{jp} \right) \boldsymbol{x}_j$. Also, I write $\sum_{p=1}^{K} n_{jp} = n_j$. Thus, we have

$$\nabla_{\boldsymbol{\beta}_p} \log \mathcal{L}(\boldsymbol{\beta}) = \sum_{j=1}^{J} (n_{jp} - n_j \pi_{jp}) \boldsymbol{x}_j - \hat{\boldsymbol{\gamma}} \odot \boldsymbol{\beta}_p$$

$$\nabla_{\boldsymbol{\beta}_q} \nabla_{\boldsymbol{\beta}_p} \log \mathcal{L}(\boldsymbol{\beta}) = \sum_{j=1}^{J} n_j \pi_{jp} (\pi_{jq} - I_{pq}) \boldsymbol{x}_j \boldsymbol{x}_j' - I_{pq} \cdot \mathbf{Diag}(\hat{\boldsymbol{\gamma}}),$$

where $\odot$ is the element wise vector product and $\mathbf{Diag}(\hat{\boldsymbol{\gamma}})$ is a diagonal matrix with $\hat{\boldsymbol{\gamma}}$ on the diagonal. Finally, the observed covariance matrix of $\boldsymbol{\beta}$ at the estimates equal to the negative diagonal of the inverse of the Hessian matrix. And the p values for $\boldsymbol{\beta}$ is calculated as the z scores, which are the point estimates divided by the observed standard deviations, in a standard normal distribution.

4.2.5 Implementation details

The implementation of model 4.4 and its inference is fully available in a Github project called rStrawberry (https://github.com/ruolin/rstrawberry). This particular implementation makes use of the single sample quantification function of Strawberry (1). To be

specific, each RNA-Seq sample is processed by Strawberry (https://github.com/ruolin/strawberry) with *–no-assembly* and *-f* options. *–no-assembly* means that the assembly module is disabled and Strawberry is using the input gene models, which could be the known annotation or Strawberry's assembly result. In addition, *-f* option is needed to output the subexon path count table which summarizes all the input information that is needed for the differential alternative splicing detection model. This information includes the subexon path count $\boldsymbol{Y}$, the number of isoforms for each gene K based on annotation, the compatibility matrix C, the path GC content, hexamer entropy, indicator of high GC stretches, i.e., the $\boldsymbol{W}$ covariate matrix. The *C++* implementation ensures a fast turnaround time (usually in minutes) for a typical RNA-Seq samples (100 million reads) which can take hours if otherwise implemented in *R*.

The default behavior of rStrawberry is to perform the differential splicing inference per locus. Although rStrawberry can group an arbitrary number of genes to a "super group", fitting on more than 20 loci together can lead to a large Hessian matrix which is expensive to compute. When a transcript has low expression, the read assignment uncertainty is usually large and can lead to excessive false positives. Therefore, rStrawberry has a default expression filter that will filter out transcripts with FPKM < 1. However, this threshold can be changed by the user and as long as one of the isoforms of a gene passes the expression filter, all isoforms from that gene will be kept.

For optimization of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in the posterior model 4.4, I choose to use a well-developed and highly optimized statistical computational code base, called STAN (https://github.com/stan-dev/stan). It is worth mentioning that STAN has its own model language and the representation of STAN language of my model is in Appendix A.1

4.3 Result

4.3.1 Correcting RNA-Seq Coverage Bias

GEUVADIS (6) is a high-quality RNA-Seq consortium for the 1000 Genomes project. GEUVADIS contains more than 600 RNA-Seq samples sequenced at 8 different European labs, which allows us to study RNA-Seq technical bias, biological variations, etc. Fig. 4.2 is a Sashimi plot on gene USF2 from two RNA-Seq samples (ERR188021 and ERR188114) from the GEUVADIS dataset. ERR188021 and ERR188114 are people from the same ethnic group who were sequenced at different labs, *UNIGE* and *CNAG-CRG* respectively. Following (8), *CNAG-CRG* is called center 1 and UNIGE is called center 2. Based on the RefSeq hg19 gene model, USF2 contains two isoforms and ten unique exons. The two isoforms differ by an exon skipping event at chr19:35760706-35760906, which is the second exon in Fig. 4.2. This cassette exon happens to be a high-GC exon (GC-content 73). The consecutive exon before the skipping exon is also a high-GC exon (GC-content 66).

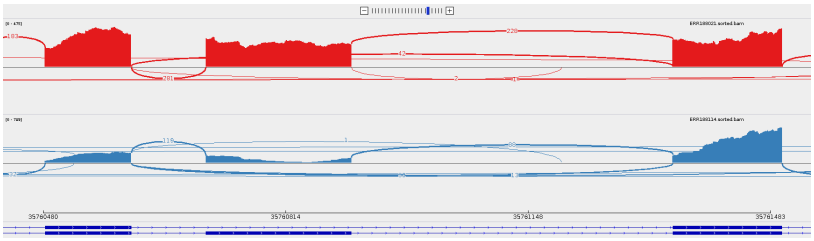


Figure 4.2 GC Bias in GEUVADIS. Sashimi plot of ERR188021 and ERR188114 on USF2 gene. Sashimi plots quantitatively visualize splice junctions for multiple samples from RNA-Seq alignments. This plot is produced by IGV (https://software.broadinstitute.org/software/igv/Sashimi). The bottom track is the genomic coordinates and the USF2 gene annotation (only 3 exons). The middle track (shown in light blue) is the junction alignments of ERR188114 (from center 1) and the top track (shown in red) is for ERR188021 (from center 2). This plot shows that samples from center 1 suffer more coverage drops for high GC exons.

The PSI (Percent Splice In, percentage of junction reads supporting the cassette exon being spliced in as opposed to being spliced out) for ERR188021 and ERR188114 on the

USF2 gene are 0.858 and 0.941 respectively, which clearly indicates the spliced-in, long isoform NM003367.2 expressed more compared to the short isoform NM207291.1 on both samples. Without bias correction, however, Strawberry's result shows that the short isoform is more abundant than the long isoform on ERR188114, Fig. 4.3. A similar result has also been observed for other software (8). The reason for this type of estimation bias in ERR188114 is that there are significant coverage drops on high-GC exons including the cassette exon. For reads that align to both isoforms, the generative model used in (1; 12) assumes a higher probability that these reads come from the short isoform. After bias correction, Fig. 4.3 clearly shows that rStrawberry has better expression estimation if some transcripts contain high GC exons, which can lead to significant quantification bias in Strawberry and Cufflinks.

To further show that rStrawberry's bias model is able to capture and even predict the variability of exon coverages, the model that is learned from the previous 6 samples is used to predict the exon coverage for sample ERR188297, which was not used in the training. Fig 4.4 shows the observed versus predicted subexon coverage density of a single isoform gene NUP107 after normalization. Gene NUP107 contains 120 subexon paths and the observed coverage of a transcribed position i is caculated as $\sum_{l \in T_i} y_l$ where T_i is the set of subexons paths that cover position i and y_l is the subexon path count as described in section 4.2. To normalize the data, I convert the observed count to the frequency such that the area under the curve is equal to 1. The plot shows that the bias model is able to capture, to some degree, the variability of read coverage on transcripts.

4.3.2 Controlling false discovery rate

As previously mentioned, GEUVADIS can be a good negative control dataset and indeed, (8) uses this dataset to claim that the transcript expression levels estimated by Cufflinks (12) and RSEM leads to a large number of false positives when their estimates are used for differentially expression analysis.

Three samples from center 1, ERR188204, ERR188317, ERR188453 and three samples from center 2, ERR188021, ERR188052, ERR188088 are were selected as negative control dataset. These samples are from the same ethnic group but were sequenced at two different sequencing centers. The raw reads were aligned against the GRCh37 human genome using HISAT2 (4). And I first ran Cufflinks v2.2.1 with "-G" option to execute annotation-based transcript quantification against GRCh37 RefSeq gene annotation. Similar but not restrict to isoform switching in (8), loci having more than one isoform and at least one isoform expressed in all samples ($>$ 0.1 FPKM) were selected in this study. A total 4468 genes, found by Cufflinks, fulfill this requirement. I then compared three methods, Cuffdiff, DEXSeq, and rStrawberry, using three samples from center 1 as group 1 and the other three samples from center 2 as group 2 to identify differential alternative splicing on these 4468 genes. These three methods report different sets of genes due to their built-in filters. In addition, DSGseq is excluded from this comparison since it does output significant values.

Cuffdiff v2.2.1 was executed on the six samples using default parameters. A total of 3114 genes passed through its filter and were reported by Cuffdiff 2. These 3114 tested genes intersected with the 4468 expressed genes to yield a final evaluation set which contains 2895 genes. Cuffdiff 2 produces a lot of "NOTEST" results. The reason why Cuffdiff 2 produces "NOTEST" is not clear and its website describes "NOTEST" as caused by not enough alignments for testing. Among the 2895 genes, only 17 of them (0.58%) are significant at 1% q value cutoff (table 4.1). This shows that Cuffdiff 2 is able to control the false discover rate (FDR) below the expected level.

For DEXSeq, two Python scripts (download together with DEXSeq Bioconductor R package) were first used to generate the exon count table where each row represents a unique exon segment and each column corresponds to a sample. The DEXSeq R package takes the count table and tests for differentially expressed (in DEXSeq paper they called differentially used) exons. Only the exons from the 4468 expressed multi-isoform genes were evaluated. Out of 56350 total tested exons, 2798 (5%) have multiple testing adjusted

p values less than 0.01 (table 4.1), which indicated that DEXSeq failed to control FDR at the expected value.

Finally, rStrawbery was run on this dataset both with and without the bias correction. rStrawberry uses the *R* package *qvalue* (10) to convert the nominal p values to q values. After intersecting with the 4468 genes, 12769 tested transcripts were obtained, 40 of which are called significant at q value 0.01 using bias correction. The FDR level of rStrawberry is around 0.29% which is well below the expected 1% level. To validate the effect of bias correction, I also include the result of no bias correction (Table 4.1). The result shows that without bias correction, rStrawberry produces a large number of false positives.

In summary, this result shows that rStrawberry and Cuffdiff 2 are able to control the FDR at or below the expected level while DEXSeq fails to control the FDR using the 6-sample GEUVAIDS data. However, these three methods have entirely different test units and they also filter genes differentially. rStrawberry tests for differentially spliced transcripts; Cuffdiff 2 tests for differentially spliced genes and DEXSeq tests for differentially spliced exons.

Table 4.1 Multiple testing adjusted p values cumulative table. rStrawberry, Cuffdiff 2 and DEXSeq were compared using 6 sample GEUVADIS data as a negative control, where no differential alternative splicing are expected.

Method	**<1e-03**	**<0.01**	**<0.025**	**<0.05**	**<=1**
rStrawberry	14	37	61	98	12769
rStrawberry(no bias correction)	122	287	420	556	12769
Cuffdiff 2	0	17	21	23	2895
DEXSeq	1876	2798	3436	4107	56350

4.3.3 A sensitivity analysis

To determine if the differential alternative splicing detection model of rStrawberry can recover true positives, I performed a sensitivity analysis using a simulated RNA-Seq dataset. This dataset was first used to benchmark differential alternative splicing methods (7), where two conditions, mocked Arabidopsis heat-stress time points, each with three replicates were

generated using our simulation pipeline. Around 10 million reads are mapped to multi-isoform genes. rStrawberry, DEXSeq, DSGseq (14), and Cuffdiff 2 were ran in the same way as for GEUVADIS data (section 4.3.2). All programs were run with the default parameters. Because the four methods have different test units, we convert the transcript p values from rStrawberry and exon p values from DEXSeq to genewise p values. For rStrawberry and DEXSeq, the minimum p value among the isoforms or the exons that belong to a gene is chosen to be the gene-level p value. Although DSGseq is considered as an event-centric model (7), it generates genewise test statistics.

Fig. 4.5 shows the Receiver operating characteristic (ROC) curve using the p values (or test statistic for DSGseq). Usually, people are more interested in ROC curve in the range of false positive rate 0 - 0.2. Therefore we also show the partial ROC in Fig. 4.6. Table 4.2 shows the number of tested genes, full AUC (Area Under the Curves) statistics and partial AUC of the four methods. From these results, we can see rStrawberry outperforms the other three methods. The distance between the top 2 methods, rStrawberry and Cuffdiff 2, is small, partial AUC at 0.9648 vs. 0.9547. However, Cufflinks filters more genes than rStrawberry, yielding a total 3603 tested genes vs. 3986 by rStrawberry. Transcript-centric methods rely on accurate transcript expression estimations. And when a gene is lowly expressed, those estimations are usually nor reliable and should not be used for differential analysis. On the other hand, event-centric methods, such as DEXSeq and DSGseq, do not need to deal with read assignment uncertainty and can test more genes.

Table 4.2 Area under the ROC curve (AUC) of the 4 methods were compared using simulated Arabidopsis data. Different methods have a different test units and filters which leads to different number of tested genes.

	Cuffdiff 2	**DEXSeq**	**DSGseq**	**rStrawberry**
Full AUC	0.9721	0.9082	0.8682	0.9778
Partial AUC	0.9547	0.8857	0.8214	0.9648
Number of genes	3603	5048	5564	3986

4.4 Conclusions and Discussion

Here, we present a novel method and R package called rStrawberry, which can detect differential alternative splicing from two groups of RNA-Seq samples. To our knowledge, rStrawberry is the first method that truly detects differential splicing at transcriptome level. In other words, rStrawberry report exactly which transcripts are differentially spliced across conditions. While other methods only reports either which genes are differentially spliced or which AS events are detected, where an AS event usually involves more than one transcripts. In addition, unlike other methods which process one gene at a time, rStrawberry generalizes the concept of "gene". For example, rStrawberry can group transcripts from an arbitrary number of loci and treat them as in a "super group" and compare the relative abundance of a transcript in that "super group" across conditions. As the knowledge about RNA increases, the scope of alternative splicing might expand. For example, the "super group" can be paralogues or gene families or pathways.

We have also demonstrated rStrawberry's performance on alternative splicing detection using both simulated data and real data. We benchmark its performance against Cuffdiff 2, DEXSeq and DSGseq, the top 3 methods based on our previous study (7). Only rStrawberry and Cuffdiff 2 are able to control the false discovery rate at expected levels using real data and rStrawberry outperform Cuffdiff 2 using simulated data where we have the ground truth.

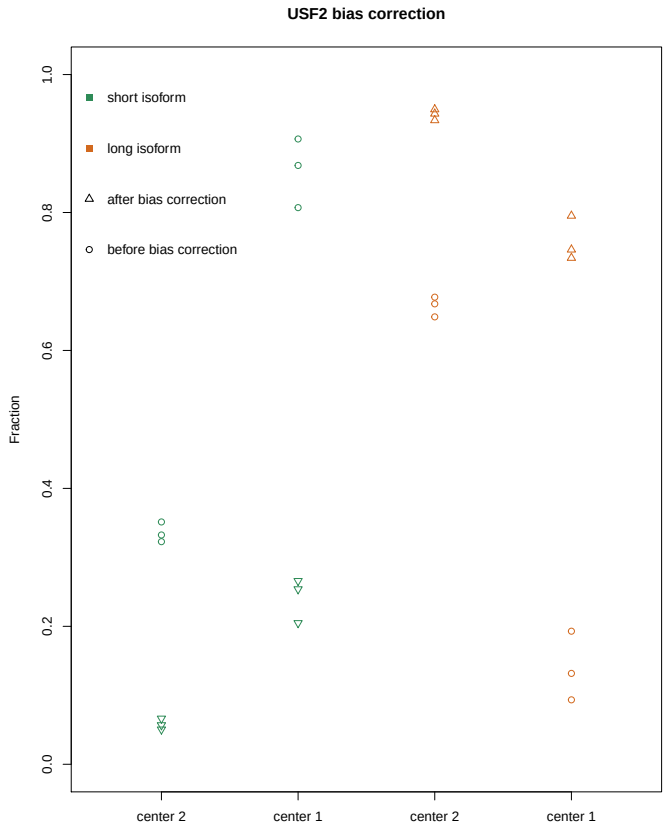


Figure 4.3 Predicted isoform fractions of gene USF2 before and after bias correction. Totally 6 samples are used for comparison, ERR188021, ERR188052, and ERR188088 from center 2 and ERR188204, ERR188 -317 and ERR188453 from center 1. The colors sea green and chocolate represent short isoform and long isoform respectively. The x-axis is a 2 by 2 factorial table of isoforms and centers. Thus a total of 4 x-axis ticks are displayed, short isoform on the first two ticks and long isoform on the last two ticks. For each tick, there are three samples and a total of 6 points. The y-values of them represent the predicted isoform fractions before and after bias correction. The open circle indicates before bias correction and the open triangle represents after bias correction. The point-up or point-down of the triangles indicate the relative isoform fraction is increased or decreased, respectively, after bias correction. It is clear that the fraction of long isoform increases after bias correction and the amount of increase is larger for center 1 than center 2. The paired t-test of the FPKM values before and after bias correction for the long isoform is 0.02607 vs. 0.04808 for center 1 and center 2 respectively. (8) has pointed out that the samples from center 1 suffer a more dramatic loss of coverage than center 2 when it comes to high-GC exons.

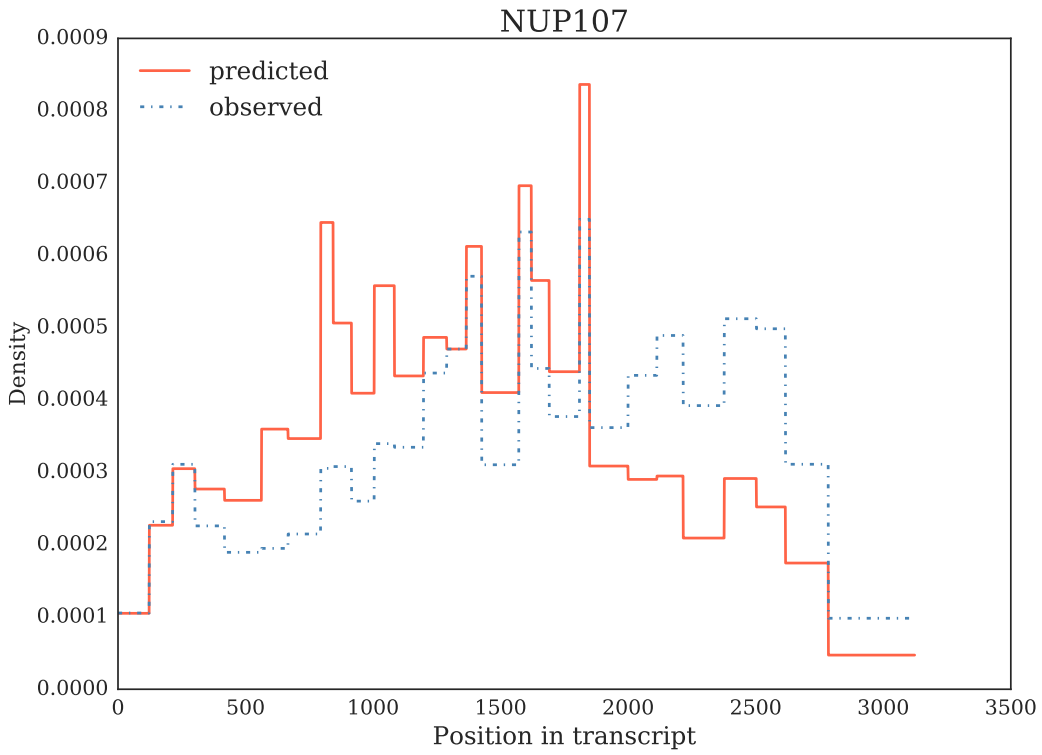


Figure 4.4 Predicted subexon coverage on NUP107 gene of ERR188297 sample using rStrawberry. The x-axis is the transcript position. And y-axis is the density so that the area under the curve is 1. The coverage is predicted only using the sequence signals, such as GC-content. And we can see these signals can explain, to some extent, the coverage variabilities of RNA-Seq.

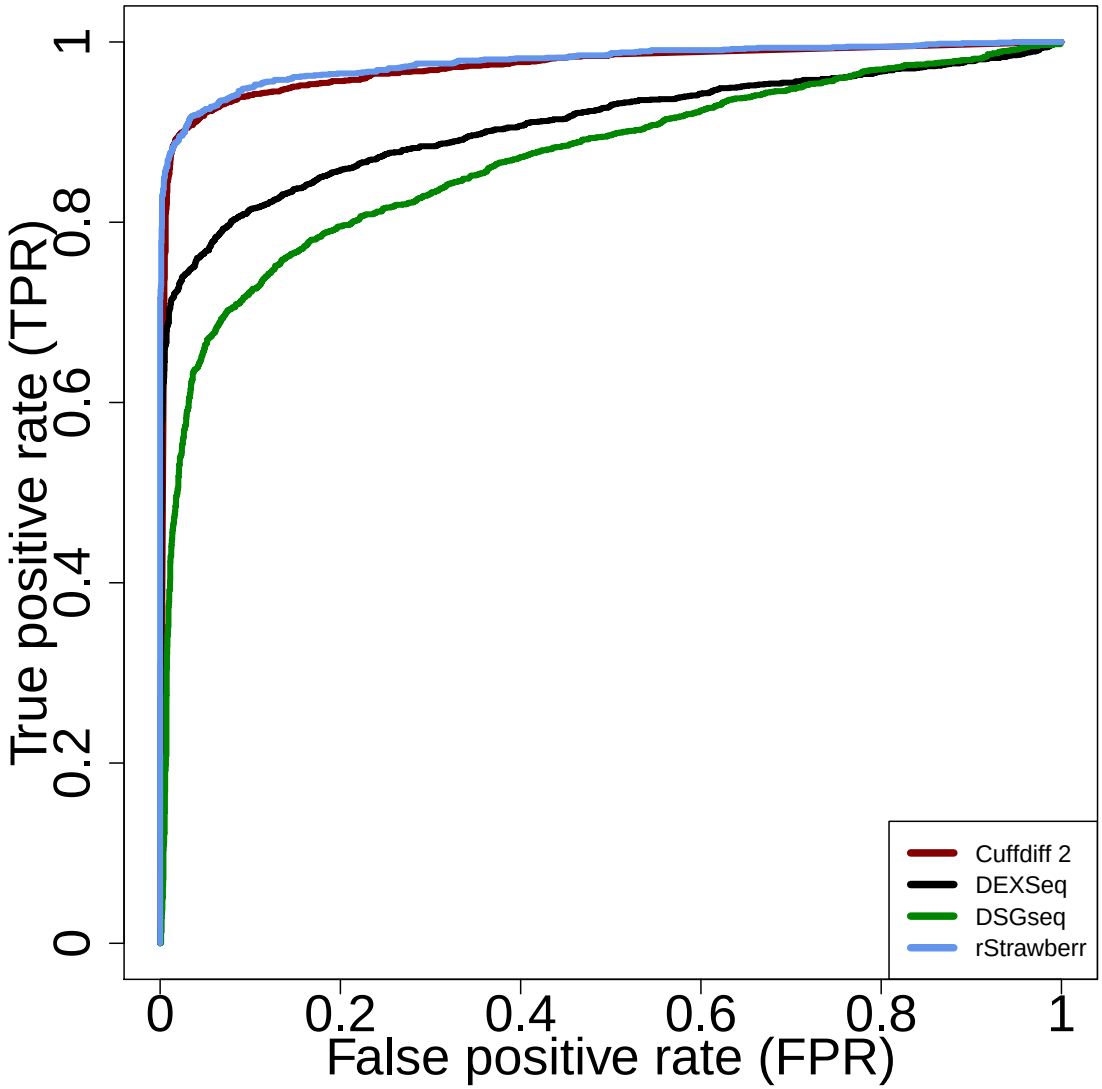


Figure 4.5 Full ROC curves of the differential alternative splicing detection results of 4 methods.

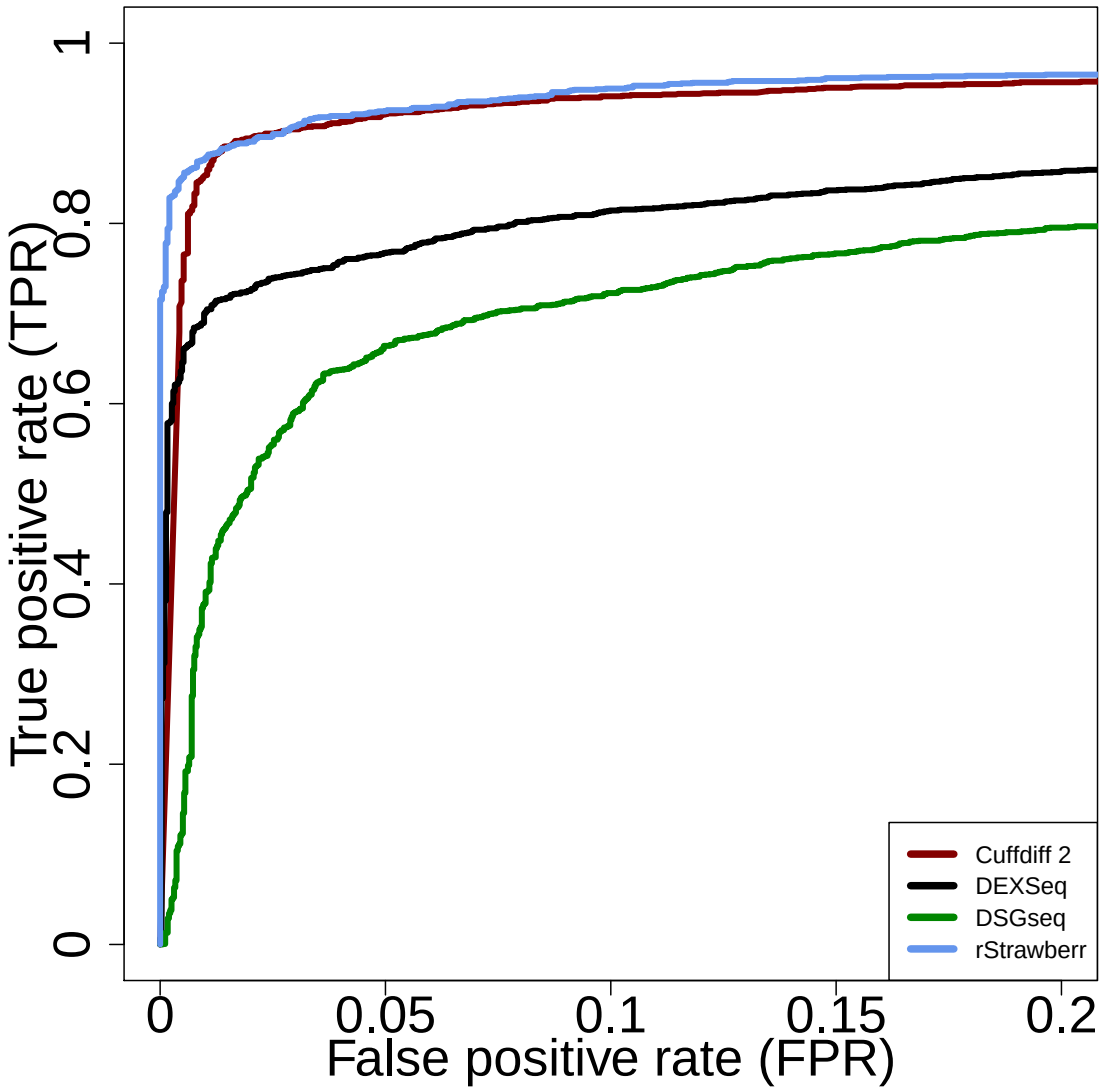


Figure 4.6 Partial ROC curves (False positive rate 0 - 0.2) of the differential alternative splicing detection results of 4 methods.

Bibliography

[1] R. Liu, and J. A. Dickerson. Strawberry: Fast and accurate genome-guided transcript reconstruction and quantification from RNA-Seq.. *PLOS Comp. Bio.*, accepted, Nov 2017.

[2] K. D. Hansen, S. E. Brenner, and S. Dudoit. Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Res.*, 38(12):e131, Jul 2010.

[3] D. C. Jones, W. L. Ruzzo, X. Peng, and M. G. Katze. A new approach to bias correction in RNA-Seq. *Bioinformatics*, 28(7):921–928, Apr 2012.

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

[5] N. F. Lahens, I. H. Kavakli, R. Zhang, K. Hayer, M. B. Black, H. Dueck, A. Pizarro, J. Kim, R. Irizarry, R. S. Thomas, G. R. Grant, and J. B. Hogenesch. IVT-seq reveals extreme bias in RNA sequencing. *Genome Biol.*, 15(6):R86, 2014.

[6] T. Lappalainen, M. Sammeth, M. R. Friedlander, P. A. 't Hoen, J. Monlong, M. A. Rivas, M. Gonzalez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, M. Barann, T. Wieland, L. Greger, M. van Iterson, J. Almlof, P. Ribeca, I. Pulyakhina, D. Esser, T. Giger, A. Tikhonov, M. Sultan, G. Bertier, D. G. MacArthur, M. Lek, E. Lizano, H. P. Buermans, I. Padioleau, T. Schwarzmayr, O. Karlberg, H. Ongen, H. Kilpinen, S. Beltran, M. Gut, K. Kahlem, V. Amstislavskiy, O. Stegle, M. Pirinen, S. B. Montgomery, P. Donnelly, M. I. McCarthy, P. Flicek, T. M. Strom, H. Lehrach, S. Schreiber, R. Sudbrak, A. Carracedo, S. E. Antonarakis, R. Hasler, A. C. Syvanen, G. J. van Ommen, A. Brazma, T. Meitinger, P. Rosenstiel, R. Guigo, I. G. Gut, X. Estivill, E. T. Dermitzakis, X. Estivill, R. Guigo, E. Dermitzakis, S. Antonarakis, T. Meitinger, T. M. Strom, A. Palotie,

J. F. Deleuze, R. Sudbrak, H. Lerach, I. Gut, A. C. Syvanen, U. Gyllensten, S. Schreiber, P. Rosenstiel, H. Brunner, J. Veltman, P. A. Hoen, G. J. van Ommen, A. Carracedo, A. Brazma, P. Flicek, A. Cambon-Thomsen, J. Mangion, D. Bentley, and A. Hamosh. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, Sep 2013.

[7] R. Liu, A. E. Loraine, and J. A. Dickerson. Comparisons of computational methods for differential alternative splicing detection using RNA-seq in plant systems. *BMC Bioinformatics*, 15(1):364, Dec 2014.

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

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

[10] J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. U.S.A.*, 100(16):9440–9445, Aug 2003.

[11] S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome Res.*, 22(10):2008–2017. Oct 2012.

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

[13] C. Trapnell, D. G. Hendrickson, M. Sauvageau, L. Goff, J. L. Rinn and L. Pachter. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nat. Biotechnol.*, 31:46–53. Dec 2012.

[14] W. Wang, Z. Qin, Z. Feng, X. Wang, and X. Zhang. Identifying differentially spliced genes from two groups of RNA-seq samples. *Gene*, 518(1):164–170. 2013

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

CHAPTER 5. GENERAL CONCLUSIONS

5.1 Conclusions

This dissertation is focused on three RNA-Seq applications, transcript assembly, transcript quantification and detection of differential alternative splicing. The short read length (usually 100-250 bp) of current RNA-Seq technology imposes multiple challenges for accurately solving these three problems. First for assembly, due to the overlapping isoforms, the transcript assembly graphs inevitably have more bubbles and branches, if compared to genome assembly. And because no read can cover a whole transcript, phasing distant reads into overlapping transcripts is very challenging. Thus, the complexity of transcript assembly grows exponentially as the number of isoforms increases. To address these challenges, Strawberry first utilizes the reference genome and state-of-the-art splice-aware alignment algorithms. Based on the alignments, Strawberry first partitions the reads into non-overlapping loci and assembles them simultaneously using multiple threads. More importantly, the exon-intron structures and transcription direction can all be inferred from the gapped alignments. Next, Strawberry converts splicing graphs into flow networks that are tailored for paired-end reads to construct a parsimonious set of transcripts. The distant reads are phased by the flow network algorithm. Based on our simulation results, Strawberry's genome dependent assembly recovers more true transcripts while achieving the same false discovery rate compared to two other leading methods.

The short reads also create challenges for transcript quantification as many reads align ambiguously to the overlapping isoforms. This read assignment challenge often requires solving a high-dimensional mixture model. In addition to the ambiguous assignment, new gene isoforms are often discovered in RNA-Seq experiments and the annotation-dependent transcript quantification methods described in chapter 2 fail to account for new isoforms and

are thus significantly bias against known isoforms. Compared to the annotation-dependent methods, the assembly step of strawberry can not only detect new isoform but also minimize the annotation bias. In addition, the graph structures used in assembly can be seamlessly propagated to quantification step to yield a highly efficient assemble-then-quantify workflow. To assign reads to overlapping transcripts, the quantification step uses a latent class model, where each read is assumed as coming from a mixture of classes (transcripts). Conditioning on the latent class, the reads are assumed to be generated from a subexon path. Strawberry uses a parametric distribution to model the subexon path probabilities, which allows being further extended to a regression model. Using the same simulated data that is used for benchmarking assembly, Strawberry outperforms Cufflinks and StringTie in terms of all three metrics. Using the real data from a highly cited method comparison study, Strawberry also beats Cufflinks and StringTie by convincing margins.

Altogether, Strawberry's transcript assembly and quantification algorithms described in chapter 3 are accurate, fast and scalable for a single sample of RNA-Seq, makes it an intriguing candidate when processing large data set (e.g., $>$ 100 million reads). It takes 12.35 min for Strawberry to process 100 million input RNA-Seq reads while a simple Linux program *wc* takes 8.69 min. However, the quantification model in chapter 3 have certain drawbacks when dealing with multiple samples and detecting differential splicing. First of all, it is unable to account for coverage bias that can adversely affect the differential splicing analysis. In addition, the model in chapter 3 is designed for single RNA-Seq sample and is thus unable to borrow information across biological replicates and, most importantly, address the count overdispersion problem. Thereby, to detect differential alternative splicing, chapter 4 presents a more comprehensive and complex model that is based on the generative model in chapter 3. The new model uses a double multinomial logit regressions. One multinomial logit regression is used to predict the differentially spliced transcripts across conditions. The other is used to account for coverage bias that is often observed in real RNA-Seq data. To overcome count overdispersion, Strawberry builds up a hierarchical

model on the first multinomial logit regression model. The hyperpriors are estimated using an empirical Bayes approach which borrows information across transcripts from multiple loci. Finally, in chapter 4, I have shown that Strawberry is able to control the false discovery rate using real data and recover true positives using simulated data when calling the differential spliced transcripts.

The differential alternative splicing model of Strawberry has many novelties. First, it is the first that simultaneously detects different splicing and estimates transcript abundances. Secondly, Strawberry combines a bias correction step into the detection of differential alternative splicing. Although many transcript quantification methods employee bias correction step, this feature has not been observed in differential alternative splicing detection methods. And the effect of coverage bias correction on differential splicing detection has not been studied until Strawberry. Thirdly, unlike other methods, Strawberry does not need to process one gene at a time. This allows Strawberry, potentially, to detect differentially spliced transcripts within paralogues or gene families.

5.2 Future works

In chapter 4, I use a two-step estimation of the hyperpriors. However, it might be better to integrate out the precision parameters. Currently, I have found that different priors have a considerable effect on the power of detection. Therefore, the method for estimating hyperpriors needs to be improved. Also, I have not fully utilized the power of the Bayesian modeling by avoiding drawing samples from posteriors. Therefore it might be worth trying the full Bayesian approach with MCMC sampling or variational Bayes if the sampling is too slow.

Secondly, detecting changes in transcript relative abundances in a gene family or paralogues might be an interesting topic. Especially for plant's genomes, where gene duplication is much more common than, say, mammalian genome. And, currently, the transcript quantification model of Strawberry in chapter 3 is restricted to uniquely aligned reads so

that the paralogues are avoided. However, the model in chapter 4 can group loci together to form a "super locus" which can contain all duplicated genes. And this can be done without a prior knowledge of the gene annotation by looking at non-uniquely mapped reads so might be applicable to non-model organisms. However, the challenge is finding a meaningful biological data set and hypothesis to test.

In terms of the implementation, the current implementation of the differential alternative splicing detection model is written in R which is known to be slow. And the users have to run two separate Strawberry C++ first and then rStrawberry, which is not user-friendly. In the future, I will re-implement rStrawberry in C++ to yield a single unified software.

APPENDIX A. ADDITIONAL MATERIAL

A.1 STAN model

STAN is used to estimate the posterior model 4.4. In particular, the point estimates of parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, as well as the hessian of $\boldsymbol{\beta}$ are obtained by STAN, which is written in STAN language as follow:

```
data {
int<lower = 2> J; // num samples
int<lower = 1> K; // num transcripts
vector<lower = 0>[K] Y[J]; // counts
vector[J] X[J]; //predictors
real<lower = 0> a0; // fixed prior
real<lower = 0> b0; // fixed prior
}

parameters {
matrix[K,J] beta;
vector<lower = 0> [K] gamma;
}

transformed parameters {
simplex[K] pi[J];
for (j in 1:J) {
pi[j] = softmax(beta * X[j]);
}
```

```
for (k in 1:K) {
sigma[k] = 1 / gamma[k];
}
}

model {
//priors
gamma ~ gamma(a0, b0);
for (k in 1:K) {
beta[k] ~ normal(0, sigma[k]);
}

for (j in 1:J) {
for (k in 1:K) {
target += Y[j][k] * log (pi[j][k]);
}
}
}.
```

A.2 Overdispersed RNA-Seq read counts

GEUVADIS (6) is a useful dataset to investigate technical variations, dispersion and etc. because it contains five cell line samples which were sequenced 8 times (8 replicated RNA-Seq samples) at 7 different labs.

I have taken one of the five biological samples, HG00117, and plotted gene-wise and transcript-wise dispersion pattern. The dispersion pattern is illustrated in a mean-variance relationship. In Fig. A.1, I plot the genewise log TPM variance vs. log TPM mean. Again, TPM refers to Transcript Per Kilobase Million. The gene-wise TPM is directly calculated

through the number of uniquely mapped reads. On the log scale, there seems to be a strong linear trend. If we treat the variance, y, as a function of mean, x, it seems to me that it is reasonable to use log linear model $\log y = a \log x + b$. It implies $Var(Y) = B\mu^a$, where Y are the counts, $B = e^b$, and μ is the mean.

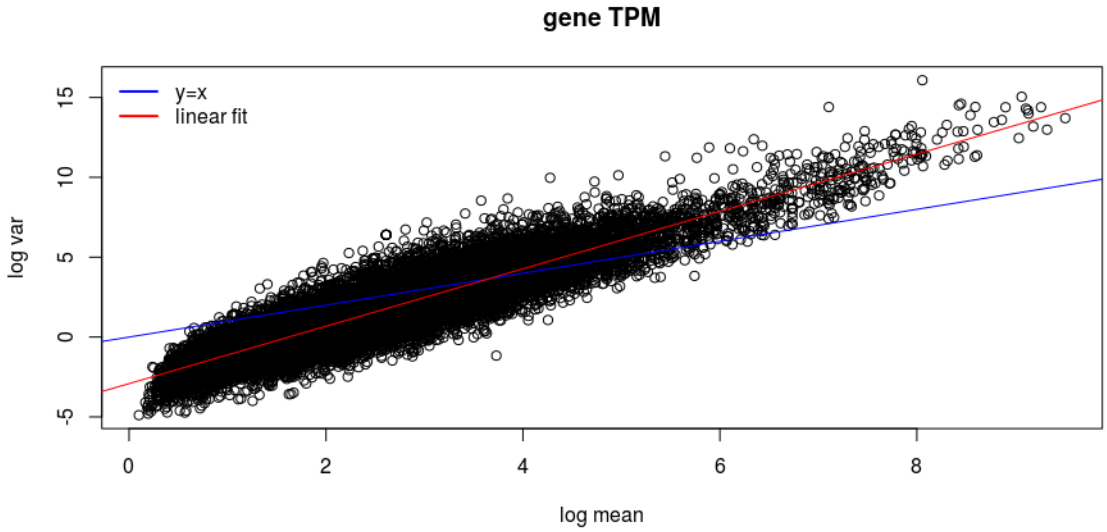


Figure A.1 Gene-wise dispersion on biological sample HG00117. Each data point represents a gene. Mean and variance of TPM is calculated using all 8 replicates. The red line is the best linear regression of variance on mean and the blue line is the 45-degree angle straight line which indicates no overdispersion.

The same analyses is repeated for transcript-level TPM and I include only two-isoform genes based on the RefSeq annotation (see Fig. A.2 The transcripts TPM is estimated by Strawberry (1). The options "–no-assembly" and "-g" are used. This will skip the reference based assembly and used the provided annotation (in this case, human RefSeq annotation) for quantification. Again, I can see a similar pattern as gene-wise expression. The slope of red line is larger than 45-degree and thus indicates overdispersion. Compare Fig. A.1 to A.2, I see no obvious different between the gene-level dispersion and transcript-level dispersion. My postulation is that because RNA-Seq directly sequence transcripts, the gene-level count

is merely an aggregation effect of transcript-level count. I also replaced RPKM with TPM, the result seems to be very similar (data not shown). Note that for all analyses, I excluded the low expressed data by setting FPKM/RPKM and TPM cutoff at 1.

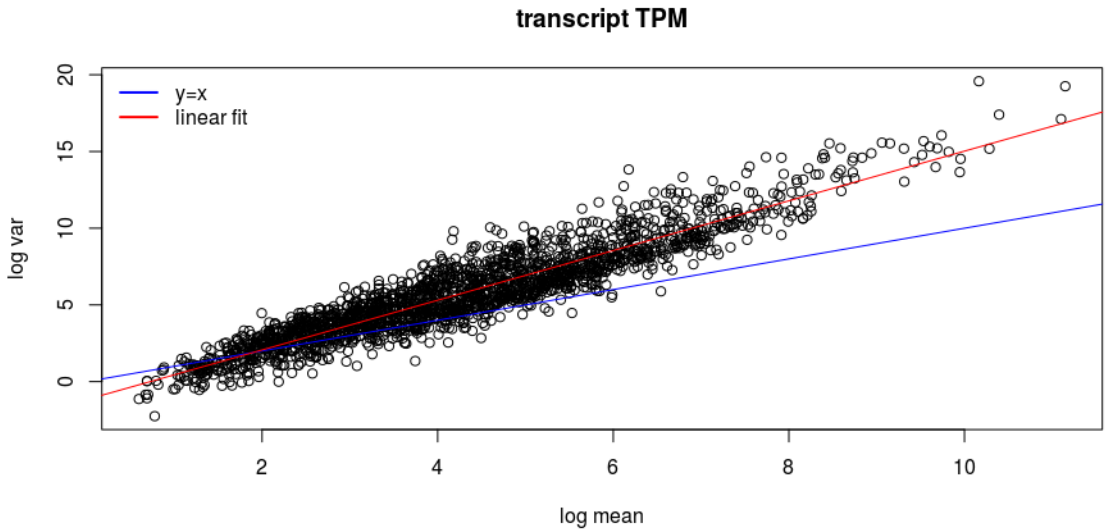


Figure A.2 Transcript-wise dispersion on biological sample HG00117. Each data point represents a transcript. And only the transcripts from two-isoform genes are used. Mean and variance of TPM is calculated using all 8 replicates. The red line is the best linear regression of variance on mean and the blue line is the 45-degree angle straight line which indicates no overdispersion.